Structuring of image databases for the suggestion of products for online advertising

Lixuan Yang

le cnam

École Doctorale d'informatique, Télécommunications et Electronique

Centre d'Études et de Recherche en Informatique et Communications

THÈSE DE DOCTORAT

présentée par : **Lixuan YANG**

soutenue le : **10 Juillet 2017**

pour obtenir le grade de : **Docteur du Conservatoire National des Arts et Métiers**

Discipline / Spécialité : **Informatique**

Structuring of image databases for the suggestion of products for online advertising

THÈSE DIRIGÉE PAR

M. CRUCIANU Michel — *Professeur des Universités, CNAM*

RAPPORTEURS

M. GOSSELIN Philippe-Henri. — *Professeur des Universités, ENSEA*
M. CHEN Liming — *Professeur des Universités, Ecole centrale de Lyon*

EXAMINATEURS

M. LEFEVRE Sébastien - Président du jury — *Professeur des Universités, IRISA*
M. FERECATU Marin - Encadrant — *Maitre de Conference, CNAM*
Mme. VINCENT Nicole — *Professeur des Universités, Paris Descartes*
Mme. RODRIGUEZ Héléna — *Directrice Technique, Shopedia*

To my family

Acknowledgments

I would like to express my gratitude to my supervisors, Assoc. Pr. Marin Ferecatu and Pr. Michel Crucianu, for their expertise and patient guidance during my thesis. Thanks to their immense experience in the field of the image processing and machine learning, they have provided considerable help to advance my research. They always gave me the freedom to choose which path to take. I was priviledged to recieve their insights which expanded my ideas. Every final result described in this thesis is proof of their excellent guidance. Futhurmore, I appreciete the advice they gave me, which broadened my overall vision. In addition, I would like to thank my labmates in team Vertigo for great conversation over coffee. I've learned a lot from their vast knowledge. Since my level in English and French was poor, I would like to acknowledge my professors for helping me with developmental editing.

I would like to acknowledge Shopedia SAS for funding the project and making this thesis happen. I want to thank Héléna Rodriguez who was the technical director, and she played an important role for the communication exchange between the CNAM and Shopedia. I also want to thank her for providing the valuable advices and help for advancing the research. I also want to thank the young brillant engineers, Anis Krayn, Ines Chebil, Kevin TAN, Zo Razafiarison, for providing a great amount of help for creating an enjoyable work enviroment. They provided an upbeat atmosphere, which was conducive to ideas being shared more freely. I cherish the memories of playing ping-pong and table football with them. A special thanks to Zo Razafiarison for helping to construct the image database. I want to thank the office manager Martina Cocco and the CEO Francesco Maio for their administrative help.

A special thanks to my cat Milou for giving me lots of trouble and wonderful company. I want to thank my friends for standing by me and cheering me up. I want to thank my parents, Kangshe YANG and Jingfen MA. Depite the huge distance between China and France, they always cared for me and gave me constant love and support to pursue my studies. And finally I would like to thank my girlfriend, Gu Chencheng, her support gave me tremendous energy. She made life very colorful with her guitar and beautiful paintings.

Résumé

Le sujet de cette thèse est l'extraction et la segmentation des vêtements dans les images fixes en utilisant des techniques de vision par ordinateur et apprentissage statistique, pour la recommandation de manière non intrusive aux utilisateurs des produits similaires provenant d'une base de données de produits. Nous proposons tout d'abord un extracteur d'objets dédié à la segmentation des vêtements qui combine des informations spécifiques locales avec un apprentissage préalable. Un détecteur de personnes localise les sites de l'image ou se trouve l'objet. Ensuite, un processus d'apprentissage intra-image en deux étapes est développé pour séparer les pixels de l'objet du fond. L'objet est finalement segmenté en utilisant un algorithme de contour actif qui prend en compte la segmentation grossière précédente et qui injecte des connaissances spécifiques sur la courbure locale dans la fonction énergie. Dans une deuxième étape, nous proposons ensuite un framework pour l'extraction des vêtements qui utilise une procédure d'ajustement globale et locale à trois étapes. Un ensemble de modèles initialise un processus d'extraction de l'objet par un alignement global du modèle, suivi d'une recherche locale en minimisant une mesure de l'inadéquation par rapport aux contours locaux dans le voisinage. Les résultats fournis par chaque modèle sont agrégés, mesuré par un critère d'ajustement global, pour choisir la segmentation finale. Dans notre dernier travail, nous étendons la sortie d'un réseau de neurones FCN (Fully Convolutional Network) pour l'inférence du contexte à partir d'unités locales de contenu (superpixels). Pour ce faire, nous optimisons une fonction d'énergie, qui combine la structure à grande échelle de l'image avec la structure local des superpixels. De plus, nous proposons une nouvelle base de données, appelée RichPicture, constituée de 1000 images annotées manuellement pour l'extraction de vêtements à partir des images de mode. Nos propositions

sont validées sur plusieurs bases de données et se comparent favorablement à plusieurs méthodes état de l'art en ce moment.

Mots clés : Segmentation des vêtements, Contour Actif, Réseau de neurones, Apprentissage profond

Resume

The topic of the thesis is the extraction and segmentation of clothing items from still images using techniques from computer vision, machine learning and image description, in view of suggesting non intrusively to the users similar items from a database of retail products. We firstly propose a dedicated object extractor for dress segmentation by combining local information with a prior learning. A person detector is applied to localize sites in the image that are likely to contain the object. Then, an intra-image two-stage learning process is developed to roughly separate foreground pixels from the background. Finally, the object is finely segmented by employing an active contour algorithm that takes into account the previous segmentation and injects specific knowledge about local curvature in the energy function.

We then propose a new framework for extracting general deformable clothing items by using a three stage global-local fitting procedure. A set of template initiates an object extraction process by a global alignment of the model, followed by a local search minimizing a measure of the misfit with respect to the potential boundaries in the neighborhood. The results provided by each template are aggregated, with a global fitting criterion, to obtain the final segmentation.

In our latest work, we extend the output of a Fully Convolution Neural Network to infer context from local units(superpixels). To achieve this we optimize an energy function, that combines the large scale structure of the image with the local low-level visual descriptions of superpixels, over the space of all possible pixel labellings. In addition, we introduce a novel dataset called RichPicture, consisting of 1000 images for clothing extraction from fashion images.

The methods are validated on the public database and compares favorably to the other methods according to all the performance measures considered.

Key words : Clothing Segmentation, Active Contour, Fully convolution network, Deep learning

Chapter 1

Annexe

RÉSUMÉ

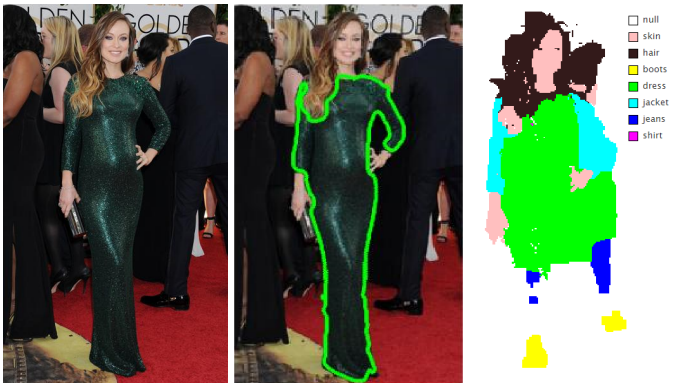
Introduction

Durant plusieurs décennies, les professionnels ont eu une forte demande pour le développement de moteurs de recherche dédiés aux bases de données pour la mode. Parce que le potentiel commercial dans l'industrie de la mode est immense. Le domaine n'a commencé à se développer qu'avec la récente prolifération massive de magasins de mode et de magasins de détail en ligne. Actuellement de plus en plus de professionnels du commerce électronique et de fournisseurs de publicité en ligne disposent de grandes bases de données d'images pour montrer leurs produits avec leurs descriptions. Les données d'images continuent à croître chaque année grâce aux utilisateurs, qu'ils soient professionnels ou non, ainsi les systèmes de recommandation devraient extraire et analyser plus rapidement et précisément ces données à grande échelle.

De nos jours, de plus en plus d'utilisateurs s'attendent de la publicité en ligne qu'elle propose des produits qui correspondent réellement à leurs attentes en termes de conception, de fabrication et d'adéquation. La localisation, l'extraction et le suivi des objets de mode lors de la navigation sur le Web permettent aux professionnels de mieux comprendre les préférences des utilisateurs et les interfaces Web proposent ainsi une meilleure expérience d'achat. Il en résulte que le système de suggestion peut aider les utilisateurs à trouver le produit désiré instantanément et, et ainsi promouvoir les ventes en ligne. Par ailleurs, ce type de système peut également être déployé pour les applications mobiles

qui pourraient également récupérer les données des images prise par les téléphones et mapper les résultats retrouvées dans le magasin physique et en ligne. Cette recherche rapide peut satisfaire les attentes des clients et augmenter le chiffre d'affaire : les acteurs ont tous à y gagner.

