Université de Montréal

Estimation de cartes d'énergie du bruit apériodique de la marche humaine avec une caméra de profondeur pour la détection de pathologies et Modèles légers de détection d'objets saillants basés sur l'opposition de couleurs

par

Didier Ndayikengurukiye

Département d'informatique et de recherche opérationnelle
Faculté des arts et des sciences

Thèse présentée à la Faculté des arts et des sciences
en vue de l'obtention du grade de Philosophiæ Doctor (Ph.D.)
en informatique

Juin, 2023

Université de Montréal
Faculté des arts et des sciences

Cette thèse intitulée:

Estimation de cartes d'énergie du bruit apériodique de la marche humaine avec une caméra de profondeur pour la détection de pathologies et Modèles légers de détection d'objets saillants basés sur l'opposition de couleurs

présentée par:

Didier Ndayikengurukiye

a été évaluée par un jury composé des personnes suivantes:

_____

Jean Meunier
président-rapporteur

_____

Max Mignotte
directeur de recherche

_____

Sébatien Roy
membre du jury

Rita Noumeir
examinatrice externe

Jean Meunier
représentant du doyen de la FAS

RÉSUMÉ

Cette thèse a pour objectif l'étude de trois problèmes : l'estimation de cartes de saillance de l'énergie du bruit apériodique de la marche humaine par la perception de profondeur pour la détection de pathologies, les modèles de détection d'objets saillants en général et les modèles légers en particulier par l'opposition de couleurs.

Comme première contribution, nous proposons un système basé sur une caméra de profondeur et un tapis roulant, qui analyse les parties du corps du patient ayant un mouvement irrégulier, en termes de périodicité, pendant la marche. Nous supposons que la marche d'un sujet sain présente n'importe où dans son corps, pendant les cycles de marche, un signal de profondeur avec un motif périodique sans bruit. La présence de bruit et son importance peuvent être utilisées pour signaler la présence et l'étendue de pathologies chez le sujet. Notre système estime, à partir de chaque séquence vidéo, une carte couleur de saillance montrant les zones de fortes irrégularités de marche, en termes de périodicité, appelées énergie de bruit apériodique, de chaque sujet. Notre système permet aussi de détecter automatiquement les cartes des individus sains et ceux malades.

Nous présentons ensuite deux approches pour la détection d'objets saillants. Bien qu'ayant fait l'objet de plusieurs travaux de recherche, la détection d'objets saillants reste un défi. La plupart des modèles traitent la couleur et la texture séparément et les considèrent donc implicitement comme des caractéristiques indépendantes, à tort.

Comme deuxième contribution, nous proposons une nouvelle stratégie, à travers un modèle simple, presque sans paramètres internes, générant une carte de saillance robuste pour une image naturelle. Cette stratégie consiste à intégrer la couleur dans les motifs de texture pour caractériser une micro-texture colorée, ceci grâce au motif ternaire local (LTP) (descripteur de texture simple mais puissant) appliqué aux paires de couleurs. La dissemblance entre chaque paire de micro-textures colorées est calculée en tenant compte de la non-linéarité des micro-textures colorées et en préservant leurs distances, donnant une carte de saillance intermédiaire pour chaque espace de couleur. La carte de saillance finale est leur combinaison pour avoir des cartes robustes.

Le développement des réseaux de neurones profonds a récemment permis des per-

formances élevées. Cependant, il reste un défi de développer des modèles de même performance pour des appareils avec des ressources limitées.

Comme troisième contribution, nous proposons une nouvelle approche pour un modèle léger de réseau neuronal profond de détection d'objets saillants, inspiré par les processus de double opposition du cortex visuel primaire, qui lient inextricablement la couleur et la forme dans la perception humaine des couleurs. Notre modèle proposé, CoSOV1net, est entraîné à partir de zéro, sans utiliser de "backbones" de classification d'images ou d'autres tâches. Les expériences sur les ensembles de données les plus utilisés et les plus complexes pour la détection d'objets saillants montrent que CoSOV1Net atteint des performances compétitives avec des modèles de l'état-de-l'art, tout en étant un modèle léger de détection d'objets saillants et pouvant être adapté aux environnements mobiles et aux appareils à ressources limitées.

Mots clés : analyse clinique de la marche, modèle léger, détection d'objets saillants, réseau neuronal profond léger, opposition de signaux de cônes, opposition de couleur, motif local ternaire, FastMap, bruit apériodique, Kinect.

ABSTRACT

The purpose of this thesis is to study three problems: the estimation of saliency maps of the aperiodic noise energy of human gait using depth perception for pathology detection, and to study models for salient objects detection in general and lightweight models in particular by color opposition.

As our first contribution, we propose a system based on a depth camera and a treadmill, which analyzes the parts of the patient's body with irregular movement, in terms of periodicity, during walking. We assume that a healthy subject gait presents anywhere in his (her) body, during gait cycles, a depth signal with a periodic pattern without noise. The presence of noise and its importance can be used to point out presence and extent of the subject's pathologies. Our system estimates, from each video sequence, a saliency map showing the areas of strong gait irregularities, in terms of periodicity, called aperiodic noise energy, of each subject. Our system also makes it possible to automatically detect the saliency map of healthy and sick subjects.

We then present two approaches for salient objects detection. Although having been the subject of many research works, salient objects detection remains a challenge. Most models treat color and texture separately and therefore implicitly consider them as independent feature, erroneously.

As a second contribution, we propose a new strategy through a simple model, almost without internal parameters, generating a robust saliency map for a natural image. This strategy consists in integrating color in texture patterns to characterize a colored micro-texture thanks to the local ternary pattern (LTP) (simple but powerful texture descriptor) applied to the color pairs. The dissimilarity between each colored micro-textures pair is computed considering non-linearity from colored micro-textures and preserving their distances. This gives an intermediate saliency map for each color space. The final saliency map is their combination to have robust saliency map.

The development of deep neural networks has recently enabled high performance. However, it remains a challenge to develop models of the same performance for devices with limited resources.

As a third contribution, we propose a new approach for a lightweight salient objects detection deep neural network model, inspired by the double opponent process in the primary visual cortex, which inextricably links color and shape in human color perception. Our proposed model, namely CoSOV1net, is trained from scratch, without using any image classification backbones or other tasks. Experiments on the most used and challenging datasets for salient objects detection show that CoSOV1Net achieves competitive performance with state-of-the-art models, yet it is a lightweight detection model and it is a salient objects detection that can be adapted to mobile environments and resource-constrained devices.

Keywords: gait clinic analysis, lightweight salient object detection, salient object detection, lightweight neural network, cone opponent, color opponent, local ternary pattern, FastMap, aperiodic noise, Kinect.

TABLE DES MATIÈRES

LISTE DES FIGURES

LISTE DES TABLEAUX

LISTE DES ALGORITHMES

LISTE DES SIGLES

2D	2 dimensions.
CIELAB	commission internationale de l'éclairage L^* a^* b^* : espace de couleur.
FPR	taux de faux positifs ("False Positive Rate").
HC	contraste basé sur l'histogramme ("histogram-based contrast").
ICA	analyse en composantes indépendantes ("Independent Component Analysis").
LBP	motifs binaires locaux ("local binary pattern").
LDA	analyse discriminante linéaire ("Linear Discriminant Analysis").
MDS	positionnement multidimensionnel ("multi-dimensional scaling").
PCA	analyse en composante principale ("principal component analysis").
RC	contraste basé sur la région ("region-based contrast").
ReLU	fonction d'activation de neuronnes ("rectified linear unit").
SVH	système visuel humain.
TPR	taux de vrais positifs ("True Positive Rate").
HVS	Human Visual System
LTP	Local Ternary Patterns
LBP	Local Binary Patterns
SLICO	Simple Linear Iterative Clustering with zero parameter
SLIC	Simple Linear Iterative Clustering
MDS	Multi-dimensional Scaling
RGB	Red–Green–Blue
HSL	Hue–Saturation–Luminance
CMY	Cyan–Magenta–Yellow
MAE	Mean Absolute Error
ECSSD	Extended Complex Scene Saliency Dataset
MSRA10K	Microsoft Research Asia 10,000 dataset
DUT-OMRON	Dalian University of Technology—OMRON Corporation dataset
SED2	Segmentation evaluation database with 2 salient objects dataset
HS	Hierarchical saliency detection model

CHS	Hierarchical image saliency detection on extended CSSD model
RR	Red-Red
RG	Red-Green
RB	Red-Blue
GR	Green-Red
GG	Green-Green
GB	Green-Blue
BR	Blue-Red
BG	Blue-Green
BB	Blue-Blue
GR	Graph-regularized saliency detection
MNP	Saliency for image manipulation
LBI	Looking beyond the image
LMLC	Bayesian saliency via low and mid level cues
SVO	Fusing generic objectness and visual saliency
SWD	spatially weighted dissimilarity
HC	Histogram-based contrast
SEG	Segmenting salient objects
CA	Context-aware saliency detection
FT	Frequency-tuned salient region detection
AC	Achanta *et al.* : Salient region detection and segmentation

NOTATION

x, β	scalaires
$\mathbf{x}, \beta$	vecteurs
$\lvert x \rvert$	valeur absolue de x
$\lVert \mathbf{x} \rVert_2$	norme euclidienne du vecteur $\mathbf{x}$
$\lVert \mathbf{x} \rVert$	norme euclidienne du vecteur $\mathbf{x}$

Béni soit l'Éternel Dieu, le Dieu d'Israël, qui Seul fait des prodiges ! Béni soit à jamais son Nom glorieux ! Que toute la Terre soit remplie de Sa gloire ! Amen ! Amen !

(Psaumes 72 : 18-19)

À mon épouse,

nos enfants,

ma famille,

tous ceux qui m'ont soutenu depuis l'enfance.

REMERCIEMENTS

Je tiens à remercier très chaleureusement mon directeur de thèse, le professeur Max Mignotte pour avoir accepté de diriger mes travaux de recherche. Je le remercie pour son encadrement, ses conseils, ses excellentes connaissances en traitement de signaux et d'images en particulier qu'il m'a partagées, tous les efforts qu'il a fournis pour que ce travail arrive à cette étape.

Je remercie les membres du jury pour m'avoir fait l'honneur d' accepter d'évaluer cette thèse.

Mes remerciements sont adressés aussi à tous les professeurs de l'université de Montréal spécialement ceux du département d'informatique et de recherche opérationnelle. Je remercie tout le personnel de l'université de Montréal en particulier celui du département d'informatique et de recherche opérationnelle qui a rendu mes études académiques possibles. Je voudrais aussi remercier mon épouse, mes enfants et toute ma famille pour leur soutien, chacun de sa façon. Je remercie particulièrement mon épouse pour sa patience dans les exigences académiques auxquelles j'ai fait face. Que tous ceux qui m'ont soutenu de près ou de loin trouvent ici ma profonde gratitude.

CHAPITRE 1

INTRODUCTION GÉNÉRALE

Le système visuel humain (SVH) normal possède la capacité de percevoir la couleur et le relief des objets, la distance entre les objets ainsi que la distance qui sépare les objets de celui qui les observe. Aussi, le système visuel humain (SVH) a fait l'objet de plusieurs études qui ont mis en évidence que le SVH est doté de mécanismes de perception de couleurs [1–5] et de perception de profondeur [6–8].

Ainsi, la compréhension des données visuelles permet à l'être humain de traiter plusieurs tâches les unes plus complexes que les autres.

Grâce à une approche similaire à celle du systèmes visuel humain [9, 10], la science de la vision par ordinateur cherche à permettre aux ordinateurs d'acquérir une compréhension détaillée des données visuelles pour aider les humains et faciliter ainsi des tâches souvent complexes et difficiles.

Dans cette thèse, nous proposons trois modèles de vision par ordinateur, basés sur des approches nouvelles permettant de comprendre des données visuelles, séquences vidéo de profondeur pour le premier modèle proposé et images pour les deux autres, afin de relever le défi d'analyse de la marche à un faible coût et de façon non invasive et facile à utiliser ou de faire la détection d'objets saillants avec des modèles pratiques sur des appareils à ressources limitées. Le premier utilise la perception de profondeur et les deux autres la perception de couleurs.

1. Dans le premier modèle, nous proposons un nouveau système d'analyse de la marche, à partir d'une caméra de profondeur placée devant un sujet marchant sur un tapis roulant classique, capable de détecter les anomalies de la marche humaine et de quantifier leur sévérité et aussi de localiser les différentes parties du corps endommagées ou altérées du patient. Ceci se fait grâce une carte couleur de saillance montrant les zones de fortes irrégularités de marche, en termes de périodicité (également appelée énergie de bruit apériodique), pour chaque séquence vidéo du sujet

en observation pour la détection de pathologies.

2. Dans le deuxième modèle, nous proposons une stratégie originale et efficace, à travers un modèle simple, presque sans paramètres internes, qui génère une carte de saillance robuste pour une image naturelle. Cette stratégie consiste à intégrer des informations de couleur dans des motifs de texture locaux pour caractériser une micro-texture de couleur.

3. Dans le troisième modèle, nous proposons une nouvelle approche pour un modèle léger de réseau neuronal profond de détection d'objets saillants, inspiré par les processus d'opposition de signaux des cônes individuellement et spatialement dans le cortex visuel primaire (V1), qui lient inextricablement la couleur et la forme dans la perception humaine des couleurs. Ce modèle peut être adapté aux environnements mobiles et aux appareils à ressources limitées.

1.1 Modèle estimant les cartes d'énergie du bruit apériodique de la marche humaine avec une caméra de profondeur pour la détection de pathologies

La marche se comporte plus comme une tâche motrice complexe que comme une tâche motrice automatisée et rythmée [11]. Elle implique l'équilibre, le rythme et la coordination de la vue et des autres sens ; elle est aussi difficile à décrire quantitativement et qualitativement [12].

L'étude de la marche humaine a de nombreux domaines d'application tels que l'identification humaine [13–20], la robotique, les sciences du sport [21, 22], la vidéosurveillance, la médecine [23, 24], etc.

La marche humaine n'est plus considérée comme une tâche purement motrice, mais comme un indicateur de la santé globale et un important outils de prédiction de l'état de santé et de la survie des personnes âgées selon Morris *et al.* [25].

Le trouble de la marche est le principal symptôme utilisé par un clinicien pour diagnostiquer et évaluer l'évolution de maladies neurodégénératives comme la sclérose laté-

rale amyotrophique (SLA), la maladie de Parkinson (MP) et la maladie de Huntington (MH) [26].

La plupart des travaux de recherche sur l'analyse de la marche pour d'éventuelles applications cliniques utilisent des systèmes à marqueurs et caméras multiples [27–29]. Ces systèmes sont les plus précis pour l'acquisition de données dans l'étude de la marche humaine [30, 31] mais aussi éparses (et généralement non réparties de manière équidistante) sur le corps. De plus, le coût élevé de ces systèmes haut de gamme [32] empêche leur utilisation généralisée pour les pratiques cliniques de routine ce qui fait que le besoin de systèmes sans marqueurs se fait également sentir [33].

Ainsi d'autres systèmes, moins invasifs et moins difficiles à manipuler, comme les semelles [28, 34], les chaussures sans fil [35], les accéléromètres [36–38] et les caméras de profondeur Microsoft Kinect™ [39] sont également de plus en plus utilisés.

Dans ce dernier cas, Kinect peut fournir une image de profondeur utile pour le développement de systèmes de surveillance à domicile qui détectent automatiquement les chutes ou lorsque le risque de chute augmente [40, 41] ou fournir directement les informations de données du squelette [42] pour l'estimation automatisée des paramètres de marche (spatiaux et temporels) des enfants [43] ou d'extraire un ensemble précis et fiable de caractéristiques de marche [44] grâce à une approche de régression (basée sur un ensemble d'arbres de régression). Plus généralement, les informations sur le squelette complet du corps fournies par Kinect peuvent également être efficacement utilisées pour la marche humaine [45] ou la reconnaissance de posture [46] ou le suivi corporel dans le cadre d'applications de santé (coaching ou exercice physique de la population âgée) [47], etc [48].

D'autres modèles [49, 50] ont utilisé une caméra de profondeur Microsoft Kinect ™ et un tapis roulant pour l'analyse de la marche humaine en soutenant l'hypothèse selon laquelle une démarche symétrique est généralement attendue chez les personnes en bonne santé, l'asymétrie indiquant par conséquent une marche anormale.

Dans cette thèse, nous proposons un modèle basé sur la quantité de bruit altérant un mouvement de profondeur périodique idéal de chaque partie du corps humain pendant la marche. Dans notre modèle, nous supposons donc que la démarche d'un sujet sain

présente, n'importe où dans son corps, au cours des cycles de marche, un signal de profondeur (dépendant du temps et d'un capteur Kinect^{TM}) avec un motif périodique sans bruit. À partir de chaque séquence vidéo, le système proposé est capable d'estimer une carte couleur visualisant les zones de fortes irrégularités de marche, en termes de périodicité, sur la silhouette corporelle du patient.

Afin d'obtenir une estimation fiable de cette carte d'énergie de bruit apériodique, nous avons décidé de l'estimer, de deux manières totalement complémentaires, à savoir l'estimation dans le domaine temporel et celle dans le domaine fréquentiel. Cette stratégie nous permet d'obtenir deux estimations différentes conduisant à des erreurs d'estimation différentes qui peuvent ensuite être efficacement combinées afin d'améliorer le résultat de la classification en les fusionnant de manière complémentaire, en termes d'interaction d'informations [51].

Cette carte est clairement informative et très discriminante pour un classement visuel direct même pour un non-spécialiste. Ici, l'emplacement et le degré de bruit (représentant le degré d'irrégularités de la marche) peuvent être utilisés comme un bon indicateur d'une pathologie possible ou pour fournir des informations sur la présence et l'étendue des problèmes médicaux. Un système automatique, basé sur l'extraction/classification des caractéristiques de chaque carte obtenue est également proposé et permet de détecter automatiquement les cartes représentant les individus sains et celles représentant les individus ayant des problèmes orthopédiques.

1.2 Modèle de détection d'objets saillants basé sur OCLTP, SLICO et FastMap

Les humains — ou les animaux en général — ont un système visuel doté de mécanismes attentionnels. Ces mécanismes permettent au système visuel humain (SVH) de sélectionner parmi la grande quantité d'informations reçues celle qui est pertinente et de ne traiter en détail que les aspects pertinents [52]. Ce phénomène s'appelle l'attention visuelle. Cette mobilisation de ressources pour le traitement d'une partie seulement de l'ensemble de l'information permet son traitement rapide. Ainsi le regard se dirige rapidement vers certains objets d'intérêt. Pour les êtres vivants, cela peut parfois être vital

car ils peuvent décider s'ils sont face à une proie ou à un prédateur [53]. La zone d'intérêt choisie par le SVH comme information pertinente correspond généralement à une forme, un ensemble de formes avec une couleur, un mélange de couleurs, un mouvement ou une texture discriminante dans la scène qui se différencie sensiblement du reste de l'image.

L'attention visuelle est réalisée de deux manières, à savoir *l'attention ascendante ("Bottom-up attention")* et *l'attention descendante ("top-down attention")* [54]. L'*attention ascendante* est un processus rapide, automatique, involontaire et dirigé par les propriétés de l'image presque exclusivement [52]. L'*attention descendante* est un mécanisme volontaire plus lent dirigé par des phénomènes cognitifs tels que les connaissances, les attentes, les récompenses et les objectifs actuels [55]. Dans ce travail, nous nous concentrons sur le *mécanisme attentionnel ascendant* qui est basé sur l'image.

L'attention visuelle a fait l'objet de plusieurs travaux de recherche dans les domaines de la psychologie cognitive [56, 57] et des neurosciences [58], pour n'en citer que quelques-uns. Les chercheurs en vision par ordinateur ont également utilisé les avancées de la psychologie cognitive et des neurosciences pour mettre en place des modèles informatiques de saillance visuelle qui exploitent cette capacité du système visuel humain à comprendre rapidement et efficacement une image ou une scène. Ainsi, de nombreux modèles informatiques de saillance visuelle ont été proposés et sont principalement subdivisés en deux catégories : les modèles conventionnels (par exemple, le modèle de Yan et al. [59]) et les modèles d'apprentissage profond (par exemple, le modèle de Gupta et al. [60]). La plupart des modèles peuvent être trouvés dans Gupta *et al.* [61], Borji *et al.* [62] and Borji *et* Itti [63].

Les modèles informatiques de saillance visuelle ont plusieurs applications telles que la compression d'images/vidéos [64], la correction d'images [65], l'analyse d'illustrations iconographiques [66], la récupération d'images [67], l'optimisation des publicités [68], l'évaluation de l'esthétique [69], l'évaluation de la qualité d'image [70], le reciblage d'images [71], le montage d'images [72], le collage d'images [73], la reconnaissance, le suivi et la détection d'objets [74], pour n'en citer que quelques-uns.

Les modèles informatiques de saillance visuelle sont orientés soit vers la prédiction

de la fixation de l'œil, soit vers la segmentation ou la détection d'objets saillants. C'est cette dernière orientation qui fait l'objet de ce travail. La détection d'objets saillants est matérialisée par des cartes de saillance. Une carte de saillance est représentée par une image en niveaux de gris dans laquelle une région de l'image doit être plus blanche car elle diffère significativement du reste de l'image en termes de forme, d'ensemble de formes avec une couleur, un mélange de couleurs, un mouvement ou une texture discriminante ou généralement tout attribut perçu par le système visuel humain.

Ici, nous proposons un modèle simple et presque sans paramètre qui nous donne une carte de saillance efficace pour une image naturelle en utilisant une nouvelle stratégie. Le modèle proposé, contrairement aux méthodes de détection de saillance classiques, utilise les caractéristiques de texture et de couleur d'une manière qui intègre la couleur dans les caractéristiques de texture à l'aide d'algorithmes simples et efficaces. En effet, la texture est un phénomène omniprésent dans les images naturelles : les images des montagnes, des arbres, des buissons, de l'herbe, du ciel, des lacs, des routes, des bâtiments, etc. apparaissent sous différents types de texture. Haidekker [75] soutient que la texture et l'analyse de forme sont des outils très puissants pour extraire des informations d'image de manière non supervisée. Cet auteur ajoute que l'analyse de la texture est devenue une étape clé dans l'analyse quantitative et non supervisée des images biomédicales [75]. D'autres auteurs, tels que Knutsson et Granlund [76], Ojala et al. [77], conviennent que la texture est une caractéristique importante pour l'analyse de scène d'images. Knutsson et Granlund affirment également que la présence d'une texture quelque part dans une image est plus une règle qu'une exception. Ainsi, la texture dans l'image s'est avérée d'une grande importance pour la segmentation de l'image, l'interprétation des scènes [78], la reconnaissance du visage, la reconnaissance de l'expression faciale, l'authentification du visage, la reconnaissance du genre, la reconnaissance de la démarche et l'estimation de l'âge, pour n'en nommer que quelques uns [79]. De plus, les images naturelles sont généralement aussi des images en couleur et il est alors important de prendre également en compte ce facteur. Dans notre application, la couleur est prise en compte et intégrée de façon originale, *via* l'extraction des caractéristiques texturales faites sur les couples de canaux de couleurs opposés.

Bien qu'il y ait beaucoup de travaux concernant la texture, il n'y a pas de définition formelle de la texture [76]. Il n'y a pas non plus d'accord sur une technique unique de mesure ou de description de la texture [78, 79]. Notre modèle utilise le descripteur de texture appelé le motif ternaire local ("Local Ternary Patterns" : LTP) [80]. Le LTP ("Local Ternary Patterns") est une extension du motif binaire local ("Local Binary Pattern" : LBP) avec trois valeurs de code au lieu de deux pour LBP. Ce dernier est connu pour être un puissant descripteur de texture [79, 81]. Ses principales qualités sont l'invariance vis-à-vis des changements de niveaux de gris monotones et la simplicité de calcul et son inconvénient est qu'il est sensible au bruit dans des régions uniformes de l'image.

En revanche, LTP est plus discriminant et moins sensible au bruit dans les régions uniformes. Le LTP ("Local Ternary Patterns") est donc mieux adapté pour résoudre notre problème de détection de saillance. Certes, la présence dans les images naturelles de plusieurs motifs rend complexe la détection d'objets saillants. Cependant, le modèle que nous proposons ne se concentre pas uniquement sur les motifs de l'image en les traitant séparément des couleurs comme le font la plupart des modèles [82, 83], mais il prend en compte à la fois la présence dans les images naturelles de plusieurs motifs et de couleur, pas séparément. Cette tâche d'intégration de la couleur dans les caractéristiques de texture est accomplie par le descripteur de texture LTP ("Local Ternary Patterns") appliqué à des paires de couleurs opposées d'un espace de couleurs donné, nous l'appelons alors OCLTP ("Opponent Color Local Ternary Patterns"). Le LTP décrit les motifs de texture locaux pour une image en niveaux de gris à travers un code attribué à chaque pixel de l'image en le comparant avec ses voisins. Lorsque LTP est appliqué à une paire de couleurs opposées, devenant ainsi OCLTP, le principe est similaire à celui utilisé pour une image en niveaux de gris. Cependant, pour LTP sur une paire de couleurs opposées, les motifs texturaux locaux sont obtenus grâce à un code attribué à chaque pixel, mais la valeur du pixel de la première couleur de la paire est comparée aux équivalents de ses voisins dans la deuxième couleur de la paire. La couleur est ainsi intégrée aux motifs texturaux locaux. De cette manière, on caractérise les micro-textures colorées de l'image sans séparer les textures dans l'image et les couleurs dans cette même image. Les frontières des micro-textures de couleur correspondent aux superpixels obtenus grâce à

l'algorithme SLICO ("Simple Linear Iterative Clustering with zero parameter") [84] qui est plus rapide et présente une adhérence aux frontières de pointe. Nous tenons à souligner qu'il existe d'autres algorithmes de superpixels qui ont de bonnes performances comme l'algorithme AWkS [85]. Cependant, nous avons choisi SLICO car il est rapide et presque sans paramètre. Un vecteur caractéristique représentant la micro-texture couleur est obtenu par la concaténation des histogrammes du superpixel (définissant la micro-texture) de chaque paire de couleurs opposées. Chaque pixel a ensuite été caractérisé par un vecteur représentant la micro-texture couleur à laquelle il appartient. Nous avons ensuite comparé les micro-textures colorées caractérisant chaque paire de pixels de l'image en cours de traitement grâce à la version rapide de la méthode de positionnement multidimensionnel MDS ("multi-dimensional scaling") *FastMap* [86]. Cette comparaison nous permet de saisir le degré d'unicité d'un pixel ou la rareté d'un pixel. La méthode FastMap permettra cette capture en tenant compte des non-linéarités dans la représentation de chaque pixel. Enfin, comme il n'y a pas d'espace de couleur unique adapté à l'analyse de la texture des couleurs [87], nous avons combiné les différentes cartes générées par FastMap à partir de différents espaces de couleur à savoir RGB, HSL, LUV et CMY, pour exploiter les forces de chacun dans la carte de saillance finale.

1.3 Modèle léger de détection d'objets saillants CoSOV1Net

Les modèles de détection d'objets saillants se répartissent généralement en deux catégories, à savoir les modèles conventionnels et les modèles basés sur l'apprentissage profond, qui diffèrent par le processus d'extraction des caractéristiques. Les premiers utilisent des caractéristiques obtenues par l'ingéniosité de leurs concepteurs, tandis que les seconds utilisent des caractéristiques apprises à partir d'un réseau de neurones profond. Notre précédent modèle de détection d'objets saillants fait partie de la première catégorie de modèles. Le second modèle de détection d'objets saillants que nous proposons dans cette thèse fait partie de la deuxième catégorie, les modèles basés sur l'apprentissage profond.

Grâce aux puissantes méthodes d'apprentissage des représentations, les modèles de

détection d'objets saillants basés sur l'apprentissage profond ont récemment montré des performances supérieures aux modèles conventionnels [61, 88]. La haute performance des modèles de détection d'objets saillants basés sur l'apprentissage profond est indéniable. Cependant, ils sont aussi généralement lourds si l'on considère leur nombre de paramètres et la mémoire occupée en plus de leur coût de calcul élevé et de leur vitesse de détection lente. Cela rend ces modèles moins pratiques pour les capteurs de vision à ressources limitées, pour les appareils mobiles qui ont de nombreuses contraintes sur leur mémoire, leurs capacités de calcul et pour les applications en temps réel [89, 90]. D'où le besoin de modèles légers de détection d'objets saillants, dont les performances sont comparables aux modèles de l'état de l'art, avec l'avantage de pouvoir être déployés sur des capteurs de vision ou des appareils mobiles à ressources limitées et d'avoir une vitesse de détection permettant leur utilisation dans les applications en temps réel.

Ces dernières années, des modèles légers de détection d'objets saillants ont été proposés avec différentes stratégies et architectures. Certains s'inspirent de modèles de segmentation [91, 92], d'autres sont basés sur une architecture rationalisée [93–96]. Certains auteurs se sont inspirés de la perception visuelle hiérarchique des primates ou du système visuel humain [97] et d'autres du mécanisme d'attention stéréoscopique du système visuel humain [90], pour n'en citer que ceux-là.

Dans cette thèse, nous avons proposé une approche originale pour un nouveau modèle de réseau neuronal léger de détection d'objets saillants qui peut être adapté aux appareils mobiles ou aux dispositifs à ressources limitées avec les propriétés intéressantes de pouvoir être entraîné à partir de zéro, sans avoir à utiliser des "backbones" d'autres modèles et avec peu de paramètres mais tout en ayant des performances comparables aux modèles de l'état de l'art. Nous voudrions signaler qu'un "backbone" est la partie de base d'un réseau de neurones, qui encode les données d'entrée sous une représentation quelconque afin de passer cette représentation de caractéristiques à la partie qui va la décoder pour une certaine tâche comme la classification ou la détection. Les parties de base les plus utilisées sont celles des réseaux de neurones déjà entraînés, les plus connus pour leur efficacité comme ResNet [98], Xception [99], MobileNets [93], etc.

Étant donné que la détection d'objets saillants est une capacité du système visuel

humain, et qu'un système visuel humain normal le fait rapidement et correctement, nous avons utilisé des images ou des scènes encodant les progrès de la recherche en neurosciences, en particulier pour le système visuel humain au stade précoce [2, 3, 100]. Notre stratégie dans ce modèle s'inspire donc de deux découvertes en neurosciences dans la perception humaine des couleurs, à savoir :

1. l'encodage en opposition de couleur dans le stade précoce du SVH (Système Visuel Humain) [4, 5, 101, 102] ;

2. le fait que la couleur et le motif sont inextricablement liés dans la perception humaine des couleurs [1, 100].

Ces deux découvertes en neurosciences nous ont inspirés pour concevoir un module CoSOV1 ("Cone- and Spatial-Opponency Primary Visual cortex") qui extrait les caractéristiques au niveau spatial et entre les canaux de couleur en même temps pour intégrer la couleur dans les motifs. Il peut aussi accomplir cette tâche entre les canaux de cartes de caractéristiques ("feature maps").

Ainsi, dans cette thèse, nous proposons CoSOV1Net, un réseau de neurones profond basé sur le module CoSOV1 ("Cone- and Spatial-Opponency Primary Visual cortex") pour un modèle léger de détection d'objets saillants, qui s'articule sur deux idées principales.

Premièrement, au début du réseau de neurones, notre modèle oppose les canaux de couleur deux à deux en les groupant (R-R, R-G, R-B, G-G, G-B, B-B) et extrait de chaque paire de canaux les caractéristiques au niveau spatial des canaux et entre les canaux de couleur en même temps pour intégrer la couleur dans les motifs. De cette façon, au lieu de faire une comparaison soustractive ou un OCLTP (motif ternaire local de couleur opposée) comme dans le deuxième modèle de cette thèse, nous laissons le réseau de neurones apprendre les caractéristiques qui représentent la comparaison des deux paires de couleurs.

Deuxièmement, cette idée de grouper puis d'extraire les caractéristiques au niveau spatial des canaux et entre les canaux de couleur en même temps est appliquée sur les

canaux de cartes de caractéristiques (“feature maps”) à chaque niveau du réseau de neurones jusqu'à ce que les cartes de saillance soient obtenues. Ce processus permet au modèle proposé d'imiter la capacité du système visuel humain à lier inextricablement la couleur et le motif dans la perception des couleurs [1, 100], mais surtout d'obtenir un modèle léger de détection d'objets saillants pouvant être adaptée aux environnements mobiles et aux appareils à ressources limitées.

1.4 Organisation du travail

Après ce chapitre d'introduction, le chapitre deux explique les concepts utilisés dans ce travail comme les motifs binaires locaux, le positionnement multidimensionnel FastMap et les mesures de validation de méthodes.

Dans le troisième chapitre, nous présentons un système de vision par ordinateur basé sur une caméra de profondeur (Microsoft Kinect^{TM}) et un tapis roulant conventionnel pour une détection et une analyse visuelles rapides et fiables des parties du corps du patient qui ont un schéma de mouvement irrégulier, en termes de périodicité, pendant la marche. Le système estime, à partir de chaque séquence vidéo, une carte couleur de saillance montrant les zones de fortes irrégularités de marche, en termes d'énergie de bruit apériodique, de chaque individu en observation. Ainsi, il fournit des informations sur la présence et l'étendue d'une maladie ou de problèmes (orthopédiques, musculaires ou neurologiques) chez le patient. En plus, le système présenté détecte automatiquement les cartes représentant les individus sains et ceux représentant des personnes ayant des problèmes orthopédiques.

Le chapitre quatre présente le modèle que nous proposons dans cette thèse pour la détection d'objets saillants qui est un modèle simple, presque sans paramètres internes. Ce chapitre décrit la stratégie originale et efficace que ce modèle utilise c'est-à-dire intégrer des informations de couleur dans des motifs de texture locaux pour caractériser une micro-texture de couleur grâce au descripteur de textures OCLTP (motifs ternaires locaux d'opposition de couleur). Il décrit aussi comment grâce aux superpixels obtenus par SLICO et à la méthode FastMap, notre modèle génère une carte de saillance robuste

pour une image naturelle.

Le cinquième chapitre présente le modèle de réseau de neurones profond que nous proposons pour un modèle léger de détection d'objets saillants CoSOV1Net. Ce chapitre explique la nouvelle stratégie utilisée pour obtenir ce modèle léger. Ce modèle s'inspire du processus d'opposition de signaux de cônes individuellement et spatialement dans le cortex visuel primaire pour intégrer la couleur dans les formes, étant donné que la couleur et la forme sont inextricablement liées dans la perception de couleurs par le système visuel humain. Ce chapitre explique les détails de ce réseau de neurones de type "encoder – decoder" pour un modèle léger de détection d'objets saillants qui peut être adaptée aux environnements mobiles et aux appareils à ressources limitées.

Le chapitre six donne la conclusion générale de ce travail et quelques perspectives de recherche envisagées.

1.5 Publications

- Ndayikengurukiye, Didier, and Max Mignotte. 2019. "Periodicity Irregularity Map Estimation of Human Gait with a Depth Camera for Pathology Detection" International Journal of Emerging Technology and Advanced Engineering, no. 9 : 139-154.
  https://ijetae.com/files/Volume9Issue5/IJETAE_0519_28.pdf

- Ndayikengurukiye, Didier, and Max Mignotte. 2022. "Salient Object Detection by LTP Texture Characterization on Opposing Color Pairs under SLICO Superpixel Constraint" Journal of Imaging 8, no. 4 : 110.
  https://doi.org/10.3390/jimaging8040110

- Ndayikengurukiye, Didier, and Max Mignotte. 2023. "CoSOV1Net : A Cone- and Spatial-Opponent Primary Visual Cortex-Inspired Neural Network for Lightweight Salient Object Detection" Sensors 23, no. 14 : 6450.
  https://doi.org/10.3390/s23146450

BIBLIOGRAPHIE

[1] Robert Shapley. Physiology of color vision in primates. In *Oxford Research Encyclopedia of Neuroscience*. 2019.

[2] Valerie Nunez, Robert M Shapley, and James Gordon. Cortical double-opponent cells in color perception : perceptual scaling and chromatic visual evoked potentials. *i-Perception*, 9(1) :2041669517752715, 2018.

[3] Norbert Kruger, Peter Janssen, Sinan Kalkan, Markus Lappe, Ales Leonardis, Justus Piater, Antonio J Rodriguez-Sanchez, and Laurenz Wiskott. Deep hierarchies in the primate visual cortex : What can we learn for computer vision ? *IEEE transactions on pattern analysis and machine intelligence*, 35(8) :1847–1871, 2012.

[4] Bevil R Conway. Color vision, cones, and color-coding in the cortex. *The neuroscientist*, 15(3) :274–290, 2009.

[5] Bevil R Conway. Spatial structure of cone inputs to color cells in alert macaque primary visual cortex (v-1). *Journal of Neuroscience*, 21(8) :2768–2783, 2001.

[6] Andrew J Parker. Binocular depth perception and the cerebral cortex. *Nature Reviews Neuroscience*, 8(5) :379–391, 2007.

[7] Bruce G Cumming and Gregory C DeAngelis. The physiology of stereopsis. *Annual review of neuroscience*, 24(1) :203–238, 2001.

[8] F Gonzalez and R Perez. Neural mechanisms underlying stereoscopic vision. *Progress in neurobiology*, 55(3) :191–224, 1998.

[9] Shuyuan Xu, Jun Wang, Wenchi Shou, Tuan Ngo, Abdul-Manan Sadick, and Xiangyu Wang. Computer vision techniques in construction : a critical review. *Archives of Computational Methods in Engineering*, 28 :3383–3397, 2021.

[10] George Bebis, Dwight Egbert, and Mubarak Shah. Review of computer vision education. *IEEE Transactions on education*, 46(1) :2–21, 2003.

[11] Jeffrey M Hausdorff, Galit Yogev, Shmuel Springer, Ely S Simon, and Nir Giladi. Walking is more like catching than tapping : gait in the elderly as a complex cognitive task. *Experimental brain research*, 164 :541–548, 2005.

[12] Jessica Rose and James Gibson Gamble. *Human walking*. Lippincott Williams & Wilkins Philadelphia, 2006.

[13] Vipul Narayan, Shashank Awasthi, Naushen Fatima, Mohammad Faiz, and Swapnita Srivastava. Deep learning approaches for human gait recognition : A review. In *2023 International Conference on Artificial Intelligence and Smart Communication (AISC)*, pages 763–768. IEEE, 2023.

[14] Anil Jain, Ruud Bolle, and Sharath Pankanti. *Biometrics : personal identification in networked society*, volume 479. Springer Science & Business Media, 2006.

[15] David Cunado, Mark S Nixon, and John N Carter. Automatic extraction and description of human gait models for recognition purposes. *Computer Vision and Image Understanding*, 90(1) :1–41, 2003.

[16] Rong Zhang, Christian Vogler, and Dimitris Metaxas. Human gait recognition. In *Conference on Computer Vision and Pattern Recognition Workshop, 2004. CVPRW'04.*, pages 18–18. IEEE, 2004.

[17] Davrondzhon Gafurov, Kirsi Helkala, and Torkjel Søndrol. Biometric gait authentication using accelerometer sensor. *Journal of computers*, 1(7) :51–59, 2006.

[18] Tanmay Tulsidas Verlekar, Luís Ducla Soares, and Paulo Lobato Correia. Gait recognition in the wild using shadow silhouettes. *Image and Vision Computing*, 76 :1–13, 2018.

[19] Yumi Iwashita and Adrian Stoica. Gait recognition using shadow analysis. In *Bio-inspired Learning and Intelligent Systems for Security, 2009. BLISS'09. Symposium on*, pages 26–31. IEEE, 2009.

[20] Yohan Dupuis, Xavier Savatier, and Pascal Vasseur. Feature subset selection applied to model-free gait recognition. *Image and vision computing*, 31(8) :580–591, 2013.

[21] Christina Strohrmann, Holger Harms, Cornelia Kappeler-Setz, and Gerhard Troster. Monitoring kinematic changes with fatigue in running using body-worn sensors. *IEEE Transactions on Information Technology in Biomedicine*, 16(5) :983–990, 2012.

[22] R. Klette and G. Tee. Understanding human motion : A historic review. *3d Imaging for Safety and Security*, 1 :22–40, 2007.

[23] Claudia Ferraris, Gianluca Amprimo, Giulia Masi, Luca Vismara, Riccardo Cremascoli, Serena Sinagra, Giuseppe Pettiti, Alessandro Mauro, and Lorenzo Priano. Evaluation of arm swing features and asymmetry during gait in parkinson's disease using the azure kinect sensor. *Sensors*, 22(16) :6282, 2022.

[24] Christian Bauckhage, John K Tsotsos, and Frank E Bunn. Automatic detection of abnormal gait. *Image and Vision Computing*, 27(1-2) :108–115, 2009.

[25] Rosie Morris, Sue Lord, Jennifer Bunce, David Burn, and Lynn Rochester. Gait and cognition : mapping the global and discrete relationships in ageing and neurodegenerative disease. *Neuroscience & Biobehavioral Reviews*, 64 :326–345, 2016.

[26] Mingjing Yang, Huiru Zheng, Haiying Wang, and Sally McClean. Feature selection and construction for the discrimination of neurodegenerative diseases based on gait analysis. In *2009 3rd International Conference on Pervasive Computing Technologies for Healthcare*, pages 1–7. IEEE, 2009.

[27] Michael W Whittle. Clinical gait analysis : A review. *Human Movement Science*, 15(3) :369–387, 1996.

[28] Adam M Howell, Takehiko Kobayashi, Heather A Hayes, K Bo Foreman, and Stacy JM Bamberg. Kinetic gait analysis using a low-cost insole. *IEEE Transactions on Biomedical Engineering*, 60(12) :3284–3290, 2013.

[29] Motion capture systems from vicon.

[30] Marc Bächlin, Meir Plotnik, Daniel Roggen, Inbal Maidan, Jeffrey M Hausdorff, Nir Giladi, and Gerhard Tröster. Wearable assistant for parkinson's disease patients with the freezing of gait symptom. *IEEE Transactions on Information Technology in Biomedicine*, 14(2) :436–446, 2010.

[31] Miikka Ermes, Juha Parkka, Jani Mantyjarvi, and Ilkka Korhonen. Detection of daily activities and sports with wearable sensors in controlled and uncontrolled conditions. *IEEE Transactions on Information Technology in Biomedicine*, 12(1) :20–26, 2008.

[32] Xin Ma, Haibo Wang, Bingxia Xue, Mingang Zhou, Bing Ji, and Yibin Li. Depth-based human fall detection via shape features and improved extreme learning machine. *IEEE Journal of Biomedical and Health Informatics*, 18(6) :1915–1922, 2014.

[33] Lars Mündermann, Stefano Corazza, and Thomas P Andriacchi. Markerless motion capture for biomechanical applications. In *Human Motion*, pages 377–398. Springer, 2008.

[34] Paulo Lopez-Meyer, George D Fulk, and Edward S Sazonov. Automatic detection of temporal gait parameters in poststroke individuals. *IEEE Transactions on Information Technology in Biomedicine*, 15(4) :594–601, 2011.

[35] Stacy J Morris Bamberg, Ari Y Benbasat, Donna Moxley Scarborough, David E Krebs, and Joseph A Paradiso. Gait analysis using a shoe-integrated wireless sensor system. *IEEE Transactions on Information Technology in Biomedicine*, 12(4) :413–423, 2008.

[36] Rolf Moe-Nilssen. A new method for evaluating motor control in gait under real-life environmental conditions. part 1 : The instrument. *Clinical Biomechanics*, 13(4) :320–327, 1998.

[37] Ryo Takeda, Shigeru Tadano, Masahiro Todoh, Manabu Morikawa, Minoru Nakayasu, and Satoshi Yoshinari. Gait analysis using gravitational acceleration measured by wearable sensors. *Journal of biomechanics*, 42(3) :223–233, 2009.

[38] Dean M Karantonis, Michael R Narayanan, Merryn Mathie, Nigel H Lovell, and Branko G Celler. Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring. *IEEE Transactions on Information Technology in Biomedicine*, 10(1) :156–167, 2006.

[39] MJ Landau, BY Choo, and PA Beling. Simulating kinect infrared and depth images. *IEEE Transactions on cybernetics*, 2015.

[40] Erik E Stone and Marjorie Skubic. Fall detection in homes of older adults using the microsoft kinect. *IEEE Journal of Biomedical and Health Informatics*, 19(1) :290–301, 2015.

[41] Erik Stone and Marjorie Skubic. Evaluation of an inexpensive depth camera for in-home gait assessment. *Journal of Ambient Intelligence and Smart Environments*, 3(4) :349–361, 2011.

[42] Zhengyou Zhang. Microsoft kinect sensor and its effect. *IEEE Multimedia*, 19(2) :4–10, 2012.

[43] Saeid Motiian, Paola Pergami, Keegan Guffey, Corrie A Mancinelli, and Gianfranco Doretto. Automated extraction and validation of children's gait parameters with the kinect. *BioMedical Engineering OnLine*, 14(11), 2015.

[44] Moshe Gabel, Ran Gilad-Bachrach, Erin Renshaw, and Assaf Schuster. Full body gait analysis with kinect. In *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 1964–1967. IEEE, 2012.

[45] Milos Milovanovic, Miroslav Minovic, and Dusan Starcevic. Walking in colors : human gait recognition using kinect and cbir. *IEEE MultiMedia*, 20(4) :28–36, 2013.

[46] Thi-Lan Le, Minh-Quoc Nguyen, et al. Human posture recognition using human skeleton provided by kinect. In *Computing, Management and Telecommunications (ComManTel), 2013 International Conference on*, pages 340–345. IEEE, 2013.

[47] Štěpán Obdržálek, Gregorij Kurillo, Ferda Ofli, Ruzena Bajcsy, Edmund Seto, Holly Jimison, and Michael Pavel. Accuracy and robustness of kinect pose estimation in the context of coaching of elderly population. In *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 1188–1193. IEEE, 2012.

[48] Jungong Han, Ling Shao, Dong Xu, and Jamie Shotton. Enhanced computer vision with microsoft kinect sensor : A review. *IEEE Transactions on cybernetics*, 43(5) :1318–1334, 2013.

[49] Caroline Rougier, Edouard Auvinet, Jean Meunier, Max Mignotte, and Jacques A de Guise. Depth energy image for gait symmetry quantification. In *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE*, pages 5136–5139. IEEE, 2011.

[50] E. Auvinet, F. Multon, and J. Meunier. Lower limb movement asymmetry measurement with a depth camera. In *Engineering in Medicine and Biology Society (EMBC), 2012 Annual International Conference of the IEEE*, pages 6793–6796, Aug 2012.

[51] Guang-Zhong Yang. *Body Sensor Networks*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[52] Derrick Parkhurst, Klinton Law, and Ernst Niebur. Modeling the role of salience in the allocation of overt visual attention. *Vision research*, 42(1) :107–123, 2002.

[53] Laurent Itti. Models of bottom-up attention and saliency. In *Neurobiology of attention*, pages 576–582. Elsevier, 2005.

[54] Laurent Itti and Christof Koch. Computational modelling of visual attention. *Nature reviews neuroscience*, 2(3) :194–203, 2001.

[55] Farhan Baluch and Laurent Itti. Mechanisms of top-down attention. *Trends in neurosciences*, 34(4) :210–224, 2011.

[56] Anne Treisman. Features and objects : The fourteenth bartlett memorial lecture. *The quarterly journal of experimental psychology*, 40(2) :201–237, 1988.

[57] Jeremy M Wolfe, Kyle R Cave, and Susan L Franzel. Guided search : an alternative to the feature integration model for visual search. *Journal of Experimental Psychology : Human perception and performance*, 15(3) :419, 1989.

[58] Christof Koch and Shimon Ullman. Shifts in selective visual attention : towards the underlying neural circuitry. In *Matters of intelligence*, pages 115–141. Springer, 1987.

