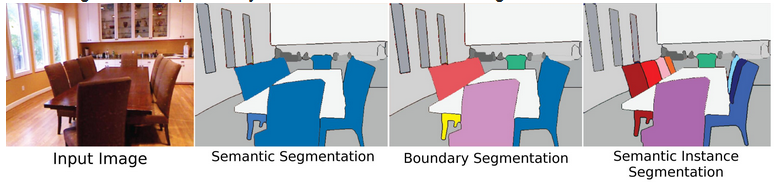
Notes sur A survey of semantic segmentation

# I. Introduction

Segmentation sémantique : regrouper des parties de l’image qui appartiennent à la même classe d’objets (ex : détection & interprétation signalétique routière)

Segmentation non-sémantique : regrouper des pixels basés sur les caractéristiques des objets

En gardant à l’esprit que la détection d’objets est différente de la segmentation (voir ci-dessous)



# II. Taxonomie des algorithmes de segmentation

Différents types d’algos peuvent être triés en fonction de:

* Types de données traitées,
* Types de segmentations (classes fixées, affiliation classe unique, colorimétrie de l’image, automatisme de la méthode)

## II.A. Classes autorisées

Classif peut être binaire (ex humain / pas humain)

Certains algos non-supervisés ne distinguent pas de classes du tout (utilisation d’une classe vide pour tout ce qui appartient à des classes non-définies) → réduit le temps d’entraînement et d’annotation

## II. B. Affiliation de classes de pixels

Segmentation en affiliation de multiples classes

Les études récentes utilisent des modèles en couches

## II.C. Données initiales

* Niveaux de gris VS colorée
* Profondeur incluse ou excluse
* Image simple VS image stéréographique VS image de co-segmentation (intéressant de voir si on peut reconstrurie le bus entier à partir des 3 caméras (mise en correspondance de points communs dans plusieurs images))

## II.D. Etat d’opération

Actif VS passif (intervention nécessaire d’un humain → give up passive form)

# III. Evaluation & données

## III.A. Mesures de qualité

La + commune : mesure de performance

Accuracy (exactitude)

Vitesse (la plupart des papiers ne mentionnent ni la vitesse ni le HW)

Latence & cadence (lequel est le plus important pour notre UC ??)

Stabilité : dépend de la qualité de l’image

Usage mémoire : bride le HW

## III.B. Sets de données

Y en a beaucoup de publics



# IV. Pipeline de segmentation

Les images d’entraînement sont de taille fixe

On peut gagner en vitesse en 1 + grand pas ou en interpolant

Les réseaux de neurones utilisent un réseau entraîné comme convolution & appliquent ladite convolution à l’ensemble de l’image

# V. Approches traditionnelles

Bcp d’algos reposent sur le deep learning

Les méthodes peuvent être combinées

Les auto-encodeurs peuvent apprendre des caractéristiques qui sont ensuite utilisées par un classificateur

## V.A. Caractéristiques & méthodes de pré-traitement

Peut changer d’espace colorimétrique

2 célèbres : RGB (le standard, supporté par tous les langages)

HSI (sympa car invariant à la luminosité)

On peut aussi égaliser l’histogramme pour améliorer le contraste

### V.A.1. Histogramme des gradients orientés (HOG)

Association position/couleur

Calcul des dérivées partielles selon x et y

Les dp donnent des cartes de caractéristiques (divisées en patchs & calcul de leur histogramme)

### V.A.2. Scale Invariant Feature Transform

→ divisé en blocs 4\*4 pixels : donnent histogrammes sur 8 directions

Au final : 128 vecteurs de caractéristiques dimensionnelles

### V.A.3. Bag-Of-Visual words (BOV)

Lié à la notion de vecteurs de quantification

Compte le nombre de patterns dans un patch de l’image

Peut être utilisé en combinaison avec le SIFT

### V.A.4. Texton

Un texton est un bloc minimal constituant la vision

Méthodes de détection des contours

V.A.5. Réduction dimensionnelle

Idée : diminuer la résolution de l’image (ou faire une principal component analysis)

But : trouver un hyperplan dans lequel tous les vecteurs de caractéristiques peuvent être projetés en perdant le moins d’information possible

Le problème de cette méthode : ne différencie pas les classes

## V.B. Segmentation non supervisée

→ Affiner une segmentation

→ Ajouter de l’information

/ ! \ Ce ne sont pas des méthodes sémantiques !