La collecte et l'analyse de l'historique de navigation et d'achat de l'utilisateur aideront à comprendre le comportement du client pour l'industrie de la mode. D'autre part, les données recueillies auprès des professionnels de la mode peuvent être analysées pour découvrir les tendances dans les collections de chaque année. Par exemple, les attentes en matière de couleur ou de coupe sont typiquement des objets qui intéresseront les concepteurs et les fabricants de vêtements. Un système d'extraction et de marquage peut produire cette analyse automatiquement et donc aider les professionnels de la mode.



(a) (b) (c)

Figure 1.1: Nous envisageons de produire une segmentation précise des articles de mode, comme illustrée dans la figure (b). L'état de l'art [79] produit le résultat dans la figure (c), ce qui est insuffisant pour fournir une description précise de l'objet (robe dans le cas présenté).

Cette thèse est financée par un contrat CIFRE [1] entre Check Lab SAS et ANRT [2] (le Cedric lab. [3] au CNAM [4]). Dans ce contexte, le sujet de recherche de cette thèse est directement lié au secteur d'activité de Check Lab S.A.S. Check Lab est une startup qui fédère les offres de plusieurs magasins de mode en ligne et se spécialise dans la détec-

[1] http://www.anrt.asso.fr/fr/espace_cifre/accueil.jsp
[2] http://www.anrt.asso.fr/fr/espace_cifre/accueil.jsp
[3] http://cedric.cnam.fr/
[4] http://www.cnam.fr/

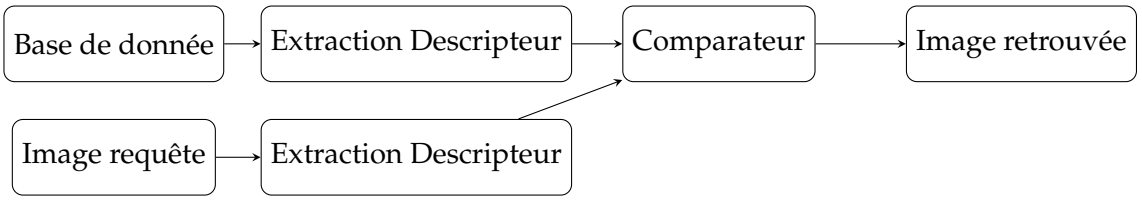


Figure 1.2: illustration du système CBIR.

tion de vêtements et de produits à partir de médias photo dans le but de présenter des suggestions commerciales proches de l'intention et du style de celles détectées dans les images.

De nos jours, la plupart des moteurs de recherche sont basés sur le texte. Les images dans les bases de données sont associées aux étiquettes des attributs. Par exemple, les méta-données contiennent des informations sur les produits et la description de contenu partiel, comme le nom de la marque, le type, la couleur principale, etc. La recherche des produits est effectuée en fournissant des mots-clés comme attributs. C'est un procédé très inefficace, car les étiquettes d'attributs ne sont pas tout à fait fiables car elles souffrent d'erreurs humaines et sont parfois incomplètes. De plus, annoter manuellement la base de donnée est un processus particulièrement long. La description textuelle et l'utilisation de mots-clés contenant beaucoup moins d'informations que le contenu visuel d'une image, la recherche par contenu visuel peut donc conduire à des résultats plus pertinents.

Un système typique de recommandation d'image basé sur le contenu (CBIR), comme illustré dans la figure 2.2 comporte deux modules: un module d'extraction et un module d'adaptation. Le module d'extraction extrait les descripteurs pour décrire le contenu de l'image. Après l'étape d'extraction, l'image de requête et les images dans la base de données sont indexées les descripteurs. L'indexation de la base de données d'images est coûteuse et se déroule habituellement hors ligne. L'étape de comparateur calcule la similarité entre les descripteurs en utilisant la mesure de la distance du vecteur comme L_1, L_2. Pour réaliser une recherche personnalisée, une procédure d'apprentissage peut être ajouté à cette étape afin d'apprendre un ensemble de poids de pondération qui minimise la distance aux résultats préférés des utilisateurs et maximise la distance par rapport aux produits indésirables.

Le travail dans cette thèse se concentre sur un système de type CBIR (Recherche d'images par le contenu) conçu spécifiquement pour les articles de mode, et modifié pour correspondre au scénario de la publicité en ligne. L'utilisateur consulte une page Web contenant une image de personne portant plusieurs articles de mode / vêtements (par exemple: t-shirt, chaussures, sac, etc.). Le système doit ensuite détecter la présence de ces éléments dans l'image de la requête, et les extraire, puis rechercher des produits dans une base de données de produits pour trouver des éléments similaires (ou identiques) à ceux présents dans l'image de requête. La partie *extraction* est une étape clé pour que le système puisse proposer des résultats pertinents en terme d'attentes de haut niveau. Comme nous envisageons de traiter l'image dans des conditions incontrôlées, une extraction du vêtement est nécessaire pour obtenir des objets proprement segmentés tout en évitant de mélanger avec l'arrière-plan. Ensuite, les descripteurs peuvent être extraits dans la région de l'objet et chercher des produites grâce à des techniques de recommandation existantes. La segmentation des vêtements est donc le module le plus important du système, mais la méthode existante n'a pas réussi à produire des résultats satisfaisants en raison de la mauvaise précision de segmentation.

Dans notre cas d'usage, les difficultés sont classées de la façon suivantes: grande diversité au sein de la même catégorie, la variété de pose humaine, occlusion, arrière plan sans contrôle, déformation de vêtement, base de donnés limitée.

Face aux difficultés présentées ci-dessus, les méthodes de l'état de l'art ne parviennent pas à produire un résultat satisfaisant. Par exemple, nous montrons dans la figure Fig. 4.1, les résultats obtenus sur l'image de gauche en utilisant la méthode de l'état de l'art [79]. Même si cette méthode est capable de fournir des segmentations multi-étiquettes, le résultat de la segmentation est inadéquat pour une description fine de la robe. Au lieu de cela, nous voulons obtenir une description plus fine comme celle de la Fig. 4.1 (b) (qui est aussi produite par la méthode décrite dans cette thèse).

Le travail de cette thèse se concentre sur la résolution du problème de segmentation des vêtements afin d'accroître la précision de l'ensemble de la chaine de traitement. Étant donné que les approches génériques de segmentation d'image ne fonctionnent pas bien pour l'extraction d'objets, la méthode que nous concevons devrait être adaptée à la

spécificité de chaque objet. Ainsi, nous injectons des connaissances spécifiques dans les modèles d'objet afin de mieux les adapter aux conditions d'image réelles. Puisque notre objectif est d'extraire un ensemble d'objets différents, la procédure conçue devrait être facile à adapter aux nouveaux objets avec peu d'intervention humaine. En outre, des algorithmes spéciaux devraient être conçus pour les petits objets. Par exemple, la diversité de formes que les accessoires peuvent prendre rend ces objets difficiles à extraire. Enfin, les algorithmes devraient éventuellement être optimisés pour des applications en temps réel (l'utilisateur ne consultera pas une page Web pendant longtemps).

Nous avons d'abord construit une base de donnée spécialement pour la segmentation de vêtement.

Rich Picture La base de données RichPicture contient 1000 images avec des annotations de segmentation au niveau des pixels. Les images sont collectées en utilisant la requête comme "mode / street / girls" dans google [5] et bing[6]. Pour notre recherche, nous sélectionnons des images de bonne qualité et seulement les mannequins en vue de face. Nous éliminons ensuite les doublons ayant la même taille d'image. Le script, qui a des difficultés pour supprimer des doublons ayant été redimensionnés et de compressés, est ajouté à la fin pour supprimer les derniers éléments grâce à l'intervention humaine.

Après avoir communiqué avec la responsable commerciale, nous avons finalement décidé de travailler sur dix objets: Bottes, Jeans, Chemises, T-shirts, Manteaux, Gilets, Pulls, Robes courtes, Robes mi-longues, Robes longues. Les éléments choisis ont un potentiel énorme de vente. Étant donné que les chaussures sont généralement trop petites pour proposer des résultats pertinents, nous avons décidé de travailler sur des bottes qui sont suffisamment larges pour obtenir une bonne description. Les robes sont un élément de mode particulièrement important, qui nécessite plus d'investissement. Pour obtenir une meilleure performance, nous incluons ensuite plus de types de robes dans l'ensemble des données: robes longues, robes mi-longues et robes courtes.