[59] Qiong Yan, Li Xu, Jianping Shi, and Jiaya Jia. Hierarchical saliency detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1155–1162, 2013.

[60] Ashish Kumar Gupta, Ayan Seal, Pritee Khanna, Enrique Herrera-Viedma, and Ondrej Krejcar. Almnet : Adjacent layer driven multiscale features for salient object detection. *IEEE Transactions on Instrumentation and Measurement*, 70 :1–14, 2021.

[61] Ashish Kumar Gupta, Ayan Seal, Mukesh Prasad, and Pritee Khanna. Salient object detection techniques in computer vision—a survey. *Entropy*, 22(10) :1174, 2020.

[62] Ali Borji, Ming-Ming Cheng, Qibin Hou, Huaizu Jiang, and Jia Li. Salient object detection : A survey. *Computational visual media*, pages 1–34.

[63] Ali Borji and Laurent Itti. State-of-the-art in visual attention modeling. *IEEE transactions on pattern analysis and machine intelligence*, 35(1) :185–207, 2012.

[64] Laurent Itti. Automatic foveation for video compression using a neurobiological model of visual attention. *IEEE transactions on image processing*, 13(10) :1304–1318, 2004.

[65] Jinjiang Li, Xiaomei Feng, and Hui Fan. Saliency-based image correction for colorblind patients. *Computational Visual Media*, 6(2) :169–189, 2020.

[66] Nicolò Oreste Pinciroli Vago, Federico Milani, Piero Fraternali, and Ricardo da Silva Torres. Comparing cam algorithms for the identification of salient image features in iconography artwork analysis. *Journal of Imaging, MDPI*, 7(7) :106, 2021.

[67] Yuan Gao, Miaojing Shi, Dacheng Tao, and Chao Xu. Database saliency for fast image retrieval. *IEEE Transactions on Multimedia*, 17(3) :359–369, 2015.

[68] Rik Pieters and Michel Wedel. Attention capture and transfer in advertising : Brand, pictorial, and text-size effects. *Journal of Marketing*, 68(2) :36–50, 2004.

[69] Lai-Kuan Wong and Kok-Lim Low. Saliency-enhanced image aesthetics class prediction. In *2009 16th IEEE International Conference on Image Processing (ICIP)*, pages 997–1000. IEEE, 2009.

[70] Hantao Liu and Ingrid Heynderickx. Studying the added value of visual attention in objective image quality metrics based on eye movement data. In *2009 16th IEEE international conference on image processing (ICIP)*, pages 3097–3100. IEEE, 2009.

[71] Li-Qun Chen, Xing Xie, Xin Fan, Wei-Ying Ma, Hong-Jiang Zhang, and He-Qin Zhou. A visual attention model for adapting images on small displays. *Multimedia systems*, 9(4) :353–364, 2003.

[72] Tao Chen, Ming-Ming Cheng, Ping Tan, Ariel Shamir, and Shi-Min Hu. Sketch2photo : Internet image montage. *ACM transactions on graphics (TOG)*, 28(5) :1–10, 2009.

[73] Hua Huang, Lei Zhang, and Hong-Chao Zhang. Arcimboldo-like collage using internet images. In *Proceedings of the 2011 SIGGRAPH Asia Conference*, pages 1–8, 2011.

[74] Arnold WM Smeulders, Dung M Chu, Rita Cucchiara, Simone Calderara, Afshin Dehghan, and Mubarak Shah. Visual tracking : An experimental survey. *IEEE transactions on pattern analysis and machine intelligence*, 36(7) :1442–1468, 2013.

[75] Mark Haidekker. *Advanced biomedical image analysis*. John Wiley & Sons, 2011.

[76] H. Knutsson and G Granlund. Texture analysis using two-dimensional quadrature filters. In *IEEE Comput. Soc. Workshop on Computer Architecture for Pattern Analysis and Image Database Management*, pages 206–213, 1983.

[77] Timo Ojala, Matti Pietikäinen, and David Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern recognition*, 29(1) :51–59, 1996.

[78] Kenneth I Laws. *Textured image segmentation*. PhD thesis, University of Southern California Los Angeles Image Processing INST, 1980.

[79] Matti Pietikäinen, Abdenour Hadid, Guoying Zhao, and Timo Ahonen. *Computer vision using local binary patterns*, volume 40. Springer Science & Business Media, 2011.

[80] Xiaoyang Tan and Bill Triggs. Enhanced local texture feature sets for face recognition under difficult lighting conditions. *IEEE transactions on image processing*, 19(6) :1635–1650, 2010.

[81] Timo Ahonen, Abdenour Hadid, and Matti Pietikäinen. Face recognition with local binary patterns. In *European conference on computer vision*, pages 469–481. Springer, 2004.

[82] Ran Margolin, Ayellet Tal, and Lihi Zelnik-Manor. What makes a patch distinct ? In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1139–1146, 2013.

[83] Qing Zhang, Jiajun Lin, Yanyun Tao, Wenju Li, and Yanjiao Shi. Salient object detection via color and texture cues. *Neurocomputing*, 243 :35–48, 2017.

[84] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. Slic superpixels compared to state-of-the-art superpixel

methods. *IEEE transactions on pattern analysis and machine intelligence*, 34(11) :2274–2282, 2012.

[85] Ashish Kumar Gupta, Ayan Seal, Pritee Khanna, Ondrej Krejcar, and Anis Yazidi. Awks : adaptive, weighted k-means-based superpixels for improved saliency detection. *Pattern Analysis and Applications*, 24(2) :625–639, 2021.

[86] Christos Faloutsos and King-Ip Lin. *FastMap : A fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets*, volume 24. ACM, 1995.

[87] Alice Porebski, Nicolas Vandenbroucke, and Ludovic Macaire. Haralick feature extraction from lbp images for color texture classification. In *2008 First Workshops on Image Processing Theory, Tools and Applications*, pages 1–8. IEEE, 2008.

[88] Wenguan Wang, Qiuxia Lai, Huazhu Fu, Jianbing Shen, Haibin Ling, and Ruigang Yang. Salient object detection in the deep learning era : An in-depth survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(6) :3239–3259, 2021.

[89] Shang-Hua Gao, Yong-Qiang Tan, Ming-Ming Cheng, Chengze Lu, Yunpeng Chen, and Shuicheng Yan. Highly efficient salient object detection with 100k parameters. In *Computer Vision–ECCV 2020 : 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VI*, pages 702–721. Springer, 2020.

[90] Yun Liu, Xin-Yu Zhang, Jia-Wang Bian, Le Zhang, and Ming-Ming Cheng. Samnet : Stereoscopically attentive multi-scale network for lightweight salient object detection. *IEEE Transactions on Image Processing*, 30 :3804–3814, 2021.

[91] Xuebin Qin, Zichen Zhang, Chenyang Huang, Masood Dehghan, Osmar R Zaiane, and Martin Jagersand. U2-net : Going deeper with nested u-structure for salient object detection. *Pattern recognition*, 106 :107404, 2020.

[92] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net : Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015 : 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015.

[93] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets : Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv :1704.04861*, 2017.

[94] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2 : Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.

[95] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet : An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6848–6856, 2018.

[96] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2 : Practical guidelines for efficient cnn architecture design. In *Proceedings of the European conference on computer vision (ECCV)*, pages 116–131, 2018.

[97] Yun Liu, Yu-Chao Gu, Xin-Yu Zhang, Weiwei Wang, and Ming-Ming Cheng. Lightweight salient object detection via hierarchical visual perception learning. *IEEE Transactions on Cybernetics*, 51(9) :4439–4449, 2020.

[98] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[99] François Chollet. Xception : Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017.

[100] Robert Shapley and Michael J Hawken. Color in the cortex : single-and double-opponent cells. *Vision research*, 51(7) :701–717, 2011.

[101] Robert William Gainer Hunt and Michael R Pointer. *Measuring colour*. John Wiley & Sons, 2011.

[102] Stephen Engel, Xuemei Zhang, and Brian Wandell. Colour tuning in human visual cortex measured with functional magnetic resonance imaging. *Nature*, 388(6637) :68–71, 1997.

CHAPITRE 2

DÉFINITION DES CONCEPTS UTILISÉS

2.1 LBP : Motifs binaires locaux

Le motif binaire local (LBP) est un opérateur simple mais très efficace de description de la texture d'une image. Cet opérateur étiquette les pixels d'une image en établissant un seuil pour le voisinage de chaque pixel. L'étiquette est alors le nombre binaire constitué par les résultats du seuillage.

L'opérateur LBP possède une importante propriété utile pour les applications du monde réel : l'invariance vis-à-vis des variations monotones du niveau de gris causées, par exemple, par des variations d'éclairage. Sa simplicité de calcul lui permet d'analyser des images dans des conditions difficiles en temps réel. Le LBP peut également être facilement adapté à différents types de problèmes et utilisé avec d'autres descripteurs d'image, ce qui démontre sa flexibilité [1].

Introduits en 1996 [2], les motifs binaires locaux (LBP) pour un voisinage de pixels 3×3 (voir Figure 2.1), ont ensuite été généralisés par Ojala *et al.* pour un voisinage quelconque (voir Figure 2.2).

240	245	248
230	235	238
239	233	232

Voisinage original

5	10	13
-5		3
4	-2	-3

Après seuillage

1	1	1
0		1
1	0	0

Motif : 01001111

1	2	4
128		8
64	32	16

Poids (LBP code= 79)

FIGURE 2.1 : Opérations pour l'obtention du code LBP par la méthode LBP originale (classique) ; le seuil est le niveau de gris du pixel du centre.

P=8 pixels voisins et le rayon R=2 P=8 pixels voisins et le rayon R=4

FIGURE 2.2 : Exemple de P pixels voisins sur un rayon R pour LBP avec voisinage quelconque.

Ainsi, selon ces auteurs, la texture T dans un voisinage local d'un pixel d'une image de texture monochrome est la distribution jointe des niveaux de gris de ce pixel et de ses P ($P > 1$) voisins dans l'image [3], [1] :

$$T = t(g_c, g_0, g_1, \ldots, g_{P-1}) \tag{2.1}$$

g_c : valeur du niveau de gris du pixel au centre du voisinage local, soit (x_c, y_c) ses coordonnées. g_p est le niveau de gris du pixel de coordonnées (x_p, y_p) telles que :
$x_p = x_c + R\cos(\frac{2\pi p}{P})$ et $y_p = y_c - R\sin(\frac{2\pi p}{P})$ avec $(p = 0, \ldots, P-1)$.

Comme cela est montré par Ojala *et al.* [4], la distribution précédente peut aussi être représentée par :

$$T = t(g_c, g_0 - g_c, g_1 - g_c, \ldots, g_{P-1} - g_c) \tag{2.2}$$

En supposant l'indépendance des $g_p - g_c$ par rapport à g_c [4], la distribution de cette texture est approximée par :

$$T \approx t(s(g_0 - g_c), s(g_1 - g_c), \ldots, s(g_{P-1} - g_c)) \tag{2.3}$$

Avec :

$$s(z) = \begin{cases} 1 & \text{si } z \geq 0 \\ 0 & \text{si } z < 0 \end{cases}$$

Chaque pixel reçoit alors un nombre, LBP code, comme étiquette calculée en fonction de ses voisins :

$$LBP_{P,R}(x_c, y_c) = \sum_{p=0}^{P-1} s(g_p - g_c) 2^p \tag{2.4}$$

La caractérisation de la texture de l'image est approximée par une distribution discrète de LBP codes de 2^P bins telle que :

$$T \approx t(LBP_{P,R}(x_c, y_c)) \tag{2.5}$$

$x_c \in \{0, \ldots, M-1\}$ et $y_c \in \{0, \ldots, N-1\}$ pour une image $M \times N$.

Ainsi, après l'obtention des codes LBP, les occurrences des codes LBP dans une image sont rassemblées dans un histogramme (voir les images dans la Figure 2.3).

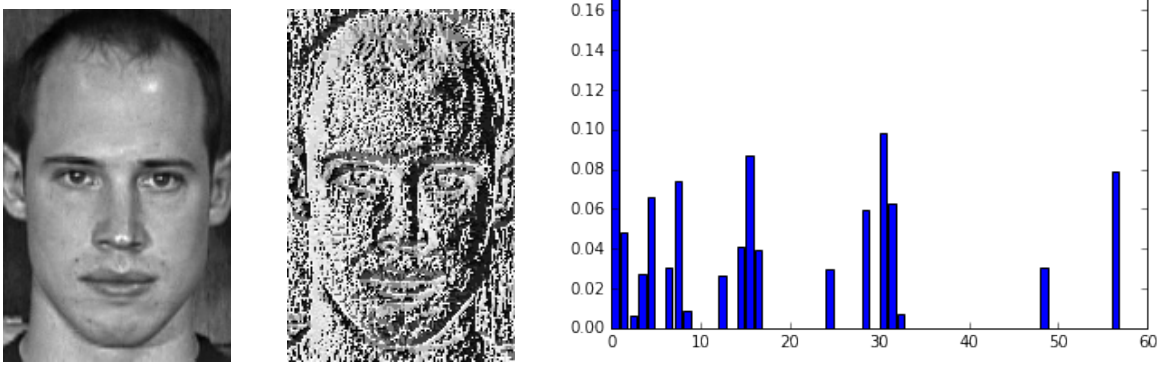


image monochrome image de code LBP Histogramme (caractéristiques)

FIGURE 2.3 : Exemple de codes et de caractéristiques LBP pour une image monochrome (l'image monochrome provient de la base de données "Georgia Tech face database" [5]).

Les motifs binaires locaux (LBP) ont connu plusieurs applications [6–9] et possèdent plusieurs variantes [10–19]. Parmi celles-ci, il y a le "OCLBP : Opponent Color LBP" [1, 20]. Celle-ci généralise le LBP classique, qui normalement s'applique sur des textures monochromes, aux motifs en couleur. Ainsi, au lieu d'un seul code LBP, un pixel obtient un code pour chaque combinaison de deux couleurs (R-R : Rouge-Rouge, R-G : Rouge-Vert, R-B : Rouge-Bleu, G-R : Vert-Rouge, G-G : Vert-Vert, G-B : Vert-Bleu, B-R : Bleu-Rouge, B-G : Bleu-Vert, B-B : Bleu-Bleu), soit 9 codes. Le pixel central est dans la première couleur de la combinaison et les voisins dans la deuxième. Les histogrammes obtenus sont concaténés pour obtenir le vecteur de caractéristiques de la texture.

2.2 FastMap : Positionnement multidimensionnel

Le positionnement multidimensionnel FastMap [21] est une très bonne approximation, en complexité linéaire, de l'algorithme de positionnement multidimensionnel MDS ("Multi-Dimensional Scaling") qui, lui-même est non seulement un algorithme de réduction de dimensions, mais aussi une méthode de cartographie ("mapping") qui a la propriété très intéressante de projeter des objets considérés comme des points dans un espace de n-dimensions dans un espace de dimension k ($n \gg k$) défini tout en préservant les distances (Euclidiennes) existant entre les paires d'objets. Dans les sections qui suivent, nous allons décrire le MDS classique, le PCA ("Principal Components Analysis") et le FastMap. Nous expliquons les raisons qui nous ont poussé de choisir le FastMap plutôt que les deux autres bien qu'ils soient les plus populaires. Ces raisons sont :

- la complexité quadratique pour le MDS classique alors que FastMap a une complexité linéaire,

- le fait que PCA cherche une transformation linéaire qui va projeter les éléments d'un espace de dimension n dans un espace de dimension k ($n \gg k$). PCA cherche par conséquent les dépendances linéaires entre les caractéristiques ("features") des points de l'espace de dimension n alors que FastMap fait cette projection à l'aide

d'une transformation non-linéaire. Ainsi, FastMap génère moins de perte d'information pendant le processus de projection car il tient en considération les dépendances non linéaires entre les caractéristiques ("features") des objets de l'espace de dimension n.

2.2.1 MDS de base

Il existe plusieurs variantes de MDS ("Multi-Dimensional Scaling"). La variante basique de MDS, intuitivement, traite chaque distance entre paires de points correspondant aux projections des objets dans l'espace de dimension k comme un ressort entre les deux points. L'algorithme tente alors de réorganiser les positions des points dans l'espace de k dimensions en minimisant la contrainte ("stress") exercée par les ressorts.

$$stress = \sqrt{\frac{\sum_{i,j}(\hat{d}_{ij} - d_{ij})^2}{\sum_{i,j} d_{ij}^2}} \tag{2.6}$$

Avec d_{ij} : la dissimilarité ou similarité entre les objets i et j

$\hat{d}_{ij}$: la distance euclidienne entre les projections des objets i et j dans l'espace de dimension k.

Le "*stress*" est l'erreur relative moyenne des distances entre les paires des projections des objets dans l'espace de dimension k par rapport aux distances entre les paires des objets leur correspondant dans l'espace d'origine (de dimension n).

L'inconvénient de l'algorithme MDS est qu'il est de complexité quadratique ($O(N^2)$, N étant le nombre d'objets). Il n'est donc pas pratique pour une grande quantité d'objets. D'où la nécessité d'un algorithme de plus faible complexité.

2.2.2 PCA : analyse en composantes principales

Un autre outils largement utilisé pour la réduction de dimensionnalité, l'extraction de caractéristiques, la compression de données et la visualisation de données est l'analyse en composantes principales ("Principal Components Analysis", PCA) [22]. La PCA

peut être définie comme la projection orthogonale linéaire des données d'un espace de dimension n sur un espace de dimension k avec ($n > k$) appelé sous-espace principal tout en maximisant la variance des données projetées. Elle peut aussi être définie comme une projection linéaire qui minimise le coût de projection moyen (c'est-à-dire la moyenne de la distance au carré entre les données et leurs projections) [22]. Cette technique est utilisée dans plusieurs domaines sous différentes appellations comme la Transformation Karhunen-Loeve ("Karhunen-Loeve Transform", KLT) en traitement d'images [23]. Par cette dernière technique, on calcule les vecteurs propres de la matrice de covariance des données et les valeurs propres qui leur sont correspondantes. Ces vecteurs propres sont triés par ordre décroissant de leurs valeurs propres. Ensuite chaque donnée (vecteur de donnée) dans l'espace de dimension n est approximée par ses projections sur les k premiers vecteurs propres. L'opération est étroitement liée à la décomposition en valeurs singulières(" SVD : Singular Value Decomposition") [21].

Ainsi, soient N le nombre d'objets dans l'espace de dimension n et X la représentation de ces données centrées. La matrice de covariance sans biais est donnée par

$$S = \frac{1}{N-1} X^T X \tag{2.7}$$

En utilisant la décomposition en valeurs singulières SVD, la matrice X est décomposée de la façon suivante : $X = U\Sigma V^T$ avec U une matrice orthogonale $N \times N$, V une matrice orthogonale $n \times n$ et Σ une matrice $N \times n$ de la forme :

$$\begin{bmatrix} D & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}$$

D étant une matrice $r \times r$ avec r : le rang de matrice Σ, les $N - r$ lignes et les $n - r$ colonnes sont composées de zéros.

Les colonnes de la matrice U sont appelées les vecteurs singuliers de gauche de la matrice X alors que les colonnes de la matrice V sont les vecteurs singuliers de droite de la matrice X. Les vecteurs singuliers de gauche de X (les colonnes de la matrice U) sont les vecteurs propres de XX^T. Les vecteurs singuliers de droite de la matrice X sont les vecteurs propres de X^TX. Les valeurs singulières non nulles de X sont les racines carrées des valeurs propres de X^TX mais aussi de XX^T [24].

Nous sommes intéressés par les vecteurs propres de X^TX, donc les colonnes de la matrice V car les principales composantes de la matrice X sont les vecteurs propres de X^TX. La matrice de covariance S de l'équation 2.7 peut alors être exprimée ainsi :

$$S = \frac{1}{N-1} X^T X \tag{2.8}$$

$$= \frac{1}{N-1} (U\Sigma V^T)^T U\Sigma V^T \tag{2.9}$$

$$= \frac{1}{N-1} V\Sigma U^T U\Sigma V^T \tag{2.10}$$

($U^TU = I$ car U est une matrice orthogonale)

$$= \frac{1}{N-1} V\Sigma^2 V^T \tag{2.11}$$

Donc

$$X^TX = V\Sigma^2 V^T \tag{2.12}$$

L'analyse en composantes principales PCA obtient la projection de chaque élément $\mathbf{x}$ de l'espace dans un nouveau repère en un élément $\mathbf{z}$ par une transformation linéaire telle que :

$$\mathbf{z} = V^T\mathbf{x} \tag{2.13}$$

Nous pouvons constater que la matrice de covariance dans ce nouveau repère est une matrice diagonale.

En effet, soit la matrice Z de tous les objets dans le nouveau repère, la matrice de covariance leur correspondant est S' telle que :

$$S' = \frac{1}{N-1} Z^T Z \tag{2.14}$$
$$= \frac{1}{N-1} (XV)^T (XV) \tag{2.15}$$
$$= \frac{1}{N-1} V^T X^T X V \tag{2.16}$$
$$= \frac{1}{N-1} V^T V \Sigma^2 V^T V \tag{2.17}$$

(car $X^T X = V \Sigma^2 V^T$ voir l'équation 2.12)

$$= \frac{1}{N-1} \Sigma^2 \tag{2.18}$$

(car V est une matrice orthogonale $V^T V = I$)

$$\tag{2.19}$$

Donc

$$S' == \frac{1}{N-1} \Sigma^2 \tag{2.20}$$

En triant les vecteurs propres obtenus pour $X^T X$ par ordre décroissant de leurs valeurs propres et en séléctionnant les k premiers vecteurs propres, nous avons les vecteurs $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k$

avec

$$\mathbf{v}_i = \begin{bmatrix} c_{i,1}, & c_{i,2}, & \cdots, & c_{i,n} \end{bmatrix}^T$$

$(i = 1, 2, \cdots, k)$

Soient

$$\mathbf{x} = \begin{bmatrix} x_1, & x_2, & \cdots, & x_n \end{bmatrix}^T$$

un vecteur dans l'espace d'origine (de dimension n) et

$$\mathbf{y} = \begin{bmatrix} y_1, & y_2, & \cdots, & y_k \end{bmatrix}^T$$

le vecteur projection de $\mathbf{x}$ dans l'espace de dimension k (sous-espace principal),
Nous avons alors pour $i = 1, 2, \cdots, k$

$$y_i = \mathbf{v}_i^T \mathbf{x}$$

$$y_i = c_{i,1}x_1 + c_{i,2}x_2 + \cdots + c_{i,n}x_n$$

L'élément y_i $(i = 1, 2, \cdots, k)$ du vecteur $\mathbf{y} = \begin{bmatrix} y_1, & y_2, & \cdots, & y_k \end{bmatrix}^T$ projection de la donnée $\mathbf{x} = \begin{bmatrix} x_1, & x_2, & \cdots, & x_n \end{bmatrix}^T$ de l'espace de dimension n dans le sous-espace de dimension k est une **combinaison linéaire** des éléments $x_1, x_2, \cdots, x_n$ du vecteur $\mathbf{x}$, les éléments du vecteur propre $\mathbf{v}_i$ étant les poids de la **combinaison linéaire**. Ainsi, PCA dans sa réduction de la dimension des données produit des éléments qui n'ont pas de corrélation linéaire entre eux (voir l'équation 2.20). Le problème est que la distribution des données dans l'ensemble des données est généralement **non-linéaire**. Dans ce cas la réduction de la dimensionnalité par des modèle comme PCA, ICA ("Independent Component Analysis"), LDA ("Linear Discriminant Analysis") ou d'autres modèles qui utilisent des transformations linéaires des caractéristiques ("features") pour y arriver, ne sont pas les mieux adaptées pour tenir compte des dépendances complexes dans les caractéristiques ("features") comme la **non-linéarité**. D'où la nécessité de transformations qui tiennent compte de la **non-linéarité** dans les données dans le processus de réduction de dimensionalité.

2.2.3 FastMap

Le FastMap est un outils de réduction de dimensionnalité de complexité linéaire et qui tient compte de la non-linéarité de données. Ainsi, l'utilisation de l'algorithme FastMap est un bon choix du fait qu'il n'a pas les désavantages des méthodes MDS basique et PCA cités ci-haut (voir leurs sections respectives 2.2.1 et 2.2.2).

Le FastMap est un algorithme qui au départ, était destiné à fournir un outil permettant de retrouver des objets similaires à un objet donné, de trouver des paires d'objets les plus similaires et de visualiser des distributions d'objets dans un espace désiré pour pouvoir déceler les principales structures dans les données, une fois que la fonction de similarité ou de dissimilarité est déterminée. Cet outil reste efficace même pour de grandes collections d'ensembles de données contrairement au positionnement multidimensionnel classique (MDS classique).

Cet algorithme fait correspondre aux objets d'une certaine dimension des points d'un espace de dimension k tout en préservant les distances (les dissimilarités ou les similarités) entre les paires d'objets.

Cette représentation d'objets d'un espace de grande dimension n vers un espace de dimension plus petite (dimension 1 ou 2 ou 3) permet la visualisation des structures des distributions dans les données ou l'accélération du temps de recherche des requêtes [21].

Comme *Faloutsos et Lin* [21] le décrivent, le problème résolu par FastMap peut être représenter de deux façons. Premièrement, à partir de N objets avec des informations sur la distance entre les objets (par exemple une matrice $N \times N$ de distance entre les paires d'objets ou la fonction de distance D(objet i, objet j), $i, j = 1, \ldots, N$), nous cherchons N points correspondant à ces objets dans un espace de k dimensions tout en préservant les distances entre les paires d'objets. La distance D() est positive, symétrique et vérifie l'inégalité triangulaire. La distance dans l'espace de dimension k peut être n'importe quelle métrique L_p. Deuxièment l'algorithme FastMap peut aussi être utilisé dans la réduction de la dimensionalité tout en préservant les distances entre les paires de vecteurs. Ce qui revient à chercher, étant donnés N vecteurs possédant n caractéristiques chacun, N vecteurs dans un espace de dimension k avec ($n \gg k$) tout en préservant les distances

entre les paires de vecteurs. La distance dans les deux espaces peut être n'importe quelle métrique L_p.

Dans toutes les deux formes du problème, FastMap cherche à trouver les coordonnées des points représentant les objets donnés dans l'espace de dimension k désiré dont les axes de coordonnées sont mutuellement orthogonaux. Le premier axe de coordonnées est la droite qui relie les objets appelés pivots. Les pivots sont choisis de telle sorte que la distance séparant ces derniers est maximale. Ainsi, pour obtenir ces pivots, l'algorithme suit les étapes ci-dessous :

- choisir arbitrairement un objet comme deuxième pivot, soit l'objet O_b
- choisir comme premier pivot l'objet le plus éloigné de O_b selon la distance en vigueur (la distance pour le premier axe est la distance donnée mais pour les autres axes, son expression est une fonction de la précédente selon l'équation 2.27), soit l'objet O_a.
- remplacer le deuxième pivot par l'objet le plus éloigné de O_a, soit l'objet O_b.
- retourner les objets O_a et O_b comme pivots.

Tous les points représentant les objets sont alors projetés orthogonalement sur cet axe, qui est la droite reliant les pivots obtenus (voir l'illustration de la Figure 2.4).

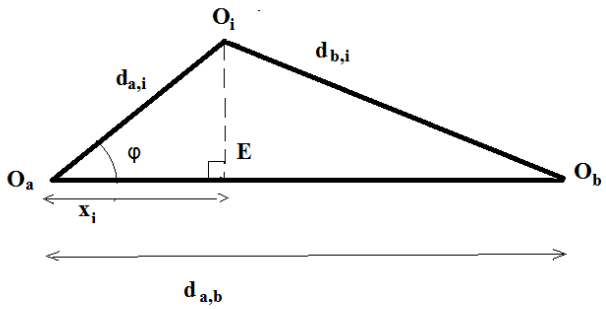


FIGURE 2.4 : Projection orthogonale de l'objet O_i en E sur l'axe reliant les objets pivots O_a et O_b pour obtenir la coordonnée x_i grâce à la loi des cosinus.

Soient les objets pivots O_a et O_b, l'objet O_i $(i = 1, \cdots, N)$; avec φ : l'angle formé par le côté $O_i\,O_a$ et le côté $O_a\,O_b$ du triangle $O_a\,O_i\,O_b$, $d_{a,i} = D(O_a, O_i)$: distance entre l'objet pivot O_a et l'objet O_i, $d_{b,i} = D(O_b, O_i)$: distance entre l'objet pivot O_b et l'objet O_i et $d_{a,b} = D(O_a, O_b)$: distance entre les objets pivots O_a et l'objet O_b.

En appliquant la loi des cosinus sur le triangle $O_a\,O_i\,O_b$, nous avons

$$d_{b,i}^2 = d_{a,i}^2 + d_{a,b}^2 - 2d_{a,i}d_{a,b}\cos\varphi \tag{2.21}$$

E étant la projection orthogonale de O_i, en considérant le triangle rectangle $O_i\,O_a\,E$, nous avons :

$$\cos\varphi = \frac{x_i}{d_{a,i}} \tag{2.22}$$

Les équations 2.21 et 2.22 donnent

$$d_{b,i}^2 = d_{a,i}^2 + d_{a,b}^2 - 2x_i d_{a,b}$$

La coordonnée de l'objet O_i (avec $i = 1, \cdots, N$) sur l'axe reliant les objets pivots O_a

et O_b pour la distance $D(O_i, O_j) = d_{O_i,O_j}$ est alors donnée par x_i :

$$x_i = \frac{d_{a,i}^2 + d_{a,b}^2 - d_{b,i}^2}{2d_{a,b}} \tag{2.23}$$

Il faut chercher ensuite les autres coordonnées sur les $k-1$ axes mutuellement orthogonaux et orthogonaux à l'axe précédent.

Pour y arriver, tous les objets (les objets sont encore considérés comme des points dans leur espace original de dimension n par ($n \gg k$)) sont projetés sur un hyper plan H de dimension $n-1$ orthogonal à la droite reliant les pivots O_a et O_b. Nous appelons O_i' la projection de l'objet O_i dans l'hyper plan H ($i = 1, \cdots, N$). Nous avons besoin d'obtenir les coordonnées des objets sur un axe orthogonal à l'axe précédent. Ceci est fait en projetant orthogonalement ces objets sur ce nouvel axe qui est dans l'hyper plan H. Cette idée peut être matérialisée par une fonction de la distance $D'()$ qui exprime la projection des objets sur un axe perpendiculaire à l'axe précédent. La distance entre deux projections est obtenue de façon suivante pour k=2 : Soient E la projection de l'objet O_i sur la droite reliant les objets pivots O_a et O_b, D la projection de l'objet O_j sur la droite reliant les mêmes pivots et C : l'intersection de la droite passant par D et parallèle à la droite $O_i\ E$ et la droite $O_i\ O_i'$ (voir la Figure 2.5).

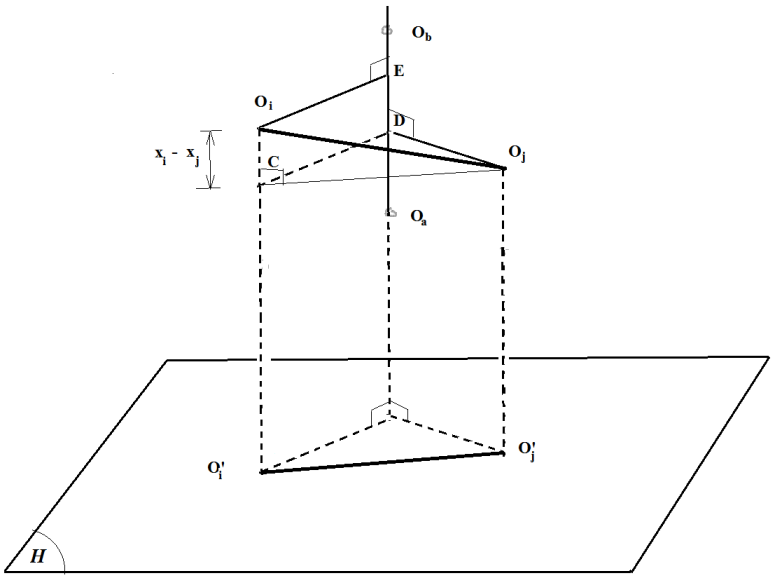


FIGURE 2.5 : Projection orthogonale des objets dans l'hyper-plan H perpendiculaire à la droite reliant les objets pivot O_a et O_b suivant la direction de cette dernière droite.

Le théorème de Pythagore appliqué au triangle $O_i\,C\,O_j$ rectangle en C donne l'équation :

$$(CO_j)^2 = (O_iO_j)^2 - (O_iC)^2 \tag{2.24}$$

Or

$$(CO_j) = (O'_i, O'_j) \tag{2.25}$$

De plus

$$(O_iC) = (ED) = ||x_i - x_j||_2 \tag{2.26}$$

L'expression (AB) étant la longueur du segment AB.

Les équations 2.24, 2.25 et 2.26 donnent :

$$(D'(O'_i, O'_j))^2 = (D(O_i, O_j))^2 - (x_i - x_j)^2 \tag{2.27}$$

De plus,

$$D'(O'_i, O'_j) = d'_{O'_i, O'_j}$$

Ayant l'expression de la fonction distance $D'()$ (voir l'équation 2.27), le problème de trouver l'axe suivant devient identique à celui de trouver le premier axe en utilisant les objets projetés dans l'hyper plan H de dimension $n-1$ et $k-1$ étant le nombre d'axes. C'est-à-dire que nous recherchons les pivots avec cette nouvelle fonction de distance et la coordonnée x'_i de l'objet O_i ($i = 1, \cdots, N$) suivant ce nouvel axe telle que :

$$x'_i = \frac{d'^2_{a',i} + d'^2_{a',b'} - d'^2_{b',i}}{2d'_{a',b'}} \tag{2.28}$$

$O_{a'}$ et $O_{b'}$ étant les nouveaux pivots selon la nouvelle distance $D'()$ définie.

Les coordonnées de l'objet O_i ($i = 1, \cdots, N$) dans l'espace de dimension k sont alors obtenues récursivement jusqu'à la $k^{\text{ième}}$ coordonnée en commençant par la première coordonnée x_i pour chaque objet O_i grâce à la distance en cours $D()$ et en définissant ensuite la nouvelle distance D'.

La complexité de FastMap est $O(kN)$. Aussi, les coordonnées dans l'espace de dimension k d'un objet ne sont pas des combinaisons linéaires de celles de l'espace d'origine comme pour PCA ("Principal Component Analysis"). De plus, FastMap préserve les distances entre les objets.

Cet outil algorithmique puissant nous permet, dans ce travail, d'exprimer et tester, pour chaque caractéristique de texture individuelle, la dissimilarité texturale existant entre deux pixels, en la résumant en une image (en échelle de gris) dans lequel deux pixels ont deux valeurs (de niveaux de gris) d'autant plus différentes que la dissimilarité texturale (entre ces deux pixels) est grande, au sens de la caractéristique texturale choisie. Les objets sont donc, dans le cas de cette étude, des pixels et chaque pixel (chaque

objet) est caractérisé par un histogramme (vecteur caractéristique) de dimension n.

L'algorithme de positionnement multidimensionnel FastMap est plus rapide que le MDS classique car il a une complexité linéaire alors que le MDS classique a une complexité quadratique. Il est ainsi mieux adapté pour une grande quantité d'objets [21]. Dans le cas de ce travail, par exemple pour une image de résolution 400×400, nous avons $400 \times 400 = 160\,000$ pixels, donc 160 000 objets pour cette image. Il a été utilisé dans l'analyse de la marche humaine [25], la segmentation [26, 27], la détection d'axes de symétrie dans les images [28], la reconnaissance des actions humaines [29], dans la détection de changements dans les images provenant de plusieurs capteurs [30], etc.

2.3 Mesures de performance de méthodes

Nous avons utilisé pour mesures de validation les mesures reconnues et les plus utilisées pour l'évaluation des modèles de détection des objets saillants [31–34]. Nous avons choisi d'une part les mesures qui ne nécessitent pas de transformer l'image probabiliste obtenue en masque binaire (selon un certain seuil, la valeur de chaque pixel est soit 0 soit 255) pour pouvoir le comparer au vérité de terrain ("Ground truth") et d'autre part les mesures qui nécessitent cette transformation pour faire ladite comparaison. Parmi ces dernières mesures, nous avons la F_β mesure, la courbe précision-Rappel (PR : "precision-recall"). Dans le groupe de mesures ne nécessitant pas de masque binaire, nous avons choisi la F_β^w mesure et la mesure de la moyenne de l'erreur absolue : MAE ("Mean Absolute Error").

La plupart de ces mesures sont fonction soit de la précision et du rappel(ou Sensibilité), soit du taux de vrais positifs ou Sensibilité (TPR : "True Positive Rate") et du taux de faux positifs (FPR : "False Positive Rate") ou (1 - Spécificité).

La comparaison de la sortie de notre modèle avec la vérité de terrain ("Ground truth") peut être considérée comme un problème de décision binaire dans la mesure où, après avoir obtenu un masque binaire de la sortie du modèle en choisissant un seuil, chaque pixel va être étiqueté de "positif" (s'il est blanc) ou "négatif" (s'il est noir). Si nous prenons M comme le masque binaire obtenu pour une image probabiliste et G la vérité

de terrain (“Ground truth”) correspondante, nous aurons la matrice de confusion et les mesures de performance suivantes :

TABLE 2.1 : Matrice de confusion pour la vérité de terrain G et le masque binaire M.

		Vérité de terrain : G	
		Blancs	Noirs
Masque binaire : M	Blancs	Vrais Positifs	Faux Positifs
	Noirs	Faux Négatifs	Vrais Négatifs

$$\text{Précision} = \frac{\text{Vrais Positifs}}{\text{Vrais Positifs} + \text{Faux Positifs}} \tag{2.29}$$

$$\text{Rappel ou Sensibilité} = \frac{\text{Vrais Positifs}}{\text{Vrais Positifs} + \text{Faux Négatifs}} \tag{2.30}$$

ou

$$\text{Précision} = \frac{|M \cap G|}{|M|} \tag{2.31}$$

$$\text{Rappel ou Sensibilité} = \frac{|M \cap G|}{|G|} \tag{2.32}$$

$$\text{TPR ou Sensibilité} = \frac{\text{Vrais Positifs}}{\text{Vrais Positifs} + \text{Faux Négatifs}} \tag{2.33}$$

$$\text{FPR} = \frac{\text{Faux Positifs}}{\text{Faux Positifs} + \text{Vrais Négatifs}} \tag{2.34}$$

$$\text{Spécificité} = 1 - \text{FPR} \tag{2.35}$$

ou

$$\text{TPR ou Sensibilité} = \frac{|M \cap G|}{|G|} \tag{2.36}$$

$$\text{FPR} = \frac{|M \cap \overline{G}|}{|\overline{G}|} \tag{2.37}$$

$$\text{Spécificité} = 1 - \text{FPR} \tag{2.38}$$

$\overline{G}$: est le complément de G

|.| : le nombre de pixels dont les valeurs ne sont pas des zéros

Les taux de vrais positifs (sensibilité) et les taux de faux positifs (1 - spécificité) sont les plus importantes proportions des quatres proportions obtenues dans la matrice de confusion 2.1 car les deux autres peuvent être obtenues à partir des premières étant leurs compléments respectifs [35].

2.3.1 La F_β mesure

La F_β-mesure (F_β) est la moyenne harmonique pondérée (avec un poids non-négatif) de la précision et du rappel :

$$F_\beta = \frac{1+\beta^2}{\frac{\beta^2}{\text{Rappel}} + \frac{1}{\text{Précision}}} \tag{2.39}$$

$$\Leftrightarrow F_\beta = \frac{(1+\beta^2) \times \text{Précision} \times \text{Rappel}}{\beta^2 \times \text{Précision} + \text{Rappel}} \tag{2.40}$$

La F_β-mesure est une mesure de performance globale [36]. Pour donner moins d'importance au rappel, qui est par exemple 100% si tous les pixels sont blancs, on donne à la valeur β^2 une valeur < 1. Ainsi, certains auteurs ont utilisé $\beta^2 = 0.5$ [36, 37] et $\beta^2 = 0.3$ [38].

2.3.2 La courbe précision-rappel (courbe PR)

Chaque point de la courbe précision-rappel correspond à un couple (rappel, précision) obtenu pour un seuil donné. Le rappel est à l'abscisse et la précision à l'ordonnée du graphe.

La courbe PR est plus efficace dans l'évaluation de la performance de modèle pour des jeux de données déséquilibrées, c'est-à-dire pour des jeux de données où le nombre d'instances négatives est largement plus grand que celui des instances positives. En effet, un changement important dans le nombre de faux positifs peut résulter en faible changement dans le taux de faux positifs à cause du grand nombre de vrais négatifs (Taux de faux positifs $= \frac{\text{Faux positifs}}{\text{Faux positifs+Vrais négatifs}}$). Cependant, ce changement aura un effet important sur la précision qui n'est pas calculée sur la base des vrais négatifs (Précision $= \frac{\text{Vrais positifs}}{\text{Vrais positifs+Faux positifs}}$) utilisée dans la courbe PR [39, 40].

L'interpolation de la précision pour des niveaux standards de rappel est faite de la façon suivante. Soient $r_j \in \{0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$ et r un rappel obtenu par l'équation 2.30 décrite ci-haut(ou l'équation 2.32). Soit $P(r)$ la précision correspondante à r calculée suivant l'équation 2.29 ou 2.31. La précision correspondante au rappel r_j est donnée par $P(r_j)$ telle que :

$$P(r_j) = \max_{r \geq r_j} P(r) \tag{2.41}$$

2.3.3 La mesure de la moyenne de l'erreur absolue (MAE : "Mean Absolute Error")

Comme son nom l'indique, la mesure de la moyenne de l'erreur absolue (MAE : "Mean Absolute Error") est la moyenne de l'erreur absolue entre les valeurs des pixels de la sortie du modèle à l'étude et celles de l'image vérité de terrain. Soient $\overline{S}$ l'image de sortie du modèle dont les valeurs des pixels sont normalisées pour être dans l'intervalle $[0.0, 1.0]$ et $\overline{G}$, la vérité de terrain, dont les valeurs de pixels normalisées sont 0 ou 1. Soient W et H, la largeur et la hauteur de l'image respectivement (en pixels).

La mesure de la moyenne de l'erreur absolue (MAE : "Mean Absolute Error") est donnée par :

$$MAE = \frac{1}{W \times H} \sum_{x=1}^{W} \sum_{y=1}^{H} |\overline{S}(x,y) - \overline{G}(x,y)| \tag{2.42}$$

La MAE mesure la moyenne des amplitudes des erreurs et ne tient pas compte des directions des erreurs. Cette mesure a l'avantage de ne pas favoriser les erreurs de plus grande amplitude par rapport à celles de plus faible amplitude.

2.3.4 La F_β^w mesure

La F_β^w est une mesure d'évaluation de la performance d'algorithmes de vision en informatique qui ont pour sortie des cartes à comparer avec une carte de vérité de terrain binaire. Cette mesure a été proposée par Margolin *et al.* [41]. A l'aide de cette mesure, ces derniers ont voulu corriger les défauts des mesures obtenues grâce aux quantités suivantes : vrais positifs, faux positifs, vrais négatifs, faux négatifs. Il s'agit entre autres de la F_β mesure, la courbe Précision-Rappel, etc. En effet, si nous considérons une image de N pixels, G un vecteur $1 \times N$ des valeurs des pixels de sa vérité de terrain ("Ground Truth") et D un vecteur de pixels de l'image de sortie de l'algorithme dont évalue la performance :

- Le premier défaut est l'interpolation pour les mesures qui sont basées sur une

courbe interpolée entre les points obtenus grâce aux différents seuils utilisés pour binariser une carte non-binaire obtenue par l'algorithme dont on mesure la performance. La courbe Précision-Rappel est une courbe interpolée. Ce défaut est corrigé en permettant aux pixels classés d'être partiellement corrects. Ceci veut dire qu'au lieu de binariser l'image de sortie de l'algorithme, le calcul des vrais positifs "True Positive" : TP), faux positifs ("False Positive" : FP), vrais négatifs ("True Negative"), faux négatifs ("False Negative" : FN) est directement effectué avec les valeurs des pixels :

$$TP' = D.G^T \tag{2.43}$$

$$TN' = (1-D).(1-G^T) \tag{2.44}$$

$$FP' = D.(1-G^T) \tag{2.45}$$

$$FN' = (1-D).G^T \tag{2.46}$$

. : étant l'opération de produit scalaire
G^T : étant la transposée du vecteur G.

- L'autre défaut relevé par Margolin *et al.* [41] est que ces mesures assument l'indépendance des pixels pour calculer les vrais positifs "True Positive" : TP), faux positifs ("False Positive" : FP), vrais négatifs ("True Negative"), faux négatifs ("False Negative" : FN). Toutefois, la dispersion des faux négatifs joue un rôle important. Par exemple, si le même nombre de pixels blancs (mais inférieur au nombre total de pixels blancs dans la vérité de terrain) sont groupés ensemble sur une partie de l'objet saillant ou s'ils sont dispersés de façon à couvrir tout l'objet saillant, les mesures qui utilisent les vrais positifs "True Positive" : TP), faux positifs ("False Positive" : FP), vrais négatifs ("True Negative"), faux négatifs ("False Negative" : FN) donneront la même valeur de performance dans les deux cas. Cependant, c'est le deuxième cas qui donne une meilleure performance pour le système visuel humain.

- Le troisième défaut relevé est que pour les mesures qui utilisent les vrais positifs "True Positive" : TP), faux positifs ("False Positive" : FP), vrais négatifs ("True Negative"), faux négatifs ("False Negative" : FN) comme quantités de base, on assume que les erreurs par rapport à l'objet saillant ont la même importance. Cependant, la localisation des faux positifs dans la carte de saillant est très importante pour la qualité de cette dernière. En effet, si les faux positifs sont distribués en suivant les contours de l'objet saillant ou s'ils sont éloignés de cet objet, on remarque aisément que la qualité de la carte de saillance est meilleure dans le premier cas.

  Pour corriger ces deux derniers défauts, Margolin *et al.* [41] ont proposé d'exprimer les TP', TN', FP' et FN' en fonction du vecteur erreur $E = |G - D|$ et en pondérant ces erreurs. Comme la vérité de terrain est une image binaire, les composantes de G sont soit 0, soit 1. Ainsi pour une composante i, $E(i) = \begin{cases} 1 - D(i) & \text{si } G(i) = 1 \\ D(i) & \text{si } G(i) = 0 \end{cases}$

  Les équations 2.43 à 2.46 s'écrivent alors de la façon suivante :

$$TP' = (1-E).G^T \tag{2.47}$$

$$TN' = (1-E).(1-G^T) \tag{2.48}$$

$$FP' = E.(1-G^T) \tag{2.49}$$

$$FN' = E.G^T \tag{2.50}$$

  Pour pondérer les erreurs le vecteur erreur E devient E^w défini par

$$E^w = \min(E, E.\mathbb{A}).\mathbb{B} \tag{2.51}$$

  Avec $\mathbb{A}$ une matrice $N \times N$ qui capture les dépendances entre pixels telle que

$$\mathbb{A}(i,j) = \begin{cases} \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{d(i,j)^2}{2\sigma^2}) & \text{si } \forall i,j \; G(i) = 1 \text{ et } G(j) = 1 \\ 1 & \text{si } \forall i,j \; G(i) = 0 \text{ et } i = j \\ 0 & \text{sinon} \end{cases}$$

Et $\mathbb{B}$ un vecteur colonne $N \times 1$ qui donne un poids aux faux positifs en fonction de leur distance de l'objet saillant :

$$\mathbb{B}(i) = \begin{cases} 1 & \text{si } \forall i \; G(i) = 1 \\ 2 - \exp(\alpha . \min_{G(j)=1, j=1}^{N} d(i,j)) & \text{sinon} \end{cases}$$

N : le nombre de pixels de l'image

$d(i,j)$: la distance euclidiennne entre les pixels i et j

$\sigma^2 = 5$ pour Margolin *et al.* [41]

$\alpha = \frac{\log(0.5)}{5}$

Les équations 2.47 à 2.50 s'écrivent alors de la fçon suivante :

$$TP^w = (1 - E^w).G^T \tag{2.52}$$

$$TN^w = (1 - E^w).(1 - G^T) \tag{2.53}$$

$$FP^w = E^w.(1 - G^T) \tag{2.54}$$

$$FN^w = E^w.G^T \tag{2.55}$$

La F_β^w est donnée par :

$$F_\beta^w = (1 + \beta^2) \frac{\text{Précision}^w \times \text{Rappel}^w}{\beta^2 \times \text{Précision}^w + \text{Rappel}^w} \tag{2.56}$$

Avec

$$\text{Précision}^w = \frac{TP^w}{TP^w + FP^w} \tag{2.57}$$

$$\text{Rappel}^w = \frac{TP^w}{TP^w + FN^w} \tag{2.58}$$

BIBLIOGRAPHIE

[1] Matti Pietikäinen, Abdenour Hadid, Guoying Zhao, and Timo Ahonen. *Computer vision using local binary patterns*, volume 40. Springer Science & Business Media, 2011.