Les algorithmes non-supervisés cherchent des régions constantes ou bien des limites de région

### V.B.1. Algorithmes de clustering

#### k-mean algorithme

En entrée on donne le nombre de clusters trouvés

On place le k-centroïde de manière aléatoire

On associe chaque pixel de donnée au centroïde le plus proche

On bouge le centroïde vers le centre du cluster

Ce processus est poursuivi jusqu’à atteinte du critère d’arrêt

#### Mean-shift

Les centroïdes sont placés aléatoirement

On déplace itérativement jusqu’à atteindre la coordonnée moyenne

→ Le centre des clusters se trouve aux endroits avec la densité locale de points la plus forte

### V.B.2. Segmentation d’image basée sur les graphes

Les pixels et dissimilarités entre pixels adjacents forment un graphe

Enlever les branches de l’arbre se fait grâce à un spann ou en enlevant les branches de poids supérieur à un certain seuil

Les composants connectés sont donc les segments.

Pb de cette méthode : lourd en mémoire (env. 10GB)

### V.B.3. Chemins aléatoires

On place une seed sur les différents objets présents dans l’image

Calcul de tous les chemins aléatoires possibles pour rejoindre la seed (basé sur les gradients, comme pour le HOG)

Classe du pixel : celle pour laquelle la proba de rejoindre la seed est la plus importante

### V.B.4. Contours actifs

Segmentation grossière le long des contours qui est ensuite raffinée (minimiser la fonction d’énergie)

### V.B.5. Segmentation par watershed

Input : image en niveaux de gris interprétée comme une carte de hauteur (profondeur ?)

Points bas : bassins de rétention

Valeurs + hautes entre les bassins : watersheds (sources/fontaines)

Bassins de rétention : généralement ce qu’on veut segmenter dans l’image (deswegen ça doit être sombre/noir dans l’image)

On commence à remplir les bassins à partir des minimums locaux

Quand 2 bassins se connectent, la ligne de watershed est atteinte

On arrête quand le maximum est atteint

Défauts du watershed :

→ over-segmentation

→ watersheds épaisses

## V.C. Random Decision Forests

On entraîne les classificateurs et on combine certaines hypotheses (*ensemble learning*)

Les classificateurs st ici des arbres de décision (chq nœud interne utilise 1 ou plusieurs caractéristiques pour savoir dans quelle branche descendre : une feuille égal une classe)

Avantage du RDF : l’échelle de mesure des caractéristiques peut être arbitraire (c’est + rapide que SVMs pour l’entraînement & la classification)

Il existe **beaucoup** d’implémentations

On arrête l’algo quand :

→ Il n’est plus possible d’ajouter des nœuds dans l’arbre, ou

→ Chaque feuille contient tous les nœuds d’une seule classe

C’est censé être rapide mais ça prend 415ms pour une image 3xx\*3xx px avec un i7 à 2,7GHz !! (avec 67% d’exactitude en best case)

## V.D. SVM, Markov Random Fields, Champs Aléatoires Conditionnels

C’est expliqué de manière vachement compliquée dans l’article donc on va aller chercher ailleurs

## V.E. Méthodes en post-processing

Le but est d’enlever les erreurs évidentes

Enlever du bruit → maths morphos

Opening : enlever les petits segments

Closing : enlever les trous dans les régions supposément uniformes

Les contours actifs ça marche aussi

# VI. Les réseaux de neurones pour la segmentation sémantique

Chaque neurone artificiel a plrs entrées pondérées qu’il somme.

Il applique ensuite une fonction d’activation & donne une sortie

Entrée du neurone : vecteur de caractéristiques ou la sortie d’autres neurones

Une fonction d’erreur est nécessaire pour la descente de gradient pour l’apprentissage (typiquement MSE ou cross-entropy)

### CNNs

Apprennent des filtres d’image

Réduisent le nombre de params nécessaires mais restent pour autant suffisamment généraux

Fonction d’activation :

(bien + rapide que la fonction sigmoïde)

# VII. Problèmes possibles sur la donnée pour les algorithmes de segmentation

## VII.A. Diffusion parasite de la lumière (lens flare)

Artéfacts type réflexion interne ou dispersion (dû à des poussières ou rayures sur la lentille)

## VII.B. Vignetting

Quand l’image devient sombre dans les coins (ça peut être dû à plein de trucs, notamment à des filtres bloquant)

## VII.C. Images floues

Défaut d’accomodation, mouvement trop rapide, ou de la mousse ou fumée

## VII.D. Autres problèmes

### VII.D.1. Occlusions partielles

### VII.D.2. Camouflage

Genre les animaux ou autre

Difficile déjà pour les humains de détecter

Ou alors le système a raison et détecte bien mais pas l’humain et du coup il apprend des trucs faux

### VII.D.3. Occlusion semi transparente

Genre derrière une vitre

Comment on identifie le pixel : appartenant à du verre ou à l’objet qui se trouve derrière ?

VII.D.4. Points de vue

Changement de pt de vue : problématique si le système n’est pas entraîné