L'état de l'art Dans la chapitre 3, nous présentons l'état de l'art sur la détection d'objet, la segmentation d'image et les applications en mode. Pour la détection d'objet, nous

[5]http://www.google.fr
[6]http://www.bing.fr

présentons les méthodes traditionnelles: la classification par histogramme d'orientation de structure (Sift features [56], HOG features [13]) dans la fenêtre locale, suivie du modèle déformable [22]. La modèle inclut la déformation spatiale de la configuration des parties dans la formulation. Nous présentons ensuite les réseaux neuronaux convolutionnels (CNN) [40] qui ont récemment avancé significativement les résultats de détection. Les travaux de la localisation des objets sont présenté dans une série d'articles connexes [26; 25; 62].

Pour présenter le travail de la segmentation d'image, nous commençons par la méthode traditionnelle qui repose sur les caractéristiques locales et introduit dans un contexte plus large, par exemple Graph Cut [6] et Active Contour [37**?**]. Ensuite, nous présentons la méthode de segmentation sémantique, qui vise à attribuer à chaque pixel plusieurs étiquettes (appelée multi-étiquetage). Nous concluons cette partie avec une présentation de l'approche de segmentation d'instance qui formule l'extraction d'objet comme un problème collectif mais dans le but de produire une segmentation d'instance. Dans la troisième partie de ce chapitre, nous présentons des recherches sur le système de recommandation pour la mode, la sélection des attributs et la segmentation des vêtements. La recommandation a deux aspects: la recommandation de style qui récupère un produit ressemble au style et une recherche de street2shop qui recherche le produit à partir d'une image de rue. La sélection des attributs est une autre approche de récupération pour marquer automatiquement l'image avec les méta-données. Enfin, nous présentons la segmentation des vêtements qui est également une étape importante de notre travail dans cette thèse.

Dans le contexte présenté, nous abordons le problème avec trois méthodes, avec une complexité croissante:

Contour actif piloté par la classification pour la segmentation de la robe

Puisque nous cherchons à trouver le plus précisément possible le contour de la robe, une approche directe est probablement caduque en raison des difficultés précédemment mentionnées. Dans le chapitre 4, nous présentons notre premier travail en utilisant une processus d'extraction directe. Puisque la robe est portée par les humains, un détecteur humain peut être appliqué au premier pour localiser la zone ROI. En considérant que

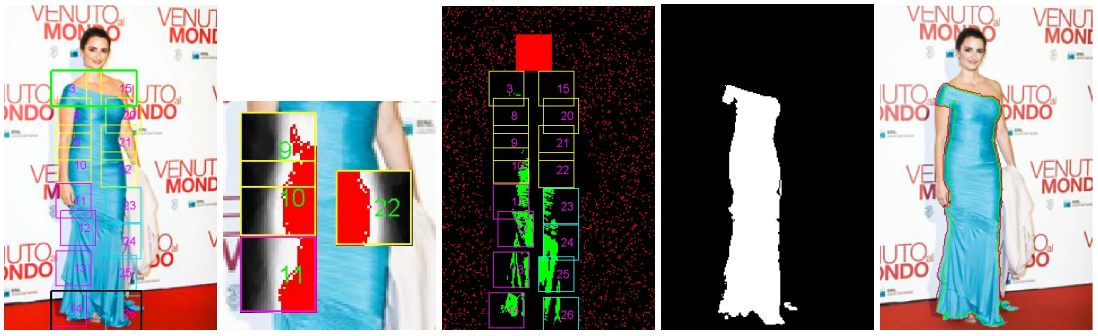
chaque boîte possède sa propriété spéciale, nous voulons explorer plusieurs informations antérieures obtenues pour soutenir l'extraction: 1. le contexte de chaque boîte: le contenu des vêtements dans les boîtes voisines sont généralement similaires ; 2. la carte de probabilité : capturer la distribution spatiale des pixels dans la boite-par exemple pour la boîte assignée comme "épaule gauche": la partie inférieure à droite est plus susceptible d'avoir des pixels de vêtements, à l'inverse de l'autre coin, l'autre coin est moins probable; 3. la courbure de la boîte : selon la nature de la partie de l'anatomie humaine, la courbure du contour est variable dans chaque boite, par exemple la courbure près des jambes est généralement petite, tandis que le bas de la robe est relativement plus grand lorsque les extrémités sont étalées sur le sol. L'idée principale de notre méthode consiste à fédérer les informations mentionnées pour extraction et finalement segmentation.

Pour atteindre notre objectif, nous combinons un détecteur de personne avec une classification SVM en deux étapes pour obtenir une estimation approximative du contour du vêtement (séparation de l'objet de l'arrière-plan). Nous adoptons une méthode en trois étapes, chaque étape préparant la suivante:

1.**Detecteur de personne.** Le détecteur de personne a été introduit en 3.2.2. Nous formons d'abord un détecteur de personne sur une base de données annotée manuellement pour trouver les régions de l'image les plus susceptibles de contenir le contour de l'objet. Nous utilisons le modèle de détection humain articulé avec un mélange flexible de parties présentées dans [83], qui fonctionne bien pour la détection de personne et l'estimation de pose, et qui a été testé avec succès dans plusieurs autres travaux liés à la mode (voir Sec.3). La sortie du détecteur de personne illustré dans Fig.1.3 est un ensemble de boîtes rectangulaires centrées sur les articulations du corps et orientées correctement.

2.**détection grossière de premier plan.** Nous employons les données d'apprentissage pour proposer une carte de probabilités sur une base qui estime que chaque pixel à l'intérieur d'une boîte appartient à l'objet. La carte est utilisée pour initialiser une SVM d'une classe estimant le support de la distribution d'exemples positifs (pixels qui appartiennent à l'objet). ensuite, un SVM de deux classes est formée pour améliorer la détection (grossière) des pixels appartenant à l'objet, en prenant comme négatif des exemples de pixels de fond aléatoires.

3.**Contour actif.** Le résultat du SVM à deux classes est utilisé comme entrée pour une procédure de contour actif en deux étapes qui produit la segmentation finale. Nous incluons plusieurs termes spécifiques dans la fonction énergétique qui guide le contour actif: le premier terme utilise les résultats de la phase d'apprentissage pour pousser le contour vers la frontière de séparation SVM (c'est-à-dire le contour de l'objet selon le SVM), le second terme prend en compte la courbure locale pondérée par l'emplacement sur l'objet. Cela garantit un bon équilibre entre le comportement local des pixels et les informations injectées par l'apprentissage, produisant de bons résultats dans la plupart des situations.



(a) (b) (c) (d) (e)

Figure 1.3: Différentes étapes de notre approche: (a) la sortie du détecteur de personne chevauche sur l'image avec la zone de délimitation numérotée, (b) sortie de la classe SVM, (c) et (d) entrée et sortie des deux Classe SVM, rouge pour échantillons négatifs et vert pour échantillons positifs; (E) résultat final après les premières étapes de contour actif (vert) et deuxième (rouge).

Chaque composant a également été testé séparément et s'est montré efficace. En comparant avec GrabCut, la méthode montre une résultat prometteur. La méthode peut facilement être étendue à de nouvelles classes d'objets en segmentant manuellement des objets de ces classes.

Une approche globale-locale de l'extraction des objets de mode Dans le chapitre 5, nous étendons notre segmentation précédente à une sélection d'objets plus large. La méthode précédente nécessite l'utilisation d'informations à priori pour définir les paramètres. Même si les paramètres peuvent être trouvés par la validation croisée, la meilleure configuration ne peut pas toujours être trouvée et ce malgré un investissement lourd en terme

de temps. Dans ce chapitre, nous proposons une méthode pour analyser automatiquement les informations préalables spécifiques aux vêtements recueillies par les templates de l'image et la projection de l'information, pour aider à segmenter de nouvelles images.

Pour extraire des objets dans les conditions difficiles et sans intervention de l'utilisateur, il est peu probable que les méthodes utilisant uniquement l'optimisation d'un critère local (ou une classification des pixels basée sur les caractéristiques locales) soient bonnes. Une certaine connaissance de la forme globale de la classe d'objets à extraire est nécessaire pour aider une analyse locale à converger vers une limite d'objet correcte. Nous utilisons cette intuition pour développer un cadre qui tient compte de la dualité locale / globale pour sélectionner la segmentation d'objet la plus probable. Nous proposons une approche globale-locale, fondée sur l'idée qu'une recherche locale est susceptible de converger avec une meilleure adéquation si l'état initial est harmonieux avec l'aspect global attendu de l'objet.