[2] Timo Ojala, Matti Pietikäinen, and David Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern recognition*, 29(1) :51–59, 1996.

[3] Timo Ojala, Matti Pietikäinen, and Topi Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on pattern analysis and machine intelligence*, 24(7) :971–987, 2002.

[4] Timo Ojala, Kimmo Valkealahti, Erkki Oja, and Matti Pietikäinen. Texture discrimination with multidimensional distributions of signed gray-level differences. *Pattern Recognition*, 34(3) :727–739, 2001.

[5] AV Nefian. Georgia tech face database, 2013 (accessed April 7, 2018). "`http://www.anefian.com/research/face_reco.htm`".

[6] Topi Mäenpää and Matti Pietikäinen. Classification with color and texture : jointly or separately ? *Pattern recognition*, 37(8) :1629–1640, 2004.

[7] Caifeng Shan, Shaogang Gong, and Peter W McOwan. Robust facial expression recognition using local binary patterns. In *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, volume 2, pages II–370. IEEE, 2005.

[8] Timo Ahonen, Abdenour Hadid, and Matti Pietikäinen. Face description with local binary patterns : Application to face recognition. *IEEE transactions on pattern analysis and machine intelligence*, 28(12) :2037–2041, 2006.

[9] Abdelmalik Ouamane, Bengherabi Messaoud, Abderrezak Guessoum, Abdenour Hadid, and Mohamed Cheriet. Multi scale multi descriptor local binary features and exponential discriminant analysis for robust face authentication. In *Image Processing (ICIP), 2014 IEEE International Conference on*, pages 313–317. IEEE, 2014.

[10] Timo Ojala, Matti Pietikäinen, and Topi Mäenpää. Gray scale and rotation invariant texture classification with local binary patterns. In *European Conference on Computer Vision*, pages 404–420. Springer, 2000.

[11] Di Huang, Yunhong Wang, and Yiding Wang. A robust method for near infrared face recognition based on extended local binary pattern. In *International Symposium on Visual Computing*, pages 437–446. Springer, 2007.

[12] Guoying Zhao and Matti Pietikäinen. Dynamic texture recognition using local binary patterns with an application to facial expressions. *IEEE transactions on pattern analysis and machine intelligence*, 29(6) :915–928, 2007.

[13] Shengcai Liao, Xiangxin Zhu, Zhen Lei, Lun Zhang, and Stan Z Li. Learning multi-scale block local binary patterns for face recognition. In *International Conference on Biometrics*, pages 828–837. Springer, 2007.

[14] Marko Heikkilä, Matti Pietikäinen, and Cordelia Schmid. Description of interest regions with local binary patterns. *Pattern recognition*, 42(3) :425–436, 2009.

[15] Timo Ahonen, Jiří Matas, Chu He, and Matti Pietikäinen. Rotation invariant image description with local binary pattern histogram fourier features. In *Scandinavian Conference on Image Analysis*, pages 61–70. Springer, 2009.

[16] Shu Liao, Max WK Law, and Albert CS Chung. Dominant local binary patterns for texture classification. *IEEE transactions on image processing*, 18(5) :1107–1118, 2009.

[17] Zhenhua Guo, Lei Zhang, and David Zhang. A completed modeling of local binary pattern operator for texture classification. *IEEE Transactions on Image Processing*, 19(6) :1657–1663, 2010.

[18] Di Huang, Caifeng Shan, Mohsen Ardabilian, Yunhong Wang, and Liming Chen. Local binary patterns and its application to facial image analysis : a survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 41(6) :765–781, 2011.

[19] Li Liu, Lingjun Zhao, Yunli Long, Gangyao Kuang, and Paul Fieguth. Extended local binary patterns for texture classification. *Image and Vision Computing*, 30(2) :86–99, 2012.

[20] Matti Pietikäinen, Abdenour Hadid, Guoying Zhao, and Timo Ahonen. Local binary patterns for still images. In *Computer vision using local binary patterns*, pages 13–47. Springer, 2011.

[21] Christos Faloutsos and King-Ip Lin. *FastMap : A fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets*, volume 24. ACM, 1995.

[22] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

[23] Juha Karhunen and Jyrki Joutsensalo. Generalizations of principal component analysis, optimization problems, and neural networks. *Neural Networks*, 8(4) :549–562, 1995.

[24] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*, volume 1. MIT press Cambridge, 2016.

[25] Antoine Moevus, Max Mignotte, JA de Guise, and Jean Meunier. Evaluating perceptual maps of asymmetries for gait symmetry quantification and pathology detec-

tion. In *Engineering in Medicine and Biology Society (EMBC), 2014 36th Annual International Conference of the IEEE*, pages 3317–3320. IEEE, 2014.

[26] Max Mignotte. Mds-based multiresolution nonlinear dimensionality reduction model for color image segmentation. *IEEE transactions on neural networks*, 22(3) :447–460, 2011.

[27] Max Mignotte. Mds-based segmentation model for the fusion of contour and texture cues in natural images. *Computer Vision and Image Understanding*, 116(9) :981–990, 2012.

[28] Max Mignotte. Symmetry detection based on multiscale pairwise texture boundary segment interactions. *Pattern Recognition Letters*, 74 :53–60, 2016.

[29] Redha Touati and Max Mignotte. Mds-based multi-axial dimensionality reduction model for human action recognition. In *Computer and Robot Vision (CRV), 2014 Canadian Conference on*, pages 262–267. IEEE, 2014.

[30] Redha Touati and Max Mignotte. An energy-based model encoding nonlocal pairwise pixel interactions for multisensor change detection. *IEEE Transactions on Geoscience and Remote Sensing*, 2017.

[31] Ali Borji, Ming-Ming Cheng, Huaizu Jiang, and Jia Li. Salient object detection : A survey. *ArXiv e-prints*, 2014.

[32] Ming-Ming Cheng, Jonathan Warrell, Wen-Yan Lin, Shuai Zheng, Vibhav Vineet, and Nigel Crook. Efficient salient region detection with soft image abstraction. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 1529–1536. IEEE, 2013.

[33] Radhakrishna Achanta, Sheila Hemami, Francisco Estrada, and Sabine Susstrunk. Frequency-tuned salient region detection. In *Computer vision and pattern recognition, 2009. cvpr 2009. ieee conference on*, pages 1597–1604. IEEE, 2009.

[34] Federico Perazzi, Philipp Krähenbühl, Yael Pritch, and Alexander Hornung. Saliency filters : Contrast based filtering for salient region detection. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 733–740. IEEE, 2012.

[35] John A Swets. Measuring the accuracy of diagnostic systems. *Science*, 240(4857) :1285–1293, 1988.

[36] Tie Liu, Zejian Yuan, Jian Sun, Jingdong Wang, Nanning Zheng, Xiaoou Tang, and Heung-Yeung Shum. Learning to detect a salient object. *IEEE Transactions on Pattern analysis and machine intelligence*, 33(2) :353–367, 2011.

[37] David R Martin, Charless C Fowlkes, and Jitendra Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE transactions on pattern analysis and machine intelligence*, 26(5) :530–549, 2004.

[38] Ming-Ming Cheng, Niloy J Mitra, Xiaolei Huang, and Shi-Min Hu. Salientshape : Group saliency in image collections. *The Visual Computer*, 30(4) :443–453, 2014.

[39] Takaya Saito and Marc Rehmsmeier. The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. *PloS one*, 10(3) :e0118432, 2015.

[40] Jesse Davis and Mark Goadrich. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd international conference on Machine learning*, pages 233–240. ACM, 2006.

[41] Ran Margolin, Lihi Zelnik-Manor, and Ayellet Tal. How to evaluate foreground maps ? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2014.

CHAPITRE 3

PERIODICITY IRREGULARITY MAP ESTIMATION OF HUMAN GAIT WITH A DEPTH CAMERA FOR PATHOLOGY DETECTION

Dans ce chapitre, nous exposons notre article publié dans la revue, "IJETAE (International Journal of Emerging Technology and Advanced Engineering) - Journal of Imaging", intitulé : "**Periodicity Irregularity Map Estimation of Human Gait with a Depth Camera for Pathology Detection**". Nous le présentons dans sa langue originale de publication.

3.1 Abstract

This paper describes a computer vision system based on a depth camera (Microsoft Kinect^{TM}) and a conventional treadmill for a fast and reliable visual detection and analysis of the patient's body parts that have an irregular movement pattern, in terms of periodicity, during walking. In this work, we thus assume that the gait of a healthy subject exhibits anywhere in the human body, during the walking cycles, a depth signal (depending on time and collected by a Kinect^{TM} sensor) with a periodic pattern without noise. Herein, the presence of noise and its importance can be used either as a good indicator of possible pathology in an early (and fast) diagnostic tool or to provide information about the presence and extent of disease or (orthopedic, muscular or neurological) problems in the patient. The depth videos used show, for each one, the gait cycles of either a healthy individual or simulating an orthopedic problem (with the presence of a heel under the right or left foot). The proposed system is able to estimate, from each video sequence, a saliency color map showing the areas of strong gait irregularities, in terms of periodicity (also called aperiodic noise energy), of each subject. Even if the maps obtained are informative and highly discriminant for a direct visual classification, even for a non-specialist, the proposed system, based on the extraction/classification of features from each obtained map allow us to automatically detect maps representing healthy

individuals and those representing individuals with orthopedic problems.

3.2 Introduction

The interest in human motion already existed in classical antiquity and depending on needs and tools available, its study has been progressing over the years [1]. As stated by Rose and Gamble [2], even if human walking is seemingly simple, it remains a complex activity (involving balance, timing and coordination of sight and the other senses) whose complexity is revealed if we try to make its qualitative or quantitative description. Every person walking has characteristics of its own (while also sharing several common characteristics of human gait). This is such a familiar experience that in our life we have recognized, at least one day, a person by his walk. This may explain why some researchers study human gait in perspective of human identification [3–5] using different techniques and materials for extracting the features used in classification. For example, Gafurov et al. used accelerometer as material [6], Verlekar et al. [7] et Iwashita et al. [8] used shadow information to characterize gait, Dupuis et al. [9] applied feature subset selection, to name but a few. However, the study of human gait has many other areas of application : robotics (e.g., studies on passive dynamic walking), sport sciences [10] (e.g., modeling of athlete motion) [1], video surveillance applications, advanced human computer interfaces or medicine (e.g., rehabilitation technology) etc. Bauckhage et al. [11] noticed that not only human gait analysis allows the person identification and activity recognition, but it can also help in detecting abnormalities in people's health.Thus, human gait has applications related to diseases that prevent patients from walking normally [12, 13]. The abnormalities generated by diseases and which may alter the *natural* (bio)mechanics of walking can be classified into five categories : deformity, muscle weakness, sensory loss, pain, and impaired motor control [14].

Most accurate systems, for data acquisition in the study of human movement, are systems with markers [15, 16] but the need for systems without markers also arises [17]. Indeed, these marker-based systems are able to give very accurate but also sparse (and generally not distributed equidistantly) measures over the body. In addition, the high

cost of these high-end systems [18] inhibit their widespread usage for routine clinical practices.

In this work, we will focus on how to design and implement a new gait analysis system, from a depth camera placed in front of a subject walking on a conventional treadmill, capable of detecting these above-mentioned abnormalities and to quantify their severity and also to localize the different damaged or impaired body parts of the patient. We are also interested in developing a low cost, without markers, non-invasive and simple to set-up, easy to use and fast computer vision based system while being accurate and reliable, for a rapid clinical diagnosis used as a first interesting screening for a possible (orthopedic, muscular or neurological) disease, prior to a more thorough examination by a specialist doctor.

3.3 Previous Work

Most research works on gait analysis (for possible clinical applications) use systems with markers and multiple cameras [19–21]. But other systems, less invasive and less difficult to handle, such as insoles [20, 22], wireless shoes [23], accelerometers [24–26] and Microsoft Kinect^{TM} depth cameras [27] are now also increasingly being used.

In this latter case, Kinect can provide a depth image which is useful for the development of in-home monitoring systems that automatically detect when falls have occurred or when the risk of falling is increasing [28, 29] or directly provide the skeleton [30] data information for the automated estimation of children's (spatial and temporal) gait parameters [31] or to extract an accurate and reliable set of gait features [32], thanks to a regression approach (based on an ensemble of regression trees). More generally, the full body skeleton information provided by Kinect can also be efficiently used for human gait [33] or posture recognition [34] or body tracking in the context of health applications (coaching or physical exercise of the elderly population) [35], to name a few [36].

As proposed herein, the research works introduced in [37] and in [38] also use, for

data acquisition, a treadmill and a Microsoft Kinect ^{TM} depth camera. These studies support the hypothesis that a symmetrical gait is generally expected in the case of healthy people. For example, in [37] a depth energy image (DEI) which is, in fact, the pixel-wise mean of all images in the input depth image sequence (over a gait cycle, or a longer period), is first estimated. Then, an abnormal gait is detected from a normal one because a symmetric healthy walk use to generate a DEI exhibiting a symmetric silhouette, in terms of mean depth (energy) and conversely. This two-class detection is then achieved through the measurement of asymmetry indexes, from the DEI. Although, this feasibility study gives good results, it was only tested on 6 subjects. Using the same important concept of gait symmetry in healthy subjects, walking abnormalities are also detected in [38]. In this work, a spatial and temporal registration procedure allows to divide each gait cycle in two sub-cycles (left and right steps) and the comparison between these two sub-cycles, at lower limbs, in term of depth difference, allows to efficiently detect an abnormal gait from a normal one and to also quantify the degree of asymmetry of the lower body. Let us also mention the system proposed in [39], which is not used for classification (between healthy or unhealthy gait) but that allows to estimate an interesting color map providing a quick overview (in terms of perceptual color difference) of asymmetries existing in the gait cycle of a subject.

In the same spirit, but with a single triaxial accelerometer (providing simultaneous measurements in three orthogonal directions), Moe-Nilssen and Helbostad [40] have estimated human gait parameters like cadence, step length, and measures of gait regularity and symmetry from subjects at free walking speeds.

Contrary to the aforementioned works, our model is not based on asymmetry detection (between left and right lower limbs) as an indicator of possible pathology but rather on the amount of noise altering an *ideal* periodic depth movement of each part of the human body during walking. In our model, we thus assume that the gait of a healthy subject exhibits, anywhere in the human body, during the walking cycles, a depth signal (depending on time and collected a Kinect^{TM} sensor) with a periodic pattern without noise. The proposed system is able to estimate, from each video sequence, a color map visualizing the areas of strong gait irregularities, in terms of periodicity, on the patient's

body silhouette. In order to get a reliable estimation of this aperiodic noise energy map we have decided to estimate it, in two fully complementary ways, namely in the temporal and in the frequency domains. This strategy allows us to get two different estimations leading to different estimation errors which can then be efficiently combine in order to improve the classification result in a (complementary) fusion way, in terms of information interaction [41]. This map is clearly informative and highly discriminant for a direct visual classification, even for a non-specialist. Herein, the location and degree of noise (representing the degree of gait irregularities) can be used as a good indicator of possible pathology or to provide information about the presence and extent of medical problems. An automatic system, based on the extraction/classification of features from each obtained map is also proposed and allow us to automatically detect maps representing healthy individuals and those representing individuals with orthopedic problems.

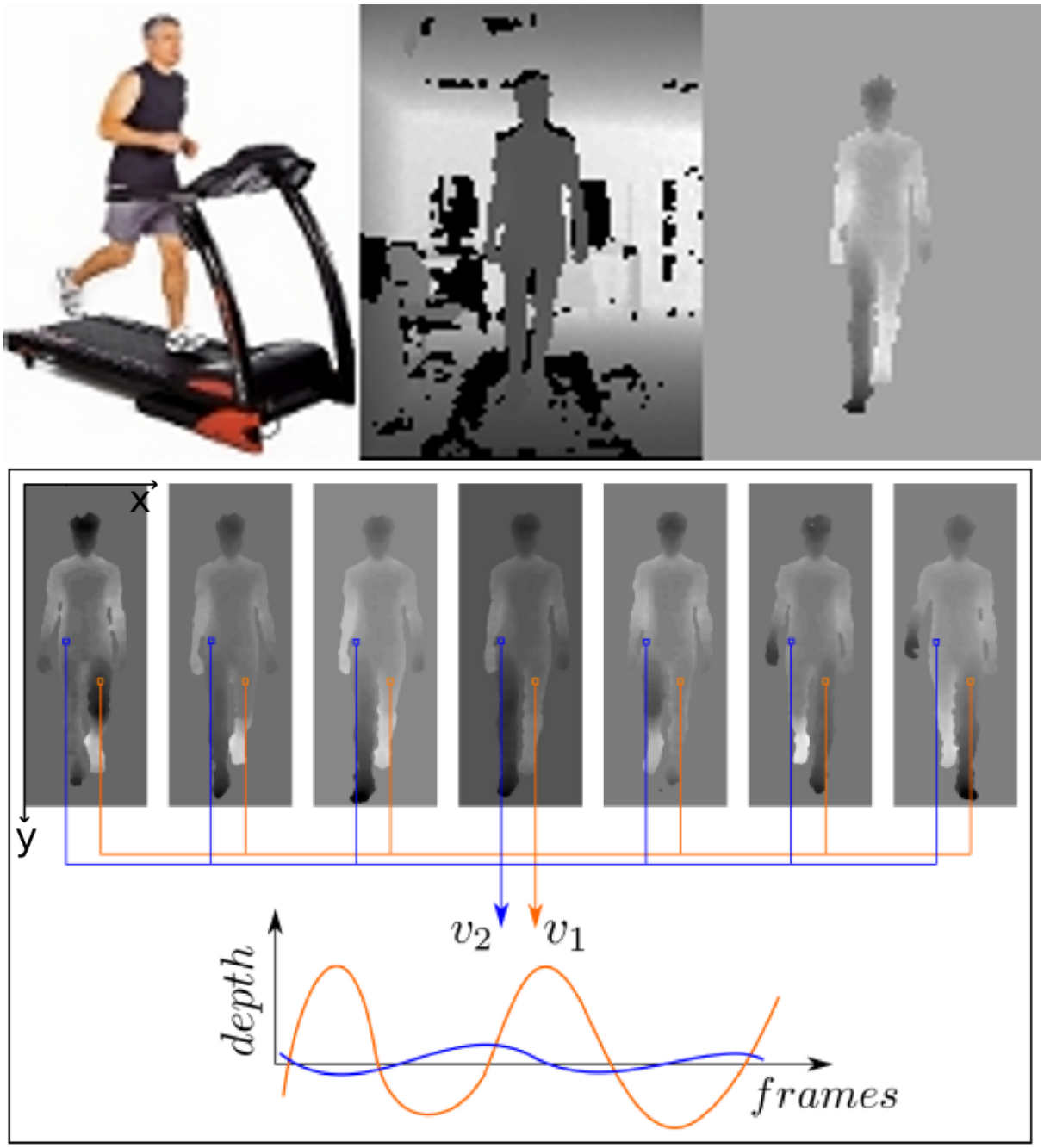


FIGURE 3.1 : From lexicographic order ; setup and pre-processing steps with respectively the subject walking on a conventional treadmill (Life Fitness F3), the original depth view recorded from Kinect^{TM} and the segmented image (after background and treadmill removal). Example of two depth signals (exhibiting a periodic pattern) for a gait cycle of an human subject.

3.4 Proposed Model

The human gait movement is a remarkable example of collaborative interactions between the articular, musculoskeletal and neurological systems working perfectly (and synchronically) together. When everything is working correctly, the healthy loco-motor system produces a smooth, energy efficient, stable gait, exhibiting, at any point of the body, during walking cycles, a (nice) periodic walking pattern without noise. That is why, gait irregularities in terms of periodicity may be a good indicator (and sometimes the first clinical manifestation) of various health problems. In our application, the degree of gait irregularities in terms of periodicity will be also referred, in the following, as being the concept of degree of aperiodicity and quantified by the aperiodic noise energy since this concept is very close to the one existing and used in speech processing[1].

For this purpose, our system first propose to estimate a saliency color map, providing both an overview of aperiodic noise energy or irregularity energy, in terms of periodicity, existing in the gait cycle of a patient and also capable of showing the areas of strongest gait irregularities. In our application, this map is estimated from the sequence of depth images captured by a Microsoft Kinect^{TM}-style (depth) sensor placed in front of the patient walking on a conventional treadmill. After a habituation period of about 2 minutes, the walking speed of each subject appeared to become constant and their gait movement is then collected by the Kinect^{TM}.

The Kinect sensor outputs 30 depth maps per second (30 frames per second), with a resolution of 640 per 480 pixels. Each pixel of the depth image cube (or video sequence) is also a 1D depth signal, evolving through time which, in fact, exhibits a periodic walking pattern, without noise, (see Fig. 3.1) for a stable, energy efficient, healthy gait.

Each video sequence (of approximately 5 minutes) shows the gait cycles of a healthy

[1] In speech or acoustic signal processing, aperiodic noise is produced by high-frequency (acoustic) energy which is distributed fairly randomly across the upper part of the spectrum. It is distinct from periodic energy, which is associated from acoustic signals such as clearly defined formants of the kind we see in vowels and other sonorants [42]. In other words, the deviations from periodicity of the signal introduce additional components on inharmonic frequencies. The energy on inharmonic frequencies (sometimes normalized by the total energy) provides a good estimation measure of noise aperiodicity [43].

person or the one simulating an orthopedic or muscle disease (presence of a troublesome heel under the right or left foot). More precisely, for the experiments, 17 (healthy) subjects (young male adults, 26.7 ± 3.8 years old, 179.1 ± 11.5 cm height and 75.5 ± 13.6 kg with no reported gait issues) were asked to walk normally on a treadmill (Life Fitness F3) with or without simulated length leg discrepancy (LLD). Every patient had to walk normally (group A), then with a 5 *cm* sole, impairing the normal walk, under the left foot (group B), then with the sole under the right foot (group C), for a total of $17 \times 3 = 51$ video sequences to be classified. The scene took place in a non-cluttered room where the treadmill is always in the same position relatively to the Kinect ^{TM} camera. In this way, a silhouette extraction strategy (background and treadmill removal) can be easily defined as proposed in [39].

To this end, in order to get a reliable estimation of this 2D spatial map of irregularities in periodicity, we have decided to estimate it in two different ways. The first one is fully (and only) estimated in the temporal domain whereas the second estimation of this map is only performed in the frequency domain. This strategy allows us to get two different estimations, regarding to the aforementioned map of irregularities in periodicity, with respect to two different criteria leading to different estimation errors. In our application, we will see later that these two different estimations could be efficiently combine in order to improve the classification result in a (complementary) fusion way, in terms of information interaction [41].

3.4.1 Estimation of the Aperiodic Noise Energy in the Temporal Domain

The first step consists in estimating the period T_c of the gait cycle (for each subject) and this can be easily achieved by using the estimation method in the temporal domain described by Cutler and Davis [44]. In this technique, a similarity matrix, comparing all frames of the video sequence, in pairs, is first computed (see Fig. 3.2). More precisely, let F^k and F^l be two frames of the video of size $H \times W$. The similarity value between

F^k and F^l, is given by :

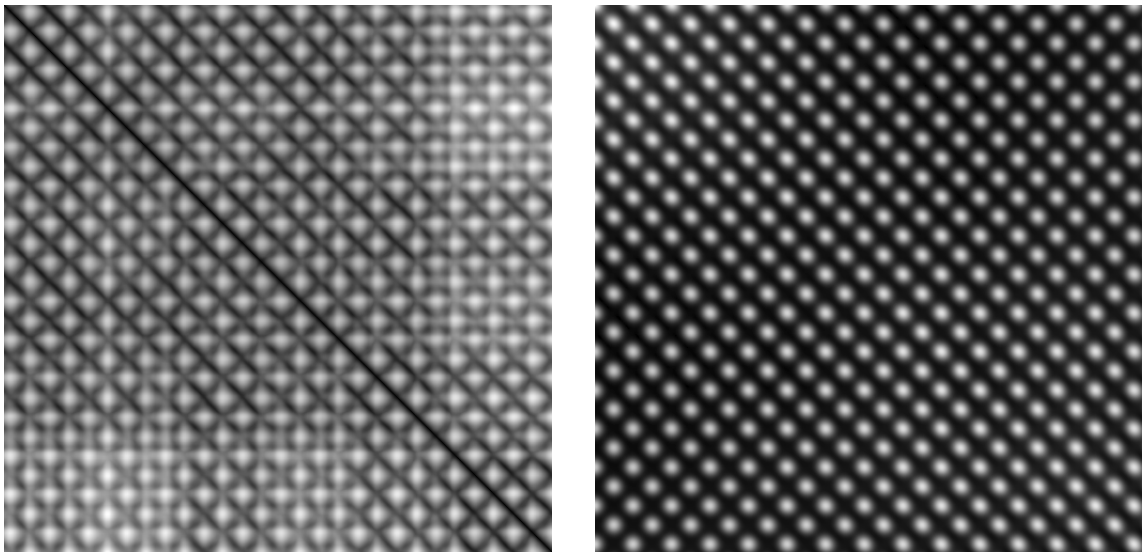
$$S_{F^k,F^l} = \sum_{i=1}^{H} \sum_{j=1}^{W} |F_{i,j}^k - F_{i,j}^l| \tag{3.1}$$

and from this squared symmetric similarity matrix S of size $N_i \times N_i$ (with N_i the number of images in the sequence used in the estimation of T_c), its auto-correlation matrix[2] $A(d_x, d_y)$ (see Fig. 3.2) allows us to highlight the peak values that will enable to estimate the gait period T_c :

$$A(d_x, d_y) = \frac{\sum(S_{x,y} - \bar{S})(S_{x+d_x,y+d_y} - \bar{S}_d)}{\sqrt{\sum(S_{x,y} - \bar{S})^2 \sum(S_{x+d_x,y+d_y} - \bar{S}_d)^2}} \tag{3.2}$$

where the different summations are over all pixel locations (x, y) in the image. $\bar{S}$ and $\bar{S}_d$ are respectively the mean of the similarity matrix and the mean of the shifted similarity matrix. d_x and d_y represents respectively the shift relative to the height and width axis.

[2] The auto-correlation 2D function is itself periodic with the same period as the similarity matrix (or the video sequence) but has the advantage of being much more robust to noise.



(a) **(b)**

FIGURE 3.2 : (a) Images of the similarity matrix S and (b) its auto-correlation function $A(d_x, d_y)$ for the 512 frames of the video and for the $S05_B$ patient (B : with a heel under the left foot).

In Fig. 3.2.(b), the peaks regularly spaced (in a planar lattice) by T_c pixels (in length or width), spatially show the strong temporal correlations existing between image frame at time k and frame at time $k + T_c$ in the video sequence. In addition, the regularity of these peaks both demonstrates that the video contains a periodic motion with a period equal to T_c, in terms of number of image frames (or $T_c\Delta$, in terms of time, with Δ denoting the time between each frame) and that the video remains stationary over the time (*i.e.*, the period T_c of the gait cycle do not sensitively change during the time study). As the autocorrelation matrix is robust to noise and stationary over the time, the period T_c of the gait cycle is simply estimated by the average distance existing between two peaks. This can be done by any procedure returning the local maxima (peaks) of an input signal (e.g., a local peak is a data sample that is larger than its two neighboring samples).

From the previously estimated gait period T_c, the following step consists of the computation of the noiseless periodic pattern for each depth signal, related to each pixel of

the subject silhouette.

1. This step(see step 1. of algorithm 1) is based, in fact, on the simple averaging, in the time domain, of multiple and (non-overlapping) consecutive segments of the depth signal of length T_c. In this context, it is well known that appropriate averaging can increase the Signal to Noise Ratio (SNR), especially for Gaussian noise processes. In the later case, averaging N samples will reduce the mean root mean square (rms) current noise by a factor of $\sqrt{N}$, or the mean noise power by a factor of $1/N$ [45]. Thus, to this end, we subdivide the depth signal s_{depth} (of length T) into successive signals of length T_c and estimate an average pattern s_{patt} which is therefore less noisy.

Estimation of the aperiodic noise energy
in the temporal domain

---

s_{depth} Depth signal for each pixel in the subject silhouette (size : $depth$) (Input)

T_c Depth signal period (Input)

n^s_{depth} Aperiodic noise energy of the depth signal (Output)

s_{patt} Noiseless periodic pattern (size : T_c)

c_{patt} Vector containing the number of times that a point of the periodic pattern occurred in the depth signal (size : T_c)

Initialization :

▷ **for** *each* $i \in [0,\ldots,T_c[$ **do**

$\quad s_{patt}[i] \leftarrow 0.0$ and $c_{patt}[i] \leftarrow 0$

1. Compute the noiseless periodic pattern

▷ **for** *each* $i \in [0,\ldots,depth[$ **do**

$\quad s_{patt}[i \bmod \mathrm{T}_c] \leftarrow s_{patt}[i \bmod \mathrm{T}_c] + \mathrm{s}_{depth}[\mathrm{i}]$

$\quad c_{patt}[i \bmod \mathrm{T}_c] \leftarrow c_{patt}[i \bmod \mathrm{T}_c] + 1$

▷ **for** *each* $i \in [0,\ldots,T_c[$ **do**

$\quad s_{patt}[i] \leftarrow s_{patt}[i] / c_{patt}[i]$

2. Compute the aperiodic noise energy

$n^s_{\text{depth}} \leftarrow 0$

▷ **for** *each* $i \in [0,\ldots,depth[$ **do**

$\quad n^s_{\text{depth}} += \left(s_{\text{depth}}[i] - s_{patt}[i \mod T_c]\right)^2$

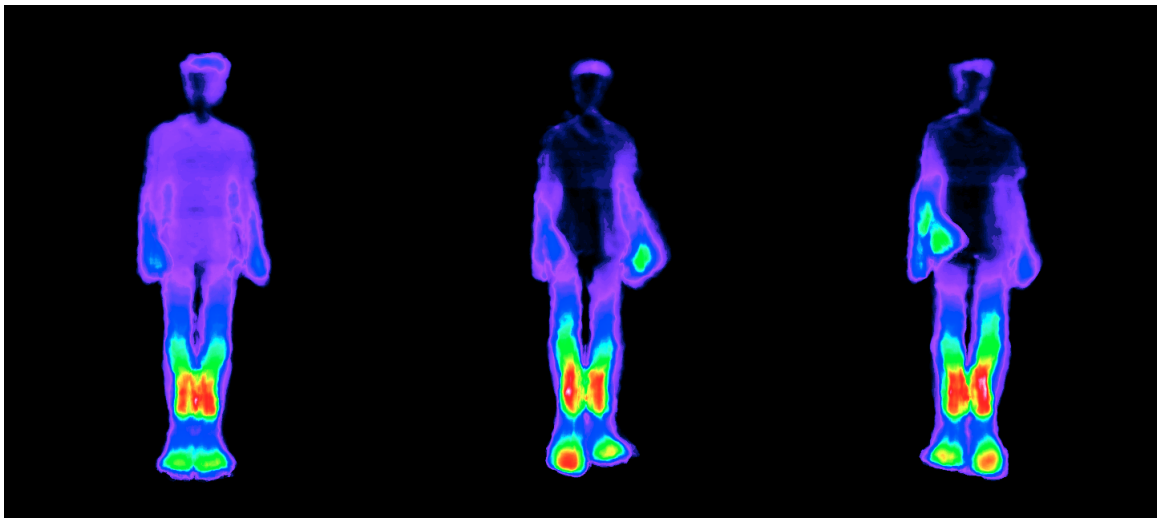
Algorithm 1 : Estimation of the aperiodic noise energy, in the temporal domain, for each pixel s in the subject silhouette

2. The final step(see step 2. of algorithm 1) consists in computing, for each pixel of the subject silhouette, the difference between the original (noised) depth signal and its related noiseless periodic pattern (modulo T_c). This difference signal represents the (aperiodic) noise signal $n(t)$ of each depth signal, which is then squared integrated over $T \gg T_c$ (T is the length of the depth signal), in order to compute the (temporal) energy of this aperiodic noise in the temporal domain. To this end, we use the classical numerical integration formula [46] :

$$\underbrace{n^s_{\text{depth}}}_{\text{Aperiodic Noise Energy}} = \sum_{t=0}^{T} \Big| \underbrace{s_{\text{depth}}(t) - s_{patt}(t \bmod T_c)}_{n(t)} \Big|^2 \tag{3.3}$$

where s_{patt} is the noiseless periodic pattern (of length T_c) estimated in the previous (first) step. The above-mentioned two steps of the estimation of the aperiodic noise energy (in the temporal domain), for each pixel of the subject silhouette, are shown, in pseudo code form, in Algorithm 1.

Each estimated noise energy value, related to each pixel of the subject silhouette, represents thus, in our application, the amount of gait irregularities, in terms of periodicity, of each depth signal, or equivalently (since the treadmill and the depth camera remains fixed during the gait), of each body part of the subject during his gait cycle. This map is also capable of showing the areas of strongest gait irregularities. In our application, this map is visualized in pseudo-color using the thermal scale (from dark blue-cold to red for white spot) in order to make some details more visible (see Fig. 3.3 and 3.4).



(a) **(b)** **(c)**

FIGURE 3.3 : Periodicity irregularity maps for S05 subject (map obtained in the temporal domain). From left to right : (a) Without a heel (b) Heel under left foot (c) Heel under right foot.

3.4.2 Estimation of the Aperiodic Noise Energy in the Frequency Domain

In order to estimate, in the frequency domain, the noiseless periodic pattern of the depth signal (related to each pixel) we rely on the Parseval's theorem which specifies that the signal energy is preserved by the Fourier transform across the time and frequency domains [46]. So, let $n(t)$ be the aperiodic noise signal in the temporal domain and $X(\nu)$, its discrete Fourier transform (DFT). Parseval's equality allows us to write that the aperiodic noise energy expressed, in the time and frequency domain, is equal, *i.e.* ; $n^s_{\text{depth}} = \sum_{t=0}^{T} |n(t)|^2 = (1/T) \cdot \sum_{\nu=0}^{T} |X(\nu)|^2 = n^f_{\text{depth}}$, where $|X(\nu)|^2$ is called the power spectrum or the power spectral density (PSD) of the aperiodic noise signal. Moreover, in the frequency domain, the energy of the aperiodic noise signal (n^f_{depth}) is simply the summation of the difference between the PSD of the (noisy) depth signal $s_{\text{depth}(t)}$ and $|X_{patt}(\nu)|^2$, the PSD of the noiseless periodic pattern $s_{patt}(t)$ according to the following

formula :

$$\underbrace{n^{f}_{\text{depth}}}_{\text{Aperiodic Noise Energy}} = \sum_{\nu=0}^{T} \left| |X(\nu)|^2 - |X_{patt}(\nu)|^2 \right| \tag{3.4}$$

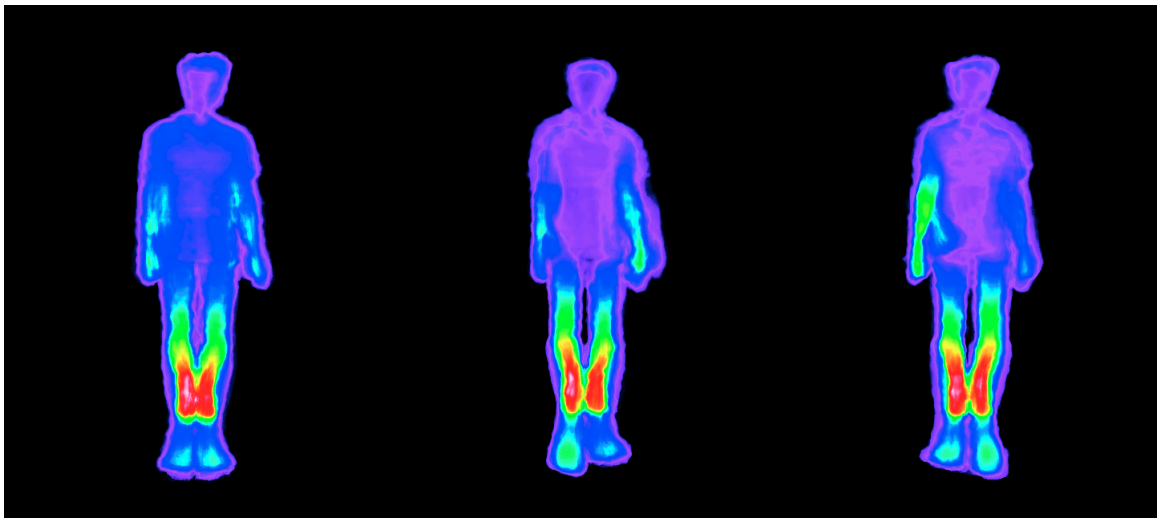
Thus, in order to efficiently estimate $|X_{patt}(\nu)|^2$, we rely on the estimation of the mean periodogram introduced by Welch [47]. This improved estimator[3] of the power spectrum density (PSD) consists of dividing the temporal signal of depth into (possibly) overlapping segments x_b. Each segment is weighted by a smooth (e.g., Hamming or Hanning) window[4] and is then processed by a DFT in order to obtain the modified periodogram $|X_b(\nu)|^2$. The averaging of these modified periodograms allows us to estimate the Welch's PSD estimate [47] $|X_W(\nu)|^2$. This averaging (of modified periodograms) tends to decrease the variance of the PSD estimate relative to a single periodogram estimate of the entire data record. By this fact, the variance is reduced by a factor of L over the periodogram, L being the number of blocks used in the averaging. In our application, we use an overlapping of 50% for segments of length T_s weighted by an Hanning window (and for a length of an entire data record $T = 512$). In addition, we found that the classification rate is also improving as the length T_s decreases until the value $T_s = 16$ (see Fig. 3.5). This gives us a total number of $L = 63$ block segments (used in the averaging) leading to a PSD estimate with a variance reduced by $L = 63$ compared to single periodogram estimation.

The integration of the difference (see Eq. (3.4)) between the Welch's mean periodogram $|X_W(\nu)|^2$ (of length $T_s = 16$) and the periodogram of each $|X_b(\nu)|^2$ allows us to give an estimation, in the frequency domain, of the irregularities in terms of periodicity or the degree of aperiodicity (the so-called aperiodic noise energy) existing in each depth signal.

---

[3] In the sense that this estimator is able to give an estimation which is very robust to noise and thus a quasi-noiseless estimation.

[4] The windowing suppress the discontinuity, and the resulting spurious high frequencies in the frequency analysis, by "tapering" the recorded signal smoothly to zero at the start and end of the recording period.



(a) **(b)** **(c)**

FIGURE 3.4 : Periodicity irregularity maps for S05 subject (map obtained in the frequency domain). From left to right : (a) Without a heel (b) Heel under left foot (c) Heel under right foot.

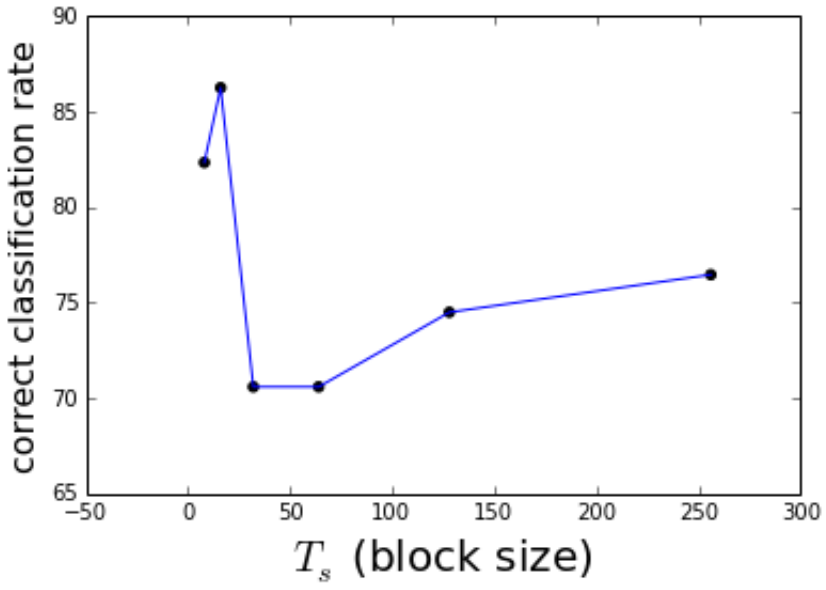


FIGURE 3.5 : Curve representing the correct classification rate as a function of T_s, the block size.

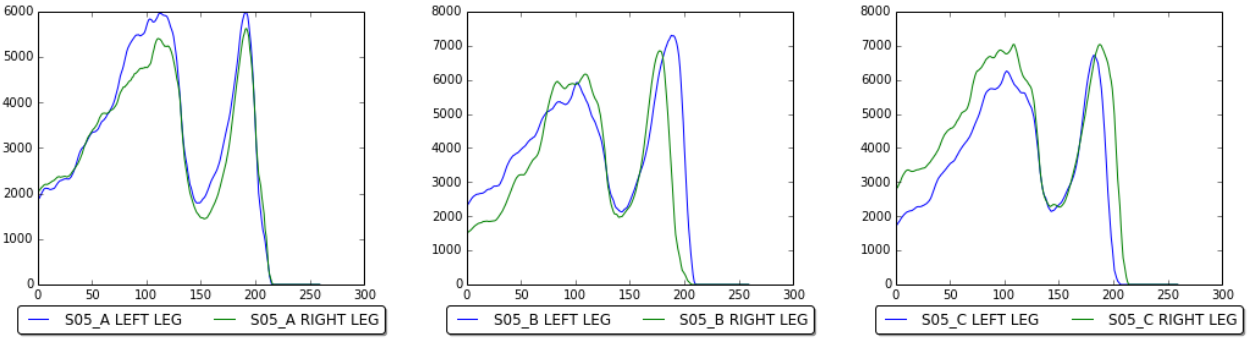
3.4.3 Automatic Classification of the Subjects

The two maps of aperiodic noise energy, estimated in the frequency and temporal domain, contain the same information but are degraded with different estimation errors. More precisely, these estimation errors are mainly due, in the temporal domain, to the estimation of the period T_c (of the gait cycle) from the autocorrelation of the similarity matrix and mainly due, in the frequency domain, both to the choice of the window (or apodization) function which slightly modified the spectrum (causing leakage[5]) and the window size $T_s < T$ which reduced the frequency resolution of the spectrum and consequently, bias the amplitudes and shape of the spectrum.

At this stage the estimated maps are highly discriminant for a direct visual classification by the clinician (which can also easily localize the problematic or aperiodic noise parts of the patient's body), or even for a non-specialist (Figures 3.3 and 3.4 show that

[5] Different types of windows will have different leakage-properties.

for a subject without a heel, the irregularities of periodicity are identically distributed on either side of the axis of symmetry of the silhouette while for subjects with heel, the irregularities of periodicity are larger for the member with heel). Nevertheless, in order to propose a fully automated gait analysis system, we have also developed a classification scheme based on the extraction of the following features obtained from each obtained map. We classified the maps into two classes, namely healthy individuals and those representing individuals with orthopedic problems (with the left or right foot) and three classes (subject without heel under foot, with heel under left foot, with heel under the right foot).



(a) **(b)** **(c)**

FIGURE 3.6 : Curves obtained by the estimation of the aperiodic noise energy achieved in the temporal domain (as a function of the pixel number) representing the two vectors of the horizontal summations of the aperiodic noise energy for different values of height (from top to bottom) of the left and the right leg, for *S*05 subject : (a) Without heel (b) Heel under left foot (c) Heel under right foot.

3.4.3.1 Feature Extraction Step

These features are divided into two major categories. The first group estimates the degree of difference of the aperiodicity noise energy level (or degree of irregularity, in terms of periodicity) between the left and the right leg for each individual and for a

height at a given vertical position (see algorithm 7). In fact, it is expected that there is a larger concentration of aperiodic noise energy on the side with the (simulated orthopedic) problem comparatively to the other side. In order to exploit this discriminant property, the horizontal summations of the aperiodic energy values, for different values of height, from top to bottom of the left and the right leg, are stored in two different vectors for each individual.

This interesting feature vector is shown by the curves that display the aforementioned vectors for a subject with or without simulated length leg discrepancy (see Fig. 3.6). The figure shows that for subjects without heel under the foot, the curves are almost identical, but the difference is very remarkable for subjects with heel. This suggests that this difference of aperiodic noise energy (for the left and the right leg) provides an interesting and a discriminant feature vector for classification.

The second group of features is related to the deformation or the shape of the silhouette generated by the walking action in the video sequence (and thus is not directly related to the aforementioned concept of aperiodic noise energy). Indeed, the presence of problems or pain in lower limbs (or in the body in general) may disrupt the body's alignment and posture. More precisely, for a healthy gait, the right arm swings in the same direction as the left leg, and conversely with a certain symmetry, in terms of arm and leg swings and approximating, in fact, a (regular and periodical) symmetrical pendular movement.

In order to now model the potential disruption related to a possible asymmetry between the left arm and right leg and vice-versa, we consider the following operations :

1. From the silhouette (given by the set of non-zero values of the) aperiodic noise energy map I (see Fig. 3.7.(a)), we estimate the silhouette asymmetry AS(I), by a simple logical "exclusive or" operation between the left and right part (and conversely) of the binary silhouette (around the preliminary estimated longitudinal axis x_{sym}) (see Algorithm 2 and Fig. 3.7.(c)).

2. We divide the silhouette (see Fig. 3.7.(a)) into two parts (the lower limbs [between

$\frac{3}{5}H$ and H with H the image height] and the upper part of the body above the lower limbs [between 0 and $\frac{3}{5}H$ with H the image height]). Each part is then further divided into two sub-parts (the left and the right parts of the longitudinal axis) for a total of four parts (P_1, P_2, P_3 and P_4) (see Fig. 3.7.(b)).

On these four parts, we compute SURFA_1 and SURFA_2, the number of pixels (or surface areas) located, above the lower limbs, respectively, to the left and to the right of x_{sym} on AS(I). We also compute SURFA_3 and SURFA_4, the number of pixels located, for the lower limbs, respectively, to the left and to the right of x_{sym} on AS(I) (see Fig. 3.7.(c)).

Concretely, the parameters SURFA_i represent (in terms of number of pixels), the asymmetry degree (or magnitude) of the facial silhouette between the right and left parts of the body during consecutive gait cycles.

3. We now compute SIL_1 and SIL_2, the two parameters estimated, in the following way :

$$\text{SIL}_1 = 2 \times (\text{SURFA}_1 + \text{SURFA}_4) \tag{3.5}$$

$$\text{SIL}_2 = 2 \times (\text{SURFA}_2 + \text{SURFA}_3) \tag{3.6}$$

Concretely speaking, the parameters $\text{SIL}_{1,2}$ represent, the degree or magnitude of asymmetry swing between the left arm and the right leg and vice-versa during gait.