D'abord, nous préparons un ensemble d'images Fig. 1.4 (a) contenant l'objet d'intérêt et nous les segmentons manuellement Fig. 1.4 (b). Ces masques d'objets initiaux (appelés template dans la suite) fournissent les connaissances antérieures utilisées par l'algorithme. Une segmentation manuelle ne correspond pas exactement à l'objet dans une image inconnue. Afin de s'adapter à une déformation diverse, nous avons prototypé les modèles en huit groupes, puis sélectionné les modèles les plus similaires dans chaque cluster pour la template et guider ainsi la segmentation. Étant donné que les vêtements ont une grande dépendance spatiale, parmi les informations antérieures obtenues, nous avons choisi cette information pour projeter la probabilité de l'occupation des vêtements, ce qui pourrait être facilement déduit par les coordonnées des articulations humaines. Nous utilisons chaque segmentation (après un alignement approprié Fig. 1.4 (d)) en tant que modèle pour lancer une procédure de contour actif (AC) (voir Fig. 1.4 (e)) qui converge plus près des limites réelles de l'objet réel dans l'image actuelle. Nous extrayons ensuite l'objet avec une procédure GrabCut appropriée pour fournir la segmentation finale(voir Fig. 1.4 (g)). Ainsi, au final, nous avons autant de segmentations candidates correspondant à chaque template utilisé. Étant donné qu'une bonne segmentation s'accroche bien au bord, la qualité de la segmentation est évaluée par l'accord de contour. Dans la

dernière étape, nous choisissons le meilleur selon un critère qui optimise la cohérence de la segmentation proposée avec les bords extraits de l'image.

(a) (b) (c) (d) (e) (f) (g)

Figure 1.4: Différentes étapes de notre approche: (a) image originale, (b) un modèle se chevauche avec la segmentation, (c) sortie du détecteur de personne, (d) résultat après l'étape d'alignement, (e) résultat après l'étape de contour actif, (f) la bande GrabCut, (g) résultat après l'étape GrabCut.

Les comparaisons avec One Cut (une version évoluée de GrabCut)et avec Paper Doll montrent que l'approche proposée est prometteuse et s'effectue favorablement par rapport aux extracteurs d'objets génériques ou plus dédiés. La méthode peut facilement être étendue à de nouvelles classes d'objets à un coût relativement faible, *i.e.* en segmentant manuellement des objets de ces classes.

Réseau Fully Convolutional avec analyse de superpixel pour la segmentation d'image Web de mode

Comme l'apprentissage profond a montré son efficacité dans une grande variété de domaines, tels que le Réseau Fully Convolutional 3.3.3 (notée FCN dans la suite), qui a déterminé l'état de l'art de segmentation sémantique. En général, le FCN produit un bon emplacement d'objet avec un résultat de segmentation brut. De même que la plupart des réseaux, le FCN souffre toujours d'une difficulté de localisation de contour. Étant donné que les informations de bas niveau ont une structure contenant les informations sur le contour, nous proposons dans le chapitre 6 un post-traitement pour tirer la prédiction à partir d'un niveau supérieur et donc corriger la localisation du contour en utilisant les informations de bas niveau.

Parmi les informations de bas niveau (comme le gradient, la texture, le superpixel, le contour), le superpixel semble inclure presque tous les points forts des informations

ci-dessus. Les superpixels s'attachent bien au bord de l'image et segmentent l'image en petites régions uniformes. Nous pouvons attendre des superpixels qu'ils améliorent la localisation du contour. Par contre, ils peuvent aussi souffrir d'une mauvaise segmentation, et donc à l'intérieur de quelques superpixels, ils pourraient y avoir un contour réel. Pour corriger certains défauts, nous pouvons utiliser le terme de finesse pixelwise pour renforcer la force de contour entre les pixels.

Pour tenir compte des informations (prédiction de haut niveau, structure de superpixels, lissage de pixel), nous étendons la sortie d'un FCN en optimisant une fonction d'objectif qui itère sur tous les étiquetage possibles au niveau des pixels. La fonction objective considère trois facteurs: la prédiction de haut niveau (étiquette), la structure à mi-échelle des unités (superpixels), ainsi que la niveau de lissage local de l'étiquetage. Les termes utilisés sont les suivants :

Terme de la convolution Ce terme contient la prédiction de haut niveau obtenue par la sortie de FCN. Le réseau est initialisé en utilisant un modèle pré- appris sur Pascal VOC [21] et ensuite affiné sur l'ensemble de données utilisées ici suite à une procédure similaire à celle décrite dans [5]. La sortie contient des prédictions de score pour chaque classe et chaque pixel. La sortie de FCN-8s est représentée dans la figure **??** (c), les articles sont grossièrement segmentés et annotés par l'étiquette correcte. Nous voulons profiter de la haute précision par l'utilisation du réseau profond. En minimisant simplement ce terme, cela a le même effet que le softmax du FCN qui prend l'étiquette du score maximal. Par conséquent, ce terme peut préserver la prédiction de FCN.

Terme de la prédiction de région Le deuxième terme dans l'équation encode le niveau d'accord entre les étiquettes voisines et l'étiquette du pixel actuel. L'image est d'abord sur-segmentée en superpixels, en suivant l'idée que tous les pixels d'un superpixel doivent être attribués à la même étiquette, puisque les superpixels regroupent les petits pixels uniformes. Ceci est raisonnable car les objets sont généralement délimités par des contours physiques, et sont ainsi obtenus en tant qu'unités disjointes de superpixels. Cette procédure devrait donc améliorer la localisation du contour, qui était l'une des faiblesses du FCN original.

Ce terme contient la prédiction de l'étiquette du superpixel oú se trouve le pixel. Les

superpixels ont des informations plus riches, comme la couleur et la texture, la prédiction par superpixel sera certainement plus précise. À cette fin, nous prédisons l'étiquette en regroupant la prédiction de tous les superpixels. Cette prédiction est pondérée par le nombre de superpixels ayant cette étiquette, cela peut éviter que l'objet plus grand ne domine pas la prédiction.

Terme de lissage Illustré dans Fig. 6.4(d), quelques superpixels ne sont pas bien segmentés. Ceci est probablement dû à une mauvaise segmentation pour laquelle le contour peut passer par les superpixels. Pour corriger cette imperfection, nous incluons le terme de lissage des pixels pour corriger la prédiction localement. Le troisième terme dans l'équation implémente une condition de lissage: deux pixels sont plus susceptibles de partager la même étiquette si elles ont des description visuelles similaires et ne sont pas très éloignées dans l'image.

(a) (b) (c) (d) (e) (f)

Figure 1.5: L'image originale (a), la vérité terrain (b) et les étapes de notre approche: (c) softmax FCN, (d) softmax avec superpixels marqués par la valeur moyenne de FCN, (e) analyse par superpixel avec la prédiction de la région , (F) résultat final en résolvant la probabilité maximale définie par la sortie FCN, la prédiction de la région et le lissage des pixels.

En combinant la prédiction de haut niveau fournie par le réseau en profondeur et la description de l'image de niveau intermédiaire, la méthode proposée améliore considérablement la localisation des contours. L'approche proposée est validée par les comparaisons avec le FCN (ajusté) seul et avec la méthode Co-parsing [51] qui est l'état actuel de l'art dans l'extraction de l'objet de mode.

Conclusion La segmentation sémantique est un composant clé dans le système de recommandation de mode pour assurer une proposition pertinente. Bien qu'il existe de

nombreuses publications sur ce sujet, il est encore problématique d'obtenir un résultat satisfaisant. C'est parce qu'il y a de nombreuses difficultés: des arrière plan complexes, des vêtements déformés et des conditions de prise de photo sans contraintes. Tout au long de nos travaux, nous envisagions non seulement de segmenter autant d'étiquettes que possible, mais aussi d'aborder le problème de segmentation précise pour conduire à une meilleure localisation des contours. Dans cette thèse, nous résolvons le problème de l'extraction du vêtement en utilisant trois méthodes. Nous avons commencé par faire une segmentation de robe pour segmenter précisément un seul objet en utilisant les informations recueillies au début du processus pour apprendre le modèle à la volée. Ensuite, nous étendons le travail pour dix étiquettes à l'aide de la segmentation basée sur les templates. Les deux méthodes sont faciles à appliquer sur les nouveaux articles de mode. Enfin, nous étendons le travail pour la segmentation sémantique qui segmentera plusieurs étiquettes dans la même image. La troisième méthode peut s'appliquer à de nouveaux articles de mode et à d'autres objets génériques. Nos méthodes se comparent favorablement à l'état de la technique.

Perspectives Ici, nous donnons plusieurs idées sur la façon d'améliorer les résultats pour l'extraction des vêtement et la nouvelle conception du système de recherche de la recommandation appliqué à la mode.

Une base de données plus grande. Une grande base de données d'images est toujours un facteur essentiel pour les méthodes d'apprentissage. En incluant plus d'images avec plus d'étiquettes, l'algorithme peut apprendre des informations plus riches. Lorsqu'une telle base de données sera disponible, il serait intéressánt de réévaluer nos algorithmes sur une base de données de mode plus grande pour valider l'algorithme dans un contexte plus large. Une autre expérience intéressante serait l'évaluation de notre travail sur d'autres bases de données ou la base de données génériques, par exemple PascalVOC.

La couche intermédiaire. Les couches intermédiaires du réseau FCN contiennent des informations riches. En extrayant des informations à partir de couches intermédiaires, plusieurs travaux ont déjà montré des résultats très prometteurs [60; 54]. Pendant la propagation vers l'avant, les couches pooling ont réduites la taille d'image, ce qui a

entraîné une carte de probabilité plus petite. Cependant, les couches intermédiaires contiennent les informations locales de l'image et ces informations ont été perdues dans les couches intermédiaires. Ces informations peuvent être récupérées pour une plus grand précision.

De plus, le réseau de segmentation peut être construit au-dessus d'autres réseaux, tels que GoogLeNet [72], ResNet [30] qui sont actuellement l'état de l'art dans la reconnaissance d'objet. Un autre effort peut être dédié à l'amélioration du réseau afin que l'architecture soit plus proche de la perception humaine et en améliorant la technique d'apprentissage pour stimuler la performance. Par exemple, les réseaux RNN et LSTM sont largement utilisés dans le domaine du traitement du langage naturel. Une idée prometteuse serait d'exploiter ces réseaux pour la segmentation en utilisant le mécanisme de la mémoire.

L'extraction des petits objets. Les petits objets ont un potentiel commercial important (Par exemple, anneaux d'oreille, sacs à main, montres, etc.), mais les méthodes existantes ont un faible taux de performance pour ces objets. Pour résoudre ce problème, une étape de post-traitement devrait être ajoutée à la fin pour segmenter spécifiquement ces objets sur les régions d'intérêt et avec une résolution plus élevée si disponible.

Système de recommandation. Un système de recommandation de mode peut largement bénéficier de notre algorithme d'extraction. Notre algorithme fournit des informations importantes sur les objets: par exemple l'étiquette de l'article, la région d'intérêt (ROI) et la région nettement segmentée. L'étiquette d'article peut d'abord réduire la zone de recherche dans une catégorie spécifique de produits. En outre, il permet la recherche spécifique intra-catégorie, c'est-à-dire que différentes catégories ont des expressions différents de la similarité (par exemple, la similarité des t-shirts repose principalement sur le contenu visuel puisqu'ils ont plûtot la même forme). En outre, le retour sur les régions bien segmentées aideront à décrire le contenu visuel en éliminant l'influence de l'arrière-plan.

De plus, lors de la recherche d'articles de mode, il existe également plusieurs critères de récupération tels que la récupération de style et les préférences personnelles. La récupération de style vise à associer le concept de style aux fonctionnalités d'image afin

de proposer des produits similaires. Cela peut se faire en transformant d'un espace des descripteur visuel à un espace décrit le style. Dans l'espace de style, le produit similaire devrait être étroitement lié au style, quelle que soit la catégorie de produit.

Et la recommandation personnelle devrait pouvoir recevoir des retours et former le systéme de recherche itérativement avec les retours de l'utilisateur [63; 76]. En général, la préférence peut être acquise en donnant l'ordre de navigation de l'utilisateur. La photo affichée précédement et la photo visualisée ensuite peuvent être organisées en paires pertinentes, et le reste en paires non liées. Un modèle peut être appris au fur et à mesure qui maximise la distance entre les paires non liées et minimise la distance pour les paires liées.

Contents

1 Annexe **10**

Contents **25**

List of Tables **27**

List of Figures **28**

2 Introduction **2**

Introduction **2**

2.1 The context of the work 4

2.2 Contributions of the thesis 8

2.3 Structure of the manuscript 10

3 State of the Art Research **14**

3.1 Machine learning 14

3.2 Object Detection and classification 22

3.3 Image Segmentation 26

3.4 Fashion application 36

3.5 Dataset 42

4 Classification-Driven Active Contour for Dress Segmentation **47**

4.1 Introduction 47
4.2 Our proposal 50
4.3 Experimental results 58
4.4 Conclusion 62

5 A Gloabal-Local approach to fashion items extraction 63

5.1 Our proposal 64
5.2 Experimental results 73
5.3 Conclusion 78

6 Fully convolutional network with superpixel parsing for fashion Web image segmentation 79

6.1 Introduction 79
6.2 Our proposal 81
6.3 Experimental results 88
6.4 Conclusion 91

7 Conclusion 94

7.1 Summary of contributions 95
7.2 Perspectives for future research 96

Bibliography 98

A List of Publications 108

List of Tables

4.1 Evaluation of different configurations by statistical measure for our method and comparison with GrabCut. 59

5.1 Segmentation selection from the results based on the 8 templates of the class, using the corresponding fit values. The test image is given top left, with the extracted edges shown bottom left. The best score is the smallest (outlined in boldface). 73

5.2 Comparison with the One Cut algorithm. The comparison measure is the Jaccard index. 76

5.3 Comparison with Paper Doll and One Cut on Fashionista. The comparison measure is the Jaccard index. 76

6.1 Class-by-class comparison with the FCN [55] using the average F_1 score on CFPD database. 89

6.2 Global comparison of Paper Doll [79], Co-parsing [51], FCN [55] and our method on CFPD database. 90

List of Figures

1.1 Nous envisageons de produire une segmentation précise des articles de mode, comme illustrée dans la figure (b). L'état de l'art [79] produit le résultat dans la figure (c), ce qui est insuffisant pour fournir une description précise de l'objet (robe dans le cas présenté). 11

1.2 illustration du système CBIR. 12

1.3 Différentes étapes de notre approche: (a) la sortie du détecteur de personne chevauche sur l'image avec la zone de délimitation numérotée, (b) sortie de la classe SVM, (c) et (d) entrée et sortie des deux Classe SVM, rouge pour échantillons négatifs et vert pour échantillons positifs; (E) résultat final après les premières étapes de contour actif (vert) et deuxième (rouge). 17

1.4 Différentes étapes de notre approche: (a) image originale, (b) un modèle se chevauche avec la segmentation, (c) sortie du détecteur de personne, (d) résultat après l'étape d'alignement, (e) résultat après l'étape de contour actif, (f) la bande GrabCut, (g) résultat après l'étape GrabCut. 19

1.5 L'image originale (a), la vérité terrain (b) et les étapes de notre approche: (c) softmax FCN, (d) softmax avec superpixels marqués par la valeur moyenne de FCN, (e) analyse par superpixel avec la prédiction de la région , (F) résultat final en résolvant la probabilité maximale définie par la sortie FCN, la prédiction de la région et le lissage des pixels. 21

2.1 An example of Shop Match based on shape descriptors. The user inputs a query images and then choose a type of similarity: color, shape or texture. 5

2.2 An illustration of CBIR system. 6

2.3 Examples from a fashion dataset. This illustrates the huge diversity of clothing items and backgrounds that are present, and also the intra-class variability due to clothing deformation and occlusions. 7

3.1 Illustration of (a) object detection (image from [62]), (b, c) image segmentation (image from [2]). 15

3.2 Illustration of SVM for 2 class 17

3.3 Illustration of LENET-5 architecture 19

3.4 Illustration of CNN filters 20

3.5 Illustration of Alexnet architecture 22

3.6 The principle of the model in [83] : (a) Approximation of the deformations by the mixture, (b) Local template, (c) Tree structure. 24

3.7 Illustration of (a) Faster RCNN, (b) RPN Network(image from [62]). . . . 26

3.8 Illustration of (a) RGB, (b) HSV. 27

3.9 Illustration of conversion from CNN to fully convolutional network 33

3.10 Illustration of DAG nets 33

3.11 Illustration of DAG nets 34

3.12 Illustration of LG-LSTM 36

3.13 Illustration of street-to-shop clothing retrieval system(Image from [50]) . 37

3.14 Illustration of attributes(Image from [16]) 37

3.15 Illustration of matchnet in [38] 38

3.16 Illustration of network in [12] 39

3.17 Illustration of PaperDoll 41

3.18 Illustration of network human parsing in [46](image from [46]) 41

3.19 Illustration of image annotation by labelme tools by marking the foreground(red) and background(blue), the result overlaps on the image. . . . 44

3.20 Annotation of key points by Matlab tools : (a) click on the key points; (b) The 14 key points. 46

4.1 We aim to produce a precise segmentation of the fashion items as in (b). Recent state of the art [79] produces the result in right figure (c), which is insufficient to provide a precise description of the object (dress in this case). 48