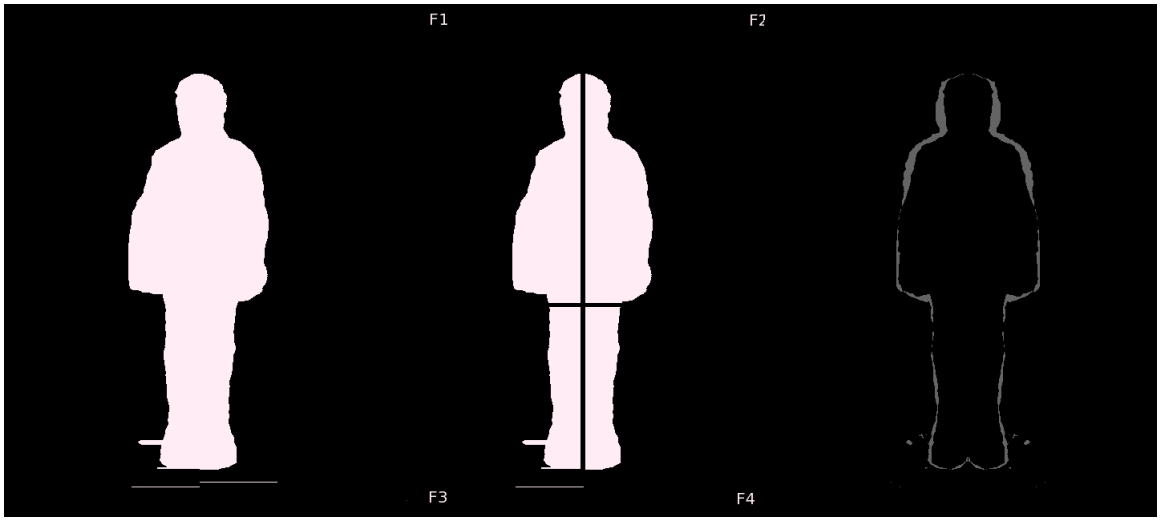
4. In addition, we estimate the following two features :

$$\text{Feat}_2 = \frac{(\text{SIL}_1 - \text{SIL}_2)}{\text{SIL}_1} \text{ if } (\text{SIL}_1 > \text{SIL}_2) \tag{3.7}$$

$$\text{or :} \quad \text{Feat}_2 = \frac{(\text{SIL}_2 - \text{SIL}_1)}{\text{SIL}_2} \text{ if } (\text{SIL}_2 > \text{SIL}_1) \tag{3.8}$$

Finally, Feat_2 quantify the maximum amplitude, in terms of number of pixels, of irregularity, or asymmetry existing in the left arm swing and right leg movement

(or conversely) during consecutive gait cycles (or the asymmetric arm and leg swings motion). We multiply the feature value Feat_2 by 1000 in order to equal weight its importance relatively to the first feature vector related to the asymmetry of the aperiodic noise energy (see algorithm 7).



(a) **(b)** **(c)**

FIGURE 3.7 : Estimation of AS(I), the (asymmetry) deformation of the silhouette of the aperiodic energy map I (subject $S17_A$). From left to right. (a) The silhouette (defined by the set of non-zero values of aperiodic noise energy I) on either side of the preliminary estimated longitudinal axis x_{sym} (see Algorithm 2). (b) The silhouette divided into four parts : P_1, P_2, P_3 and P_4. (c) Estimation of the silhouette asymmetry AS(I), between the left and right part of the silhouette (given by a simple logical "exclusive or" operation between the left and right part of the binary silhouette and then symmetrized around the longitudinal axis x_{sym}).

In all, the dimensionality size of the feature vector, used as input for classification, for the temporal and frequency domain separately, is 261 (*i.e.*, 260 features for the first group of feature, corresponding to the curve (of length 260) of difference between the left and the right's periodicity irregularity of the lower limb (see Fig. 3.6) plus 1 feature,

the 261st, belonging to the second group (see Section 3.4.3.1, and corresponding to the deformation of the silhouette (see Eq. (3.7) or Eq. (3.8)). When the feature vector, in temporal and in frequency domain, is placed side by side ; the vector size is 522. After using PCA, the dimensionality size is reduced to 51.

3.5 Experimental Results

In our application, the depth of the video cube, to be processed is $T = 512$, which corresponds to a gait video sequence of about 17 seconds (for a Kinect capturing 30 frames per second).

First, concerning the estimation of the aperiodic noise energy in the temporal domain, we can notice on the similarity matrix (see Fig. 3.2) a dark line on the diagonal, that emphasizes the fact that each frame of the video is similar to itself, and also a periodic pattern identical in the vertical and horizontal direction, showing that the human gait exhibits a clear periodic motion [44]. The auto-correlation function of the similarity matrix (see Fig. 3.2) is itself periodic with the same period T_c as the similarity matrix (or the video sequence) but has the important advantage of being much more robust to noise since each value of the auto-correlation function results from the integration over all values of the similarity matrix. The period T_c of the gait cycle is, of course, slightly different for each subject, but remains about 36 ± 4. [6]

Consequently, the estimation of the noiseless periodic pattern of each depth signal (related to each pixel of the subject silhouette and required in the estimation of the aperiodic noise energy map in the temporal domain, see Section 3.4.1), is based on the averaging of $T/T_c \approx 14.22$ depth signals which thus leads to a periodic pattern altered by a mean noise power approximately reduced by a factor around 15 (or a periodic pattern with less than $100/14.22 \approx 7\%$ of aperiodic (irregularity) energy.

[6] 36 ± 4 means that 99% of the gait period estimations, obtained on the set of 17 subjects, are, in terms of number of frames, within the confidence interval $[32, ..., 40]$ among the subjects (*i.e.*, the shortest gait period is 32 frames and the longest is 40 frames). In terms of seconds, by remembering that the kinect outputs 30 frames per second, it defines the following confidence interval : [≈ 1.06 seconds, ..., $\approx$ 1.34 seconds].

Second, concerning the estimation of the aperiodic noise energy in the frequency domain, we recall that we use an overlapping of 50% for segments of length $T_s = 16$, weighted by an Hanning window, for a total of $L = 63$ block segments, used in the averaging, and leading to a PSD estimate with a variance reduced by $L = 63$ compared to single periodogram estimation.

The examples shown in Fig. 3.6, 3.8 and 3.9 confirm that the difference of aperiodicity curves, along with the parameter quantifying the asymmetric arm and leg swings motion, provide complementary and relevant features for our classification problem. More precisely, the curve for the right leg and the curve for the left leg tends to coincide for subjects without heel whereas the curve tends to move away from one another for subjects with a heel. In Fig. 3.8, the curve of the difference of the curves in Fig. 3.6, tends to touch the x-axis for subjects with heel while for subjects without their heel, the curves are less close to the x-axis. In Fig. 3.9, the deformation, evidenced by the white pixels, is larger for subjects with heel and smaller for subjects without a heel.

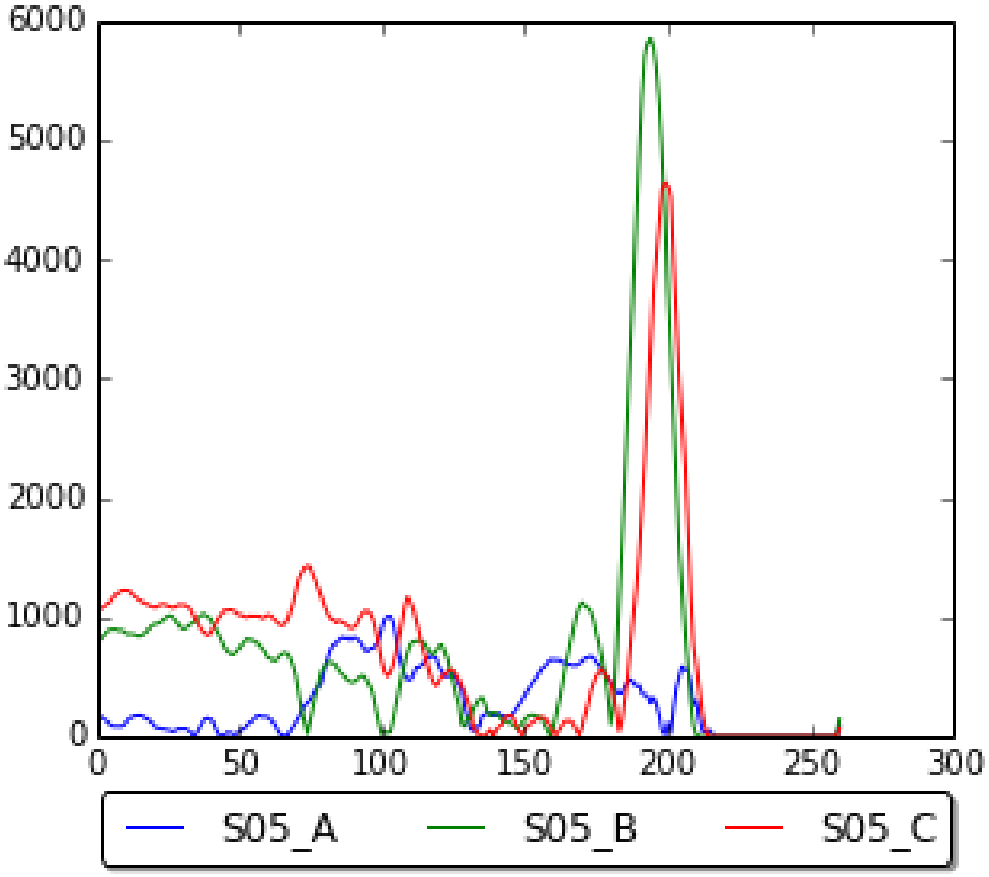
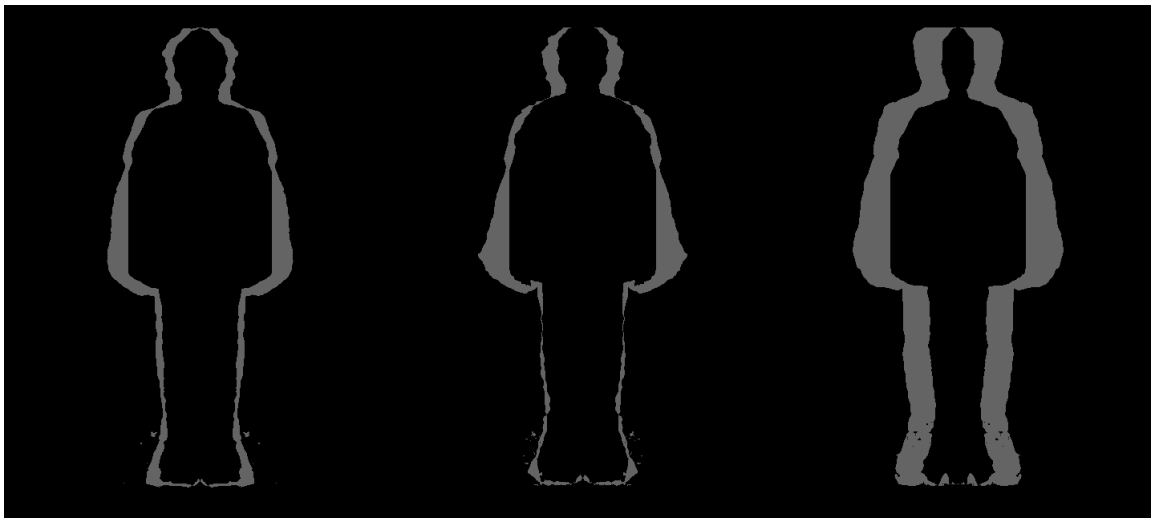


FIGURE 3.8 : Curves obtained by the estimation of the aperiodic noise energy achieved in the temporal domain (as a function of the pixel number) representing the absolute value of the difference of the two vectors of the horizontal summations (i.e., summation over the columns) of the aperiodic noise energy for different values (from top to bottom) of height, of the left and right leg, of the subject $S05$, for the three considered cases, namely ; $S05_A$: subject without the heel ; $S05_B$: subject with the heel under the left foot ; $S05_C$: subject $S05$ with the heel under the right foot.



(a) **(b)** **(c)**

FIGURE 3.9 : Silhouette deformation maps for the *S*06 subject. From left to right ; (a) without heel (b) heel under the left foot (c) heel under the right foot.

TABLE 3.1 : Classification rates of 51 periodicity irregularity maps of 17 subjects into two classes (normal or not) (LOOCV : Leave One Out Cross-Validation, LR : Logistic Regression, SVM : Support Vector Machine, KNN : K-Nearest Neighbors, GNB : Gaussian Naive Bayes, SGDC : Stochastic Gradient Descent Classifier.
PCA : Principal Component Analysis)

TEMPORAL DOMAIN					
LOOCV with :	**Accuracy :**	**Precision :**	**Sensitivity : (Recall)**	**Specificity :**	**F_1 Score :**
KNN with scaling	90.20%	96.77%	88.23%	94.12%	0.9231
SVM (kernel :RBF) with scaling	88.23%	91.18%	91.18%	82.35%	0.9118
GNB with PCA preprocessing	84.31%	84.21%	94.12%	64.70%	0.8889
LR with scaling	82.35%	87.88%	85.29%	76.47%	0.8657

FREQUENCY DOMAIN					
LOOCV with :	**Accuracy :**	**Precision :**	**Sensitivity : (Recall)**	**Specificity :**	**F_1 Score :**
SVM(kernel :polynomial) with PCA preprocessing	86.27%	90.91%	88.23%	82.35%	0.8955
SVM (kernel :RBF) with scaling	84.31%	80.95%	100.0%	52.94%	0.8947
GNB with PCA preprocessing	82.35%	87.88%	85.29%	76.47%	0.8657

TABLE 3.2 : Classification (for majority voting) rates of 51 periodicity irregularity maps of 17 subjects into two classes (normal or not) (LOOCV : Leave One Out Cross-Validation, LR : Logistic Regression, SVM : Support Vector Machine, KNN : K-Nearest Neighbors, GNB : Gaussian Naive Bayes, SGDC : Stochastic Gradient Descent Classifier.
PCA : Principal Component Analysis)

MAJORITY VOTING					
LOOCV with :	**Accuracy :**	**Precision :**	**(Recall) Sensitivity :**	**Specificity :**	**F_1 Score :**
KNN (spatial), SVM(kernel :RBF) (spatial), SVM(kernel :RBF) (spectral) with scaling	94.12%	94.29%	97.06%	88.23%	0.9565
SVM (kernel :linear)(spatial), KNN(spectral), LR(spectral) with scaling	90.20%	93.94%	91.18%	88.23%	0.9254
KNN (spatial), KNN(spectral), GNB (spatial), GNB(spectral), SVM(kernel :polynomial)(spectral) with PCA preprocessing	88.23%	93.75%	88.23%	88.23%	0.9091

TABLE 3.3 : Classification rates of 51 periodicity irregularity maps of 17 subjects into three classes (A : without heel under foot, B : with heel under left foot, C : with heel under right foot) with the same classifiers and data processing like above classification. linSVM mentioned here is a SVM with linear kernel but implemented in a different way than the other SVM (linear).

TEMPORAL DOMAIN					
LOOCV with :	**Accuracy :**	**Precision :**	**(Recall) Sensitivity :**	**Specificity :**	**F_1 Score :**
SVM (linSVM) with scaling	68.62%	69%	68.67%	68.62%	0.6867
LR with PCA preprocessing	64.70%	65%	64.67%	64.70%	0.64
SVM (kernel :polynomial) without preprocessing	60.78%	61%	60.67%	60.78%	0.6033

FREQUENCY DOMAIN					
LOOCV with :	**Accuracy :**	**Precision :**	**(Recall) Sensitivity :**	**Specificity :**	**F_1 Score :**
SVM(kernel :linear) with PCA preprocessing	66.66%	66.33%	66.66%	66.66%	0.6633
LR with PCA preprocessing	64.70%	66.67%	64.67%	64.70%	0.64
SGDC with PCA preprocessing	60.78%	61%	61.33%	60.78%	0.58

The automatic classification of the subjects walking on the treadmill is studied through

Gaussian Naive Bayes (GNB), k-Nearest Neighbors (KNN), Logistic Regression (LR), Support Vector Machine (SVM) and Stochastic Gradient Descent (SGDC) classifiers from the Python's scikit-learn library [48]. The 51 examples are randomly ordered and we use the cross-validation (leave one cross-validation) technique because we have relatively small data set (51 examples). The kernel for SVM with the best rates for two classes is the radial basis function (rbf kernel). For the ternary (*i.e.*, three classes) classification problem, it is the cubic polynomial kernel that gave the best rates.

The classification results into two or three classes are respectively shown in Table 3.1, Table 3.2 and Table 3.3 both for individual state-of-the-art classification model used in the literature and exploiting the features extracted from either our aperiodic energy map obtained in the temporal domain or in the frequency domain and also for a simple multiple classifier system whose decisions are then combined through the combination of the estimates obtained from each individual base classifier with the simple majority voting decision rule. The experiments have been tried and tested with or without preprocessing by Principal Component Analysis (PCA) and with or without scaling (the PCA and the scaler are from scikit-learn library [48]).

In addition, we show the classification results into two or three classes with a different fusion strategy (between the feature vector extracted from the aperiodic energy map estimated in the frequency and temporal domains) and consisting simply in concatenating the two feature vectors into a single feature vector (see Table 3.4).

We can also notice that the performance of classifiers to the aperiodic energy maps obtained in the temporal or frequency domains are also somewhat different which in fact makes a multiple classifier based strategy a reliable classification algorithm for our two class classification problem. It also shows that these two maps are complementary and can be effectively combined.

Finally, the results show that the two-class classification problem ("normal" or "abnormal" gait) gives excellent results, especially with a multiple classifier system and the first fusion strategy (see Table 3.1 and Table 3.2). On the other hand, the three-class classification problem is more complex and the results are disappointing (whatever the fusion strategy used). This can be explained by the fact that the physical behavior of each

individual, in terms of response to the presence of a heel (placed below the right or left foot) may be very different across individuals (some of them answer to the problem by a greater oscillation of one arm (or the two arms), or a greater (or smaller or different) stride length, etc.

TABLE 3.4 : Classification rates from features extracted from the periodicity irregularity maps in frequency and temporal domain and placed side by side (51 examples for 17 subjects). LOOCV : Leave One Out Cross-Validation, LR : Logistic Regression, SVM : Support Vector Machine, KNN : K-Nearest Neighbors, GNB : Gaussian Naive Bayes and SGDC : Stochastic Gradient Descent Classifier, PCA : Principal Component Analysis ; linSVM mentioned here is a SVM with linear kernel but but implemented in a different way than the other SVM (kernel : linear).

TWO CLASSES					
LOOCV with :	**Accuracy :**	**Precision :**	**(Recall) Sensitivity :**	**Specificity :**	**F_1 Score :**
KNN with or without preprocessing	88.23%	93.75%	88.23%	88.23%	0.9091
GNB with PCA processing	88.23%	88.89%	94.12%	76.47%	0.9143
SVM (kernel :linear) with PCA processing	84.31%	90.63%	85.29%	82.35%	0.8788
LR with scaling	84.31%	93.33%	82.35%	88.23%	0.875

THREE CLASSES					
LOOCV with :	**Accuracy :**	**Precision :**	**(Recall) Sensitivity :**	**Specificity :**	**F_1 Score :**
LR with PCA preprocessing	68.62%	69.33%	68.66%	68.62%	0.68
KNN with scaling	66.66%	65.33%	66.66%	66.66%	0.6433
SVM(linSVM) with scaling	66.66%	66%	66.66%	66.66%	0.6566
SVM (kernel :linear) with scaling	64.70%	64.33%	64.66%	64.70%	0.6466

3.6 Conclusion

In this work, we have presented a new gait analysis system based on features extracted from the estimation of an aperiodic noise energy map which aims at showing the areas of strong irregularities of the gait, in terms of periodicity, of each subject walking on a treadmill, and also allows to quantify the degree of asymmetrical (opposite arm and leg) movement patterns. This 2D spatial map is estimated in two complementary ways, namely in the temporal and in the frequency domains in order to get an estimation of the useful information with two different noises and to subsequently provide complementary decisions from different individual classifier which will then be combined. This map also allows for the clinician to visually and quickly localize and quantify the gait abnormalities and for the rehabilitation of patients, to the evolution of these abnormalities over the time. With further analysis across other population including real patients, we also hope this system could be useful or a good indicator to quickly detect a possible disease, or for a rapid but reliable diagnosis, prior to a more thorough examination by a specialist doctor. The system proposed in this paper is also inexpensive, marker-less, non-invasive, easy to set up and requiring a small room. These characteristics qualify it for the daily activities in a clinic. This system also makes an automatic classification of healthy patients and those who are unhealthy with good classification rates.

BIBLIOGRAPHIE

[1] R. Klette and G. Tee. Understanding human motion : A historic review. *3d Imaging for Safety and Security*, 1 :22–40, 2007.

[2] Jessica Rose and James Gibson Gamble. *Human walking*. Lippincott Williams & Wilkins Philadelphia, 2006.

[3] Anil Jain, Ruud Bolle, and Sharath Pankanti. *Biometrics : personal identification in networked society*, volume 479. Springer Science & Business Media, 2006.

[4] David Cunado, Mark S Nixon, and John N Carter. Automatic extraction and description of human gait models for recognition purposes. *Computer Vision and Image Understanding*, 90(1) :1–41, 2003.

[5] Rong Zhang, Christian Vogler, and Dimitris Metaxas. Human gait recognition. In *Conference on Computer Vision and Pattern Recognition Workshop, 2004. CVPRW'04.*, pages 18–18. IEEE, 2004.

[6] Davrondzhon Gafurov, Kirsi Helkala, and Torkjel Søndrol. Biometric gait authentication using accelerometer sensor. *Journal of computers*, 1(7) :51–59, 2006.

[7] Tanmay Tulsidas Verlekar, Luís Ducla Soares, and Paulo Lobato Correia. Gait recognition in the wild using shadow silhouettes. *Image and Vision Computing*, 76 :1–13, 2018.

[8] Yumi Iwashita and Adrian Stoica. Gait recognition using shadow analysis. In *Bioinspired Learning and Intelligent Systems for Security, 2009. BLISS'09. Symposium on*, pages 26–31. IEEE, 2009.

[9] Yohan Dupuis, Xavier Savatier, and Pascal Vasseur. Feature subset selection applied to model-free gait recognition. *Image and vision computing*, 31(8) :580–591, 2013.

[10] Christina Strohrmann, Holger Harms, Cornelia Kappeler-Setz, and Gerhard Troster. Monitoring kinematic changes with fatigue in running using body-worn sensors. *IEEE Transactions on Information Technology in Biomedicine*, 16(5) :983–990, 2012.

[11] Christian Bauckhage, John K Tsotsos, and Frank E Bunn. Automatic detection of abnormal gait. *Image and Vision Computing*, 27(1-2) :108–115, 2009.

[12] Luca Palmerini, Laura Rocchi, Sabato Mellone, Franco Valzania, and Lorenzo Chiari. Feature selection for accelerometer-based posture analysis in parkinson's disease. *IEEE Transactions on Information Technology in Biomedicine*, 15(3) :481–490, 2011.

[13] Daniel TH Lai, Pazit Levinger, Rezaul K Begg, Wendy Lynne Gilleard, and Marimuthu Palaniswami. Automatic recognition of gait patterns exhibiting patellofemoral pain syndrome using a support vector machine approach. *IEEE Transactions on Information Technology in Biomedicine*, 13(5) :810–817, 2009.

[14] Jacquelin Perry and Judith M Burnfield. *Gait analysis : normal and pathological function*. Slack Incorporated, 2010.

[15] Marc Bächlin, Meir Plotnik, Daniel Roggen, Inbal Maidan, Jeffrey M Hausdorff, Nir Giladi, and Gerhard Tröster. Wearable assistant for parkinson's disease patients with the freezing of gait symptom. *IEEE Transactions on Information Technology in Biomedicine*, 14(2) :436–446, 2010.

[16] Miikka Ermes, Juha Parkka, Jani Mantyjarvi, and Ilkka Korhonen. Detection of daily activities and sports with wearable sensors in controlled and uncontrolled conditions. *IEEE Transactions on Information Technology in Biomedicine*, 12(1) :20–26, 2008.

[17] Lars Mündermann, Stefano Corazza, and Thomas P Andriacchi. Markerless motion capture for biomechanical applications. In *Human Motion*, pages 377–398. Springer, 2008.

[18] Xin Ma, Haibo Wang, Bingxia Xue, Mingang Zhou, Bing Ji, and Yibin Li. Depth-based human fall detection via shape features and improved extreme learning machine. *IEEE Journal of Biomedical and Health Informatics*, 18(6) :1915–1922, 2014.

[19] Michael W Whittle. Clinical gait analysis : A review. *Human Movement Science*, 15(3) :369–387, 1996.

[20] Adam M Howell, Takehiko Kobayashi, Heather A Hayes, K Bo Foreman, and Stacy JM Bamberg. Kinetic gait analysis using a low-cost insole. *IEEE Transactions on Biomedical Engineering*, 60(12) :3284–3290, 2013.

[21] Motion capture systems from vicon.

[22] Paulo Lopez-Meyer, George D Fulk, and Edward S Sazonov. Automatic detection of temporal gait parameters in poststroke individuals. *IEEE Transactions on Information Technology in Biomedicine*, 15(4) :594–601, 2011.

[23] Stacy J Morris Bamberg, Ari Y Benbasat, Donna Moxley Scarborough, David E Krebs, and Joseph A Paradiso. Gait analysis using a shoe-integrated wireless sensor system. *IEEE Transactions on Information Technology in Biomedicine*, 12(4) :413–423, 2008.

[24] Rolf Moe-Nilssen. A new method for evaluating motor control in gait under real-life environmental conditions. part 1 : The instrument. *Clinical Biomechanics*, 13(4) :320–327, 1998.

[25] Ryo Takeda, Shigeru Tadano, Masahiro Todoh, Manabu Morikawa, Minoru Nakayasu, and Satoshi Yoshinari. Gait analysis using gravitational acceleration measured by wearable sensors. *Journal of biomechanics*, 42(3) :223–233, 2009.

[26] Dean M Karantonis, Michael R Narayanan, Merryn Mathie, Nigel H Lovell, and Branko G Celler. Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring. *IEEE Transactions on Information Technology in Biomedicine*, 10(1) :156–167, 2006.

[27] MJ Landau, BY Choo, and PA Beling. Simulating kinect infrared and depth images. *IEEE Transactions on cybernetics*, 2015.

[28] Erik E Stone and Marjorie Skubic. Fall detection in homes of older adults using the microsoft kinect. *IEEE Journal of Biomedical and Health Informatics*, 19(1) :290–301, 2015.

[29] Erik Stone and Marjorie Skubic. Evaluation of an inexpensive depth camera for in-home gait assessment. *Journal of Ambient Intelligence and Smart Environments*, 3(4) :349–361, 2011.

[30] Zhengyou Zhang. Microsoft kinect sensor and its effect. *IEEE Multimedia*, 19(2) :4–10, 2012.

[31] Saeid Motiian, Paola Pergami, Keegan Guffey, Corrie A Mancinelli, and Gianfranco Doretto. Automated extraction and validation of children's gait parameters with the kinect. *BioMedical Engineering OnLine*, 14(11), 2015.

[32] Moshe Gabel, Ran Gilad-Bachrach, Erin Renshaw, and Assaf Schuster. Full body gait analysis with kinect. In *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 1964–1967. IEEE, 2012.

[33] Milos Milovanovic, Miroslav Minovic, and Dusan Starcevic. Walking in colors : human gait recognition using kinect and cbir. *IEEE MultiMedia*, 20(4) :28–36, 2013.

[34] Thi-Lan Le, Minh-Quoc Nguyen, et al. Human posture recognition using human skeleton provided by kinect. In *Computing, Management and Telecommunications (ComManTel), 2013 International Conference on*, pages 340–345. IEEE, 2013.

[35] Štěpán Obdržálek, Gregorij Kurillo, Ferda Ofli, Ruzena Bajcsy, Edmund Seto, Holly Jimison, and Michael Pavel. Accuracy and robustness of kinect pose estimation in the context of coaching of elderly population. In *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 1188–1193. IEEE, 2012.

[36] Jungong Han, Ling Shao, Dong Xu, and Jamie Shotton. Enhanced computer vision with microsoft kinect sensor : A review. *IEEE Transactions on cybernetics*, 43(5) :1318–1334, 2013.

[37] Caroline Rougier, Edouard Auvinet, Jean Meunier, Max Mignotte, and Jacques A de Guise. Depth energy image for gait symmetry quantification. In *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE*, pages 5136–5139. IEEE, 2011.

[38] E. Auvinet, F. Multon, and J. Meunier. Lower limb movement asymmetry measurement with a depth camera. In *Engineering in Medicine and Biology Society (EMBC), 2012 Annual International Conference of the IEEE*, pages 6793–6796, Aug 2012.

[39] A. Moevus, M. Mignotte, J. de Guise, and J. Meunier. Evaluating perceptual maps of asymmetries for gait symmetry quantification and pathology detection. In *36th International Conference of the IEEE Engineering in Medicine and Biology Society, EMBC'2014*, pages 3317–3320, Chicago, Illinois, USA, August 2014.

[40] Rolf Moe-Nilssen and Jorunn L Helbostad. Estimation of gait cycle characteristics by trunk accelerometry. *Journal of biomechanics*, 37(1) :121–126, 2004.

[41] Guang-Zhong Yang. *Body Sensor Networks*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[42] P. Backley. *An introduction to element theory*. Edinburgh University Press, 2011.

[43] H Kawahara, J Estill, and O Fujimura. Aperiodicity extraction and control using mixed mode excitation and group delay manipulation for a high quality speech analysis, modification and synthesis system straight. In *Conference MAVEBA*, 2001.

[44] Ross Cutler and Larry S Davis. Robust real-time periodic motion detection, analysis, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8) :781–796, 2000.

[45] National Engineering Laboratory (U.S.). Electromagnetic Technology Division. *Optical Fiber Characterization*. Number vol. 1 ;vol. 637 in NBS special publication. U.S. Department of Commerce, National Bureau of Standards, 1982.

[46] A Rihaczek. Signal energy distribution in time and frequency. *IEEE Transactions on Information Theory*, 14(3) :369–374, 1968.

[47] P.D. Welch. The use of fast fourier transform for the estimation of power spectra : A method based on time averaging over short, modified periodograms. *IEEE Trans. Audio Electroacoust.*, AU-15 :70–73, 1967.

[48] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn : Machine learning in python. *The Journal of Machine Learning Research*, 12 :2825–2830, 2011.

Appendix : Algorithms

Estimation of the longitudinal axis estimation

I	Aperiodic noise map (size : $height \times width$) (Input)
x_{sym}	Longitudinal axis estimation (Output)
r	Size of the search interval
G_I	Gradient magnitude map of I
Vct	Vector of floats of size *height*
x_{sym}	Column coord. estimation of the longitudinal axis

Initialization : $G_I \leftarrow$ gradient magnitude map of I

1. Longitudinal Axis Estimation

▷ **for** *each* $i \in [0,\ldots,height[$ **do**

  grdMx $\leftarrow 0$

  ▷ **for** *each* $j \in [(width/2) - r,\ldots,(width/2) + r[$ **do**

    ▷ **for** *each* $m \in [0,\ldots,width/2[$ **do**

      grd $\leftarrow G_I[i][j-m] + G_I[i][j+m]$

      if (grd $>$ grdMx) { pos $\leftarrow$ j grdMx $\leftarrow$ grd }

  Vct$[i] \leftarrow$ pos

$x_{\text{sym}} \leftarrow$ median value of the vector elements Vct[]

Algorithm 2 : Estimation of the longitudinal axis estimation

Estimation of depth signal noise in frequency domain using the averaged Welch's periodogram

---

s_{depth} Depth signal in temporal domain (size : $depth$) (Input)

b_dim Block dimension (Input)

n^f_{depth} Aperiodic noise energy of the depth signal (Output)

X_{awp} Averaged Welch's periodogram (size : b_dim)

N_{depth} Noise vector in freq. domain (size : $depth$)

n_b Number of blocks

h_w Hanning window (size : b_dim)

x_b A sub-block of s_{depth} in temporal domain (size : b_dim)

N_t Temporary vector (size : $depth$)

Initialization :

$n\_b \leftarrow 0 \quad k \leftarrow 0$

for *each* $i \in [0,\ldots,b\_dim[$ **do**

$h_w[i] \leftarrow 0.54 - 0.46\cos(2\pi\frac{i}{b\_dim})$

$X_{awp}[i] \leftarrow 0$

Algorithm 3 : Estimation of the depth signal noise in frequency domain using averaged Welch's periodogram with a 50% overlapping between blocks of size b_dim

(continued on next page)

1. Averaged Welch's periodogram estimation

▷**for** *each j = 0 to (depth - b_dim) with j+= b_dim* **do**

$n\_b \leftarrow n\_b + 1$

▷ **for** *each* $i \in [0, \ldots, b\_dim[$ **do**

$x_b[i] \leftarrow s_{depth}[i+j] \times h_w[i]$

end

$X_{mdl} \leftarrow |\text{FFT1D}(x_b)|$

▷ **for** *each* $i \in [0, \ldots, b\_dim[$ **do**

$N_t[k] \leftarrow (X_{mdl}[i])^2$

$X_{awp}[i] \leftarrow X_{awp}[i] + (X_{mdl}[i])^2$

$k \leftarrow k+1$

end

end

▷ **for** *each* $j = \frac{b\_dim}{2}$ *to (depth -* $\frac{b\_dim}{2}$*) with j+= b_dim* **do**

$n\_b \leftarrow n\_b + 1$

▷ **for** *each* $i \in [0, \ldots, b\_dim[$ **do**

$x_b[i] \leftarrow s_{depth}[i+j] \times h_w[i]$

end

$X_{mdl} \leftarrow |\text{FFT1D}(x_b)|$

▷ **for** *each* $i \in [0, \ldots, b\_dim[$ **do**

$X_{awp}[i] \leftarrow X_{awp}[i] + (X_{mdl}[i])^2$

end

end

▷ **for** *each* $i \in [0, \ldots, b\_dim[$ **do**

$X_{awp}[i] \leftarrow \frac{X_{awp}[i]}{n\_b}$

end

Algorithm 4 : Estimation of the depth signal noise in frequency domain using averaged Welch's periodogram with a 50% overlapping between blocks of size *b_dim*
(continued on next page)

2. Depth signal noise estimation in frequency domain

▷ **for** *each* $i \in [0,\ldots,depth[$ **do**

$N_{depth}[i] \leftarrow (N_t[i] - X_{awp}[\text{i modulo b\_dim}])$

end

$n^f_{\text{depth}} \leftarrow 0$

▷ **for** *each* $i \in [0,\ldots,depth[$ **do**

$n^f_{\text{depth}} += \ |N_{depth}[i]|$

end

Algorithm 5 : Estimation of the depth signal noise in frequency domain using averaged Welch's periodogram with a 50% overlapping between blocks of size *b_dim*

Estimation of the features

---

0. Definitions

I	Aperiodic noise energy map (size : $height \times width$) (Input)
x_{sym}	Longitudinal axis estimation (column coordinate) (Input)
C_l, C_r	Vector of the horizontal summation of the aperiodic noise energy for the left (right) area of the longitudinal axis of the lower limbs (size : $height - h$) (Output)
C	Vector for the difference of C_l and C_r; and the value of the silhouette asymmetry (size : $height - h + 1$) (Output)
h	Row coord. of the beginning of the lower limbs of I ($h = 380$)
AS(I)	Silhouette asymmetry map of I
$\text{SURFA}_{1,2}$	Number of pixels located to the left and to the right of x_{sym} respectively above the lower limbs (on $SA(I)$)
$\text{SURFA}_{3,4}$	Number of pixels located to the left and to the right of x_{sym} respectively for lower limbs part (on $SA(I)$)

Algorithm 6 : Estimation of the features : Definitions (continued on next page)

1. Computation of vectors C_l, C_r and C

$k \leftarrow 0$

▷ **for** *each* $i \in [h, \ldots, height[$ **do**

$m_l \leftarrow 0$

▷ **for** *each* $j \in [0, \ldots, x_{sym}[$ **do**

$m_l \leftarrow m_l + I[i][j]$

$C_l[i-h] \leftarrow m_l$

$m_r \leftarrow 0$

▷ **for** *each* $j \in [x_{sym}, \ldots, width[$ **do**

$m_r \leftarrow m_r + I[i][j]$

$C_r[i-h] \leftarrow m_r$

$C[k] \leftarrow |m_l - m_r|$

$k \leftarrow k+1$

2. Computation of the silhouette asymmetry features

$x_{\text{sym}} \leftarrow$ Longitudinal axis estimation $[I]$ (cf. Algorithm)

AS(I) $\leftarrow$ "exclusive or" operation around the column coordinate x_{sym} of the binary silhouette I

$\text{SURFA}_{1,2,3,4} \leftarrow$ number of pixels located respectively above (right and left) and on the lower limbs

$\text{SIL}_1 \leftarrow 2\ (\text{SURFA}_1 + \text{SURFA}_4)$

$\text{SIL}_2 \leftarrow 2\ (\text{SURFA}_2 + \text{SURFA}_3)$

if $(\text{SIL}_1 > \text{SIL}_2)$ $\quad \text{C[k]} \leftarrow 1000\,(\ \text{SIL}_1\text{-SIL}_2)/\text{SIL}_1$

else $\quad \text{C[k]} \leftarrow 1000\ (\text{SIL}_2\text{-SIL}_1)/\text{SIL}_2$

Algorithm 7 : Estimation of the features

CHAPITRE 4

SALIENT OBJECT DETECTION BY LTP TEXTURE CHARACTERIZATION ON OPPOSING COLOR PAIRS UNDER SLICO SUPERPIXEL CONSTRAINT

Dans ce chapitre, nous exposons notre article publié dans la revue, " MDPI (Multidisciplinary Digital Publishing Institute) - Journal of Imaging ", intitulé : " **Salient Object Detection by LTP Texture Characterization on Opposing Color Pairs under SLICO Superpixel Constraint** ". Nous le présentons dans sa langue originale de publication.

4.1 Abstract

The effortless detection of salient objects by humans has been the subject of research in several fields, including computer vision, as it has many applications. However, salient object detection remains a challenge for many computer models dealing with color and textured images. Most of them process color and texture *separately* and therefore implicitly consider them as independent features which is not the case in reality. Herein, we propose a novel and efficient strategy, through a simple model, almost without internal parameters, which generates a robust saliency map for a natural image. This strategy consists of integrating color information into local textural patterns to characterize a color micro-texture. It is the simple, yet powerful LTP (Local Ternary Patterns) texture descriptor applied to opposing color pairs of a color space that allows us to achieve this end. Each color micro-texture is represented by a vector whose components are from a superpixel obtained by the SLICO (Simple Linear Iterative Clustering with zero parameter) algorithm, which is simple, fast and exhibits state-of-the-art boundary adherence. The degree of dissimilarity between each pair of color micro-textures is computed by the FastMap method, a fast version of MDS (Multi-dimensional Scaling) that considers the color micro-textures' non-linearity while preserving their distances. These degrees of dissimilarity give us an intermediate saliency map for each RGB (Red–Green–Blue), HSL (Hue–Saturation–Luminance), LUV (L for luminance, U and V represent chroma-

ticity values) and CMY (Cyan–Magenta–Yellow) color space. The final saliency map is their combination to take advantage of the strength of each of them. The MAE (Mean Absolute Error), MSE (Mean Squared Error) and F_β measures of our saliency maps, on the five most used datasets show that our model outperformed several state-of-the-art models. Being simple and efficient, our model could be combined with classic models using color contrast for a better performance.

4.2 Introduction

Humans—or animals in general—have a visual system endowed with attentional mechanisms. These mechanisms allow the human visual system (HVS) to select from the large amount of information received that which is relevant and to process in detail only the relevant aspects [1]. This phenomenon is called visual attention. This mobilization of resources for the processing of only a part of whole information allows its rapid processing. Thus the gaze is quickly directed towards certain objects of interest. For living beings, this can sometimes be vital as they can decide whether they are facing prey or a predator [2].

Visual attention is carried out in two ways, namely *bottom-up attention* and *top-down attention* [3]. *Bottom-up attention* is a process which is fast, automatic, involuntary and directed by the image properties almost exclusively [1]. The *top-down attention* is a slower, voluntary mechanism directed by cognitive phenomena such as knowledge, expectations, rewards, and current goals [4]. In this work, we focus on the *bottom-up attentional mechanism* which is image-based.

Visual attention has been the subject of several research works in the fields of cognitive psychology [5, 6] and neuroscience [7], to name a few. Computer vision researchers have also used the advances in cognitive psychology and neuroscience to set up computational visual saliency models that exploit this ability of the human visual system to quickly and efficiently understand an image or a scene. Thus, many computational visual saliency models have been proposed and are mainly subdivided into two categories : conventional models (e.g., Yan et al. model [8]) and deep learning models (e.g.,

Gupta et al. model [9]). For more details, most of the models can be found in these works [10–12]).

Computational visual saliency models have several applications such as image/video compression [13], image correction [14], iconography artwork analysis [15], image retrieval [16], advertisements optimization [17], aesthetics assessment [18], image quality assessment [19], image retargeting [20], image montage [21], image collage [22], object recognition, tracking, and detection [23], to name but a few.

Computational visual saliency models are oriented to either eye fixation prediction or salient object segmentation or detection. The latter is the subject of this work. Salient object detection is materialized with saliency maps. A saliency map is represented by a grayscale image in which an image region must be whiter as it differs significantly from the rest of the image in terms of shape, set of shapes with a color, mixture of colors, movement, or a discriminating texture or generally any attribute perceived by the human visual system.

Herein, we propose a simple and nearly parameter-free model which gives us an efficient saliency map for a natural image using a new strategy. The proposed model, contrary to classical salient detection methods, uses texture and color features in a way that integrates color in texture features using simple and efficient algorithms. Indeed, the *texture* is a ubiquitous phenomenon in natural images : images of mountains, trees, bushes, grass, sky, lakes, roads, buildings, and so forth appear as different types of texture. Haidekker [24] argues that *texture* and shape analysis are very powerful tools for extracting image information in an unsupervised manner. This author adds that the *texture* analysis has become a key step in the quantitative and unsupervised analysis of biomedical images [24]. Other authors, such as Knutsson and Granlund [25], Ojala et al. [26], agree that *texture* is an important feature for scene analysis of images. Knutsson and Granlund also claim that the presence of a *texture* somewhere in an image is more a rule than an exception. Thus, *texture* in the image has been shown to be of great importance for image segmentation, interpretation of scenes [27], in face recognition, facial expression recognition, face authentication, gender recognition, gait recognition and age estimation, to just name a few [28]. In addition, natural images are usually also color

images and it is then important to take this factor into account as well. In our application, the color is taken into account and integrated in an original way, *via* the extraction of the textural characteristics made on the pairs of opposing color spaces.

Although there is much work relating to *texture*, there is no formal definition of *texture* [25]. There is also no agreement on a single technique for measuring texture [27, 28]. Our model uses the LTP (local ternary patterns) [29] texture measurement technique. The LTP (local ternary patterns) is an extension of local binary pattern (LBP) with three code values instead of two for LBP. LBP is known to be a powerful texture descriptor [28, 30]. Its main qualities are invariance against monotonic gray level changes and computational simplicity and its drawback is that it is sensitive to noise in uniform regions of the image.

In contrast, LTP is more discriminant and less sensitive to noise in uniform regions. The LTP (Local Ternary Patterns) is therefore better suited to tackle our salience detection problem. Certainly, the presence in natural images of several patterns make the detection of salient objects complex. However, the model we propose does not just focus on the patterns in the image by processing them separately from the colors as most models do [31, 32] but it takes into account both the presence in natural images of several patterns and color, not separately. This task of integrating color in texture features is accomplished through LTP (Local Ternary Patterns) applied to opposing color pairs of a givencolor space. The LTP describes the local textural patterns for a grayscale image through a code assigned to each pixel of the image by comparing it with its neighbours. When LTP is applied to an opposing color pair, the principle is similar to that used for a grayscale image. However, for LTP on an opposing color pair, the local textural patterns are obtained thanks to a code assigned to each pixel, but the value of the pixel of the first color of the pair is compared to the equivalents of its neighbours in the second color of the pair. The color is thus integrated to the local textural patterns. In this way, we characterize the color micro-textures of the image without separating the textures in the image and the colors in this same image. The color micro-textures' boundaries correspond to the superpixel obtained thanks to the SLICO (Simple Linear Iterative Clustering with zero parameter) algorithm [33] which is faster and exhibits state-of-the-art boun-

dary adherence. We would like to point out that there are other superpixels algorithms that have a good performance such as the AWkS algorithm [34] ; however, we chose SLICO because it is fast and almost parameter-free. A feature vector representing the color micro-texture is obtained by the concatenation of the histograms of the superpixel (defining the micro-texture) of each opposing color pair. Each pixel was then characterized by a vector representing the color micro-texture to which it belongs. We then compared the color micro textures characterizing each pair of pixels of the image being processed thanks to the fast version of the MDS (multi-dimensional scaling) method *FastMap* [35]. This comparison permits us to capture the degree of a pixel's uniqueness or a pixel's rarity. The FastMap method will allow this capture while taking into account the non-linearities in the representation of each pixel. Finally, since there is no single color space suitable for color texture analysis [36], we combined the different maps generated by FastMap from different color spaces (section 4.4.1), such as RGB, HSL, LUV and CMY, to exploit each other's strengths in the final saliency map.

Thus, the contribution of this work is twofold :

- we propose an unexplored approach to salient object detection. Indeed, our model *integrates* the color information into the texture whereas most of the models in the literature that use these two visual characteristics, namely color and texture, process them *separately* thus implicitly considering them as independent characteristics. Our model, on the other hand, allows us to compute saliency maps that take into account the interdependence of color and texture in an image as they are in reality ;

- we also use the *FastMap* method which is conceptually both local and global allowing us to have a simple and efficient model whereas most of the models in the literature use either a local approach or a global approach and other models combine these approaches in salient object detection.

Our model highlights the interest in opposing colors for the salient object detection problem. In addition, this model could be combined and be complementary with more

classical approaches using the contrast ratio. Moreover, our model can be parallelized (using the massively parallel processing power of GPUs : graphics processing units) by processing each opposing color pair in parallel.

The rest of this work is organized as follows : Section 5.3 presents some models related to this approach with an emphasis on the features used and how their dissimilarities are computed. Section 4.4 presents our model in detail. Section 4.5 describes the datasets used, our experimental results, the impact of the color integration in texture and the comparison of our model with state-of-the-art models. Section 5.6 discusses our results but also highlights the strength of our model related to our results. Section 4.7 concludes this work.

4.3 Related Work

Most authors define salient object detection as a capture of the uniqueness, distinctiveness, or rarity of a pixel, a superpixel, a patch, or a region of an image [11]. The problem of detecting salient objects is therefore to find the best characterization of the pixel, the patch or the superpixel and to find the best way to compare the different pixels (patch or superpixel) representation to obtain the best saliency maps. In this section, we present some models related to this work approach with an emphasis on the features used and how their dissimilarities are computed.

Thanks to studies in cognitive psychology and neuroscience, such as those by Treisman and Gelade [37], Wolfe et al. [6, 38] and Koch and Ullman [7], the authors of the seminal work of Itti et al. [39]—oriented eye fixation prediction—chose as features : color, intensity and orientation. Frintrop et al. [40], adapting the Itti et al. model [39] for salient objects segmentation—or detection—chose color and intensity as features. In the two latter models, the authors used pyramids of Gaussian and center-surround differences to capture the distinctiveness of pixels.