4.2 We present some examples to illustrate the diversity of dress items: (a) A person nearby with the same color of dress; (b) Colorful dress; (c) Highly textured dress; (d) The bottom of dress roll out the red carpet; (e) Semi-transparent dress; (f) Other confusing accessories like the scarf. 49

4.3 Different stages of our approach: (a) output of the person detector overlaps on the image with the numbered bounding box, (b) output of the one class SVM, (c) and (d) input and output of the two class SVM, red for negative samples and green for positive samples; (e) final result after first (green) and second (red) active contour steps 51

4.4 Each training image is augmented 5 times by following the transformation process: (a) original images; (b) mirrored image by flipping left right; (c, d, e, f) rotation with -15, -7.5, 7.5, 15 degrees. 53

4.5 Illustration of two modes of local mean value computation(a) For each point in the contour, compute on a fixed size window around point, (b) Approximate mean value in image grid for all the inside points. 56

4.6 Paper Doll [80] is inadequate for the precise segmentation needed by our use case: (a) original image, (b) our method obtains a satisfy result, (c) Paper Doll fails to segment the dress and miss detection. 58

4.7 More results on different types of background: (a, c, e, g, i, k) original images, (b, d, f, h, j, l) associated segmentation result. 60

4.8 Qualitative evaluation: (a, b, c) original images; (d, e, f) associated segmentation results. 61

4.9 Visual comparison with GrabCut : Our method segment perfectly the dress, while the result of GrabCut includes background and skin pixels. 62

5.1 We present some examples to illustrate the diversity of the database Rich-Picture. 65

5.2 Different stages of our approach: (a) original image, (b) a template overlap with segmentation, (c) output of the person detector, (d) result after the alignment step, (e) result after the active contour step, (f) the GrabCut band , (g) result after the GrabCut step. 66

5.3 Medoids of the 8 clusters of template overlap with the segmentations for three classes: jeans (top), long dress (middle) and coat (bottom). 69

5.4 SVM template alignment : (a,f) images and outputs of person detector, (b,g) templates and outputs of person detector, (c, h) template overlapped with the segmentation of coat, (d,i) the predicted coat occurrence, (e, j) the predicted SVM score. 70

5.5 Qualitative evaluation: (a, c, e, g, i, k, m, o, q, s, u, w) are original images, (b, d, f, h, j, l, n, p, r, t, v, x) are associated segmentation results. Segmentation for each item: (a, c) for boots, (e, g, m) for dresses, (i, k) for shirts, (o, q, s) for jeans, (u, w) for pull. 74

5.6 Comparison with OneCut: original image (left), our method (middle) and OneCut (right) 77

6.1 To illustrate the fitness of contour by the superpixels, the red contour of superpixels is overlapped on the original image. 80

6.2 Our goal is to produce a precise semantic segmentation (extraction) of the fashion items as in (b) which has been obtained by our method. Each color signifies a semantic clothing label. 81

6.3 We present some examples to illustrate the diversity of the database CFPD [51]. 82

6.4 Original image (a), ground truth (b) and the stages of our approach: (c) softmax FCN, (d) softmax with superpixels labeled by mean value of FCN, (e) superpixel parsing by region prediction, (f) final result by solving the maximum probability defined by the FCN output, region prediction and pixel smoothness. 88

6.5 Qualitative comparison with fine tuned FCN: original image (first), ground truth (second), our full method (third) and fine tuned FCN (last). 91

6.6 Qualitative evaluation: original images (a, d, g, j, m, p, s, v), ground truth (b,e, h, k, n, q, t, w) and associated segmentation results (c, f, i, l, o, r, u, x). The first three rows show the good results,while the last row for bad results. The labels are presented by colors. The white color stands for the background. And the other color's correspondence is illustrated at the bottom of figures by the color squares with the related clothing semantic labels underneath. 92

Chapter 2

Introduction

Although the interest in developing dedicated search engines for fashion databases is several decades old and the commercial potential in the fashion industry is immense, the field only started to develop with the recent massive proliferation of fashion web-stores and on-line retail shops. Indeed, more and more professional e-commerce and on-line advertising providers have large databases of images showing their products and describing their characteristics. The image data continue to grow each year by input from professional and personal users, and retrieval systems are expected to extract and analyze more quickly and precisely these large scale data.

Nowadays, more and more users expect online advertising to propose items that truly correspond to their expectations in terms of design, manufacturing and suitability. Localizing, extracting and tracking fashion items during web browsing allows the professionals to better understand the users' preferences and design web interfaces that make for a better web shopping experience. Therefore, the proposal system can help the users to find their desired product instantly and consequently promote the online sales. Furthermore the system can also be deployed for mobile applications that could also be expected to retrieve the items from the image taken by phones and map the retrieved result to the physical and online shop. This quick search can satisfy the clients' expectation and grow the turnover as a win-win situation. Collecting and analyzing the user's browsing and purchasing history will help to understand customer's behavior for the fashion industry. On the other hand, the data collected from the fashion professionals can be analyze

to discover fashion trends in each year's collections. For instance, the color trend, the cutting trend will be of interest to clothing designers and manufacturers. An extracting and tagging system can produce this analysis automatically and thus be of great help to fashion professionals.

This thesis is financed by a CIFRE contract [1] between Check Lab S.A.S and ANRT [2] (the Cedric lab.[3] at CNAM [4]). In this context, the research topic of this thesis is closely related to the Check Lab S.A.S. business sector. Check Lab is a startup company that federates the offers of several on-line fashion retail shops and specializes in the detection of clothes and products from photo media with the goal of presenting commercial suggestions that are close in intent and style to those detected in the images.

More specifically, this thesis proposes to address the learning of classes of fashion products and then investigate their detection in images such as those available on the web sites frequented by the general public. This is a first step to solving a more difficult problem inspired by the need of professionals of online advertising and fashion media: to present to the users relevant items from a database of clothes, based on the content of the web application they are consulting and its context of use. This goes far beyond the needs of a search engine: the user is not asked to interact with any search interface or formulate a query, but instead she is accompanied by automatic personal suggestions presenting in a non intrusive way a selection of products that are likely to interest her. Thus, the ultimate goal is to be able to help structuring offline database of product images such as to fast search in large scale images database and make suggestions based on user preferences and possibly on their purchase history.

For illustration, Fig. 2.1 shows an example of an existing shop match application: shape based retrieval for a product image. The query image contains a well-positioned object on a uniform background and the user is required to give the image URL address and the similarity metric in terms of color, texture or shape. Also, the usual similarity by global characteristics is not sufficient to get good result in most of the situations, and

[1] http://www.anrt.asso.fr/fr/espace_cifre/accueil.jsp
[2] http://www.anrt.asso.fr/fr/espace_cifre/accueil.jsp
[3] http://cedric.cnam.fr/
[4] http://www.cnam.fr/

a more sophisticated description is needed to achieve better retrieval results. The focus in this work is to go beyond this scenario and build an application capable of retrieving clothes worn by people in real world images without the interaction from the user, and not limited to clean product images.

The rest of this chapter is organized as follows: in the next section we present the general context of our work and the main objectives of the thesis (Sec. 2.1), followed by a summary of our contributions in Sec. 2.2. Finally in the last part we present an overview of the the structure of the manuscript in 2.3.

2.1 The context of the work

Nowadays, most search engines are based on text. The images in databases are associated with the attribute labels. For instance, the meta data contains product information and partial content description, namely the brand name, type, present color and so on. The search for desired products is performed by providing keywords as attributes. This is a very inefficient way, since the attribute labels are unreliable because they suffer from human errors it is and are probably incomplete. Futhurmore, it is time-consuming to manually annotate database. Describing by only text has less information than image visual content, thus searching by visual content can lead to more relevant results.

A typical Content Based Image Retrieval (CBIR) system illustrated in Fig. 2.2 has two important modules: a feature extraction module and a feature matching module. The feature extraction module extracts the features to describe the image content. There are two types of features : low level features and high level features. **The low level features** describe the local appearance of the image. For instance, a color descriptor may describe the color distribution in the RGB color space or in some other color space closer to human perception, a texture descriptor may describe local patterns by using Gabor filter banks, Fourier descriptors and so on. **The high level features** are built on the low level features and try to capture the semantic content of the image. A well known high level feature is Fc7 [58] extracted from the a deep Convolutional Neural Networks (CNN) just before the output layer which describes the image on a semantic level.

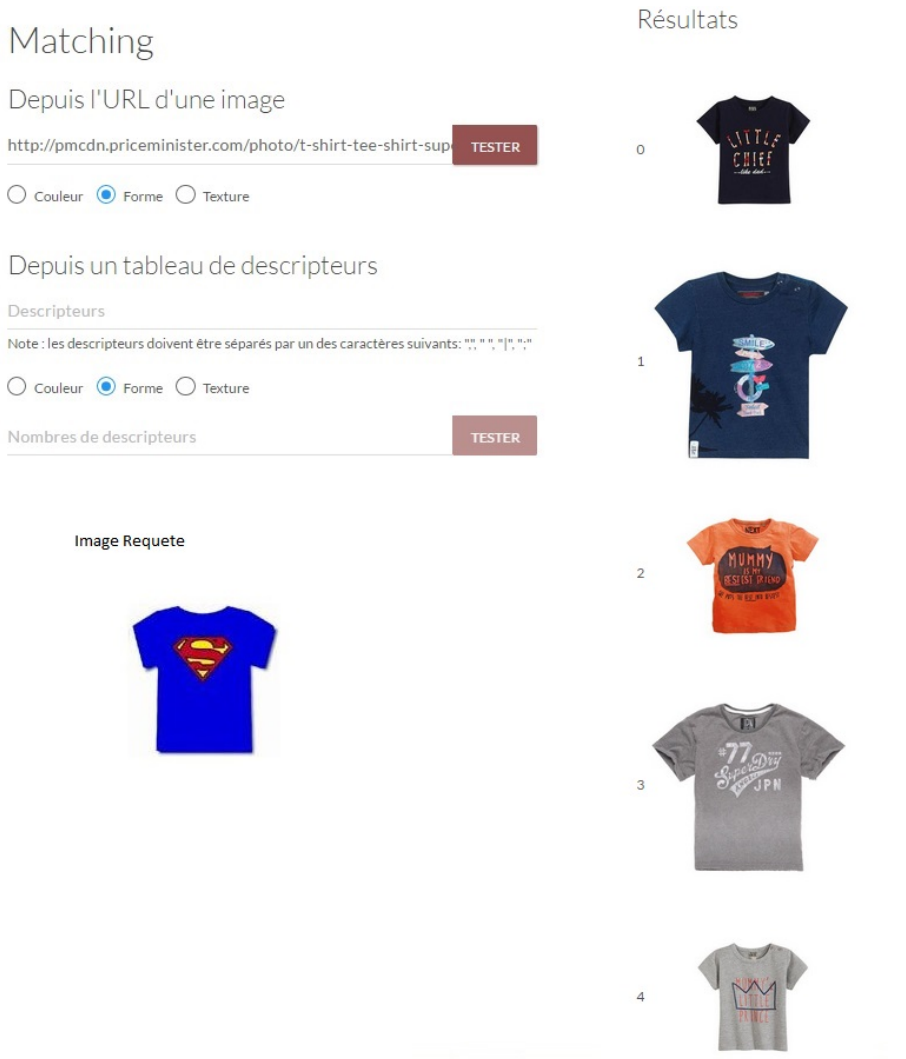


Figure 2.1: An example of Shop Match based on shape descriptors. The user inputs a query images and then choose a type of similarity: color, shape or texture.

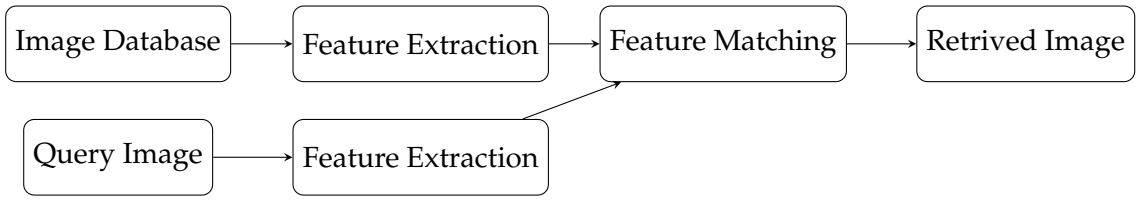


Figure 2.2: An illustration of CBIR system.

After the feature extraction stage, the query image and the images in the database images are indexed to build features. The image database indexing is costly process and is performed usually offline. The feature matching step computes the similarities between the features by using distance by the vector's distance measure like L_1, L_2. To enable a personalized search, a learning procedure can be added to this step such as to learn a set of weights that minimizes the distance to the users' favorite results and maximizes the distance to the undesired ones.

The work in this thesis focus on such a CBIR system designed specifically for fashion items, but modified to correspond to the scenario of online advertising. The user is consulting a Web page containing an image of people wearing several fashion items/clothing (for example : dress, shoes, bag etc.). The system then must detect the presence of these items in the query image, try to extract them and then search a database of retail products to find items that are similar (or identical) to those present in the query image. The feature extraction part is a key step for the system to propose meaningful results in terms of high-level expectations. Since we aim to process the image in uncontrolled conditions, a clothing extraction is necessary to obtain clean objects while avoiding mixing with the background. Afterwards, the features can be extracted from the object's region and retrieved by existing retrieval techniques. The clothing segmentation is thus the most important module in the system but the existing method failed to produce satisfying results due to the poor segmentation accuracy. The work in this thesis focuses on solving the clothing segmentation problem in order to increase the accuracy of the entire processing chain.

Our goal is thus to precisely segment the object of interest from the real life image. This is a difficult problem, because the photos are taken in an unconstrained condition.

Figure 2.3: Examples from a fashion dataset. This illustrates the huge diversity of clothing items and backgrounds that are present, and also the intra-class variability due to clothing deformation and occlusions.

Our algorithm encounters various types of obstacles, as shown in Fig. 4.7. The potential factors are summarized as the following:

- **Clothing variation.** Clothing objects have large inter-class diversity. The colors, the texture and the shape can vary significantly, even in the same category. We aim to adapt our approach deal with this large diversity. Therefore, our algorithms should rely on the global information used to resist against the local variations.

- **Human pose variation.** Since the fashion items to be extracted are used by human subjects, our proposals will have to rely on the pose estimation to localize the different parts of the human body. Even if fashion images have less pose variation than, for example, sports images, the localization of some parts is still problematic. For instance, limbs are fine structures that are hard to localize. We proposed to include

hard samples to improve human pose estimation and we devised a new ways of annotating to better handle this issue.

- **Occlusion issue.** This occurs when the clothing item is overlapping over another clothing item, a background object and another nearby person. From an object description point of view, finding the outline of the whole object is preferred to describe the shape (even if some parts of it are occluded — which implied predicting the missing parts), while a well segmented object is preferred to describe well the content (such as to avoid including the background in the description). Information that was gathered in the beginning should thus be used to get specific results.

- **Background condition.** The background takes on many forms and is therefore impossible to model mathematically. Sometimes backgrounds can be problematic for the clothing segmentation when they have a close resemblance.

- **Clothing deformation.** The very act of clothing being worn inevitably creates wrinkles. In other words, it changes the fabric, thus altering the form, but also the texture and the local structure. It also creates spurious contours in the images, that are not specific to the class of object.

- **Limited database.** The publicly available clothing databases are very limited in terms of the size and lack of the accuracy. To achieve a precise segmentation, we have built a database which contains the precise contour of the clothing objects. This database is dedicated solving the precise segmentation problem.

2.2 Contributions of the thesis

Since generic image segmentation approaches do not work well for object extraction, the method we devise should be adapted to the specificity of each object. As we shall see, we inject specific knowledge into the object models in order to get a better fit to the real image conditions. Since the goal is to extract a set of different objects, the devised procedure should be easy to adapt to new objects with little human intervention. Also, special algorithms should be designed for small objects. For instance, the forms which

accessories take make them challenging objects to extract. Finally, the algorithms should be eventually optimized for real-time applications (the user will not consult a web page for a very long time).

In this context, we approach the problem in three steps, of increasing complexity:

- First, we first devise a pixel based classification method for extracting an object from the background by using object specific information. The local information extracted using a person detector is used to devise a two class SVM pixel classifier (object vs. background). The result of the classification step is used to drive an active contour potential function to guide the local convergence. The model has a parameter which allows the user to control whether to predict the contour in the presence of small occlusions, or to follow local contours. For example, a smooth output is desired in order to better describe the shape with occlusion. On the other hand, to successfully describe the content, a well fitted contour is necessary. To this end, we inject specific knowledge about the object (local curvature) into the energy function, that guides the convergence of the active contour, which helps disambiguate the object contour in a cluttered background (see Ch. 4 and Sec. 2.3.2).

- Second, we extend the previous method by using template matching to allow multi-label segmentation. We introduce a new framework that combines local and global object characteristics together with a new active contour that optimizes the gap with respect to the global segmentation model, by measuring how well the proposed segmentation fits into the real distribution of the contours. In addition, to evaluate this work we have prepared a new benchmark database that contains contour based segmentation and we made it available to the community (see Ch. 5 and Sec. 2.3.3).