The Achanta et al. model [41] and the histogram-based contrast (HC) model [42] used color in CIELab space to characterize a pixel. In the latter model, the pixel's saliency is obtained using its color contrast to all other pixels in the image by measuring

the distance between the pixel for which they are computing saliency and all other pixels in the image; this is coupled with a smoothing procedure to reduce quantization artifacts. The Achanta et al. model [41] computed a pixel's saliency on three scales. For each scale, this saliency is computed as the Euclidean distance between the average color vectors of the inner region R_1 and that of the outer region R_2, both centered on that pixel mentioned above.

Joseph and Olugbara [43] used color histogram clustering to determine suitable homogeneous regions in image and compute each region saliency based on color contrast, spatial features, and center prior.

Guo and Zhang [44], in the phase spectrum of the Quaternion Fourier Transform model, represent each image's pixel by a Quaternion that consists of color, intensity and a motion feature. A Quaternion Fourier Transform (QFT) is then applied to that representation of each pixel. After setting the module of the result of the QFT to 1 to keep only the phase spectrum in the frequency domain, this result is used to reconstruct the Quaternion in spatial space. The module of this reconstructed Quaternion is smoothed with a Gaussian filter and this then produces the spatio-temporal saliency map of their model. For static images the motion feature is set to zero.

Other models also take color and position as features to characterize a region or patch instead of a pixel [42, 45, 46]. They differ, however, in how they obtain the salience of a region or patch. Thus, the region-based contrast (RC) model [42] measured the region saliency as the contrast between this region and the other regions of the image. This contrast is also weighted depending on the spatial distance of this region relative to the other regions of the image.

In the Perazzi et al. model [45], contrast is measured by the uniqueness rate and the spatial distribution of small perceptually homogeneous regions. The uniqueness of a region is calculated as the sum of the Euclidean distances between its color and the color of each region weighted by a Gaussian function of their relative position. The spatial distribution of a region is given by the sum of the Euclidean distances between its position and the position of each region weighted by a Gaussian function of their relative color. The region saliency is a combination of its uniqueness and its spatial distribution.

Finally, the saliency of each pixel in the image is a linear combination of the saliency of homogeneous regions. The weight for each region's saliency of this sum is a Gaussian function of the Euclidean distances between the color of the pixel and the colors of the homogeneous regions and the Euclidean distances between its spatial position and theirs. In the Goferman et al. model [46], the dissimilarity between two patches is defined as directly proportional to the Euclidean distance between the colors of the two patches and inversely proportional to their relative position normalized to be between 0 and 1. The salience of a pixel at a given scale is then 1 minus the inverse of the exponential of the mean of the dissimilarity between the patch centered on this pixel and the patches which are more similar to it ; the final saliency of the pixel being the average of the saliency of the different scales to which they add the context.

Some models focus on the patterns as features but they compute patterns separately from colors [31, 32]. For example Margolin et al. [31] defined a salient object as consisting of pixels whose local neighborhood (region or patch) is distinctive in both color and pattern. The final saliency of their model is the product of the color and pattern distinctness weighted by a Gaussian to add a center-prior.

As Frintrop et al. [40] stated, most saliency systems use intensity and color features. They are differentiated by the feature extraction and the general structure of the models. They have in common the computation of the contrast relative to the features chosen since the salient objects are so because of the importance of their dissimilarities with their environment. However, models in the literature differ on how these dissimilarities are obtained. Even though there are many salient object detection models, the detection of salient objects remains a challenge [47].

The contribution of this work is twofold :

- we propose an unexplored approach to the detection of salient objects. Indeed, we use for the first time in the salient object detection, to our knowledge, the feature *color micro-texture* in which the *color* feature is integrated *algorithmically* into the local textural patterns for salient object detection. This is done by applying LTP (Local Ternary Patterns) to each of the opposing color pairs of a chosen color

space. Thus, in salient object detection computation, we *integrate* the color information in the texture while most of the models in the literature which use these two visual features, namely color and texture, perform this computation *separately*;

- we also use the *FastMap* method which, conceptually, is both local and global while most of the models in the literature use either a local approach or a global approach and other models combine these approaches in saliency detection. FastMap can be seen as a nonlinear one-dimensional reduction of the micro-texture vector taken locally around each pixel with the interesting constraint that the (Euclidean) difference existing between each pair of (color) micro textural vectors (therefore centered on two pixels of the original image) is preserved in the reduced (one-dimensional) image and is represented (after reduction) by two gray levels separated by this same distance. After normalization, a saliency measure map (with range values between 0 and 1) is estimated in which lighter regions are more salient (higher relevance weight) and darker regions are less salient.

The model we propose in this work is both simple and efficient while being almost parameter free. Being simple and being different from the classic salience detection models which use the color contrast strategy between a region and other regions of an image, our model could therefore be effectively combined with these models for a better performance. Moreover, by processing each opposing color pair in parallel, our model can be parallelized using the massively parallel processing power of GPUs (graphics processing units). In addition, it produces good results in comparison with the state-of-the-art models in [48] for the ECSSD, MSRA10K, DUT-OMRON, THUR15K and SED2 datasets.

4.4 Proposed Model

4.4.1 Introduction

In this work, we present a model that does not require any learning basis and that highlights the interest of color opposing for the salient object detection problem. The main idea of our model is to algorithmically integrate the color feature into the textural

characteristics of the image and then to describe this vector of textural characteristics by an intensity histogram.

To incorporate the color into the texture description, we mainly relied on the opponent color theory. This theory states that the HVS interprets information about color by processing signals from the cone and rod cells in an antagonistic manner. This theory was suggested as a result of the way in which photo-receptors are interconnected neurally and also by the fact that it is made more efficient for the HVS to record differences between the responses of cones, rather than each type of cone's individual response. The opponent color theory suggests that there are three opposing channels called the cone photo-receptors, which are linked together to form three pairs of opposite colors. This theory was first computer modeled for incorporating the color into the LBP texture descriptor by Mäenpää and Pietikäinen [28, 49]. It was called Opponent-Color LBP (OC-LBP), and was developed as a joint color-texture operator, thus generalizing the classical LBP, which normally applies to monochrome textures.

Our model is locally based (for each pixel) on nine opposing color pairs and semi-locally, on the set of estimated superpixels of the input image. These nine opposing color pairs are in the RGB (Red—Green—Blue) color space channel : RR, RG, RB, GR, GG, GB, BR, BG and BB (see section 4.4.2.2).

The LTP (Local Ternary Patterns) [29] texture characterization method is then applied to each opposing color pair to capture the features of the color micro-textures. At this stage, we obtain nine grayscale texture maps which already highlight the salient objects in the image as can be seen in Figure 4.1.

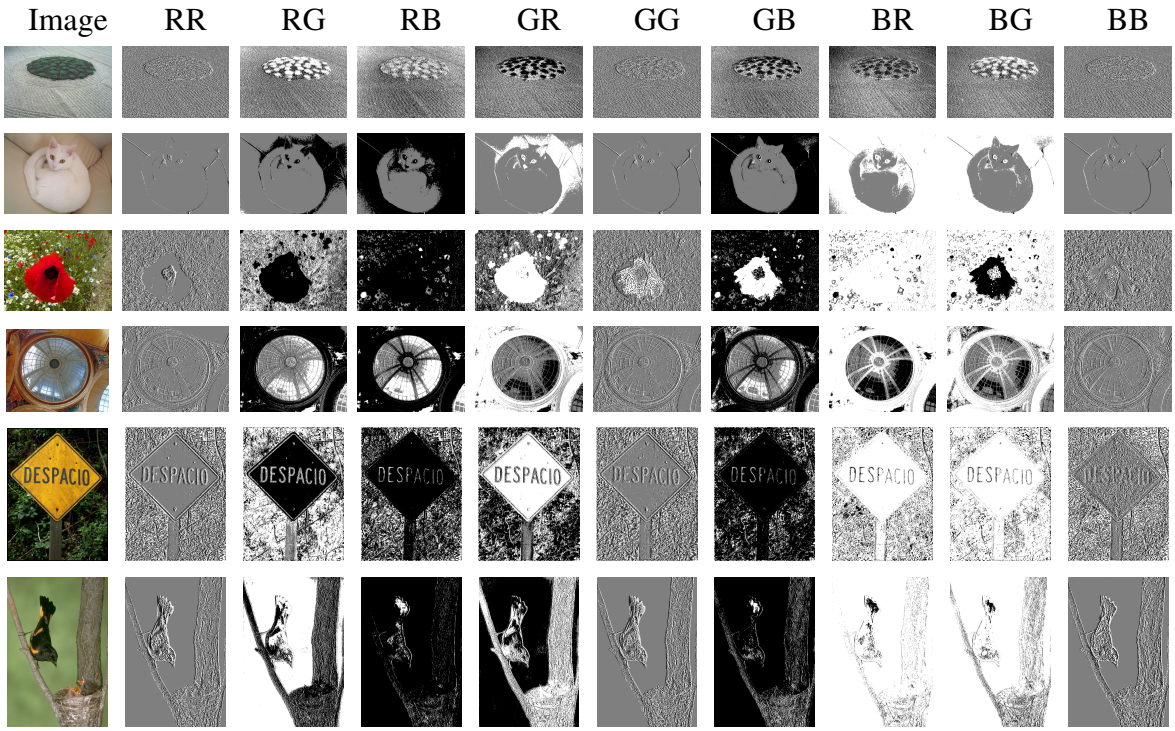


FIGURE 4.1 : Micro-texture maps given by LTP on the 9 opposing color pairs (for the RGB color space). We can notice that this LTP coding already highlights the salient objects.

We then consider each texture map as being composed of micro-textures that can be described by a gray level histogram. As it is not easy to determine in advance the size of each micro-texture in the image, we chose to use adaptive windows for each micro-texture. This is why we use superpixels in our model. To find these superpixels, our model uses the SLICO (Simple Linear Iterative Clustering with zero parameter) superpixel algorithm [33], which is a version of SLIC (Simple Linear Iterative Clustering). The SLICO is a simple, very fast algorithm that produces superpixels, which has the merit of adhering particularly well to the boundaries (see Figure 4.2) [33]. In addition, the SLICO algorithm (with its default internal parameters), has just one parameter : the number of superpixels desired.

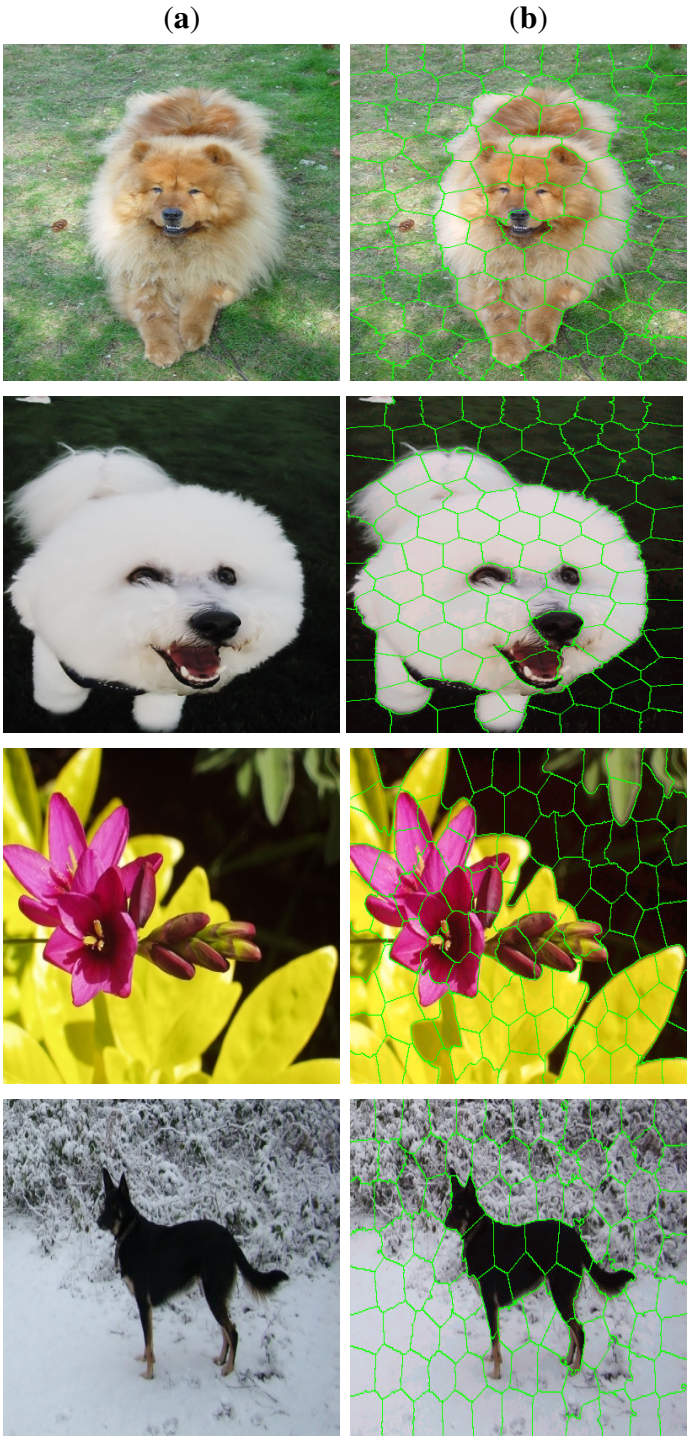


FIGURE 4.2 : Illustration of SLICO (Simple Linear Iterative Clustering with zero parameter) superpixels bounderies : (**a**) images ; (**b**) superpixels.

Thus, we characterize each pixel of each texture map by the gray level histogram of the superpixel to which it belongs. We thus obtain a histogram map for each texture map. The nine histogram maps are then concatenated pixel by pixel to have a single histogram map that characterizes the color micro-textures of the image. Each histogram of the latter is then a feature vector for the corresponding pixel.

The dissimilarity between pixels of the input color image is then given by the dissimilarity between their feature vectors. We quantify this dissimilarity thanks to the FastMap method which has the interesting property of non-linearly reducing in one dimension these feature vectors while preserving the structure in the data. More precisely, the FastMap allows us to find a configuration, in one dimension, that preserves as much as possible all the (Euclidean) distance pairs that initially existed between the different (high dimensional) texture vectors (and that takes into account the non-linear distribution of the set of feature vectors). After normalization between the range 0 and 1, the map estimated by the FastMap produces the Euclidean embedding (in near-linear time) which can be viewed as a *probabilistic* map, i.e., with a set of gray levels with high grayscale values for salient regions and low values for non-salient areas (see Figure 4.3 for the schematic architecture).

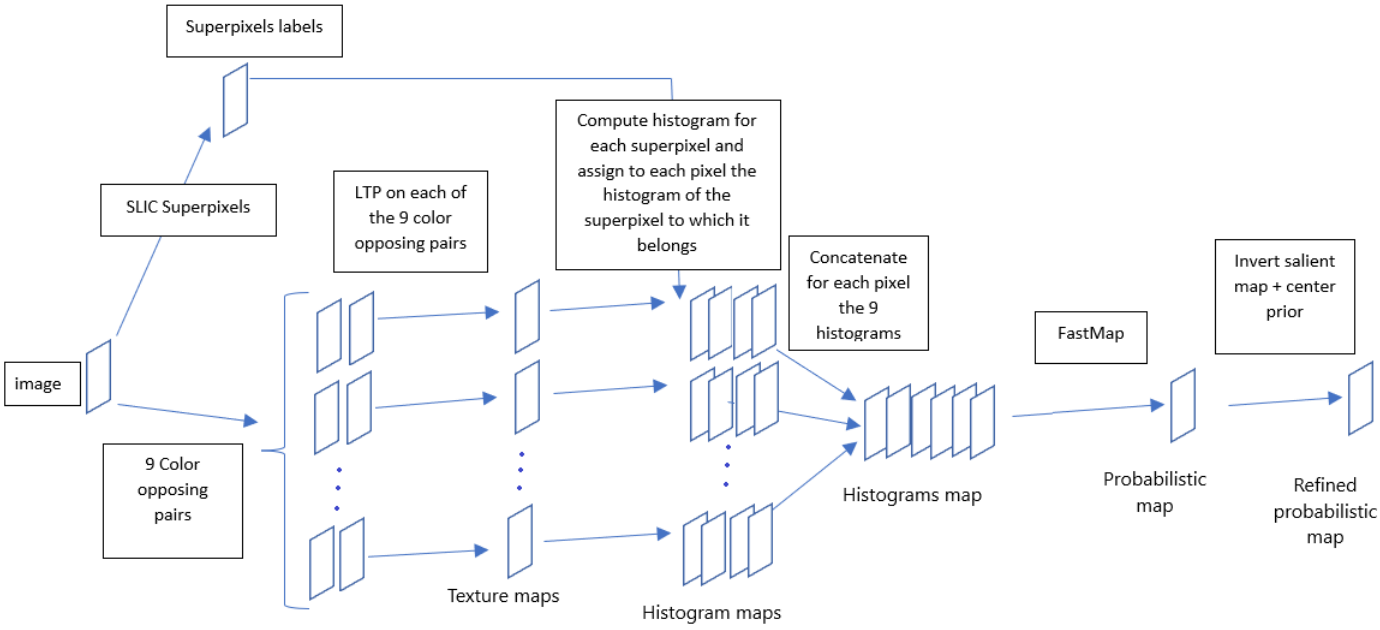


FIGURE 4.3 : Proposed model steps to obtain the refined probabilistic map from a color space (e.g., RGB : Red–Green–Blue).

As Borji and Itti [50] stated, almost all saliency approaches use just one color channel. The latter authors also argued that employing just one color space does not always lead to successful outlier detection. Thus, taking into account this argument, we used, in addition to the RGB color space the color spaces HSL, LUV and CMY. Finally, we combine the probabilistic maps obtained from these color spaces to obtain the desired saliency map. To combine the probabilistic maps from the different color spaces used, we reduce for each pixel a vector which is the concatenation of the averages of the values of the superpixel to which this pixel belongs successively in all the color spaces used. In the following section, we describe the different steps in detail.

4.4.2 LTP Texture Characterization on Opposing Color Pairs

4.4.2.1 Local Ternary Patterns (LTP)

Since LTP (*local ternary patterns*) is a kind of generalization of LBP (*local binary patterns*) [26, 51], let us first recall the LBP technique.

The local binary pattern $LBP_{P,R}$ labels each pixel of an image (see Equation (4.1)).

$$\mathrm{LBP}_{P,R}(x_c, y_c) = \sum_{p=0}^{P-1} s(g_p - g_c) 2^p, \tag{4.1}$$

with (x_c, y_c) being the pixel coordinate and :

$$s(z) = \begin{cases} 1 & \text{if } z \geq 0 \\ 0 & \text{if } z < 0, \end{cases}$$

where $z = g_p - g_c$.

The label of a pixel at the position (x_c, y_c) with g_c as gray level is a set of P binary digits obtained by thresholding each gray level value g_p of the p neighbour located at the distance R (see Figure 4.4) from this pixel by the value of the gray level g_c (p is one of the P chosen neighbors).

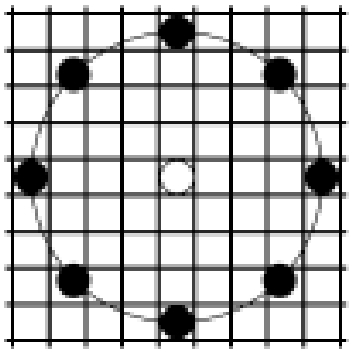
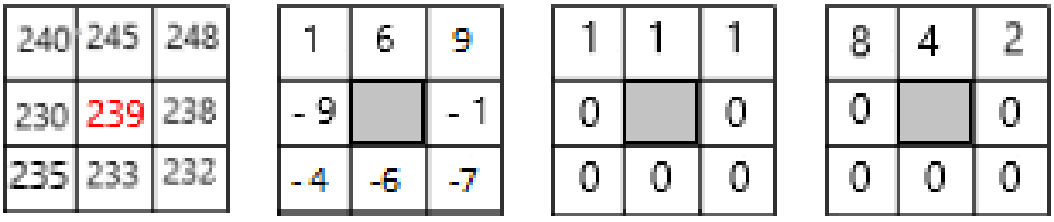


FIGURE 4.4 : Example of neighborhood (black disks) for a pixel (central white disk) for $LBP_{P,R}$ code computation : in this case $P = 8$, $R = 4$.

The set of binary digits obtained constitutes the label of this pixel or its LBP code (see Figure 4.5).



(**a**) (**b**) (**c**) (**d**)

FIGURE 4.5 : Example of LBP code computation for a pixel : LBP code is $2+4+8=14$ in this case. (**a**) pixel neighbourhood; $g_c = 239$; (**b**) after thresholding; (**c**) pattern : 00001110; (**d**) code = 14.

Once this code is computed for each pixel, the characterization of the texture of the image (within a neighborhood) is approximated by a discrete distribution (histogram) of LBP codes of 2^P bins.

The LTP (local ternary patterns) [29] is an extension of LBP in which the function $s(z)$ (see Equation (4.1)) is defined as follows :

$$s(z) = \begin{cases} 2 & \text{if } z \geq t \\ 1 & \text{if } |z| < t \\ 0 & \text{if } z \leq -t, \end{cases}$$

where $z = g_p - g_c$.

The basic coding of LTP is, thus, expressed as :

$$\text{LTP}_{P,R}(x_c, y_c) = \sum_{p=0}^{P-1} s(g_p - g_c) 3^p. \tag{4.2}$$

Another type of encoding can be obtained by splitting the LTP code into two codes, LBP : Upper LBP code and Lower LBP code (see Figure 4.6). The LTP histogram is

then the concatenation of the histogram of the upper LBP code with that of the lower LBP code [29].

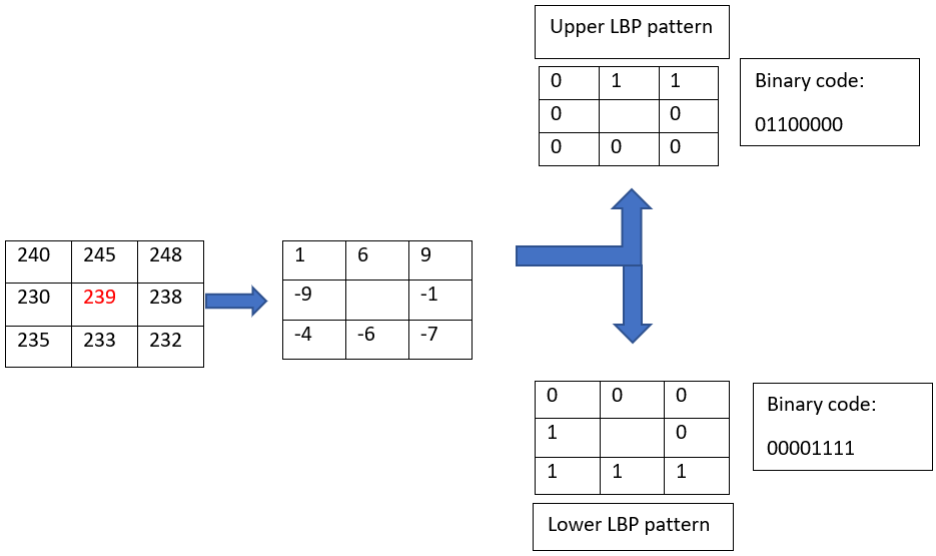


FIGURE 4.6 : Example of LTP code splitting with threshold t = 3

In our model we use the LTP basic coding because we use five neighbors for the central pixel. So the maximum size of the histograms is $3^5 = 243$. In addition, we re-quantized the histogram with levels/classes of 75 bins for computational reasons (thus greatly reducing the computational time for the next step using the FastMap algorithm while generalizing the feature vector a bit as this operation smoothes the histogram) and we have effectively noticed that this strategy produces slightly better results.

4.4.2.2 Opposing Color Pairs

To incorporate the color into the texture description, we rely on the color opponent theory. We thus used the color texture descriptor from Mäenpää and Pietikäinen [28, 49], called “Opponent Color LBP”. This one generalizes the classic LBP, which normally applies to grayscale textures. So instead of just one LBP code, one pixel gets a code for

every combination of two color channels (i.e., 9 opposing color pair codes). Example for RGB channels : RR (Red-Red), RG (Red-Green), RB (Red-Blue), GR (Green-Red), GG (Green-Green), GB (Green-Blue), BR (Blue-Red), BG (Blue-Green), BB (Blue-Blue) (see Figure 4.7).

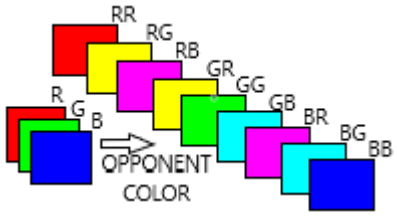


FIGURE 4.7 : Illustration of color opponent on RGB (Red Green Blue) color space with its 9 opposing color pairs (i.e., RR, RG, RB, GR, GG, GB, BR, BG, BB).

The central pixel is in the first color channel of the combination and the neighbors are picked in the second color (see Figure 4.8b).

(**a**) (**b**)

FIGURE 4.8 : (**a**) Pixel gray LBP code : the code for the central pixel (i.e., white small disk) is computed with respect to his neighbors (i.e., 8 black small disks). (**b**) Pixel opponent color LBP code for RG pair : the central pixel is in the first color channel (red) and the neighbous are picked in the second channel (green).

The histogram that describes the color micro-texture is the concatenation of the histograms obtained from each opposing color pair.

4.4.3 FastMap : Multi-Dimensional Scaling

The FastMap [35] is an algorithm which initially was intended to provide a tool allowing us to find objects similar to a given object, to find pairs of the most similar objects and to visualize distributions of objects in a desired space in order to be able to identify the main structures in the data, once the similarity or dissimilarity function is determined. This tool remains effective even for large collections of datasets, unlike classical multi-dimensional scaling (classic MDS). The FastMap algorithm matches objects of a certain dimension to points in a k-dimensional space while preserving distances between pairs of objects. This representation of objects from a large-dimensional space n to a smaller-dimensional space (dimension 1 or 2 or 3) allows the visualization of the structures of the distributions in the data or the acceleration of the search time for queries [35].

As Faloutsos and Lin [35] describe it, the problem solved by FastMap can be represented in two ways. First, FastMap can be seen as a means to represent N objects in a k-dimensional space, given the distances between the N objects, while preserving the distances between pairs of objects. Second, the FastMap algorithm can also be used in reducing dimensionality while preserving distances between pairs of vectors. This amounts to finding, given N vectors having n features each, N vectors in a space of dimension k—with $n \gg k$—while preserving the distances between the pairs of vectors. To do this, the objects are considered as points in the original space. The first coordinate axis is the line that connects the objects, called *pivots*. The pivots are chosen so that the distance separating them is at a maximum. Thus, to obtain these pivots, the algorithm follows the steps below :

- choose arbitrarily an object as the second pivot, i.e., the object O_b ;
- choose as the first pivot O_a, the object furthest from O_b according to the used distance ;
- replace the second pivot with the furthest object from O_a, that is, the object O_b ;
- return the objects O_a and O_b as pivots.

The axis of the pivots thus constitutes the first coordinate axis in the targeted k-dimensional space. All the points representing the objects are then projected orthogonally on this axis and in the H hyperplane of $n-1$ dimensions (perpendicular to the first axis already obtained) connecting the pivot objects O_a and O_b along the latter axis. The coordinates of a given object O_i on the first axis are given by :

$$x_i = \frac{d_{a,i}^2 + d_{a,b}^2 - d_{b,i}^2}{2d_{a,b}}, \tag{4.3}$$

where $d_{a,i}$, $d_{b,i}$ and $d_{a,b}$ are, respectively, the distance between the pivot O_a and object O_i, the distance between the pivot O_b and object O_i, the distance between the pivot O_a and the pivot O_b. The process is repeated up to the desired dimension, each time expressing :

1. the new distance $D'()$:

$$(D'(O'_i, O'_j))^2 = (D(O_i, O_j))^2 - (x_i - x_j)^2. \tag{4.4}$$

   For simplification,

$$D'(O'_i, O'_j) \equiv d'_{O'_i, O'_j},$$

   where x_i and x_j are the coordinates on the previous axis of respectively the object O_i and O_j.

2. the new pivots O'_a and O'_b constituting the new axis,

3. the coordinate of the projected object O'_i on the new axis :

$$x'_i = \frac{d'^2_{a',i} + d'^2_{a',b'} - d'^2_{b',i}}{2d'_{a',b'}}. \tag{4.5}$$

$O_{a'}$ and $O_{b'}$ are the new pivots according to the new distance expression $D'()$. The line that connects them is therefore the new axis.

After normalization between the range 0 and 1, the map estimated by the FastMap generates a *probabilistic* map, i.e., with a set of gray levels with high grayscale values

for salient regions and low values for non-salient areas. Nevertheless, in some (rare) cases, the map estimated by the FastMap algorithm can possibly present a set of gray levels whose amplitude values would be in completely the opposite direction (i.e., low grayscale values for salient regions and high values for non-salient areas). In order to put this grayscale mapping in the right direction (with high grayscale values associated with salient objects), we simply use the fact that a salient object/region is more likely to appear in the center of the image (or conversely unlikely on the edges of the image). To this end, we compute the Pearson correlation coefficient between the saliency map obtained by the FastMap and a rectangle, with a maximum intensity value and about half the size of the image, and located in the center of the image. If the correlation coefficient is negative (anti-correlation), we invert the signal (i.e., associate to each pixel its complementary gray value).

4.5 Experimental Results

In this section, we present our salient object detection model's results. In order to obtain the $\text{LTP}_{P,R}$ pixel's code (LTP code for simplification), we used an adaptive threshold. For a pixel at position (x_c, y_c) with value g_c, the threshold for its LTP code is a tenth of the pixel's value : $t = \frac{g_c}{10}$ (see Equation (4.2)). We chose this threshold because empirically it is this value that has given better results. The number of neighbors P around the pixel on a radius R used to find its LTP code in our model is $P = 5$ and $R = 1$. Thus the maximum value of the LTP code in our case is $3^5 - 1 = 242$. This makes the maximum size of the histogram characterizing the micro-texture in an opposing color pair to be $3^5 = 243$ which is then requantized with levels/classes of 75 bins (see Section 4.4.2). The superpixels that we use as adaptive windows to characterize the color micro-textures are obtained thanks to the SLICO (Simple Linear Iterative Clustering with zero parameter) algorithm which is faster and exhibits state-of-the-art boundary adherence. Its only parameter is the number of superpixels desired and is set to 100 in our model (which is also the value recommended by the author of the SLICO algorithm). Finally, we use in the combination to obtain the final saliency map, the color spaces RGB, HSL,

LUV and CMY.

We chose, for our experiments, images from public datasets, the most widely used in the salient object detection field [48] such as Extended Complex Scene Saliency Dataset (ECSSD) [52], Microsoft Research Asia 10,000 (MSRA10K) [42, 48], DUT-OMRON (Dalian University of Technology—OMRON Corporation) [53], THUR15K [54] and SED2 (Segmentation evaluation database with two salient objects) [55]. The ECSSD contains 1000 natural images and their ground truth. Many of its images are semantically meaningful, but structurally complex for saliency detection [52]. The MSRA10K contains 10,000 images and 10,000 manually obtained binary saliency maps corresponding to their ground truth. DUT-OMRON contains 5168 images and their binary mask. THUR15K is a dataset of images taken from the "Flickr" web site divided into five categories (butterfly, coffee mug, dog jump, giraffe, plane), each of which contains 3000 images. Only 6233 images have ground truths. The images of this dataset represent real world scenes and are considered complex for obtaining salient objects [54]. The SED2 dataset has 100 images and their ground truth.

We used for the evaluation of our salient object detection model the Mean Absolute Error (MAE), the Mean Squared Error (MSE), the Precision-Recall curve (PR), the F_β measure curve and the F_β measure with $\beta^2 = 0.3$. The MSE measure results for ECSSD, MSRA10K, DUT-OMRON, THUR15K and SED2 datasets are shown in Table 4.1. We compared the MAE (Mean Absolute Error) and the F_β measure of our model with the 29 state-of-the-art models from Borji et al. [48] and our model outperformed many of them as shown in Table 4.2. In addition, we can see that our model succeeded to obtain saliency maps close to the ground truth for each of the datasets used although for some images it failed, as shown in Figure 4.9.

TABLE 4.1 : Our model's MSE measure results for ECSSD, MSRA10K, DUT-OMRON, THUR15K and SED2 datasets (for MSE, the smaller value is the best).

	ECSSD	**MSRA10K**	**DUT-OMRON**	**THUR15K**	**SED2**
MSE	0.135	0.105	0.130	0.116	0.177

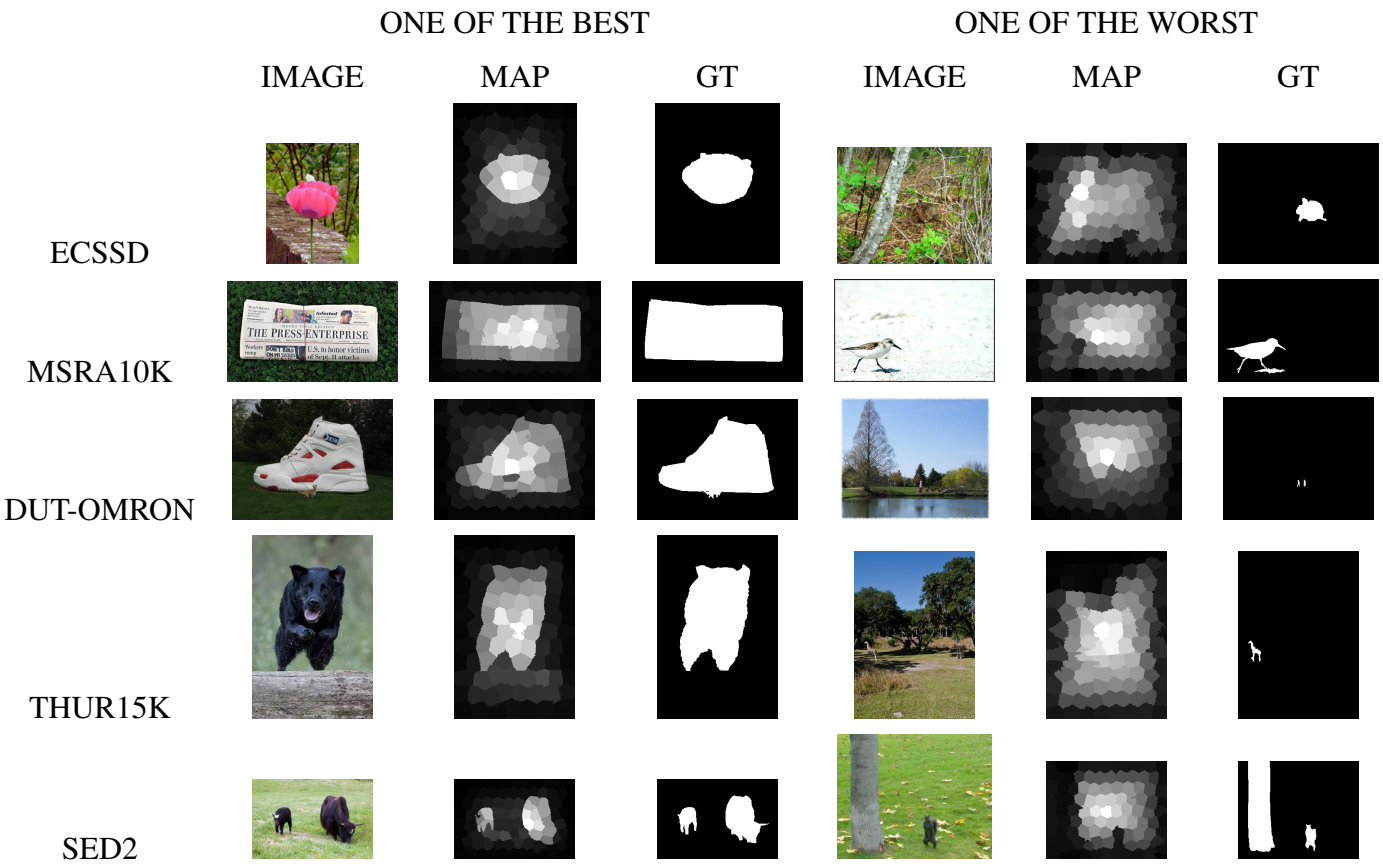


FIGURE 4.9 : One of the best and one of the worst saliency maps for each dataset used in this work.

4.5.1 Color Opposing and Colors Combination Impact

Our results show that combining the opposing color pairs improves the individual contribution of each pair to the F_β measure and the Precision-Recall as shown for the RGB color space by the F_β measure curve (Figure 4.10) and the Precision–Recall curve (Figure 4.11). The combination of the color spaces RGB, HSL, LUV and CMY also improves the final result as can be seen from the F_β measure curve and the precision–recall curve (see Figures 4.12 and 4.13).

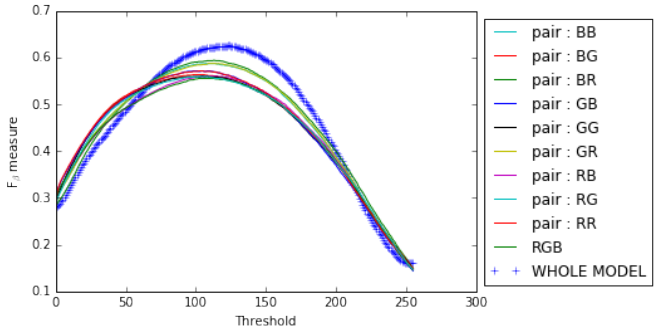


FIGURE 4.10 : F_β measure curves for opposing color pairs, RGB color space and the whole model on the ECSSD dataset.

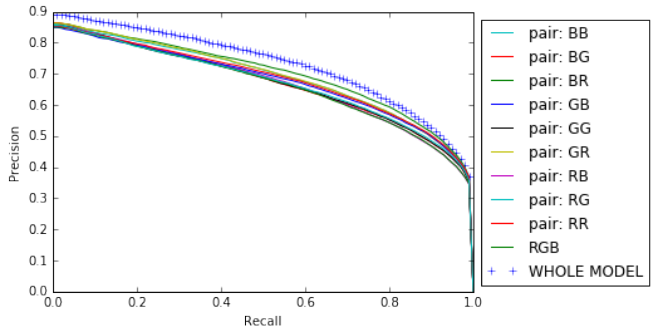


FIGURE 4.11 : Precision–Recall curves for opposing color pairs, RGB color space and the whole model on the ECSSD dataset.

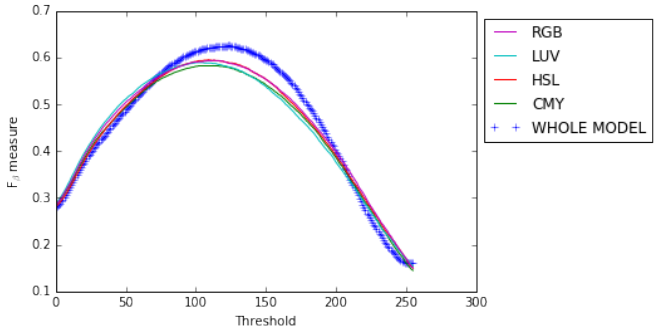


FIGURE 4.12 : F_β measure curves for color spaces RGB, HSL, LUV and CMY and the whole model on the ECSSD dataset.

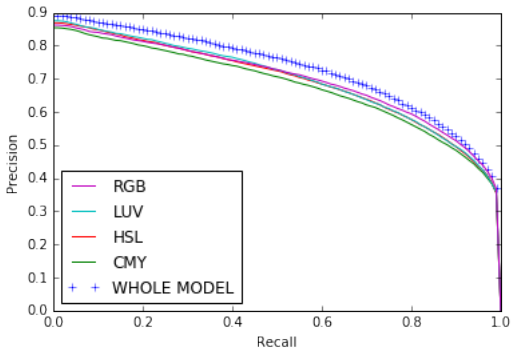


FIGURE 4.13 : Precision-Recall curves for color spaces RGB, HSL, LUV and CMY and the whole model on the ECSSD dataset.

4.5.2 Comparison with State-of-the-Art Models

In this work, we studied a method that requires no learning basis. Therefore, we did not include machine learning methods in these comparisons.

We compared the MAE (Mean Absolute Error) and F_β measure of our model with the 29 state-of-the-art models from Borji et al. [48] and our model outperformed many of them as shown in Table 4.2. Table 4.3 shows the F_β measure and Table 4.4 the Mean Absolute Error (MAE) of our model on ECSSD, MSRA10K, DUT-OMRON, THUR15K

and SED2 datasets compared to some state-of-the-art models.

TABLE 4.2 : Number of models among the 29 state-of-the-art models from Borji et al. [48] outperformed by our model on MAE and F_β measure results.

	ECSSD	**MSRA10K**	**DUT-OMRON**	**THUR15K**	**SED2**
F_β	21	11	12	17	4
MAE	11	8	6	10	3

TABLE 4.3 : Our model's F_β measure results compared with some state-of-the-art models from Borji et al. [48].

MODELS	**ECSSD**	**MSRA10K**	**DUT—OMRON**	**THUR15K**	**SED2**
GR [56]	0.664	0.816	0.599	0.551	0.798
MNP [57]	0.568	0.668	0.467	0.495	0.621
LBI [58]	0.586	0.696	0.482	0.519	0.692
LMLC [59]	0.659	0.801	0.521	0.540	0.653
SVO [60]	0.639	0.789	0.557	0.554	0.744
SWD [61]	0.624	0.689	0.478	0.528	0.548
HC [42]	0.460	0.677	0.382	0.386	0.736
SEG [62]	0.568	0.697	0.516	0.500	0.704
CA [46]	0.515	0.621	0.435	0.458	0.591
FT [63]	0.434	0.635	0.381	0.386	0.715
AC [41]	0.411	0.520	0.354	0.382	0.684
OURS	0.729	0.781	0.531	0.581	0.635

TABLE 4.4 : Our model's MAE results compared with some state-of-the-art models from Borji et al. [48] (for MAE, the smaller value is the best).

MODELS	ECSSD	MSRA10K	DUT-OMRON	THUR15K	SED2
GR [56]	0.285	0.198	0.259	0.256	0.189
MNP [57]	0.307	0.229	0.272	0.255	0.215
LBI [58]	0.280	0.224	0.249	0.239	0.207
LMLC [59]	0.260	0.163	0.277	0.246	0.269
SVO [60]	0.404	0.331	0.409	0.382	0.348
SWD [61]	0.318	0.267	0.310	0.288	0.296
HC [42]	0.331	0.215	0.310	0.291	0.193
SEG [62]	0.342	0.298	0.337	0.336	0.312
CA [46]	0.310	0.237	0.254	0.248	0.229
FT [63]	0.291	0.235	0.250	0.241	0.206
AC [41]	0.265	0.227	0.190	0.195	0.206
OURS	0.257	0.215	0.267	0.236	0.289

4.5.2.1 Comparison with Two State-of-the-Art Models HS and CHS

We have chosen to compare our model to HS [8] and CHS [52] state-of-the-art models because on the one hand they are among the best state-of-the-art models and on the other hand our model has some similarities with these two models. Indeed, our model is a combination of energy-based models MDS and SLICO and is based on the color texture while the two state-of-the-art models are energy based models. Moreover, their energy function is based on a combination of the color and the pixel coordinates.

First, the visual comparison of some of our saliency maps with those of two state-of-the-art models ("Hierarchical saliency detection" : HS [8] and "Hierarchical image

saliency detection on extended CSSD" : CHS [52] models) shows that our saliency maps are of good quality (see Figure 4.14).

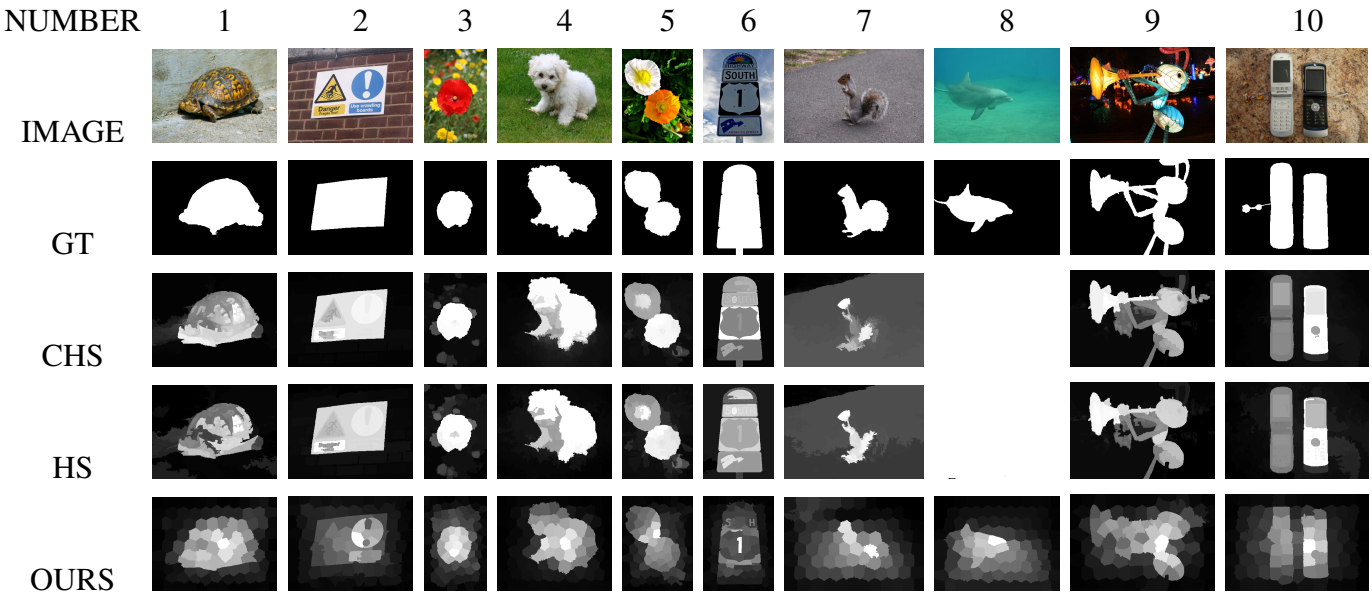


FIGURE 4.14 : Comparison of some result images for HS [8], CHS [52] and our model. For image number 8, the HS [8] and CHS [52] models find white salient maps (GT : Ground Truth).

Second, we compared our model with the two state-of-the-art HS [8] and CHS [52] models with respect to the precision-recall, F_β measure curves (see Figures 4.15 and 4.16) and MSE (Mean Squared Error). Table 4.5 shows that our model outperformed them on the MSE measure.

TABLE 4.5 : Our model's MSE measure results compared with two state-of-the-art HS [8] and CHS [52] models for the ECSSD dataset (for MSE, the smaller value is the best).

	OURS	**HS [8]**	**CHS [52]**
MSE	0.135	0.163	0.220

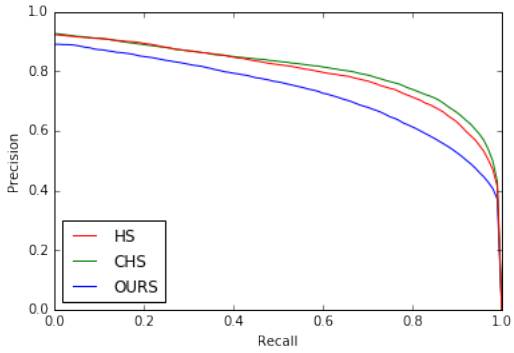


FIGURE 4.15 : Precision–Recall curves for HS [8], CHS [52] models and ours on the ECSSD dataset.

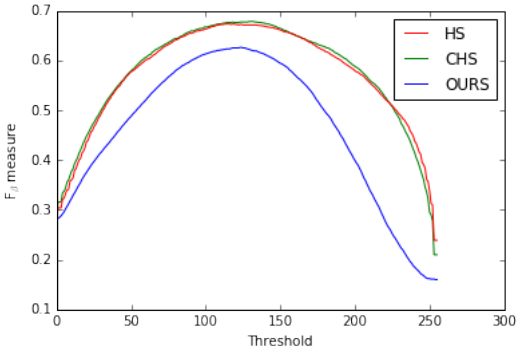


FIGURE 4.16 : F_β measure curves for HS [8], CHS [52] models and ours on the ECSSD dataset.

Thus, our model is better than HS [8] and CHS [52] for the MSE measure while both models are better for the F_β and Precision–Recall.

Our model also outperformed some of the recent methods for F_β-measure on the ECSSD dataset as shown in Table 4.6.

TABLE 4.6 : Our model's F_β-measure results compared with some of the recent models for the ECSSD dataset.

	OURS	**Wu et al. [64]**	**Yuan et al. [65]**	**Zhang et al. [66]**
F_β-measure	0.729	0.718	0.714	0.725

4.6 Discussion

Our model has less dispersed MAE measures than the HS [8] and CHS [52] models, which are among the best models of the state-of-the-art. This can be observed in Figure 4.17 but is also shown by the standard deviation which for our model is 0.071 (mean = 0.257), for HS [8] is 0.108 (mean = 0.227), and for CHS [52] is 0.117 (mean = 0.226). For HS [8] the relative error between the two standard deviations is $\frac{(0.108-0.071)\times 100}{0.071} = 52.11\%$ while for CHS [52] it is $\frac{(0.117-0.071)\times 100}{0.071} = 64.78\%$.

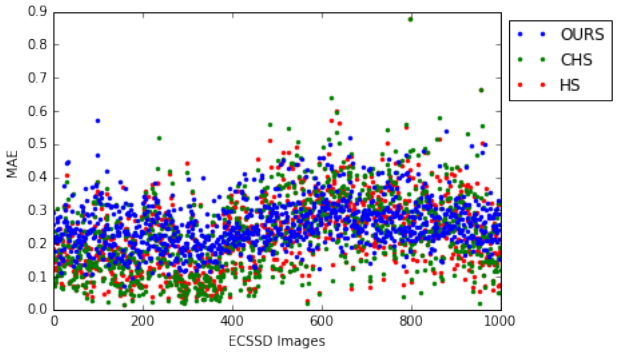


FIGURE 4.17 : Comparison of the MAE measure dispersion for our model and the HS [8], CHS [52] models on the ECSSD dataset (for MAE, the smaller value is the best).

Our model is stable on new data. Indeed, a model with very few internal parameters is supposed to be more stable for different datasets. We also noticed that nearly 500

first image numbers of the ECSSD dataset are less complex than the rest of the images in this dataset by observing the different measures (see Table 4.7 and Figures 4.17 and 4.18). However, it is clear that the drop in performance over the last 500 images from the ECSSD dataset is less pronounced for our model than for the HS [8] and CHS [52] models (see Table 4.7). This can be explained by the stability of our model (we used to compute these measures except for MAE a threshold, for each image, which gives the best F_β measure. It should also be noted that the images are ordered only by their numbers in the ECSSD dataset).

TABLE 4.7 : Performance drop for Precision and MAE measures with respect to image numbers 0 to 500 (*) and 500 to 1000 (**) of the ECSSD dataset (for MAE, the smaller value is the best).

	Precision			**MAE**		
	Ours	**HS**	**CHS**	**Ours**	**HS**	**CHS**
(*)	0.832	0.919	0.921	0.234	0.176	0.172
(**)	0.737	0.791	0.791	0.279	0.278	0.280
Gap	0.095	0.128	0.130	0.045	0.102	0.108

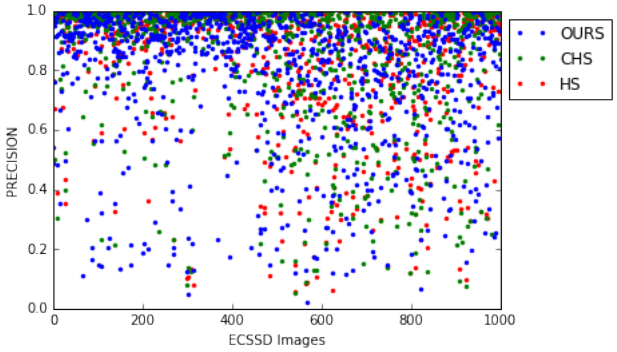


FIGURE 4.18 : Comparison of the precision measure dispersion for our model and the HS [8], CHS [52] models on the ECSSD dataset.

Our model is also relatively stable for an increase or decrease of its unique internal parameter. Indeed, by increasing or decreasing the number of superpixels, which is the only parameter of the SLICO algorithm, we find that there is almost no change in the results as shown by the MAE and F_β measure (see Table 4.8) and F_β measure and precision-recall curves for 50, 100 and 200 superpixels (see Figures 4.19 and 4.20).

TABLE 4.8 : Our model's F_β measure and MAE results for 50, 100 and 200 superpixels (ECSSD dataset).

Superpixels	**50**	**100**	**200**
F_β measure	0.722	0.729	0.725
MAE	0.257	0.257	0.257

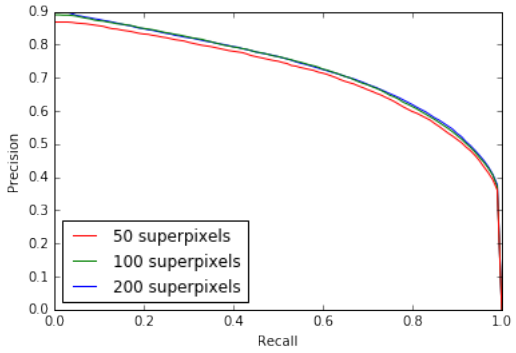


FIGURE 4.19 : Precision–Recall model's curves for 50, 100, 200 superpixels (ECSSD dataset).

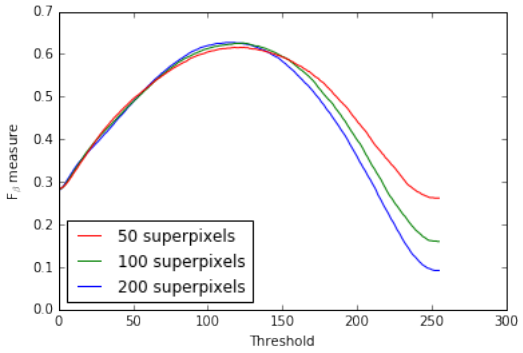


FIGURE 4.20 : F_β measure model's curves for 50, 100, 200 superpixels (ECSSD dataset).

4.7 Conclusions

In this work, we presented a simple, nearly parameter-free model for the estimation of saliency maps. We tested our model on the complex ECSSD dataset for which the average measures of MAE = 0.257 and F_β measure = 0.729, and on the MSRA10K dataset. We also tested on THUR15K, which represents real world scenes and is considered complex for obtaining salient objects, and on DUT-OMRON and SED2 datasets.

The novelty of our model is that it only uses the textural feature after incorporating the color information into these textural features thanks to the opposing color pairs theory of a given color space. This is made possible by the LTP (Local Ternary Patterns) texture descriptor which, being an extension of LBP (Local Binary Patterns), inherits its strengths while being less sensitive to noise in uniform regions. Thus, we characterize each pixel of the image by a feature vector given by a color micro-texture obtained thanks to the SLICO superpixel algorithm. In addition, the FastMap algorithm reduces each of these feature vectors to one dimension while taking into account the non-linearities of these vectors and preserving their distances. This means that our saliency map combines local and global approaches in a single approach and does so in almost linear complexity times.

In our model, we used RGB, HSL, LUV and CMY color spaces. Our model is there-

fore perfectible if we increase the number of color spaces (uncorrelated) to be merged.

As shown by the results we obtained, this strategy generates a model which is very promising, since it is quite different from existing saliency detection methods using the classical color contrast strategy between a region and the other regions of the image and, consequently, it could thus be efficiently combined with these methods for a better performance. Our model can also be parallelized (using the massively parallel processing power of GPUs) by processing each opposing color pair in parallel. In addition, it should be noted that this strategy of integrating color into local textural patterns could also be interesting to study with deep learning techniques or convolutional neural networks (CNNs) to further improve the quality of saliency maps.

BIBLIOGRAPHIE

[1] Derrick Parkhurst, Klinton Law, and Ernst Niebur. Modeling the role of salience in the allocation of overt visual attention. *Vision research*, 42(1) :107–123, 2002.

[2] Laurent Itti. Models of bottom-up attention and saliency. In *Neurobiology of attention*, pages 576–582. Elsevier, 2005.

[3] Laurent Itti and Christof Koch. Computational modelling of visual attention. *Nature reviews neuroscience*, 2(3) :194–203, 2001.

[4] Farhan Baluch and Laurent Itti. Mechanisms of top-down attention. *Trends in neurosciences*, 34(4) :210–224, 2011.

[5] Anne Treisman. Features and objects : The fourteenth bartlett memorial lecture. *The quarterly journal of experimental psychology*, 40(2) :201–237, 1988.

[6] Jeremy M Wolfe, Kyle R Cave, and Susan L Franzel. Guided search : an alternative to the feature integration model for visual search. *Journal of Experimental Psychology : Human perception and performance*, 15(3) :419, 1989.

[7] Christof Koch and Shimon Ullman. Shifts in selective visual attention : towards the underlying neural circuitry. In *Matters of intelligence*, pages 115–141. Springer, 1987.

[8] Qiong Yan, Li Xu, Jianping Shi, and Jiaya Jia. Hierarchical saliency detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1155–1162, 2013.

[9] Ashish Kumar Gupta, Ayan Seal, Pritee Khanna, Enrique Herrera-Viedma, and Ondrej Krejcar. Almnet : Adjacent layer driven multiscale features for salient object detection. *IEEE Transactions on Instrumentation and Measurement*, 70 :1–14, 2021.

[10] Ashish Kumar Gupta, Ayan Seal, Mukesh Prasad, and Pritee Khanna. Salient object detection techniques in computer vision—a survey. *Entropy*, 22(10) :1174, 2020.

[11] Ali Borji, Ming-Ming Cheng, Qibin Hou, Huaizu Jiang, and Jia Li. Salient object detection : A survey. *Computational visual media*, pages 1–34.

[12] Ali Borji and Laurent Itti. State-of-the-art in visual attention modeling. *IEEE transactions on pattern analysis and machine intelligence*, 35(1) :185–207, 2012.

[13] Laurent Itti. Automatic foveation for video compression using a neurobiological model of visual attention. *IEEE transactions on image processing*, 13(10) :1304–1318, 2004.

[14] Jinjiang Li, Xiaomei Feng, and Hui Fan. Saliency-based image correction for colorblind patients. *Computational Visual Media*, 6(2) :169–189, 2020.

[15] Nicolò Oreste Pinciroli Vago, Federico Milani, Piero Fraternali, and Ricardo da Silva Torres. Comparing cam algorithms for the identification of salient image features in iconography artwork analysis. *Journal of Imaging, MDPI*, 7(7) :106, 2021.

[16] Yuan Gao, Miaojing Shi, Dacheng Tao, and Chao Xu. Database saliency for fast image retrieval. *IEEE Transactions on Multimedia*, 17(3) :359–369, 2015.

[17] Rik Pieters and Michel Wedel. Attention capture and transfer in advertising : Brand, pictorial, and text-size effects. *Journal of Marketing*, 68(2) :36–50, 2004.

[18] Lai-Kuan Wong and Kok-Lim Low. Saliency-enhanced image aesthetics class prediction. In *2009 16th IEEE International Conference on Image Processing (ICIP)*, pages 997–1000. IEEE, 2009.

[19] Hantao Liu and Ingrid Heynderickx. Studying the added value of visual attention in objective image quality metrics based on eye movement data. In *2009*

16th IEEE international conference on image processing (ICIP), pages 3097–3100. IEEE, 2009.

[20] Li-Qun Chen, Xing Xie, Xin Fan, Wei-Ying Ma, Hong-Jiang Zhang, and He-Qin Zhou. A visual attention model for adapting images on small displays. *Multimedia systems*, 9(4) :353–364, 2003.

[21] Tao Chen, Ming-Ming Cheng, Ping Tan, Ariel Shamir, and Shi-Min Hu. Sketch2photo : Internet image montage. *ACM transactions on graphics (TOG)*, 28(5) :1–10, 2009.

[22] Hua Huang, Lei Zhang, and Hong-Chao Zhang. Arcimboldo-like collage using internet images. In *Proceedings of the 2011 SIGGRAPH Asia Conference*, pages 1–8, 2011.

[23] Arnold WM Smeulders, Dung M Chu, Rita Cucchiara, Simone Calderara, Afshin Dehghan, and Mubarak Shah. Visual tracking : An experimental survey. *IEEE transactions on pattern analysis and machine intelligence*, 36(7) :1442–1468, 2013.

[24] Mark Haidekker. *Advanced biomedical image analysis*. John Wiley & Sons, 2011.

[25] H. Knutsson and G Granlund. Texture analysis using two-dimensional quadrature filters. In *IEEE Comput. Soc. Workshop on Computer Architecture for Pattern Analysis and Image Database Management*, pages 206–213, 1983.

[26] Timo Ojala, Matti Pietikäinen, and David Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern recognition*, 29(1) :51–59, 1996.

[27] Kenneth I Laws. *Textured image segmentation*. PhD thesis, University of Southern California Los Angeles Image Processing INST, 1980.

[28] Matti Pietikäinen, Abdenour Hadid, Guoying Zhao, and Timo Ahonen. *Computer vision using local binary patterns*, volume 40. Springer Science & Business Media, 2011.

[29] Xiaoyang Tan and Bill Triggs. Enhanced local texture feature sets for face recognition under difficult lighting conditions. *IEEE transactions on image processing*, 19(6) :1635–1650, 2010.

[30] Timo Ahonen, Abdenour Hadid, and Matti Pietikäinen. Face recognition with local binary patterns. In *European conference on computer vision*, pages 469–481. Springer, 2004.

[31] Ran Margolin, Ayellet Tal, and Lihi Zelnik-Manor. What makes a patch distinct ? In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1139–1146, 2013.

[32] Qing Zhang, Jiajun Lin, Yanyun Tao, Wenju Li, and Yanjiao Shi. Salient object detection via color and texture cues. *Neurocomputing*, 243 :35–48, 2017.

[33] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE transactions on pattern analysis and machine intelligence*, 34(11) :2274–2282, 2012.

[34] Ashish Kumar Gupta, Ayan Seal, Pritee Khanna, Ondrej Krejcar, and Anis Yazidi. Awks : adaptive, weighted k-means-based superpixels for improved saliency detection. *Pattern Analysis and Applications*, 24(2) :625–639, 2021.

[35] Christos Faloutsos and King-Ip Lin. *FastMap : A fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets*, volume 24. ACM, 1995.

[36] Alice Porebski, Nicolas Vandenbroucke, and Ludovic Macaire. Haralick feature extraction from lbp images for color texture classification. In *2008 First Workshops on Image Processing Theory, Tools and Applications*, pages 1–8. IEEE, 2008.

[37] Anne M Treisman and Garry Gelade. A feature-integration theory of attention. *Cognitive psychology*, 12(1) :97–136, 1980.

[38] Jeremy M Wolfe and Todd S Horowitz. What attributes guide the deployment of visual attention and how do they do it ? *Nature reviews neuroscience*, 5(6) :495–501, 2004.

[39] Laurent Itti, Christof Koch, and Ernst Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 20(11) :1254–1259, 1998.

[40] Simone Frintrop, Thomas Werner, and German Martin Garcia. Traditional saliency reloaded : A good old model in new shape. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 82–90, 2015.

[41] Radhakrishna Achanta, Francisco Estrada, Patricia Wils, and Sabine Süsstrunk. Salient region detection and segmentation. In *International conference on computer vision systems*, pages 66–75. Springer, 2008.

[42] Ming-Ming Cheng, Niloy J Mitra, Xiaolei Huang, Philip HS Torr, and Shi-Min Hu. Global contrast based salient region detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(3) :569–582, 2015.

[43] Seena Joseph and Oludayo O Olugbara. Detecting salient image objects using color histogram clustering for region granularity. *Journal of Imaging, MDPI*, 7(9) :187, 2021.

[44] Chenlei Guo and Liming Zhang. A novel multiresolution spatiotemporal saliency detection model and its applications in image and video compression. *IEEE transactions on image processing*, 19(1) :185–198, 2009.

[45] Federico Perazzi, Philipp Krähenbühl, Yael Pritch, and Alexander Hornung. Saliency filters : Contrast based filtering for salient region detection. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 733–740. IEEE, 2012.

[46] Stas Goferman, Lihi Zelnik-Manor, and Ayellet Tal. Context-aware saliency detection. *IEEE transactions on pattern analysis and machine intelligence*, 34(10) :1915–1926, 2012.

[47] Wei Qi, Ming-Ming Cheng, Ali Borji, Huchuan Lu, and Lian-Fa Bai. Saliencyrank : Two-stage manifold ranking for salient object detection. *Computational Visual Media*, 1(4) :309–320, 2015.

[48] Ali Borji, Ming-Ming Cheng, Huaizu Jiang, and Jia Li. Salient object detection : A benchmark. *IEEE transactions on image processing*, 24(12) :5706–5722, 2015.

[49] Topi Mäenpää and Matti Pietikäinen. Classification with color and texture : jointly or separately ? *Pattern recognition*, 37(8) :1629–1640, 2004.

[50] Ali Borji and Laurent Itti. Exploiting local and global patch rarities for saliency detection. In *2012 IEEE conference on computer vision and pattern recognition*, pages 478–485. IEEE, 2012.

[51] Timo Ojala, Matti Pietikainen, and Topi Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on pattern analysis and machine intelligence*, 24(7) :971–987, 2002.

[52] Jianping Shi, Qiong Yan, Li Xu, and Jiaya Jia. Hierarchical image saliency detection on extended cssd. *IEEE transactions on pattern analysis and machine intelligence*, 38(4) :717–729, 2016.

[53] Chuan Yang, Lihe Zhang, Huchuan Lu, Xiang Ruan, and Ming-Hsuan Yang. Saliency detection via graph-based manifold ranking. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 3166–3173. IEEE, 2013.

[54] Ming-Ming Cheng, NiloyJ. Mitra, Xiaolei Huang, and Shi-Min Hu. Salientshape : group saliency in image collections. *The Visual Computer*, 30(4) :443–453, 2014.

[55] Sharon Alpert, Meirav Galun, Achi Brandt, and Ronen Basri. Image segmentation by probabilistic bottom-up aggregation and cue integration. *IEEE transactions on pattern analysis and machine intelligence*, 34(2) :315–327, 2011.

[56] Chuan Yang, Lihe Zhang, and Huchuan Lu. Graph-regularized saliency detection with convex-hull-based center prior. *IEEE Signal Processing Letters*, 20(7) :637–640, 2013.

[57] Ran Margolin, Lihi Zelnik-Manor, and Ayellet Tal. Saliency for image manipulation. *The Visual Computer*, 29(5) :381–392, 2013.

[58] Parthipan Siva, Chris Russell, Tao Xiang, and Lourdes Agapito. Looking beyond the image : Unsupervised learning for object saliency and detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3238–3245, 2013.

[59] Yulin Xie, Huchuan Lu, and Ming-Hsuan Yang. Bayesian saliency via low and mid level cues. *IEEE Transactions on Image Processing*, 22(5) :1689–1698, 2012.

[60] Kai-Yueh Chang, Tyng-Luh Liu, Hwann-Tzong Chen, and Shang-Hong Lai. Fusing generic objectness and visual saliency for salient object detection. In *2011 International Conference on Computer Vision*, pages 914–921. IEEE, 2011.

[61] Lijuan Duan, Chunpeng Wu, Jun Miao, Laiyun Qing, and Yu Fu. Visual saliency detection by spatially weighted dissimilarity. In *CVPR 2011*, pages 473–480. IEEE, 2011.

[62] Esa Rahtu, Juho Kannala, Mikko Salo, and Janne Heikkilä. Segmenting salient objects from images and videos. In *European conference on computer vision*, pages 366–379. Springer, 2010.

[63] Radhakrishna Achanta, Sheila Hemami, Francisco Estrada, and Sabine Susstrunk. Frequency-tuned salient region detection. In *2009 IEEE conference on computer vision and pattern recognition*, pages 1597–1604. IEEE, 2009.

[64] Xiyin Wu, Xiaodi Ma, Jinxia Zhang, Andong Wang, and Zhong Jin. Salient object detection via deformed smoothness constraint. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 2815–2819. IEEE, 2018.

[65] Yuchen Yuan, Changyang Li, Jinman Kim, Weidong Cai, and David Dagan Feng. Reversion correction and regularized random walk ranking for saliency detection. *IEEE transactions on image processing*, 27(3) :1311–1322, 2017.

[66] Lihe Zhang, Dandan Zhang, Jiayu Sun, Guohua Wei, and Hongguang Bo. Salient object detection by local and global manifold regularized svm model. *Neurocomputing*, 340 :42–54, 2019.

CHAPITRE 5

COSOV1NET : A CONE- AND SPATIAL-OPPONENT PRIMARY VISUAL CORTEX-INSPIRED NEURAL NETWORK FOR LIGHTWEIGHT SALIENT OBJECT DETECTION

Dans ce chapitre, nous exposons notre article publié dans la revue, " MDPI (Multidisciplinary Digital Publishing Institute) - Sensors Journal, section : Sensing and Imaging", intitulé : " **CoSOV1Net : A Cone- and Spatial-Opponent Primary Visual Cortex-Inspired Neural Network for Lightweight Salient Object Detection** ". Nous le présentons dans sa langue originale de publication.

5.1 Abstract

Salient object-detection models attempt to mimic the human visual system's ability to select relevant objects in images. To this end, the development of deep neural networks on high-end computers has recently achieved high performance. However, developing deep neural network models with the same performance for resource-limited vision sensors or mobile devices remains a challenge. In this work, we propose CoSOV1net, a novel lightweight salient object-detection neural network model, inspired by the cone- and spatial-opponent processes of the primary visual cortex (V1), which inextricably link color and shape in human color perception. Our proposed model is trained from scratch, without using backbones from image classification or other tasks. Experiments on the most widely used and challenging datasets for salient object detection show that CoSOV1Net achieves competitive performance (i.e., $F_\beta = 0.931$ on the ECSSD dataset) with state-of-the-art salient object-detection models while having a low number of parameters (1.14 M), low FLOPS (1.4 G) and high FPS (211.2) on GPU (Nvidia GeForce RTX 3090 Ti) compared to the state of the art in lightweight or nonlightweight salient object-detection tasks. Thus, CoSOV1net has turned out to be a lightweight salient object-detection model that can be adapted to mobile environments

and resource-constrained devices.

5.2 Introduction

The human visual system (HVS) has the ability to select and process relevant information from among the large amount that is received. This relevant information in an image is called salient objects [1]. Salient object-detection models in computer vision try to mimic this phenomenon by detecting and segmenting salient objects in images. This is an important task, given its many applications in computer vision, such as object tracking, recognition and detection [2], advertisement optimization [3], image/video compression [4], image correction [5], analysis of iconographic illustrations [6], image retrieval [7], aesthetic evaluation [8], image quality evaluation [9], image retargeting [10], image editing [11] and image collages [12], to name a few. Thus, it has been the subject of intensive research in recent years and is still being investigated [13]. Salient object-detection models generally fall into two categories, namely conventional and deep learning-based models, which differ by their feature extraction process. The former use hand-crafted features, while the latter use features learned from a neural network. Thanks to powerful representation learning methods, deep learning-based salient object-detection models have recently shown superior performance over conventional models [13, 14]. The high performance of these models is undeniable; however, generally, they are also heavy if we consider their number of parameters and the amount of memory occupied, in addition to their high computational cost and slow detection speed. This makes these models less practical for resource-limited vision sensors or mobile devices that have many constraints on their memory and computational capabilities, as well as for real-time applications [15, 16]. Hence, there is a need for lightweight salient object-detection models whose performance is comparable to state-of-the-art models, with the advantages of being deployed on resource-limited vision sensors or mobile devices and having a detection speed that allows them to be used in real-time applications. Existing lightweight salient object-detection models have used different methodologies, such as backbones from nonlightweight classification models [17, 18], the imitation of

primate hierarchical visual perception [19], human attention mechanisms [16, 19], etc.

In this work, we propose an original approach for a new lightweight neural network model, namely CoSOV1Net, for salient object detection, that can therefore be adapted to mobile environments and resource-limited or -constrained devices, with the additional properties of being able to be trained from scratch without having to use backbones developed from image-classification tasks and having few parameters, but with comparable performance with state-of-the-art models.

Given that detecting salient objects is a capability of the human visual system and that a normal human visual system performs this quickly and correctly, we used images or scenes encoding mechanism research advances in neuroscience, especially for the early stage of the human visual system [20–22]. Our strategy in this model is therefore inspired by two neuroscience discoveries in human color perception, namely :

1. The color-opponent encoding in the early stage of the HVS (human visual system) [23–26] ;

2. The fact that color and pattern are linked inextricably in human color perception [20, 27].

Inspired by these neuroscience discoveries, we propose a cone- and spatial-opponent primary visual cortex (CoSOV1) module that extracts features at the spatial level and between color channels at the same time to integrate color in the patterns. This process is applied first on opposing color pair channels two by two and then to grouped feature maps through our deep neural network. Thus, based on the CoSOV1 module, we build a novel lightweight encoder–decoder deep neural network for salient object detection : CoSOV1Net, which has only 1.14 M parameters but comparable performance with state-of-the-art salient object-detection models. CoSOV1Net predicts salient maps at a speed of 4.4 FPS on an Intel CPU, i7-11700F and 211.2 FPS on a Nvidia GeForce RTX 3090 Ti GPU for 384×384 images and it has a low FLOPS $= 1.4$ G. Therefore, CoSOV1net is a lightweight salient object-detection model that can be adapted for mobile environments and limited-resource devices.

Our contribution is threefold :

- We propose a novel approach to extract features from opposing color pairs in a neural network to exploit the strength of the color-opponent principle from human color perception. This approach permits the acceleration of neural network learning ;

- We propose a novel strategy to integrate color in patterns in a neural network by extracting features locally and between color channels at the same time in successively grouped feature maps, which results in a reduction in the number of parameters and the depth of the neural network, while keeping good performance ;

- We propose—for the first time, to our knowledge—a novel lightweight salient object-detection neural network architecture based on the proposed approach for learning opposing color pairs along with the strategy of integrating color in patterns. This model has few parameters, but its performance is comparable to state-of-the-art methods.

The rest of this work is organized as follows : Section 5.3 presents some lightweight models related to this approach ; Section 5.4 presents our proposed lightweight salient object-detection model ; Section 5.5 describes the datasets used, evaluation metrics, our experimental results and the comparison of our model with state-of-the-art models ; Section 5.6 discusses our results ; Section 5.7 concludes this work.

5.3 Related Work

Many salient object-detection models have been proposed and most of the influential advances in image-based salient object detection have been reviewed by Gupta et al. [13]. Herein, we present some conventional models and lightweight neural network models related to this approach.

5.3.1 Lightweight Salient Object Detection

In recent years, lightweight salient object-detection models have been proposed with different strategies and architectures. Qin et al. [28] designed U^2net, a lightweight salient object-detection model with a two-level nested Unet [29] neural network able to capture more contextual information from different scales, thanks to the mixture of receptive fields of different sizes. Its advantages are threefold : first, it increases the depth of the whole architecture without increasing the computational cost ; second, it is trained from scratch without using pretrained backbones, thus being able to keep feature maps high-resolution ; third, it has high accuracy. Its disadvantage is its number of parameters. Other models are based on streamlined architecture to build lightweight deep neural networks. MobileNets [30, 31] and ShuffleNets [32, 33], along with their variants, are among the latter models. MobileNets [30] uses architecture based on depthwise separable convolution. ShuffleNets [32] uses architecture based on pointwise group convolution and channel shuffle, as well as depthwise convolution, to greatly reduce computational cost while maintaining accuracy. Their advantages are their computational cost, accuracy and speed, while their disadvantages are their number of parameters and their input resolution. Other authors have been inspired by primate or human visual system processes. Thus, Liu et al. [19] designed HVPNet, a lightweight salient object-detection network based on a hierarchical visual perception (HVP) module that mimics the primate visual cortex for hierarchical perception learning, whereas Liu et al. [16] were inspired by human perception attention mechanisms in designing SAMNet, another lightweight salient object-detection network, based on a stereoscopically attentive multiscale (SAM) module that adopts a stereoscopic attention mechanism for effective and efficient multiscale learning. Their advantages are their computational cost and accuracy, while their disadvantages are their number of parameters and their input resolution.

5.3.2 Color-Opponent Models

Color opponency, which is a human color perception propriety, has inspired many authors who have defined channels or feature maps to tackle their image-processing

tasks. Frintrop et al. [34] used three opponent channels—*RG*, *BY* and *I*—to extract features for their salient object-detection model.

To extract features for salient object detection, Ndayikengurukiye and Mignotte [1] used nine (9) opponent channels for RGB color space (RR : red–red ; RG : red–green ; RB : red–blue ; GR : green–red ; GG : green–green ; GB : green–blue ; BR : blue–red ; BG : blue–green ; BB : blue–blue) with a nonlinear combination, thanks to the OCLTP (opponent color local ternary pattern) texture descriptor, which is an extension of the OCLBP (opponent color local binary pattern) [35, 36] and Fastmap [37], which is a fast version of MDS (multidimensional scaling).

Most authors apply the opponent color mechanism to the input image color space channels and not on the resulting feature maps. However, Jain and Healey [38] used opponent features computed from Gabor filter outputs. They computed opponent features by combining information across different spectral bands at different scales obtained via Gabor filters for color texture recognition [38]. Yang et al. [39] proposed a framework based on the color-opponent mechanisms of color-sensitive double-opponent (DO) cells in the human visual system's primary visual cortex (V1) in order to combine brightness and color to maximize the boundary-detection reliability in natural scenes. The advantages of hand-crafted models are their computational cost, number of parameters, speed and input resolution, while their disadvantage is accuracy.

In this work, we propose a model inspired by the human visual system but different from other models, because our model uses the primary visual cortex (V1) cone- and spatial-opponent principle to extract features at channels' spatial levels and between color channels at the same time to integrate color into patterns in a manner allowing for a lightweight deep neural network design with performance comparable with state-of-the-art lightweight salient object-detection models.

5.4 Materials and Methods

5.4.1 Introduction

Our model for tackling the challenge of lightweight salient object detection is inspired by the human visual system (HVS)'s early visual color process, especially its cone opponency and spatial opponency in the primary visual cortex (V1). The human retina (located in the inner surface of the eye) has two types of photoreceptors, namely rods and cones. Rods are responsible for monochromatic vision under low levels of illumination, while cones are responsible for color vision at normal levels of illumination. There are three classes of cones : L, M and S. When light is absorbed by cone photoreceptors, the L, M and S cones absorb long-, middle- and short-wavelength visible light, respectively [24, 25, 27].

The cone signals are then processed by single-opponent retina ganglion cells. The single opponent operates an antagonistic comparison of the cone signals [23, 25, 26, 40] :

- $L - M$ opponent for red–green ;
- $S - (L + M)$ opponent for blue–yellow.

The red–green and blue–yellow signals are carried by specific cells (different cells each for red–green and blue–yellow) through the lateral geniculate nucleus (LGN) to the primary visual cortex (V1).

Shapley [27] and Shapley and Hawken [20] showed that the primary visual cortex (V1) plays an important role in color perception through the combined activity of two kinds of color-sensitive cortical neurons, namely single-opponent and double-opponent cells. Single-opponent cells in V1 operate in the same manner as those of retina ganglion cells and provide neuronal signals that can be used for estimating the color of the illumination [27]. Double-opponent cells in V1 compare cone signals across space as well as between cones [21, 22, 24, 27]. Double-opponent cells thus have two opponencies : spatial opponency and cone opponency. These properties permit them to be sensitive to color edges and spatial patterns. They are thus able to inextricably link color and pattern in human color perception [20, 27].

As the primary visual cortex (V1) is known to play a major role in visual color perception, as highlighted above, in this work, we propose a deep neural network based on the primary visual cortex (V1) to tackle the challenge of lightweight salient object detection. In particular, we use two neuroscience discoveries in human color perception, namely :

1. The color-opponent encoding in the early stage of the HVS ;

2. The fact that color and pattern are inextricably linked in human color perception

These two discoveries in neuroscience inspired us to design a neural network architecture for lightweight salient object detection, which hinges on two main ideas. First, at the beginning of the neural network, our model opposes color channels two by two by grouping them (R-R, R-G, R-B, G-G, G-B, B-B) then extracting the features at the channels' spatial levels and between the color channels from each channel pair at the same time, to integrate color into patterns. Therefore, instead of performing a subtractive comparison or an OCLTP (opponent color linear ternary pattern) like Ndayikengurukiye and Mignotte [1], we let the neural network learn the features that represent the comparison of the two color pairs. Second, this idea of grouping and then extracting the features at the channels' spatial levels and between the color channels at the same time is applied on feature maps at each neural network level until the saliency maps are obtained. This process allows the proposed model to mimic the human visual system's capability of inextricably linking color and pattern in color perception [20, 27].

It is this idea that differentiates our model from other models that use depthwise convolution followed by pointwise convolution [30, 31] to extract features at each individual color channel level (or feature map) first, not through a group of color channels (or feature maps) at the same time, as our model does. This idea also differentiates our model from models that combine a group of color channels (or feature maps) pixel by pixel first and apply depthwise convolution afterwards [32, 33]. The idea of grouping color channels in pairs (or feature map groups) differentiates our model from models that consider all color channels (or feature maps) as a single group while extracting features at color channels' spatial levels and between color channels at the same time.

Our model takes into account nonlinearities in the image at the beginning as well as through our neural network. For this purpose, we use an encoder–decoder neural network type whose core is a module that we call CoSOV1 (cone- and spatial-opponent primary visual cortex).

5.4.2 CoSOV1 : Cone- and Spatial-Opponent Primary Visual Cortex Module

The CoSOV1 (cone- and spatial-opponent primary visual cortex) module is composed of two parts (see Figure 5.1).

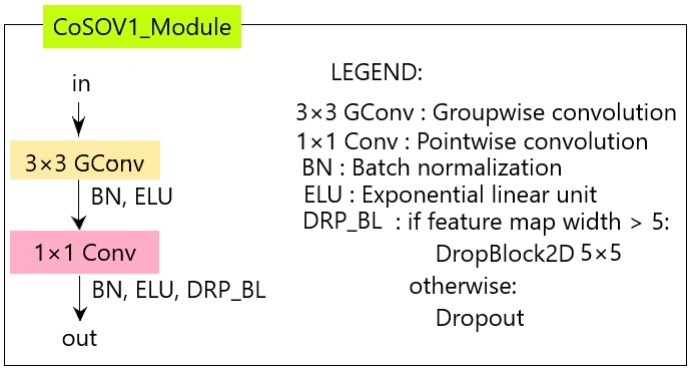


FIGURE 5.1 : The CoSOV1 (cone- and spatial-opponent primary visual cortex) module is the core of our neural network model.

In the first part, input color channels (or input feature maps) are split into groups of equal depth. Convolution (3×3) operations are then applied to each group of channels (or feature maps) in order to extract features from each group as opposing color channels (or opposing feature maps). This is performed thanks to a set of filters that convolve the group of color channels (or feature maps). Each filter is applied to the color channels (or input feature maps) through a convolution operation that detects local features at all locations on the input. Let $\mathscr{I}^g \in \mathbb{R}^{\mathscr{W} \times \mathscr{H} \times S}$ be an input group of feature maps, where $\mathscr{W}$ and $\mathscr{H}$ are the width and the height of each group's feature map, respectively, and $W \in \mathbb{R}^{3 \times 3 \times S}$, a filter with learned weights, with S being the depth of each group or the number of the channels in each group g, with $g \in \{1, \ldots, \mathscr{G}\}$ (where $\mathscr{G}$ is the number of

groups). The output feature map $\mathscr{O}^g \in \mathbb{R}^{\mathscr{W}\times\mathscr{H}}$ for this group g has a pixel value in the (k,l) position, defined as follows :

$$\mathscr{O}^g_{k,l} = \sum_{s=1}^{S}\sum_{i=0}^{2}\sum_{j=0}^{2} W_{i,j,s}\mathscr{I}^g_{k+i-1,l+j-1,s} \tag{5.1}$$

The weight matrix $W \in \mathbb{R}^{3\times3\times S}$ is the same across the whole group of channels or feature maps. Therefore, each resulting output feature map represents a particular feature at all locations in the input color channels (or input feature maps) [41]. We call the 3×3 convolution on grouped channels (or grouped feature maps) groupwise convolution. The zero padding is applied during the convolution process to keep the input channel size for the output feature maps. After groupwise convolution, we apply the batch normalization transform, which is known to enable faster and more stable training of deep neural networks [42, 43]. Let $\mathfrak{B} = \{X_1,\ldots,X_K\}$ be a minibatch that contains K examples from a dataset. The minibatch mean is

$$\mu_{\mathfrak{B}} = \frac{1}{K}\sum_{k=1}^{K} X_k \tag{5.2}$$

and the minibatch variance is

$$\sigma^2_{\mathfrak{B}} = \frac{1}{K}\sum_{k=1}^{K}(X_k - \mu_{\mathfrak{B}})^2 \tag{5.3}$$

The batch normalization transform $BN_{\gamma,\beta} : \{X_1,\ldots,X_K\} \longrightarrow \{Y_1,\ldots,Y_K\}$ (γ and β are parameters to be learned) :

$$Y_k = \gamma\widehat{X}_k + \beta \tag{5.4}$$

where $k \in \{1,\ldots,K\}$ and

$$\widehat{X_k} = \frac{X_k - \mu_{\mathfrak{B}}}{\sqrt{\sigma_{\mathfrak{B}}^2 + \varepsilon}} \tag{5.5}$$

and ε is a very small constant to avoid division by zero.

In order to take into account the nonlinearities present in the color channel input (or feature map input), given that groupwise convolution is a linear transformation, batch normalization is followed by a nonlinear function, exponential linear unit (ELU), defined as follows :

$$\text{ELU}(x) = \begin{cases} x & \text{if } x \geq 0, \\ \alpha \times (\exp(x) - 1) & \text{otherwise} \end{cases} \tag{5.6}$$

where $\alpha = 1$ by default.

The nonlinear function, which is the activation function, is placed after batch normalization, as recommended by Chollet [44].

The second part of the module searches for the best representation of the obtained feature maps. It is similar to the first part of the module, except for the groupwise convolution, which is replaced by point-wise convolution, but the input feature maps for pointwise convolution in this model are not grouped. Pointwise convolution allows us to learn the filters' weights and thus obtain feature maps that best represent the input channels (or input feature maps) for the salient object-detection task, while having few parameters.

Let $\mathscr{O} \in \mathbb{R}^{\mathscr{W} \times \mathscr{H} \times M}$ be the output of the first part of the module, with M being the number of feature maps in this output and $\mathscr{W}$ and $\mathscr{H}$ being the width and the height, respectively. Let a filter of the learned weights $V \in \mathbb{R}^M$ and $\mathscr{FM} \in \mathbb{R}^{\mathscr{W} \times \mathscr{H}}$ be its output feature map by pointwise convolution. Its pixel value $\mathscr{FM}_{k,l}$ in (k,l) position is :

$$\mathscr{FM}_{k,l} = \sum_{m=1}^{M} V_m \mathscr{O}_{k,l,m} \tag{5.7}$$

Thus, $V \in \mathbb{R}^{M}$ is a vector of learned weights that associates the input feature maps $\mathscr{O} \in \mathbb{R}^{\mathscr{W} \times \mathscr{H} \times M}$ to the feature map $\mathscr{FM} \in \mathbb{R}^{\mathscr{W} \times \mathscr{H}}$, which is the best representation of the latter-mentioned input feature maps. The pointwise convolution in this module uses many filters and thus it outputs many feature maps that are the best representation of the input feature map $\mathscr{O}$. As pointwise convolution is a linear combination, we again apply batch normalization followed by a exponential linear unit function (ELU) on the feature map $\mathscr{FM}$ to obtain the best representation of the input feature maps for the learned weights $V \in \mathbb{R}^{M}$, which takes into account nonlinearities in the feature maps $\mathscr{O} \in \mathbb{R}^{\mathscr{W} \times \mathscr{H} \times M}$.

Our scheme is different from depthwise separable convolution in that it uses the convolution of a group of channels instead of each channel individually [30, 45].

In addition, after the nonlinear function, noise is injected in the resulting feature maps during the neural network learning stage thanks to the dropout process (but not in the prediction stage) to facilitate the learning process. In this model, we use DropBlock [46] if the width of the feature map is greater than 5; otherwise, we use the common dropout [47].

The CoSOV1 module allows our neural network to have few parameters but good performance.

5.4.3 CoSOV1Net Neural Network Model Architecture

Our proposed model is built on the CoSOV1 module (see Figure 5.1). It is a neural network of the U-net encoder–decoder type [29] and is illustrated in Figure 5.2. Thus, our model consists of three main blocks :

1. The input RGB color channel pairing;
2. The encoder;
3. The decoder.

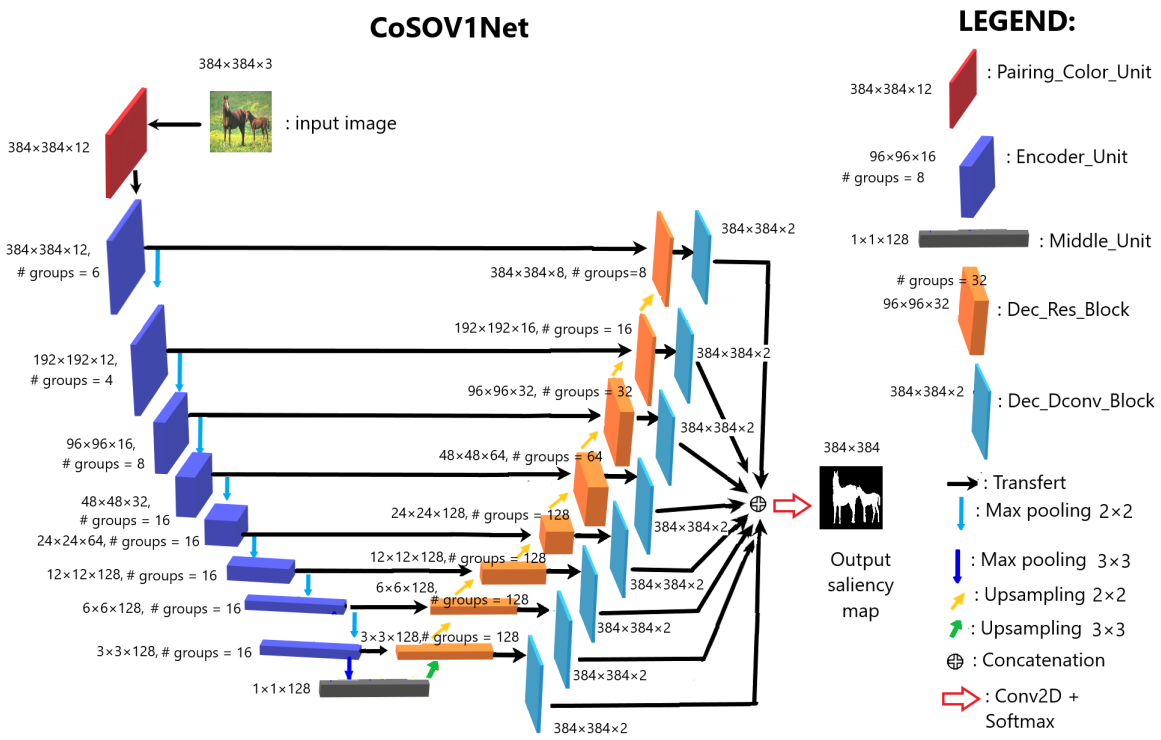


FIGURE 5.2 : Our model CoSOV1 neural network architecture consisting of 5 blocks : Pairing_Color_Unit, Encoder_Unit, Middle_Unit, Dec_Res_Block and Dec_Dconv_Block.

5.4.3.1 Input RGB Color Channel Pairing

At this stage, through Pairing_Color_Unit, the input RGB image is paired in six opposing color channel pairs : R-R, R-G, R-B, G-G, G-B and B-B [1, 35, 48]. These pairs are then concatenated, which gives 12 channels, R, R, R, G, R, B, G, G, G, B, B, B, as illustrated in Figure 5.3. This is the step for choosing the color channels to oppose. The set of concatenated color channels is then fed to the encoder.

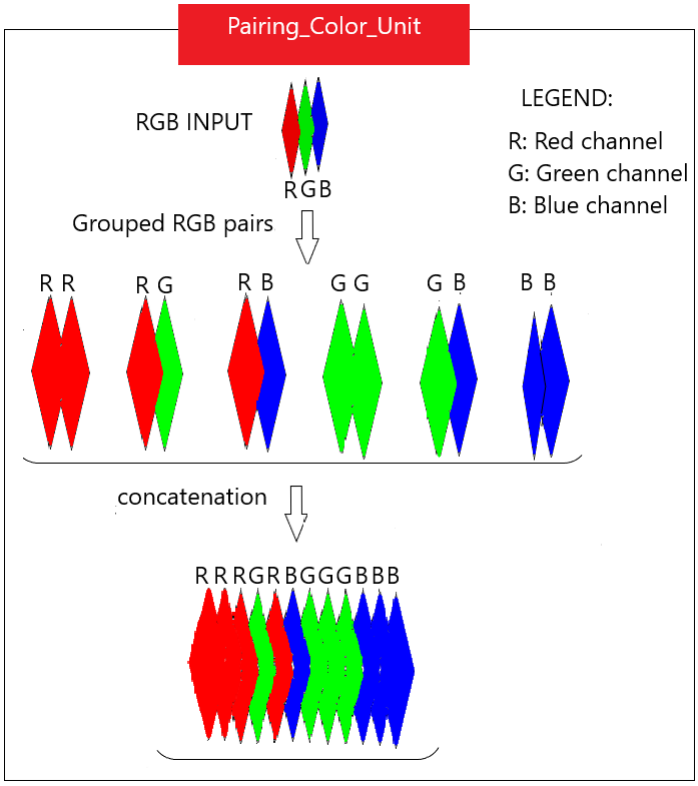


FIGURE 5.3 : Pairing_Color_Unit : input RGB color image is transformed in 6 opposing color channel pairs ; these are then concatenated to obtain 12 color channels.

5.4.3.2 Encoder

The encoder, in our proposed neural network model, is a convolutional neural network (CNN) [49] where an encoder unit (see Figure 5.2) is repeated eight times. Each encoder unit is followed by a max pooling (2×2) with strides = 2, except for the eighth neural network level, where the max pooling is 3×3 with strides = 3 (the max pooling is a downsampling operation, like a filtering with a maximum filter). While the size of each feature map is reduced by half, the depth of the feature maps is doubled, except for the first level, where it is kept at 12 and the last two levels, where it is kept at 128 to have few parameters.

The encoder unit (see Figure 5.4a) is composed of a residual block (Figure 5.4b) repeated three (3) times.

We used the residual block because this kind of block is known to improve the training of deeper neural networks [50]. The residual block consists of two CoSOV1 modules with a residual link. The reason for all these repetitions is to encode more information and thus allow our network performance to increase.

In the encoder, schematically, as explained above (Section 5.4.2), the CoSOV1 module (Figure 5.4c) splits the input channels into groups and applies groupwise convolution (3×3 convolution). Then, pointwise convolution is applied to the outputs of the concatenated groups (see Figure 5.5 for the first-level input illustration). Each of these convolutions is followed by batch normalization and a nonlinear function (ELU : exponential linear unit activation). After these layers, during the model training, regularization is performed in the CoSOV1 module using the dropout [47] method for small feature maps (dimensions smaller than 5×5) and DropBlock [46]—which is a variant of dropout that zeroes a block instead of pixels individually as dropout does—for feature maps with dimensions greater than 5×5.

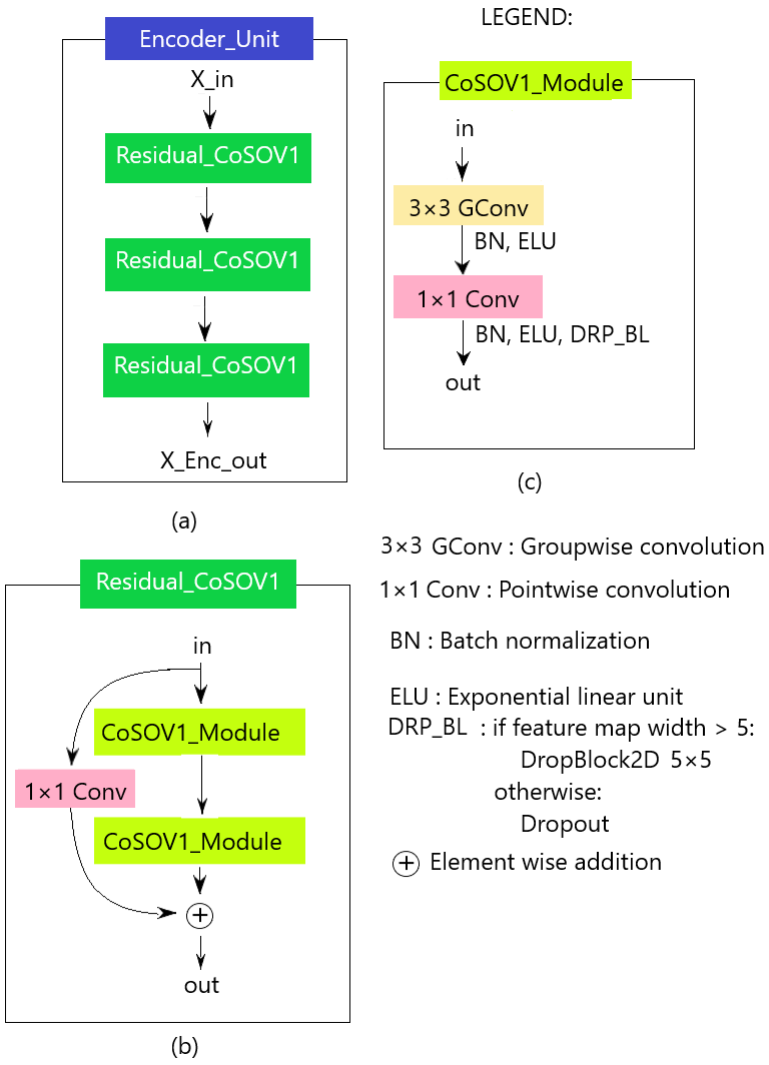


FIGURE 5.4 : Encoder unit : (**a**) encoder unit ; (**b**) the residual block ; (**c**) CoSOV1 module.

At its end, the encoder is followed by the middle unit (see Figure 5.6a), which is the CoSOV1 module (see Figure 5.6b), where we remove the groupwise convolution—since at this stage, the feature maps are $1 \times 1 \times 128$ in size—and add a residual link.

5.4.3.3 Decoder

The decoder transforms the features from the encoder to obtain the estimate of the salient object(s) present in the input image. This transformation is achieved through a repeating block, namely the decoder unit (see Figure 5.7a). The decoder unit consists of two parts : the decoder residual block (see Figure 5.7b) and the decoder deconvolution block (see Figure 5.7c). The decoder residual block is a modified CoSOV1 module that allows the model to take into account the output of the corresponding level in the encoder. The output of the decoder residual block takes two directions. On the one hand, it is passed to the next level of the decoder ; and on the other, to the second part of the decoder unit, which is the decoder deconvolution block. The latter deconvolves this output, obtaining two feature maps having the size of the input image ($384 \times 384 \times 2$ in our case). At the last level of the decoder, all the outputs from the deconvolution blocks are concatenated and fed to a convolution layer followed by a softmax activation layer, which gives the estimation of the salient object-detection map.

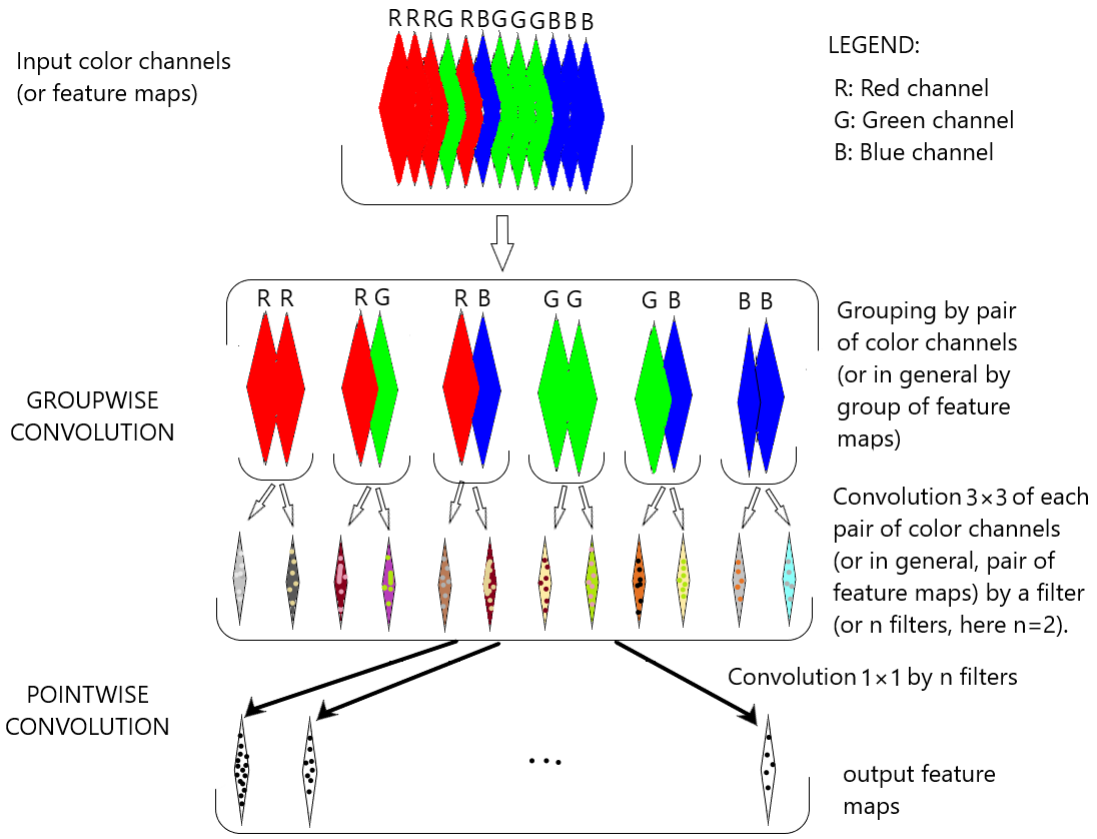


FIGURE 5.5 : Simplified flowchart in CoSOV1 module for processing pairs of opposing color pairs (or group of feature maps).

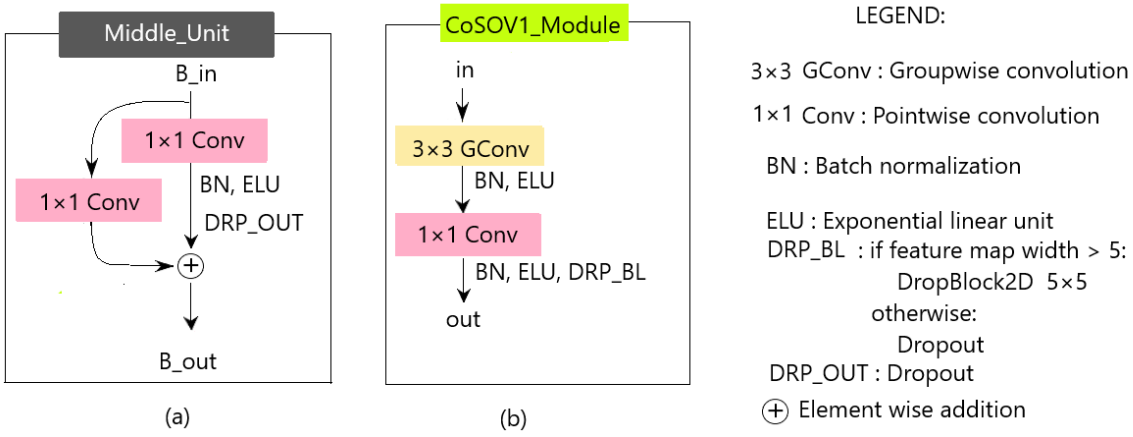


FIGURE 5.6 : (**a**) The middle unit, (**b**) the CoSOV1 module.

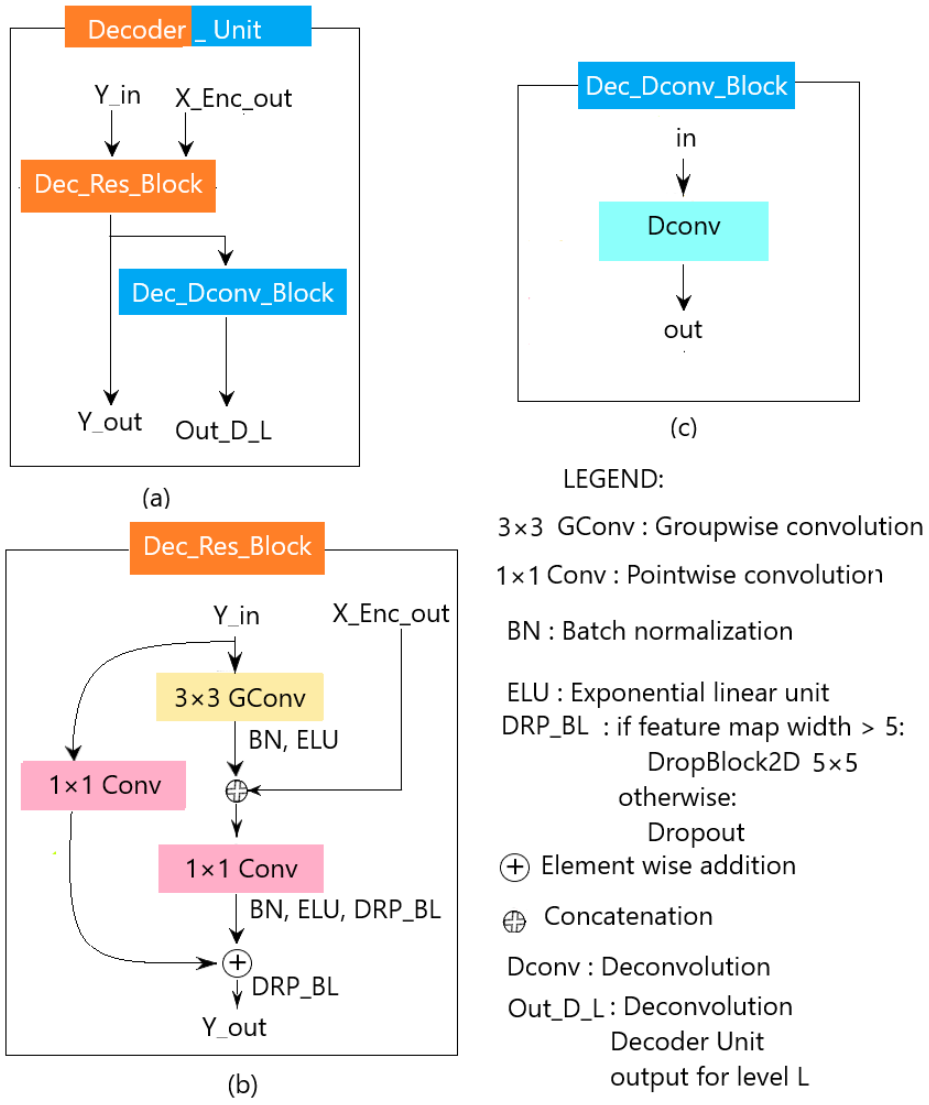


FIGURE 5.7 : (**a**) The decoder unit; (**b**) the decoder residual block; (**c**) the decoder deconvolution block.

5.5 Experimental Results

5.5.1 Implementation Details

For our proposed model implementation, we used the deep learning platform TensorFlow with the Keras deep learning application programming interface (API) [51].

All input images were resized to 384×384 and pixel values were normalized (each pixel channel value $\in [0.0, \ldots, 1.0]$ and ground truth pixels $\in \{0, 1\}$). Experiments were conducted on a single GPU, Nvidia GeForce RTX 3090 Ti (24 GB) and an Intel CPU, i7-11700F.

5.5.2 Datasets

Our proposed model's experiments were conducted on public datasets, which are the most widely used in the field of salient object detection [52]. Thus, we used the Extended Complex Scene Saliency dataset (ECSSD) [53] and the DUT-OMRON (Dalian University of Technology—OMRON Corporation) [54], DUTS [55], HKU-IS [56] and THUR15K [57] datasets.

ECSSD [53] contains 1000 natural images and their ground truths. Many of its images are semantically meaningful but structurally complex for saliency detection [53].

DUT-OMRON [54] contains 5168 images and their binary masks, with diverse variations and complex backgrounds.

The DUTS dataset [55] is divided into DUTS-TR (10,553 training images) and DUTS-TE (5019 test images). We trained and validated our proposed model on the DUTS-TR and DUTS-TE was used for tests.

HKU-IS [56] is composed of 4447 complex images, which contain many disconnected objects with different spatial distributions. Furthermore, it is very challenging for similar foreground/background appearances [58].

THUR15K is a dataset of images taken from the "Flickr" website, divided into five categories (butterfly, coffee mug, dog jump, giraffe, plane), which contains 3000 images. The images of this dataset represent real-world scenes and are considered complex for obtaining salient objects [57] (6232 images with ground truths).

5.5.3 Model Training Settings

For the reproducibility of the experiments, we set the seed = 123. We trained our proposed model on DUTS-TR (10,553 training images). We split the DUTS-TR dataset

into a train set (9472 images) and a validation set (1056 images) ; that is, approximately 90% of the dataset for the training set and 10% for the validation set. We did not use 25 images because we wanted the training set and the validation set to be divisible by batch size, which is 32.

Our proposed model was trained on scratch without pretrained backbones from image classification (i.e., VGG [59], etc.) or lightweight backbones (i.e., MobileNets [30, 31] or ShuffleNets [32, 33]). As DUTS-TR is not a big dataset, we used data augmentation during training and many epochs in order to overcome this problem. Indeed, the more epochs, the more the data-augmentation process transforms data. Thus, our proposed model training has two successive stages :

- The first stage is with data augmentation, which is applied to each batch with random transformation (40% zoom in or horizontal flip or vertical flip). This stage has 480 epochs : 240 epochs with learning rate = 0.001 and 240 epochs with learning rate = 0.0001 ;

- The second stage is without data augmentation. It has 620 epochs : 240 epochs with learning rate = 0.001, followed by 140 epochs with learning rate = 0.0001 and 240 epochs with learning rate = 0.00005.

We also used the same initializer for all layers in the neural network : the HeUniform Keras initializer [60], which draws samples from a uniform distribution within [−limit, limit], where limit = $\sqrt{\frac{6}{fan\_in}}$ (fan_in is the number of input units in the weight tensor). The dropout rate was set to 0.2. We used the RMSprop [61] Keras optimizer with default values except for the learning rate ; the centered, which was set to true ; and the clipnorm = 1. The loss function used was the “sparse_categorical_crossentropy” Keras function ; the Keras metric was “SparseCategoricalAccuracy ; the Keras check point monitor was “val_sparse_categorical_accuracy”.

5.5.4 Hyperparameters

Hyperparameters such as the ELU activation function, the optimizer, the batch size,

the filter size and the learning rates were chosen experimentally by observing the results.

The other hyperparameters were chosen as follows :

- Image size : The best image size was 384×384. We did not choose a small size because we expected to have a small salient object. As we also wanted to have a low computational cost, we did not go beyond this size.

- Number of levels for the encoder : We empirically obtained eight levels as the best number. The choice of image size permitted us to have a maximum of eight levels for the encoder part, given that $384 = 2^7 \times 3$. The size of the feature maps of each level corresponds to the size of those of the previous level divided by 2, except the last level, where the division is by 3.

- Number of levels for the decoder : Eight levels. The number of levels is the same for the encoder part and the decoder part.

- Number of layers : At each level, we chose to use an encoder unit that has an equal number of layers for all levels and a decoder unit that has an equal number of layers for all levels. The number of layers was obtained experimentally.

- Number of filters : We also experimentally chose the number of filters keeping in mind the minimum parameters ; the encoder's number of filters was 12, 16, 32, 64, 128, 128, 128 and 128, respectively, for the first, second, …, seventh and eighth levels ; the decoder residual bloc number of filters was 128, 128, 128, 128, 64, 32, 16 and 8, respectively, for the eighth, seventh, sixth, …, second and first levels. For the decoder deconvolution blocs, at each level, the number of filters was 2.

- The use of batch normalization : Batch normalization is known to enable faster and more stable training for deep neural networks [42, 43]. So, we decided to use it.

- Use of dropout : The dropout process injects noise in the resulting feature maps during the neural network learning stage (but not in the prediction stage) to facilitate the learning process. In this model, we used DropBlock [46] if the width of the feature map was greater than 5 ; otherwise, we used the common dropout [47]. The best results were obtained for DropBlock size = 5×5 and rate = 0.1 (the authors' paper suggested a value between 0.05 and 0.25). For the common dropout, the best rate was 0.2, obtained experimentally.

As our proposed model, CoSOV1Net does not use pretrained backbones and the input image is resized to 384×384 ; it has the advantage of good resolution.

5.5.5 Evaluation Metrics

5.5.5.1 Accuracy

The metrics used to evaluate our proposed model accuracy were F_β measure, *MAE* (mean absolute error) and weighted F_β^w measure [62]. We also used precision, precision–recall and F_β measure curves.

Let M be the binary mask obtained for the predicted saliency probability map, given a threshold in the range of $[0,1)$ and with G being the corresponding ground truth :

$$\text{Precision} = \frac{|M \cap G|}{|M|} \tag{5.8}$$

$$\text{Recall} = \frac{|M \cap G|}{|G|} \tag{5.9}$$

$\cap$: set intersection symbol ; $|.|$: the number of pixels whose values are not zeros.

The F_β-measure (F_β) is the weighted harmonic mean of precision and recall :

$$F_\beta = \frac{(1+\beta^2) \times \text{Precision} \times \text{Recall}}{\beta^2 \times \text{Precision} + \text{Recall}} \tag{5.10}$$

During evaluation, $\beta^2 = 0.3$, as it is often suggested [16, 58].

Let $\overline{S}$ be the saliency map estimation with pixel values normalized in order to be in $[0.0, \ldots, 1.0]$ and $\overline{G}$; its ground truth also normalized in $\{0; 1\}$. The *MAE* (mean absolute error) is :

$$MAE = \frac{1}{W \times H} \sum_{x=1}^{W} \sum_{y=1}^{H} |\overline{S}(x,y) - \overline{G}(x,y)| \tag{5.11}$$

where W and H are the width and the height, respectively, of the above maps ($\overline{S}$ and $\overline{G}$).

The F_β^w measure [62] fixes the interpolation flaw, dependence flaw and equal importance flaw in traditional evaluation metrics and its value is :

$$F_\beta^w = (1+\beta^2) \frac{\text{Precision}^w \times \text{Recall}^w}{\beta^2 \times \text{Precision}^w + \text{Recall}^w} \tag{5.12}$$

Precision^w and Recall^w are the weighted precision and the weighted recall, respectively.

5.5.5.2 Lightweight Measures

Since we propose a lightweight salient object-detection model in this work, we therefore also evaluate the model with lightweight measures : the number of parameters, the saliency map estimation speed (FPS : frames per second) and the computational cost by measuring the FLOPS (the number of floating-point operations). The FLOPS is related to the device's energy consumption (the higher the FLOPS, the higher the energy consumption). The floating-point operation numbers are computed as follows [63] :

- For a convolution layer with n filters of size $k \times k$ applied to $W \times H \times C$ feature maps (W : width ; H : height ; C : channels), with P : number of parameters :

$$\text{FLOPS} = W \times H \times P \tag{5.13}$$

- For a max-pooling layer or an upsampling layer with a window of size $sz \times sz$ on $W \times H \times C$ feature maps (W : width ; H : height ; C : channels) :

$$\text{FLOPS} = W \times H \times C \times sz \times sz \tag{5.14}$$

5.5.6 Comparison with State of the Art

We compare our proposed model with 20 state-of-the-art salient object detection and 10 state-of-the-art lightweight salient object-detection models. We divided these methods because the lightweight methods outperform others with respect to lightweight measures. However, the lightweight methods' accuracy is lower than the accuracy of those with huge parameters. We mainly used the salient object-detection results provided by Liu et al. [16], except for the F_β measure and precision–recall curves, where we used saliency maps provided by these authors. We also used saliency maps provided by the HVPNet authors [19] to compute HVPNet F_β^ω measures.

In this section, we describe the comparison with the 20 salient object-detection models, namely DRFI [64], DCL [65], DHSNet [66], RFCN [67], NLDF [68], DSS [69], Amulet [18], UCF [70], SRM [71], PiCANet [17], BRN [72], C2S [73], RAS [74], DNA [75], CPD [76], BASNet [77], AFNet [78], PoolNet [79], EGNet [80] and BANet [81].

Table 5.1 shows that our proposed model CoSOV1Net outperforms all 20 state-of-the-art salient object-detection models for lightweight measures (#parameters, FLOPS and FPS) by a large margin (i.e., the best among them for FLOPS is DHSNet [66], with FLOPS = 15.8 G and $F_\beta = 0.903$ for ECSSD ; the worst is EGNet [80], with FLOPS = 270.8 G and $F_\beta = 0.938$ for ECSSD ; meanwhile, our proposed model, CoSOV1Net, has FLOPS = 1.4 G, and its $F_\beta = 0.931$ for ECSSD) (see Table 5.1).

Table 5.1 also shows that CoSOV1Net is among the top 6 models for ECSSD, among the top 7 for DUT-OMRON and around the top 10 for the other three datasets for the F-measure. Tables 5.2 and 5.3 compare our model with the state-of-the-art models for the MAE and F_β^ω measures, respectively. From this comparison, we see that our model is

ranked around the top 10 for all four datasets and is ranked 15th for the HKU-IS dataset. This demonstrates that our model is also competitive with respect to the performance of state-of-the-art models.

Tables 5.1–5.3 show that our proposed model, CoSOV1Net, clearly has the advantage of the number of parameters, computational cost and speed over salient object detection. They also show that its performance is closer to the best among them.

TABLE 5.1 : Our proposed model F-measure ($F_\beta \uparrow$, $\beta^2 = 0.3$) compared with 20 state-of-the-art models (best value in bold) [# Param : number of parameters, $\uparrow$: great is best, $\downarrow$: small is the best].

Methods	**# Param (M)** $\downarrow$	**FLOPS (G)** $\downarrow$	**Speed (FPS)** $\uparrow$	**ECSSD**	**DUT-OMRONTE**	**DUTS-TE**	**HKU-IS**	**THUR15K**
DRFI [64]	-	-	0.1	0.777	0.652	0.649	0.774	0.670
DCL [65]	66.24	224.9	1.4	0.895	0.733	0.785	0.892	0.747
DHSNet [66]	94.04	15.8	10.0	0.903	-	0.807	0.889	0.752
RFCN [67]	134.69	102.8	0.4	0.896	0.738	0.782	0.892	0.754
NLDF [68]	35.49	263.9	18.5	0.902	0.753	0.806	0.902	0.762
DSS [69]	62.23	114.6	7.0	0.915	0.774	0.827	0.913	0.770
Amulet [18]	33.15	45.3	9.7	0.913	0.743	0.778	0.897	0.755
UCF [70]	23.98	61.4	12.0	0.901	0.730	0.772	0.888	0.758
SRM [71]	43.74	20.3	12.3	0.914	0.769	0.826	0.906	0.778
PiCANet [17]	32.85	37.1	5.6	0.923	0.766	0.837	0.916	0.783
BRN [72]	126.35	24.1	3.6	0.919	0.774	0.827	0.910	0.769
C2S [73]	137.03	20.5	16.7	0.907	0.759	0.811	0.898	0.775
RAS [74]	20.13	35.6	20.4	0.916	0.785	0.831	0.913	0.772
DNA [75]	20.06	82.5	25.0	0.935	0.799	0.865	0.930	0.793
CPD [76]	29.23	59.5	68.0	0.930	0.794	0.861	0.924	0.795
BASNet [77]	87.06	127.3	36.2	0.938	**0.805**	0.859	0.928	0.783
AFNet [78]	37.11	38.4	21.6	0.930	0.784	0.857	0.921	0.791
PoolNet [79]	53.63	123.4	39.7	0.934	0.791	0.866	0.925	**0.800**
EGNet [80]	108.07	270.8	12.7	0.938	0.794	0.870	0.928	**0.800**
BANet [81]	55.90	121.6	12.5	**0.940**	0.803	**0.872**	**0.932**	0.796
CoSOV1Net (OURS)	**1.14**	**1.4**	**211.2**	0.931	0.789	0.833	0.912	0.773

TABLE 5.2 : Our proposed model MAE (↓) compared with 20 state-of-the-art models (best performance in bold) [# Param : number of parameters, ↑ : great is the best, ↓ : small is the best].

Methods	**# Param (M) ↓**	**FLOPS (G) ↓**	**Speed (FPS) ↑**	**ECSSD**	**DUT-OMRON**	**DUTS-TE**	**HKU-IS**	**THUR15K**
DRFI [64]	-	-	0.1	0.161	0.138	0.154	0.146	0.150
DCL [65]	66.24	224.9	1.4	0.080	0.095	0.082	0.063	0.096
DHSNet [66]	94.04	15.8	10.0	0.062	-	0.066	0.053	0.082
RFCN [67]	134.69	102.8	0.4	0.097	0.095	0.089	0.080	0.100
NLDF [68]	35.49	263.9	18.5	0.066	0.080	0.065	0.048	0.080
DSS [69]	62.23	114.6	7.0	0.056	0.066	0.056	0.041	0.074
Amulet [18]	33.15	45.3	9.7	0.061	0.098	0.085	0.051	0.094
UCF [70]	23.98	61.4	12.0	0.071	0.120	0.112	0.062	0.112
SRM [71]	43.74	20.3	12.3	0.056	0.069	0.059	0.046	0.077
PiCANet [17]	32.85	37.1	5.6	0.049	0.068	0.054	0.042	0.083
BRN [72]	126.35	24.1	3.6	0.043	0.062	0.050	0.036	0.076
C2S [73]	137.03	20.5	16.7	0.057	0.072	0.062	0.046	0.083
RAS [74]	20.13	35.6	20.4	0.058	0.063	0.059	0.045	0.075
DNA [75]	20.06	82.5	25.0	0.041	**0.056**	0.044	**0.031**	0.069
CPD [76]	29.23	59.5	68.0	0.044	0.057	0.043	0.033	**0.068**
BASNet [77]	87.06	127.3	36.2	0.040	**0.056**	0.048	0.032	0.073
AFNet [78]	37.11	38.4	21.6	0.045	0.057	0.046	0.036	0.072
PoolNet [79]	53.63	123.4	39.7	0.048	0.057	0.043	0.037	**0.068**
EGNet [80]	108.07	270.8	12.7	0.044	**0.056**	0.044	0.034	0.070
BANet [81]	55.90	121.6	12.5	**0.038**	0.059	**0.040**	**0.031**	**0.068**
CoSOV1Net (OURS)	**1.14**	**1.4**	**211.2**	0.051	0.064	0.057	0.045	0.076

TABLE 5.3 : Our proposed model weighted F-measure ($F_\beta^\omega \uparrow$, $\beta^2 = 1$) compared with 20 state-of-the-art models (best value in bold) [# Param : number of parameters, $\uparrow$: great is the best, $\downarrow$: small is the best].

Methods	**# Param (M)** $\downarrow$	**FLOPS (G)** $\downarrow$	**Speed (FPS)** $\uparrow$	**ECSSD**	**DUT-OMRONTE**	**DUTS-**	**HKU-IS**	**THUR15K**
DRFI [64]	-	-	0.1	0.548	0.424	0.378	0.504	0.444
DCL [65]	66.24	224.9	1.4	0.782	0.584	0.632	0.770	0.624
DHSNet [66]	94.04	15.8	10.0	0.837	-	0.705	0.816	0.666
RFCN [67]	134.69	102.8	0.4	0.725	0.562	0.586	0.707	0.591
NLDF [68]	35.49	263.9	18.5	0.835	0.634	0.710	0.838	0.676
DSS [69]	62.23	114.6	7.0	0.864	0.688	0.752	0.862	0.702
Amulet [18]	33.15	45.3	9.7	0.839	0.626	0.657	0.817	0.650
UCF [70]	23.98	61.4	12.0	0.805	0.573	0.595	0.779	0.613
SRM [71]	43.74	20.3	12.3	0.849	0.658	0.721	0.835	0.684
PiCANet [17]	32.85	37.1	5.6	0.862	0.691	0.745	0.847	0.687
BRN [72]	126.35	24.1	3.6	0.887	0.709	0.774	0.875	0.712
C2S [73]	137.03	20.5	16.7	0.849	0.663	0.717	0.835	0.685
RAS [74]	20.13	35.6	20.4	0.855	0.695	0.739	0.849	0.691
DNA [75]	20.06	82.5	25.0	0.897	0.729	0.797	**0.889**	0.723
CPD [76]	29.23	59.5	68.0	0.889	0.715	0.799	0.879	**0.731**
BASNet [77]	87.06	127.3	36.2	0.898	**0.751**	0.802	**0.889**	0.721
AFNet [78]	37.11	38.4	21.6	0.880	0.717	0.784	0.869	0.719
PoolNet [79]	53.63	123.4	39.7	0.875	0.710	0.783	0.864	0.724
EGNet [80]	108.07	270.8	12.7	0.886	0.727	0.796	0.876	0.727
BANet [81]	55.90	121.6	12.5	**0.901**	0.736	**0.810**	**0.889**	0.730
CoSOV1Net (OURS)	**1.14**	**1.4**	**211.2**	0.861	0.696	0.731	0.834	0.688

We also compared CoSOV1Net with the state-of-the-art lightweight salient object-detection models MobileNet [30], MobileNetV2 [31], ShuffleNet [32], ShuffleNetV2 [33], ICNet [82], BiSeNet R18 [83], BiSeNet X39 [83], DFANet [84], HVPNet [19] and SAMNet [16].

For the comparison with state-of-the-art lightweight models, Table 5.4 shows that our proposed model outperforms these state-of-the-art lightweight models in parameter numbers and the F_β measure for the ECSSD dataset and is competitive for other measures and datasets. Table 5.5 shows that our model outperforms these state-of-the-art lightweight models for the MAE measure for the ECSSD and DUTS-TE datasets and is ranked first ex aequo with HVPNet for DUT-OMRON, first ex aequo with HVPNet and SAMNet for the HKU-IS dataset and second for the THUR15K dataset. Our model also outperforms these state-of-the-art lightweight models for the F_β^ω measure for ECSSD and DUTS-TE and is competitive for the three other datasets (see Table 5.6).

Tables 5.4–5.6 show that CoSOV1Net clearly has the advantage of the number of parameters over the lightweight salient object detection. They also show that its performance is closer to the best among them. Thus, CoSOV1Net has the advantage of performance.

Regarding computational cost, CoSOV1Net has an advantage over half of the state-of-the-art lightweight salient object-detection models. Overall, we can conclude that it has an advantage in terms of computational cost.

5.5.7 Comparison with SAMNet and HVPNet State of the Art

We chose to compare our CoSOV1Net model specifically with SAMNet [16] and HVPNet [19] because they are among the best state-of-the-art models.

Figure 5.8 shows that precision curves for ECSSD and HKU-IS datasets highlight that CoSOV1Net slightly dominates the SAMNet and HVPNet state-of-the-art lightweight salient object-detection models and that there is no clear domination for the DUT-OMRON, DUTS-TE and THUR15K precision curves between the three models. Therefore, the proposed model CoSOV1Net is competitive with these two state-of-the-art lightweight salient object-detection models with respect to precision.

TABLE 5.4 : Our proposed model's F-measure ($F_\beta \uparrow$, $\beta^2 = 0.3$) compared with state-of-the-art lightweight salient object-detection models (best value in bold) [# Param : number of parameters, $\uparrow$: great is the best, $\downarrow$: small is the best].

Methods	**# Param (M)** $\downarrow$	**FLOPS (G)** $\downarrow$	**Speed (FPS)** $\uparrow$	**ECSSD**	**DUT-OMRONTE**	**DUTS-TE**	**HKU-IS**	**THUR15K**
MobileNet * [30]	4.27	2.2	295.8	0.906	0.753	0.804	0.895	0.767
MobileNetV2 * [31]	2.37	0.8	446.2	0.905	0.758	0.798	0.890	0.766
ShuffleNet * [32]	1.80	0.7	406.9	0.907	0.757	0.811	0.898	0.771
ShuffleNetV2 * [33]	1.60	**0.5**	**452.5**	0.901	0.746	0.789	0.884	0.755
ICNet [82]	6.70	6.3	75.1	0.918	0.773	0.810	0.898	0.768
BiSeNet R18 [83]	13.48	25.0	120.5	0.909	0.757	0.815	0.902	0.776
BiSeNet X39 [83]	1.84	7.3	165.8	0.901	0.755	0.787	0.888	0.756
DFANet [84]	1.83	1.7	91.4	0.896	0.750	0.791	0.884	0.757
HVPNet [19]	1.23	1.1	333.2	0.925	**0.799**	**0.839**	**0.915**	**0.787**
SAMNet [16]	1.33	**0.5**	343.2	0.925	0.797	0.835	**0.915**	0.785
CoSOV1Net (OURS)	**1.14**	1.4	211.2	**0.931**	0.789	0.833	0.912	0.773

* SAMNet, where the encoder is replaced by this backbone.

TABLE 5.5 : Our proposed model MAE (↓) compared with state-of-the art lightweight salient object-detection models (best value in bold) [# Param : number of parameters, ↑ : great is the best, ↓ : small is the best].

Methods	**# Param (M) ↓**	**FLOPS (G) ↓**	**Speed (FPS) ↑**	**ECSSD**	**DUT-OMRON**	**DUTS-TE**	**HKU-IS**	**THUR15K**
MobileNet * [30]	4.27	2.2	295.8	0.064	0.073	0.066	0.052	0.081
MobileNetV2 * [31]	2.37	0.8	446.2	0.066	0.075	0.070	0.056	0.085
ShuffleNet * [32]	1.80	0.7	406.9	0.062	0.069	0.062	0.050	0.078
ShuffleNetV2 * [33]	1.60	**0.5**	**452.5**	0.069	0.076	0.071	0.059	0.086
ICNet [82]	6.70	6.3	75.1	0.059	0.072	0.067	0.052	0.084
BiSeNet R18 [83]	13.48	25.0	120.5	0.062	0.072	0.062	0.049	0.080
BiSeNet X39 [83]	1.84	7.3	165.8	0.070	0.078	0.074	0.059	0.090
DFANet [84]	1.83	1.7	91.4	0.073	0.078	0.075	0.061	0.089
HVPNet [19]	1.23	1.1	333.2	0.055	**0.064**	0.058	**0.045**	0.076
SAMNet [16]	1.33	**0.5**	343.2	0.053	0.065	0.058	**0.045**	**0.077**
CoSOV1Net (OURS)	**1.14**	1.4	211.2	**0.051**	**0.064**	**0.057**	**0.045**	0.076

* SAMNet, where the encoder is replaced by this backbone.

TABLE 5.6 : Our proposed model's weighted F-measure ($F_\beta^\omega \uparrow$, $\beta^2 = 1$) compared with lightweight salient object-detection models (best value in bold) [# Param : number of parameters, $\uparrow$: great is the best, $\downarrow$: small is the best].

Methods	**# Param (M)** $\downarrow$	**FLOPS (G)** $\downarrow$	**Speed (FPS)** $\uparrow$	**ECSSD**	**DUT-OMRON**	**DUTS-TE**	**HKU-IS**	**THUR15K**
MobileNet * [30]	4.27	2.2	295.8	0.829	0.656	0.696	0.816	0.675
MobileNetV2 * [31]	2.37	0.8	446.2	0.820	0.651	0.676	0.799	0.660
ShuffleNet * [32]	1.80	0.7	406.9	0.831	0.667	0.709	0.820	0.683
ShuffleNetV2 * [33]	1.60	**0.5**	**452.5**	0.812	0.637	0.665	0.788	0.652
ICNet [82]	6.70	6.3	75.1	0.838	0.669	0.694	0.812	0.668
BiSeNet R18 [83]	13.48	25.0	120.5	0.829	0.648	0.699	0.819	0.675
BiSeNet X39 [83]	1.84	7.3	165.8	0.802	0.632	0.652	0.784	0.641
DFANet [84]	1.83	1.7	91.4	0.799	0.627	0.652	0.778	0.639
HVPNet [19]	1.23	1.1	333.2	0.854	**0.699**	0.730	**0.839**	**0.696**
SAMNet [16]	1.33	**0.5**	343.2	0.855	**0.699**	0.729	0.837	0.693
CoSOV1Net (OURS)	**1.14**	1.4	211.2	**0.861**	0.696	**0.731**	0.834	0.688

* SAMNet, where the encoder is replaced by this backbone.

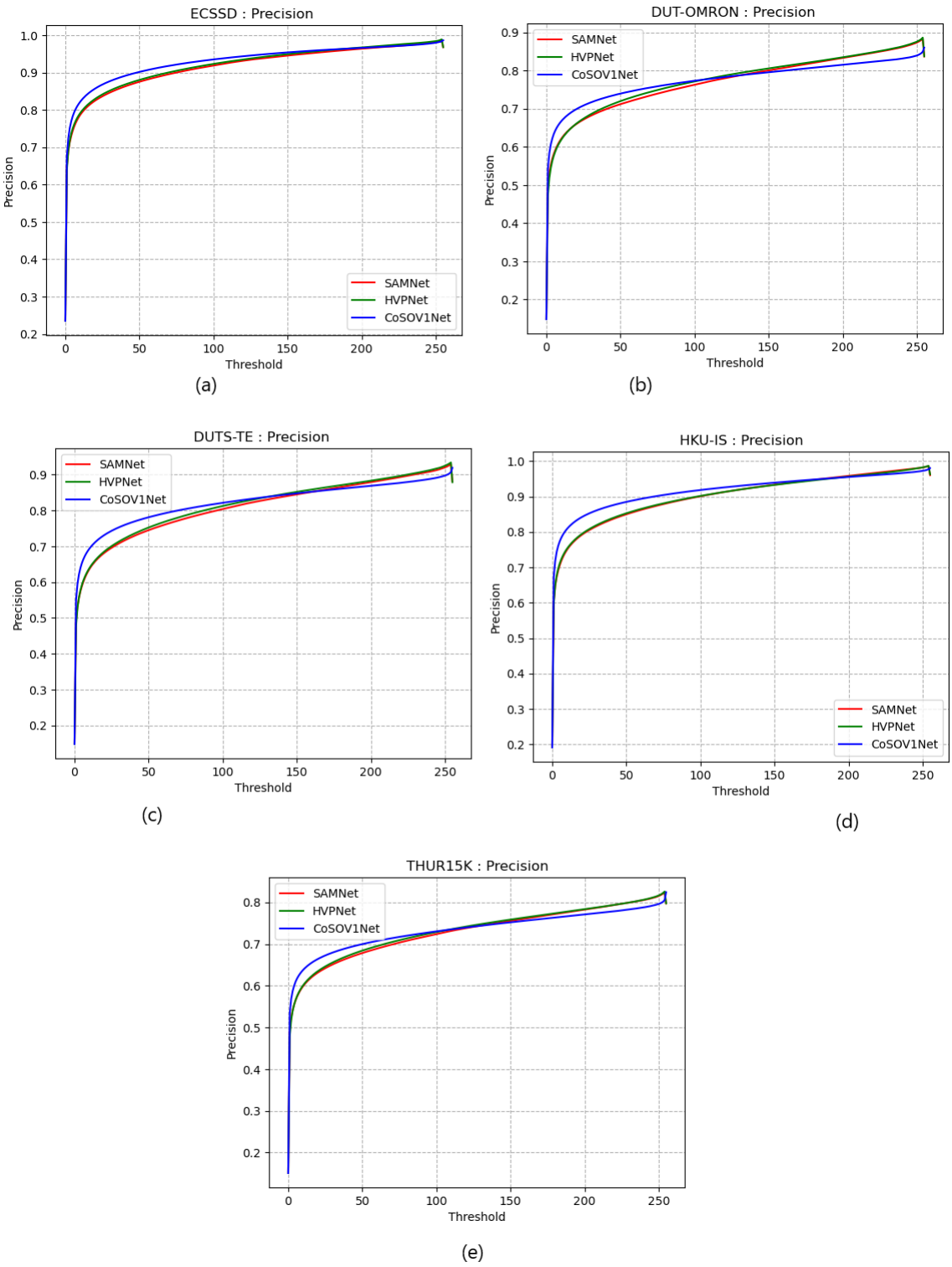


FIGURE 5.8 : Precision curves for (**a**) ECSSD, (**b**) DUT-OMRON, (**c**) DUTS-TE, (**d**) HKU-IS and (**e**) THUR15K datasets.

Figure 5.9 shows that the three models' precision–recall curves (for the five datasets used : ECSSD, DUT-OMRON, DUTS-TE, HKU-IS and THUR15K) are very close to each other. Therefore, the proposed model is competitive with these two state-of-the-art lightweight salient object-detection models with respect to precision–recall.

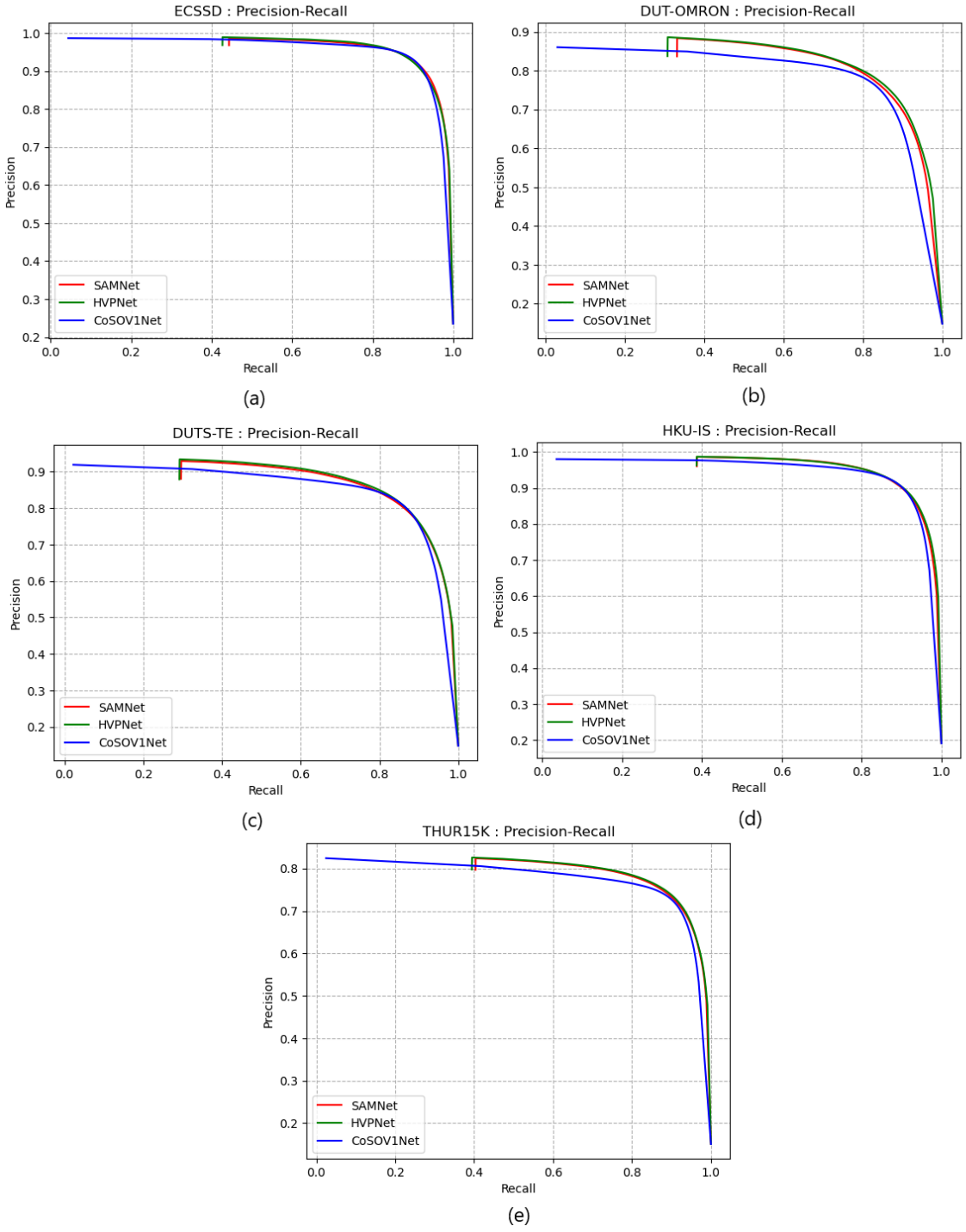


FIGURE 5.9 : Precision–recall curves for (**a**) ECSSD, (**b**) DUT-OMRON, (**c**) DUTS-TE, (**d**) HKU-IS and (**e**) THUR15K datasets.

Figure 5.10 shows that the three models' F_β measure curves (for the five datasets used : ECSSD, DUT-OMRON, DUTS-TE, HKU-IS and THUR15K) are very close to each other. The CoSOV1Net model slightly dominates the two state-of-the-art lightweight salient object-detection models for thresholds ≤ 150 and the two state-of-the-art models slightly dominate for thresholds ≥ 150. Thus, there is no clear dominance for one model among the three. This proves that our CoSOV1Net model is comparable to these state-of-the-art lightweight salient object-detection models while having the advantage of a low number of parameters compared to them.

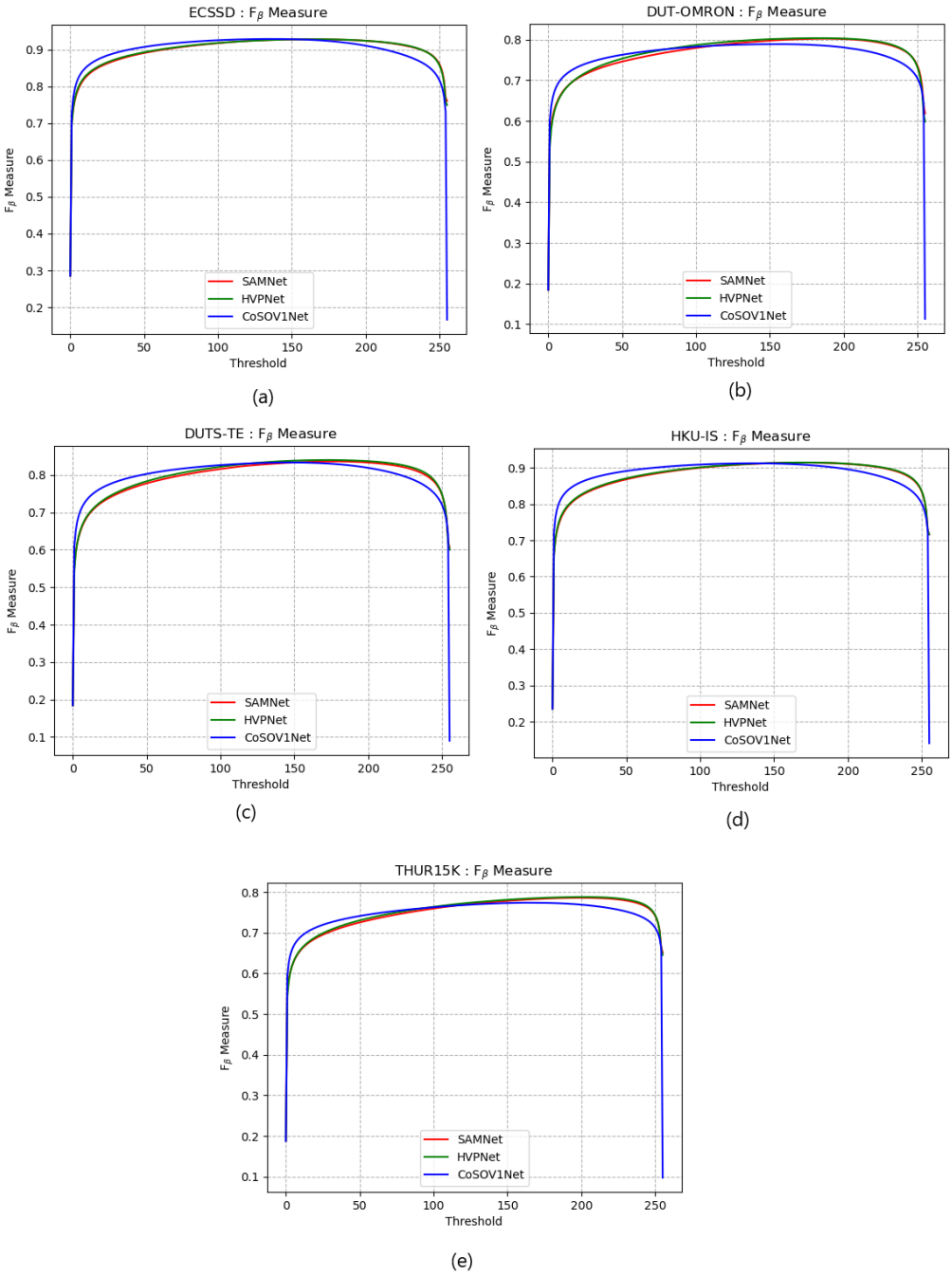


FIGURE 5.10 : F_β measure curves for (**a**) ECSSD, (**b**) DUT-OMRON, (**c**) DUTS-TE, (**d**) HKU-IS and (**e**) THUR15K datasets.

For qualitative comparison, Figure 5.11 shows some images highlighting that our

proposed model (CoSOV1Net) is competitive with regard to the state-of-the-art SAMNet [16] and HVPNet [19] models, which are among the best ones.

Images from rows 1 and 2 show a big salient object on a cloudy background and a big object on a complex background, respectively : CoSOV1Net (ours) performs better than HVPNet on these saliency maps. Row 3 shows salient objects with the same colors and row 4 shows salient objects with multiple colors : the SAMNet and CoSOV1Net saliency maps are slightly identical and the HVPNet saliency map is slightly better. Row 5 shows n image with three salient objects with different sizes and colors : two are big and one is very small ; the CoSOV1Net saliency map is better than SAMNet's and HVP-Net's. Row 6 shows red salient objects on a black and yellow background ; SAMNet's saliency map is the worst, while CoSOV1Net and HVPNet perform well on that image. Row 7 shows a complex background and multiple salient objects with different colors : CoSOV1Net performs better than SAMNet and HVPNet. Row 8 shows tiny salient objects : the three models perform well. On row 9, SAMNet has the worst performance, while CoSOV1Net is the best. Row 10 shows colored glasses as salient objects : the CoSOV1Net performance is better than SAMNet's and HVPNet's. On row 11, SAMNet has the worst performance. On row 12 and 13, CoSOV1Net has the best performance. Row 18 shows a submarine image : CoSOV1Net is better than SAMNet.

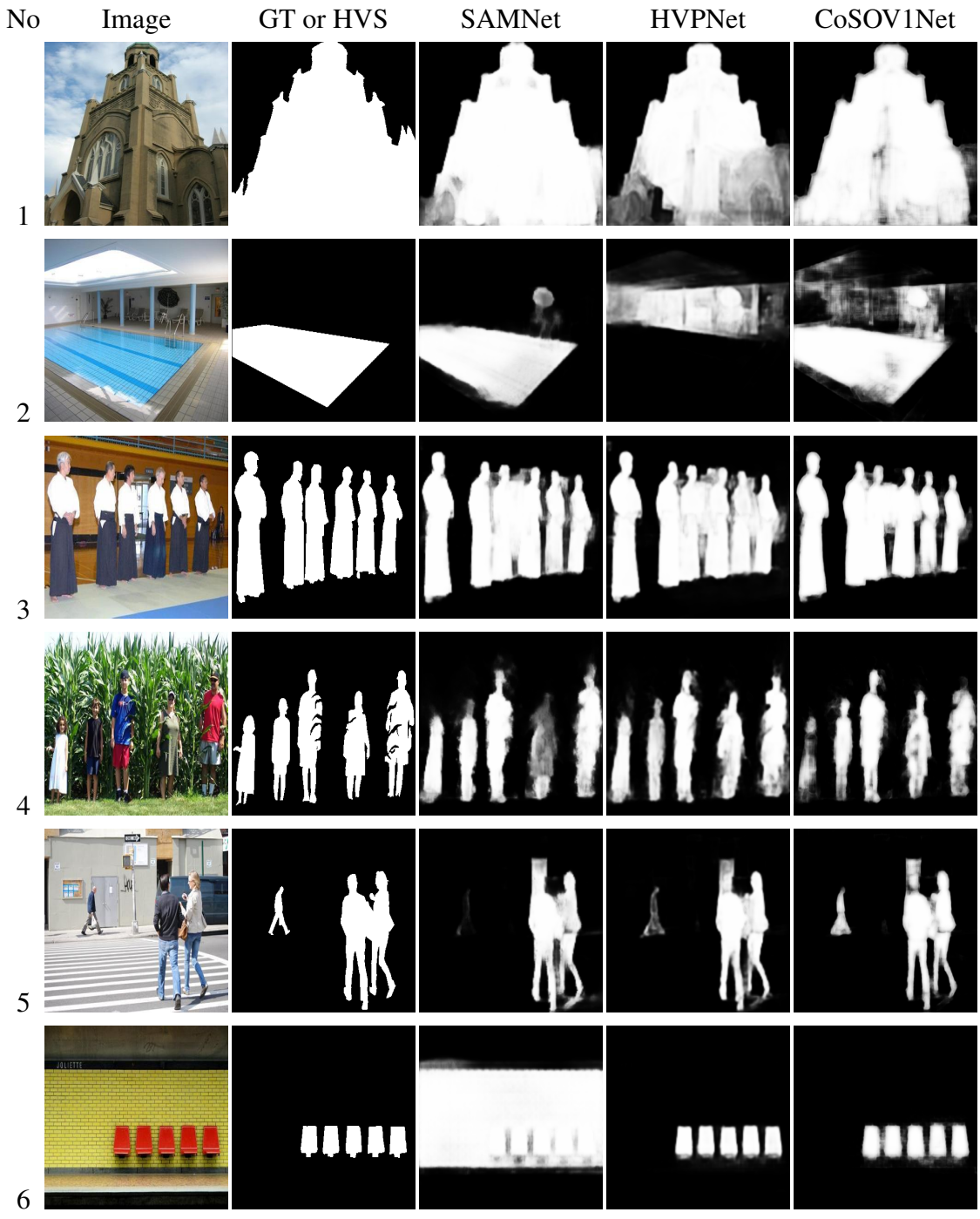


FIGURE 5.11 : *Cont.*

FIGURE 5.11 : *Cont.*

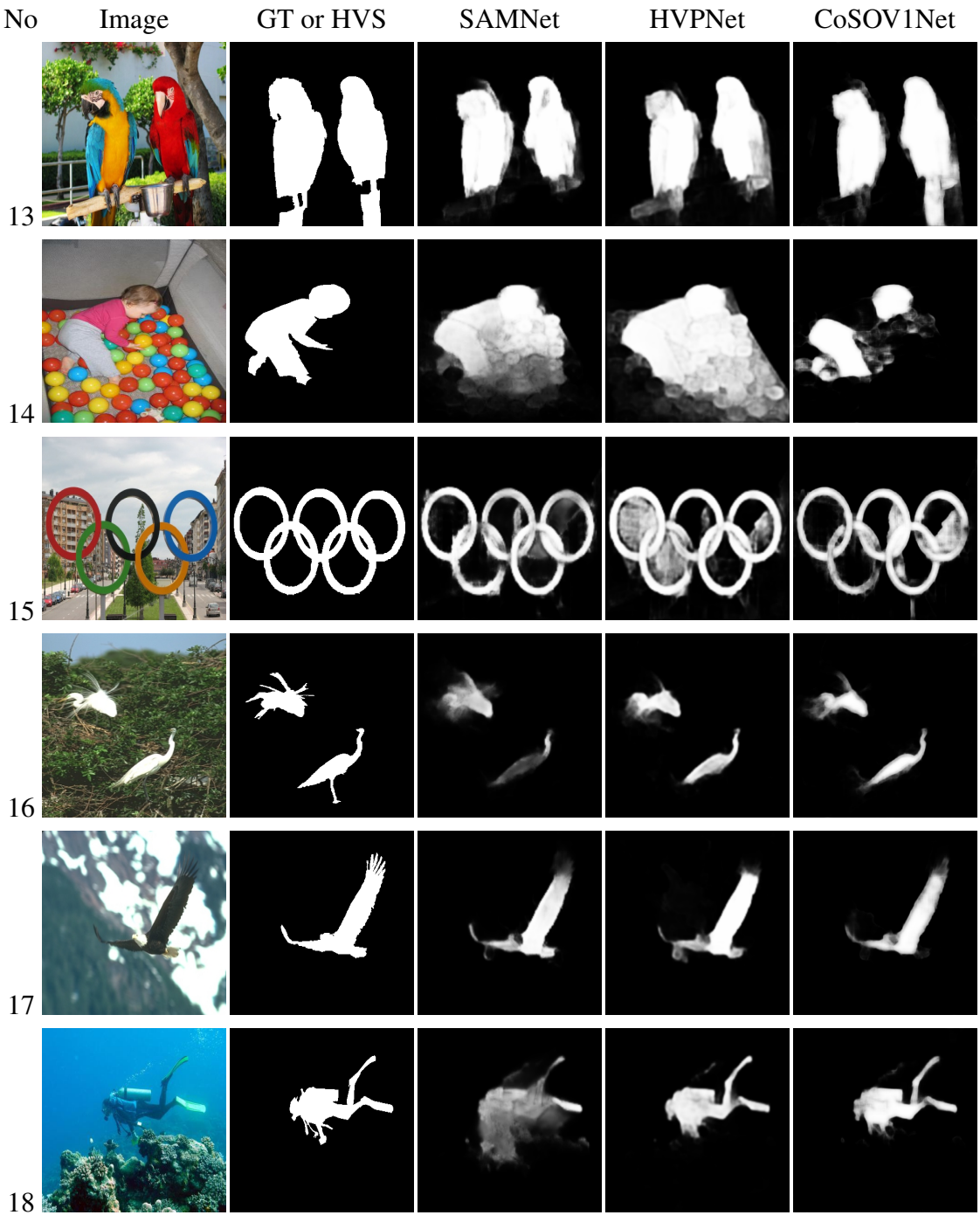


FIGURE 5.11 : Comparison between SAMNet [16], HVPNet [19] and our proposed model, CoSOV1Net, on some image saliency maps : 1st column : images ; 2nd column : ground truth or human visual system saliency map ; 3rd column : SAMNet ; 4th column : HVPNet ; 5th column : CoSOV1Net (ours).

Figures 5.8–5.11 confirm that CoSOV1Net has an advantage on performance.

5.6 Discussion

The results show the performance of our model, CoSOV1Net, for accuracy measures and lightweight measures. CoSOV1Net's rank, when compared to state-of-the-art models, shows that it behaves as a lightweight salient object-detection model by dominating lightweight measures and having good performance for accuracy measures (see Table 5.7).

TABLE 5.7 : Our proposed model (CoSOV1Net)'s ranking with respect to existing salient object detection [# Param : number of parameters, ↑ : great is the best, ↓ : small is the best].

Measure	**# Param (M) ↓**	**FLOPS (G) ↓**	**Speed (FPS) ↑**	**ECSSD**	**DUT-OMRON**	**DUTS-TE**	**HKU-IS**	**THUR15K**
F_β	1st	1st	1st	6th	7th	9th	11th	11th
MAE	1st	1st	1st	10th	10th	11th	11th	10th
F_β^ω	1st	1st	1st	11th	9th	11th	15th	11th

The results also show that when CoSOV1Net is compared to state-of-the-art lightweight salient object-detection models, its measure results are generally ranked among the best for the datasets and measures used (see Table 5.8). Thus, we can conclude that CoSOV1Net behaves as a competitive lightweight salient object-detection model.

TABLE 5.8 : Our proposed model (CoSOV1Net)'s ranking with respect to lightweight salient object-detection models [# Param : number of parameters, ↑ : great is the best, ↓ : small is the best].

Measure	**# Param (M) ↓**	**FLOPS (G) ↓**	**Speed (FPS) ↑**	**ECSSD**	**DUT-OMRON**	**DUTS-TE**	**HKU-IS**	**THUR15K**
F_β	1st	6th	7th	1st	3rd	3rd	3rd	4th
MAE	1st	6th	7th	1st	1st	1st	1st	2nd
F_β^ω	1st	6th	7th	1st	3rd	1st	3rd	3rd

As we did not use backbones from image classification (i.e., VGG [59], ...) or lightweight backbones (i.e., MobileNets [30, 31] or ShuffleNets [32, 33]), we conclude that CoSOV1Net's performance is intrinsic to this model itself.

Finally, putting together the measures for salient object-detection models and lightweight salient object-detection models in a graphic, we noticed that the CoSOV1Net model is located for F_β measures with respect to FLOPS and for the number of parameters in the top left, while for the FPS measure, it is located in the top right, thus demonstrating its performance as a lightweight salient object-detection model (see Figure 5.12). This shows that CoSOV1Net is competitive with the best state-of-the-art models used.

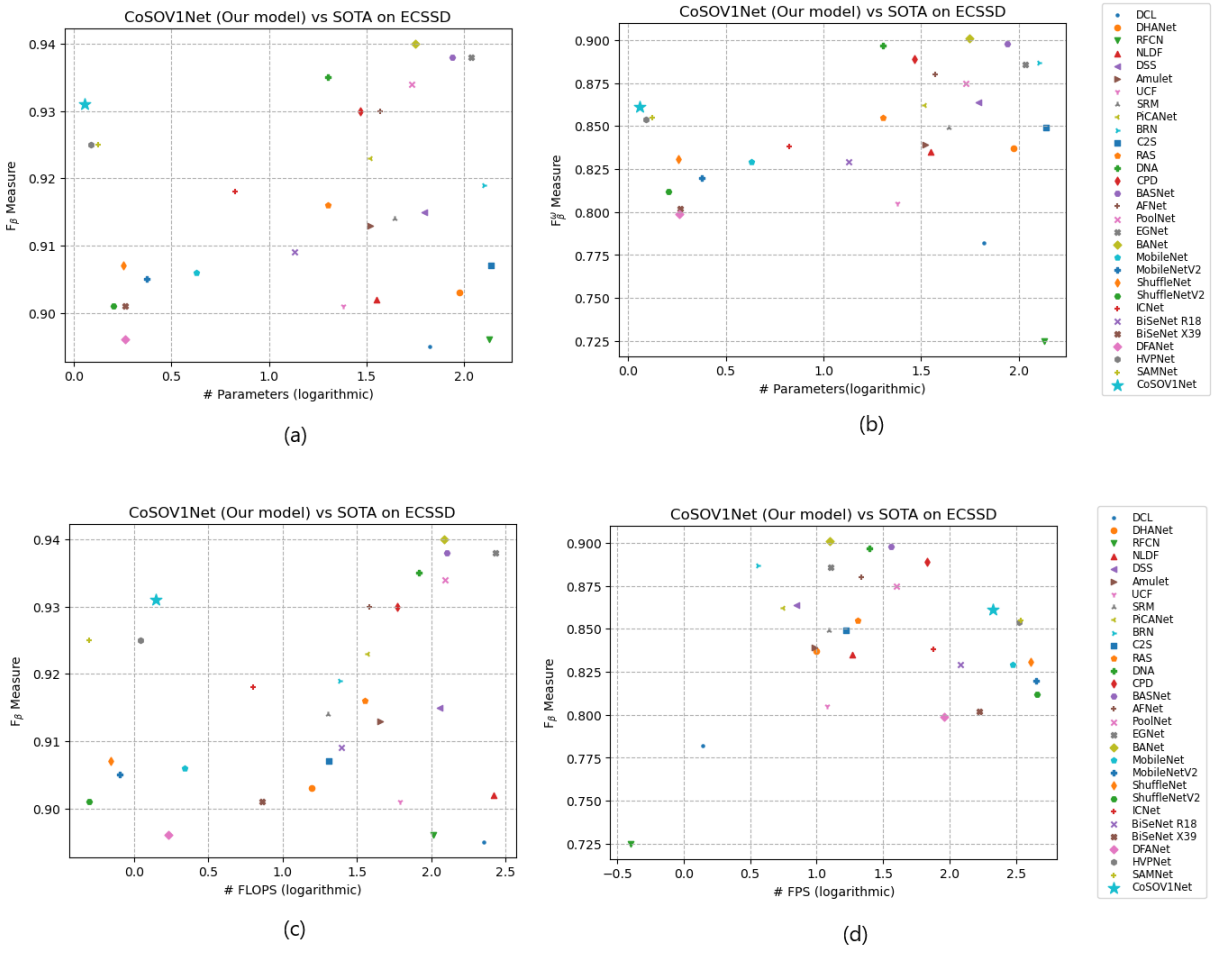


FIGURE 5.12 : Example of trade-off between (**a**) F_β measure and #parameters; (**b**) F_β^ω measure and #parameters; (**c**) F_β measure and FLOPS; (**d**) F_β measure and FPS, for ECSSD.

The quantitative and the qualitative comparisons with SAMNet [16] and HVPNet [19] showed that our proposed model has good performance, given that these state-of-the-art models are among the best ones.

5.7 Conclusion

In this work, we present a lightweight salient object-detection deep neural network, CoSOV1Net, with a very low number of parameters (1.14 M), a low floating-point ope-

rations number (FLOPS = 1.4 G) and thus low computational cost and respectable speed (FPS $=$ 211.2 on GPU : Nvidia GeForce RTX 3090 Ti), yet with comparable performance with state-of-the-art salient object-detection models that use significantly more parameters, and other lightweight salient object-detection models such as SAMNet [16] and HVPNet [19].

The novelty of our proposed model (CoSOV1Net) is that it uses the principle of integrating color in pattern in a salient object-detection deep neural network, since according to Shapley [27] and Shapley and Hawken [20], color and pattern are inextricably linked in color human perception. This is implemented by taking inspiration from the primary visual cortex (V1) cells, especially cone- and spatial-opponent cells. Thus, our method extracts features at the color channels' spatial level and between the color channels at the same time on a pair of opposing color channels. The idea of grouping color pushed us to group feature maps through the neural network and extract features at the spatial level and between feature maps, as carried out for color channels.

Our results showed that this strategy generates a model that is very promising, competitive with most state-of-the-art salient object-detection and lightweight salient object-detection models and practical for mobile environments and limited-resource devices.

In future work, our proposed CoSOV1Net model, based on integrating color into patterns, can be improved by coupling it with the human visual system attention mechanism, which is the basis of many lightweight models, to tackle its speed limitation and thus produce a more efficient lightweight salient object-detection model.

BIBLIOGRAPHIE

[1] Didier Ndayikengurukiye and Max Mignotte. Salient object detection by ltp texture characterization on opposing color pairs under slico superpixel constraint. *J. Imaging*, 8(4) :110, 2022.

[2] Arnold WM Smeulders, Dung M Chu, Rita Cucchiara, Simone Calderara, Afshin Dehghan, and Mubarak Shah. Visual tracking : An experimental survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 36(7) :1442–1468, 2013.

[3] Rik Pieters and Michel Wedel. Attention capture and transfer in advertising : Brand, pictorial, and text-size effects. *J. Mark.*, 68(2) :36–50, 2004.

[4] Laurent Itti. Automatic foveation for video compression using a neurobiological model of visual attention. *IEEE Trans. Image Processing*, 13(10) :1304–1318, 2004.

[5] Jinjiang Li, Xiaomei Feng, and Hui Fan. Saliency-based image correction for colorblind patients. *Comput. Vis. Media*, 6(2) :169–189, 2020.

[6] Nicolò Oreste Pinciroli Vago, Federico Milani, Piero Fraternali, and Ricardo da Silva Torres. Comparing cam algorithms for the identification of salient image features in iconography artwork analysis. *J. Imaging*, 7(7) :106, 2021.

[7] Yuan Gao, Miaojing Shi, Dacheng Tao, and Chao Xu. Database saliency for fast image retrieval. *IEEE Trans. Multimed.*, 17(3) :359–369, 2015.

[8] Lai-Kuan Wong and Kok-Lim Low. Saliency-enhanced image aesthetics class prediction. In *2009 16th IEEE international conference on image processing (ICIP)*, pages 997–1000. IEEE, 2009.

[9] Hantao Liu and Ingrid Heynderickx. Studying the added value of visual attention in objective image quality metrics based on eye movement data. In *2009*

16th IEEE international conference on image processing (ICIP), pages 3097–3100. IEEE, 2009.

[10] Li-Qun Chen, Xing Xie, Xin Fan, Wei-Ying Ma, Hong-Jiang Zhang, and He-Qin Zhou. A visual attention model for adapting images on small displays. *Multimedia systems*, 9 :353–364, 2003.

[11] Tao Chen, Ming-Ming Cheng, Ping Tan, Ariel Shamir, and Shi-Min Hu. Sketch2photo : Internet image montage. *ACM Trans. Graph. (TOG)*, 28(5) :1–10, 2009.

[12] Hua Huang, Lei Zhang, and Hong-Chao Zhang. Arcimboldo-like collage using internet images. In *Proceedings of the 2011 SIGGRAPH Asia Conference*, pages 1–8, 2011.

[13] Ashish Kumar Gupta, Ayan Seal, Mukesh Prasad, and Pritee Khanna. Salient object detection techniques in computer vision—a survey. *Entropy*, 22(10) :1174, 2020.

[14] Wenguan Wang, Qiuxia Lai, Huazhu Fu, Jianbing Shen, Haibin Ling, and Ruigang Yang. Salient object detection in the deep learning era : An in-depth survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(6) :3239–3259, 2021.

[15] Shang-Hua Gao, Yong-Qiang Tan, Ming-Ming Cheng, Chengze Lu, Yunpeng Chen, and Shuicheng Yan. Highly efficient salient object detection with 100k parameters. In *Computer Vision–ECCV 2020 : 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VI*, pages 702–721. Springer, 2020.

[16] Yun Liu, Xin-Yu Zhang, Jia-Wang Bian, Le Zhang, and Ming-Ming Cheng. Samnet : Stereoscopically attentive multi-scale network for lightweight salient object detection. *IEEE Trans. Image Processing*, 30 :3804–3814, 2021.

[17] Nian Liu, Junwei Han, and Ming-Hsuan Yang. Picanet : Learning pixel-wise contextual attention for saliency detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3089–3098, 2018.

[18] Pingping Zhang, Dong Wang, Huchuan Lu, Hongyu Wang, and Xiang Ruan. Amulet : Aggregating multi-level convolutional features for salient object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 202–211, 2017.

[19] Yun Liu, Yu-Chao Gu, Xin-Yu Zhang, Weiwei Wang, and Ming-Ming Cheng. Lightweight salient object detection via hierarchical visual perception learning. *IEEE Trans. Cybern.*, 51(9) :4439–4449, 2020.

[20] Robert Shapley and Michael J Hawken. Color in the cortex : Single-and double-opponent cells. *Vision research*, 51(7) :701–717, 2011.

[21] Norbert Kruger, Peter Janssen, Sinan Kalkan, Markus Lappe, Ales Leonardis, Justus Piater, Antonio J Rodriguez-Sanchez, and Laurenz Wiskott. Deep hierarchies in the primate visual cortex : What can we learn for computer vision ? *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(8) :1847–1871, 2012.

[22] Valerie Nunez, Robert M Shapley, and James Gordon. Cortical double-opponent cells in color perception : Perceptual scaling and chromatic visual evoked potentials. *i-Perception*, 9(1) :2041669517752715, 2018.

[23] Bevil R Conway. Color vision, cones, and color-coding in the cortex. *The neuroscientist*, 15(3) :274–290, 2009.

[24] Bevil R Conway. Spatial structure of cone inputs to color cells in alert macaque primary visual cortex (v-1). *J. Neurosci.*, 21(8) :2768–2783, 2001.

[25] Robert William Gainer Hunt and Michael R Pointer. *Measuring colour*. John Wiley & Sons, 2011.

[26] Stephen Engel, Xuemei Zhang, and Brian Wandell. Colour tuning in human visual cortex measured with functional magnetic resonance imaging. *Nature*, 388(6637) :68–71, 1997.

[27] Robert Shapley. Physiology of color vision in primates. In *Oxford Research Encyclopedia of Neuroscience*. 2019.

[28] Xuebin Qin, Zichen Zhang, Chenyang Huang, Masood Dehghan, Osmar R Zaiane, and Martin Jagersand. U2-net : Going deeper with nested u-structure for salient object detection. *Pattern recognition*, 106 :107404, 2020.

[29] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net : Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015 : 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015.

[30] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets : Efficient convolutional neural networks for mobile vision applications. *Arxiv Prepr. Arxiv :1704.04861*, 2017.

[31] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2 : Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.

[32] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet : An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6848–6856, 2018.

[33] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2 : Practical guidelines for efficient cnn architecture design. In *Proceedings of the European conference on computer vision (ECCV)*, pages 116–131, 2018.

[34] Simone Frintrop, Thomas Werner, and German Martin Garcia. Traditional saliency reloaded : A good old model in new shape. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 82–90, 2015.

[35] Topi Mäenpää and Matti Pietikäinen. Classification with color and texture : Jointly or separately ? *Pattern recognition*, 37(8) :1629–1640, 2004.

[36] Chi-Ho Chan, Josef Kittler, and Kieron Messer. Multispectral local binary pattern histogram for component-based color face verification. In *2007 First IEEE International Conference on Biometrics : Theory, Applications, and Systems*, pages 1–7. IEEE, 2007.

[37] Christos Faloutsos and King-Ip Lin. *FastMap : A fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets*, volume 24. ACM, 1995.

[38] Amit Jain and Glenn Healey. A multiscale representation including opponent color features for texture recognition. *IEEE Trans. Image Processing*, 7(1) :124–128, 1998.

[39] Kai-Fu Yang, Shao-Bing Gao, Ce-Feng Guo, Chao-Yi Li, and Yong-Jie Li. Boundary detection using double-opponency and spatial sparseness constraint. *IEEE Trans. Image Processing*, 24(8) :2565–2578, 2015.

[40] Leo M Hurvich and Dorothea Jameson. An opponent-process theory of color vision. *Psychological review*, 64(6p1) :384, 1957.

[41] Clement Farabet, Camille Couprie, Laurent Najman, and Yann LeCun. Learning hierarchical features for scene labeling. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(8) :1915–1929, 2012.

[42] Sergey Ioffe and Christian Szegedy. Batch normalization : Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. pmlr, 2015.

[43] Shibani Santurkar, Dimitris Tsipras, Andrew Ilyas, and Aleksander Madry. How does batch normalization help optimization ? *Adv. Neural Inf. Processing Syst.*, 31, 2018.

[44] Francois Chollet. *Deep learning with Python*. Simon and Schuster, 2021.

[45] François Chollet. Xception : Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017.

[46] Golnaz Ghiasi, Tsung-Yi Lin, and Quoc V Le. Dropblock : A regularization method for convolutional networks. *Adv. Neural Inf. Processing Syst.*, 31, 2018.

[47] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout : A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1) :1929–1958, 2014.

[48] Matti Pietikäinen, Abdenour Hadid, Guoying Zhao, and Timo Ahonen. *Computer vision using local binary patterns*, volume 40. Springer Science & Business Media, 2011.

[49] Athanasios Voulodimos, Nikolaos Doulamis, Anastasios Doulamis, and Eftychios Protopapadakis. Deep learning for computer vision : A brief review. *Comput. Intell. Neurosci.*, 2018, 2018.

[50] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[51] François Chollet et al. Keras. `https://keras.io`, 2015.

[52] Ali Borji, Ming-Ming Cheng, Huaizu Jiang, and Jia Li. Salient object detection : A benchmark. *IEEE Trans. Image Processing*, 24(12) :5706–5722, 2015.

[53] Jianping Shi, Qiong Yan, Li Xu, and Jiaya Jia. Hierarchical image saliency detection on extended cssd. *IEEE Trans. Pattern Anal. Mach. Intell.*, 38(4) :717–729, 2016.

[54] Chuan Yang, Lihe Zhang, Huchuan Lu, Xiang Ruan, and Ming-Hsuan Yang. Saliency detection via graph-based manifold ranking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3166–3173, 2013.

[55] Lijun Wang, Huchuan Lu, Yifan Wang, Mengyang Feng, Dong Wang, Baocai Yin, and Xiang Ruan. Learning to detect salient objects with image-level supervision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 136–145, 2017.

[56] Guanbin Li and Yizhou Yu. Visual saliency based on multiscale deep features. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5455–5463, 2015.

[57] Ming-Ming Cheng, Niloy J Mitra, Xiaolei Huang, and Shi-Min Hu. Salientshape : Group saliency in image collections. *Vis. Comput.*, 30 :443–453, 2014.

[58] Weijia Feng, Xiaohui Li, Guangshuai Gao, Xingyue Chen, and Qingjie Liu. Multi-scale global contrast cnn for salient object detection. *Sensors*, 20(9) :2656, 2020.

[59] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *Arxiv Prepr. Arxiv :1409.1556*, 2014.

[60] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers : Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[61] Tijmen Tieleman, Geoffrey Hinton, et al. Lecture 6.5-rmsprop : Divide the gradient by a running average of its recent magnitude. *COURSERA : Neural Netw. Mach. Learn.*, 4(2) :26–31, 2012.

[62] Ran Margolin, Lihi Zelnik-Manor, and Ayellet Tal. How to evaluate foreground maps ? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2014.

[63] Vijayakumar Varadarajan, Dweepna Garg, and Ketan Kotecha. An efficient deep convolutional neural network approach for object detection and recognition using a multi-scale anchor box in real-time. *Future Internet*, 13(12) :307, 2021.

[64] Huaizu Jiang, Jingdong Wang, Zejian Yuan, Yang Wu, Nanning Zheng, and Shipeng Li. Salient object detection : A discriminative regional feature integration approach. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2083–2090, 2013.

[65] Guanbin Li and Yizhou Yu. Deep contrast learning for salient object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 478–487, 2016.

[66] Nian Liu and Junwei Han. Dhsnet : Deep hierarchical saliency network for salient object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 678–686, 2016.

[67] Jianhuan Wei and Baojiang Zhong. Saliency detection using fully convolutional network. In *2018 Chinese Automation Congress (CAC)*, pages 3902–3907. IEEE, 2018.

[68] Zhiming Luo, Akshaya Mishra, Andrew Achkar, Justin Eichel, Shaozi Li, and Pierre-Marc Jodoin. Non-local deep features for salient object detection. In *Proceedings of the IEEE Conference on computer vision and pattern recognition*, pages 6609–6617, 2017.

[69] Qibin Hou, Ming-Ming Cheng, Xiaowei Hu, Ali Borji, Zhuowen Tu, and Philip HS Torr. Deeply supervised salient object detection with short connections. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3203–3212, 2017.

[70] Pingping Zhang, Dong Wang, Huchuan Lu, Hongyu Wang, and Baocai Yin. Learning uncertain convolutional features for accurate saliency detection. In *Proceedings of the IEEE International Conference on computer vision*, pages 212–221, 2017.

[71] Tiantian Wang, Ali Borji, Lihe Zhang, Pingping Zhang, and Huchuan Lu. A stagewise refinement model for detecting salient objects in images. In *Proceedings of the IEEE international conference on computer vision*, pages 4019–4028, 2017.

[72] Tiantian Wang, Lihe Zhang, Shuo Wang, Huchuan Lu, Gang Yang, Xiang Ruan, and Ali Borji. Detect globally, refine locally : A novel approach to saliency detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3127–3135, 2018.

[73] Xin Li, Fan Yang, Hong Cheng, Wei Liu, and Dinggang Shen. Contour knowledge transfer for salient object detection. In *Proceedings of the european conference on computer vision (ECCV)*, pages 355–370, 2018.

[74] Shuhan Chen, Xiuli Tan, Ben Wang, and Xuelong Hu. Reverse attention for salient object detection. In *Proceedings of the European conference on computer vision (ECCV)*, pages 234–250, 2018.

[75] Yun Liu, Ming-Ming Cheng, Xin-Yu Zhang, Guang-Yu Nie, and Meng Wang. Dna : Deeply supervised nonlinear aggregation for salient object detection. *IEEE Trans. Cybern.*, 52(7) :6131–6142, 2021.

[76] Zhe Wu, Li Su, and Qingming Huang. Cascaded partial decoder for fast and accurate salient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3907–3916, 2019.

[77] Xuebin Qin, Zichen Zhang, Chenyang Huang, Chao Gao, Masood Dehghan, and Martin Jagersand. Basnet : Boundary-aware salient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7479–7489, 2019.

[78] Mengyang Feng, Huchuan Lu, and Errui Ding. Attentive feedback network for boundary-aware salient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1623–1632, 2019.

[79] Jiang-Jiang Liu, Qibin Hou, Ming-Ming Cheng, Jiashi Feng, and Jianmin Jiang. A simple pooling-based design for real-time salient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3917–3926, 2019.

[80] Jia-Xing Zhao, Jiang-Jiang Liu, Deng-Ping Fan, Yang Cao, Jufeng Yang, and Ming-Ming Cheng. Egnet : Edge guidance network for salient object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8779–8788, 2019.

[81] Jinming Su, Jia Li, Yu Zhang, Changqun Xia, and Yonghong Tian. Selectivity or invariance : Boundary-aware salient object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3799–3808, 2019.

[82] Hengshuang Zhao, Xiaojuan Qi, Xiaoyong Shen, Jianping Shi, and Jiaya Jia. Icnet for real-time semantic segmentation on high-resolution images. In *Proceedings of the European conference on computer vision (ECCV)*, pages 405–420, 2018.

[83] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. Bisenet : Bilateral segmentation network for real-time semantic segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 325–341, 2018.

[84] Hanchao Li, Pengfei Xiong, Haoqiang Fan, and Jian Sun. Dfanet : Deep feature aggregation for real-time semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9522–9531, 2019.

CHAPITRE 6

CONCLUSION GÉNÉRALE ET PERSPECTIVES

Dans notre thèse, nous avons présenté notre contribution pour résoudre trois problèmes : le problème d'analyse de la marche humaine à des fins médicales, le problème de détection d'objets saillants en général et en particulier celui de trouver des modèles légers de détection d'objets saillants qui peuvent être adaptés aux environnements mobiles et aux appareils à ressources limitées comme les téléphones intelligents, capteurs intelligents, réfrigérateurs intelligents, jouets intelligents, etc.

Dans un premier temps, nous avons proposé un nouveau système d'analyse de la marche basé sur l'estimation d'une carte d'énergie de bruit apériodique qui vise à montrer les zones de fortes irrégularités de la marche, en termes de périodicité, de chaque individu marchant sur un tapis roulant. Il permet également de quantifier le degré de mouvements asymétriques (bras et jambe opposés). Cette carte spatiale 2D est estimée de deux manières complémentaires, à savoir l'estimation dans le domaine temporel et celle dans le domaine fréquentiel afin d'obtenir une estimation de l'information utile avec deux bruits différents et de fournir par la suite des décisions complémentaires à partir de différents classificateurs individuels qui seront ensuite combinés. Cette carte permet également de localiser et de quantifier visuellement et rapidement les anomalies de la marche. Le modèle peut servir pour voir l'évolution de ces anomalies dans le temps. Le système proposé dans cette thèse est également peu coûteux, sans marqueur, non invasif, facile à mettre en place et nécessitant un petit espace. Ces caractéristiques le qualifient pour les activités quotidiennes en clinique. Ce système effectue également une classification automatique des patients sains et de ceux qui ne sont pas en bonne santé avec de bons taux de classification.

Dans un deuxième temps, nous avons proposé un modèle simple, presque sans paramètres, pour l'estimation des cartes de saillance. Nous avons testé notre modèle sur l'ensemble de données complexe ECSSD pour lequel les mesures moyennes de MAE = $0,257$ et $F_\beta = 0,729$, et sur l'ensemble de données MSRA10K. Nous avons également

testé sur THUR15K, qui représente des scènes du monde réel et est considéré comme complexe pour obtenir des objets saillants, et sur des jeux de données DUT-OMRON et SED2. La nouveauté de notre modèle est qu'il n'utilise la caractéristique texturale qu'après avoir intégré les informations de couleur dans ces caractéristiques texturales grâce au principe d'opposition des paires de couleurs d'un espace de couleur donné. Ceci est rendu possible par le descripteur de texture LTP (" Local Ternary Patterns ") qui, étant une extension de LBP ("Local Binary Patterns "), hérite de ses atouts tout en étant moins sensible au bruit dans des régions uniformes. Ainsi, nous caractérisons chaque pixel de l'image par un vecteur caractéristique donné par une micro-texture couleur obtenue grâce à l'algorithme de superpixel SLICO (" Simple Linear Iterative Clustering with zero parameter "). De plus, l'algorithme FastMap réduit chacun de ces vecteurs caractéristiques à une dimension tout en prenant en compte les non-linéarités de ces vecteurs et en préservant leurs distances. Cela signifie que notre carte de saillance combine des approches locales et globales en une seule approche et le fait dans des temps de complexité presque linéaires. Dans ce modèle, nous avons utilisé les espaces de couleur RGB, HSL, LUV et CMY. Notre modèle est donc perfectible si l'on augmente le nombre d'espaces couleurs (les moins corrélés possible) à fusionner. Comme le montrent les résultats que nous avons obtenus, cette stratégie génère un modèle très prometteur, car il est assez différent des méthodes de détection de saillance existantes utilisant la stratégie classique de contraste de couleur entre une région et les autres régions de l'image.

Dans un troisième temps, nous avons proposé un réseau de neurones profond léger de détection d'objets saillants, CoSOV1Net avec un très faible nombre de paramètres (1,14 M), un faible nombre d'opérations en virgule flottante (FLOPS = 1,4 G) donc un faible coût de calcul et une vitesse de prédiction respectable ($FPS = 211,2$ sur GPU : Nvidia GeForce RTX 3090 Ti) mais avec des performances comparables avec des modèles de détection d'objets saillants de l'état de l'art qui utilisent beaucoup plus de paramètres et d'autres modèles légers de détection d'objets saillants tels que SAMNet [1] et HVP-Net [2]. La nouveauté de notre modèle proposé (CoSOV1Net) est qu'il utilise le principe d'intégration de la couleur dans la forme dans un réseau de neurones profond de détection d'objets saillants, puisque selon Shapley [3] et Shapley *et* Hawken [4] la couleur et

la forme sont inextricablement liées dans la perception humaine des couleurs. Ceci est mis en œuvre en s'inspirant du traitement de la perception de couleur et de forme dans le cortex visuel primaire (V1), en particulier l'opposition de signaux de cônes individuellement et spatialement. Ainsi, notre méthode extrait les caractéristiques au niveau spatial des canaux de couleur et entre les canaux de couleur en même temps sur une paire de canaux de couleur opposés. L'idée de regrouper les couleurs nous a poussés à regrouper les cartes de caractéristiques à travers le réseau de neurones et à extraire les caractéristiques au niveau spatial et entre les canaux des cartes de caractéristiques comme cela est fait pour les canaux de couleur. Nos résultats montrent que cette stratégie génère un modèle très prometteur, compétitif avec les modèles de l'état de l'art de la détection d'objets saillants d'une part et d'autre part les modèles légers de l'état de l'art de la détection d'objets saillants. Il est aussi pratique pour les environnements mobiles et les appareils à ressources limitées.

6.1 Perspectives

Pour notre première contribution, nous espérons faire une analyse plus poussée sur d'autres populations, y compris de vrais patients. De plus, nous aimerions dans le futur utiliser un réseau de neurones convolutionel (CNN : " Convolutional Neural Network ") qui détecte les individus sains et ceux qui sont malades. Ce réseau de neurones aurait en entrée les différents signaux de profondeur (voir Figure 3.1) en 1-dimension constituant la séquence vidéo de la caméra de profondeur. D'autres travaux ont utilisé des signaux en 1-dimension provenant de capteurs de pied mesurant la force de réaction verticale du sol ("vertical ground reaction force" : VGRF) [5]. Contrairement à ces systèmes, il serait non-invasif et les signaux proviendraient de toute la surface du corps exposée à la caméra. Nous pourrons aussi à la différence de ces travaux utiliser un réseau de neurones produisant des cartes de saillance en plus de classifier les individus comme sains ou malades.

Pour notre deuxième contribution, notre modèle proposé pourrait être efficacement combiné avec les méthodes de détection de saillance existantes utilisant la stratégie clas-

sique de contraste de couleur entre une région et les autres régions de l'image pour une meilleure performance étant donné que notre modèle utilise une stratégie différente des leurs. Notre modèle peut également être parallélisé (en utilisant la puissance de traitement massivement parallèle des GPU) en traitant chaque paire de couleurs opposées en parallèle. De plus, il convient de noter que cette stratégie d'intégration de la couleur dans les motifs texturaux locaux pourrait également être intéressante à étudier avec des techniques d'apprentissage en profondeur ou des réseaux de neurones convolutifs (CNN) pour améliorer encore la qualité des cartes de saillance.

Pour notre troisième contribution, notre modèle CoSOV1Net proposé, basé sur l'intégration de la couleur dans les motifs, peut être amélioré en le couplant avec le mécanisme d'attention du système visuel humain, qui est la base de nombreux modèles légers, pour produire un modèle léger de détection d'objets saillants plus efficace. Notre modèle, qui a été utilisé sur des images dans un espace de couleur RGB ("Red Green Blue"), pourrait être adapté pour traiter les images RGB-D ("Red Green Blue - Depth"), ajoutant ainsi la perception de profondeur dans l'image. Nous pourrons aussi augmenter d'un niveau le module de base CoSOV1 (voir Figure 6.1) pour augmenter sa capacité étant donné que le cortex visuel primaire (V1) possède plusieurs couches [6].

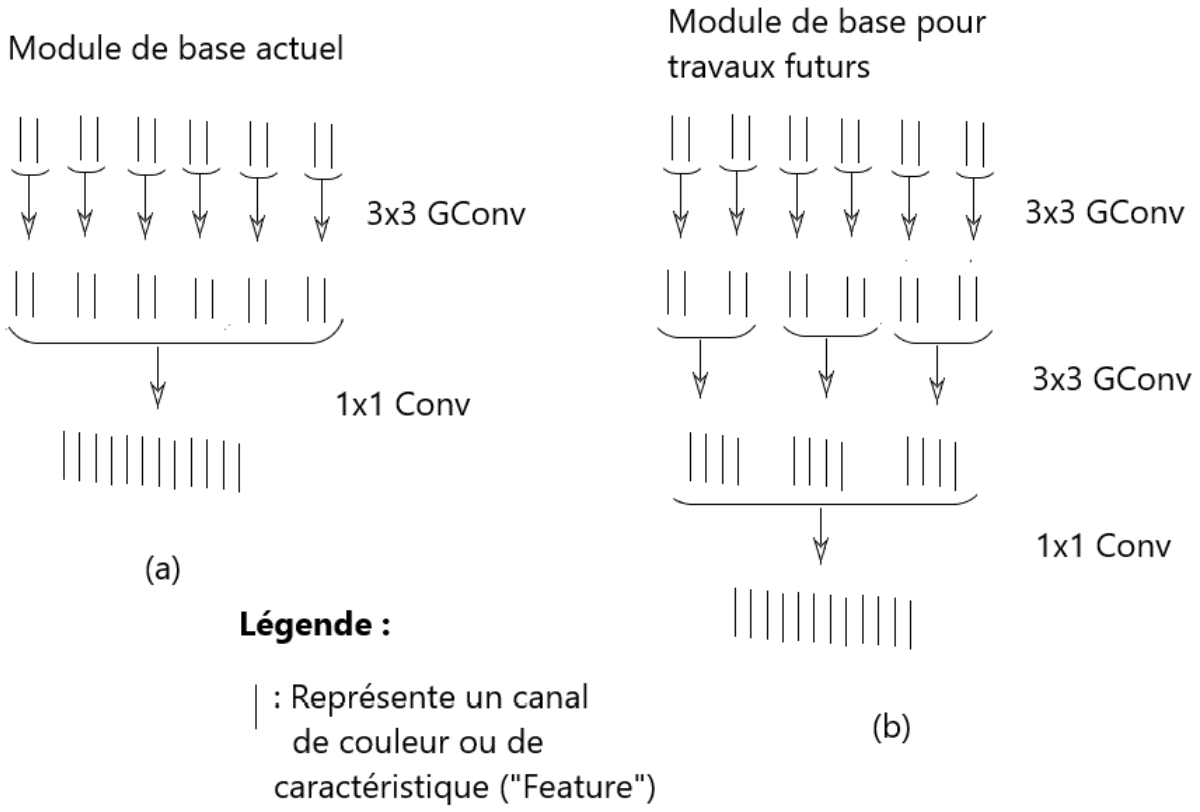


FIGURE 6.1 : Le module CoSOV1 (a) pourrait être augmenté d'un niveau comme le montre (b) pour les travaux futurs.

BIBLIOGRAPHIE

[1] Yun Liu, Xin-Yu Zhang, Jia-Wang Bian, Le Zhang, and Ming-Ming Cheng. Samnet : Stereoscopically attentive multi-scale network for lightweight salient object detection. *IEEE Transactions on Image Processing*, 30 :3804–3814, 2021.

[2] Yun Liu, Yu-Chao Gu, Xin-Yu Zhang, Weiwei Wang, and Ming-Ming Cheng. Lightweight salient object detection via hierarchical visual perception learning. *IEEE Transactions on Cybernetics*, 51(9) :4439–4449, 2020.

[3] Robert Shapley. Physiology of color vision in primates. In *Oxford Research Encyclopedia of Neuroscience*. 2019.

[4] Robert Shapley and Michael J Hawken. Color in the cortex : single-and double-opponent cells. *Vision research*, 51(7) :701–717, 2011.

[5] Imanne El Maachi, Guillaume-Alexandre Bilodeau, and Wassim Bouachir. Deep 1d-convnet for accurate parkinson disease detection and severity prediction from gait. *Expert Systems with Applications*, 143 :113075, 2020.

[6] David H Hubel and Torsten N Wiesel. Laminar and columnar distribution of geniculo-cortical fibers in the macaque monkey. *Journal of Comparative Neurology*, 146(4) :421–450, 1972.