- Fully Convoutional Networks (FCN) have recently started recently to be used with good success for many segmentation tasks. However, they still cannot produce segmentation with accurate contours, as demanded by our use case. Our next contribution is focused on improving the contour localization of FCNs by making combined use of super-pixels (the local aspect) and FCN prediction (global aspect). The idea is that super-pixels are uniform regions that tend to share the same label (do

not cross over several objects) and can be made to closely follow physical contours in the image. According to this observation, we introduced a new super-pixel potential exploited by a Conditional Random Field (CRF) approach to drive the segmentation procedure. This potential include the pixel-wise prediction from FCN, the local appearance prediction and the neighborhood smoothness (see Ch. 6 and Sec. 2.3.4).

2.3 Structure of the manuscript

Since the works are consistently based on the segmentation problem, the related state of the art process involved is presented in its entirety in Ch. 3. The work begins by using the dress segmentation to segment precisely one clothing object at a time and is presented in Ch. 4. In Ch. 5, the segmentation work is extended to a wider selection of objects by a template based segmentation method. With the success on the deep neural network, our work then aims to improve the fully convolutional neural network by inferring context from superpixels in Ch. 6. Finally, we conclude our thesis in the last chapter with an critical analysis of our results and a set of proposals to advance the presented line of research. In the remaining part of this section, we briefly present the content of each chapter.

2.3.1 Chapter 1

In this chapter, we present the state of the art research on the object detection, the image segmentation and the research related on fashion. For the object detection, we present the traditional method: classification by edge orientation feature (Sift features [56], HOG features [13])in the local window, followed by the deformable parts model [22]. It incooperates the spatial deformation of the parts configuration. We then introduce the Convolutional Neural Networks (CNN) [40] which have been found recently to advance significantly the detection results. This work has already extended on the problem of detection and location of objects in a series of related articles [26; 25; 62]. To present the work of the image segmentation, we start by the traditional method that relies on

the local features and infers that there is a larger context, for example Graph cut [6] and Active Contour [37; 8]. Then we present the semantic segmentation method, that aims to assign to each pixel several labels (approach called multi-labeling). We conclude this part with a presentation of Instance Segmentation approach which formulates the object extraction as a collective problem but with the goal of producing instance segmentation. In the third part of this chapter, we present related research covering fashion retrieval, attribute selection, and clothing segmentation. The fashion retrieval has two aspects : style retrieval that retrieves a style-alike product and street-to-shop retrieval that searches for the product from a street image. The attribute selection is another way for retrieval to automatically tag the image to acquire the meta data. Finally we present the clothing segmentation which is also the focus of our work in this thesis.

2.3.2 Chapter 4

Instead of struggling with the large sets of fashion items, we start to focus our work on one single object. The dress is a logical choice, since it's the most popular object in the fashion industry. Moreover, we aim to achieve a more precise segmentation that, three years ago, was lacking in conclusive results. We therefore propose a dedicated object extractor for dress segmentation in fashion images by combining local information with prior knowledge. First, a person detector is applied to localized sites in the image that is likely to contain the object. Then, an intra-image two-stage learning process is developed to roughly separate foreground pixels from the background. Finally, the object is finely segmented by employing an active contour algorithm that takes into account the previous segmentation and injects specific knowledge about local curvature in the energy function. The method is validated on a database of manually segmented images. We show examples of both successful and unsuccessful segmentation cases. We quantitatively analyse each component and compare them with the well-known GrabCut foreground extraction method. Our procedure has the advantage of being easy to adapt to any fashion clothing item by following a similar development approach.

2.3.3 Chapter 5

To make the previous method more general, we propose to extract the prior information from the template to better guide the segmentation. In this chapter, a new framework is built for extracting deformable clothing items from images by using a three stage global-local double switching fitting procedure. First of all, a set of initial segmentation templates are generated from a handcrafted database. Then, each template initiates an object extraction process by a global alignment of the model, followed by a local search, minimizing a measure of how badly fit is the segmentation with respect to the potential boundaries in the neighborhood. Finally, the results provided by each template are aggregated with a global fitting criterion to obtain the final segmentation. The method is validated on the Fashionista database and on a new database of manually segmented images. Our method compares favorably with the Paper Doll clothing parsing and with the recent GrabCut on the OneCut foreground extraction method. We quantitatively analyse each component, and show examples of both successful and less successful segmentation cases.

2.3.4 Chapter 6

Recently, Fully Convolutional Networks (FCN) have achieved state of art results in semantic segmentation. In general, the FCN produces a good object location with a rough segmentation result. Likewise most networks, the FCN still suffers from contour localization. Since the low level information has contour structure that has been abandoned in pooling layers, we propose a post processing to take advantage of the rough segmentation from a higher level and therefore correcting the contour localization by using the lower level information. To achieve this we extend the output of the FCN to infer context from local units (superpixels). More precisely, we optimize an energy function, which combines the large scale structure of the image with the local low-level visual descriptions of superpixels over the space of all possible pixel labellings. Our method shows promising results compared to the fine-tuned FCN network used as a baseline, as well as to the well-known Paper Doll and Co-parsing methods for fashion extraction.

2.3.5 Conclusion

We conclude the this manuscript (in Chap. 7) with an analysis of the strengths and drawbacks of the proposed methods. We also reflect on the future work concerning the image segmentation and fashion object retrieval.

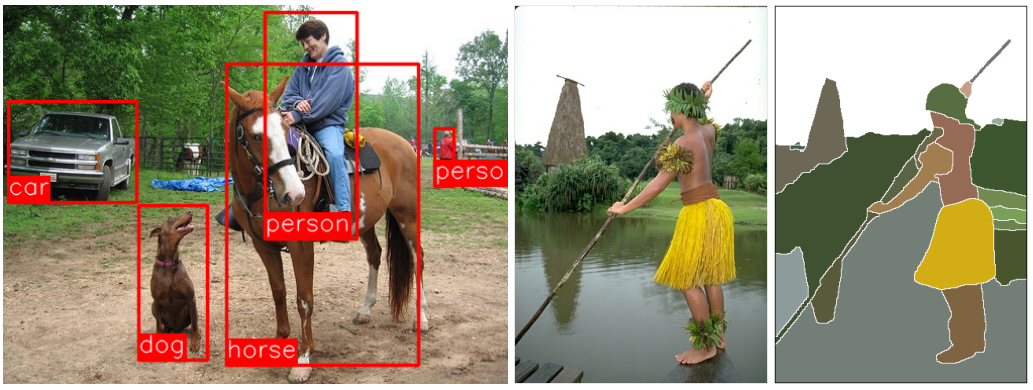
Chapter 3

State of the Art Research

The main objective in this thesis is to segment precisely the clothing's area and annotate each location with a semantic label. Our work mainly involves the following fields: object detection, and image segmentation. As shown in Fig. 3.1(a), the goal of object detection is to predict the object's presence and locate the Region of Interest (ROI) delimited by the bounding box. The image segmentation aims to separate the image in Fig. 3.1(b) into semantic regions illustrated in Fig. 3.1(c). The object detectors are used in this thesis for localizing the person and different items of clothing. The detected object can give a helpful hint for the image segmentation to focus on a smaller targeting region.
In this chapter, we first start by introducing the machine learning tools that are the foundation of these fields. Afterwards, we'll present the state of the art in object detection, image segmentation and research related to the fashion application.

3.1 Machine learning

The aim of machine learning is to understand that the structure of the data and generalize this knowledge to new data. There are mainly three types of learning: **Supervised Learning** Each training data set is associated with the label, and therefore predicts correctly a label on the test data,for example, the linear regression, Support Vector Machine, Neural network. While in **Unsupervised Learning**, the data isn't associated with any label. The algorithm aims to make a discovery of the knowledge in the space of the description, for example, in the clustering algorithm, the K-means, k-medoid, Gaussian



(a) (b) (c)

Figure 3.1: Illustration of (a) object detection (image from [62]), (b, c) image segmentation (image from [2]).

Mixture Model. In **Semi supervised Learning**, the input data is a mixture of data consisting of the labelled and unlabelled data. In this section, we introduce the most commonly used, supervised learning method in the thesis.

3.1.1 Support Vector Machine

Support Vector Machine(SVM) is a powerful machine learning tool for data classification. Presented below is the hypothesis in the form of the linear model for the observations and variables:

$$y(x) = w^{\mathsf{T}}\phi(x) + b \tag{3.1}$$

The training data set has N input vectors $x_1, .., x_N$ with target values $t_1, .., t_N$ where $tn \in -1, 1$, $\phi(x)$ denotes a fixed feature-space transformation. The objective is to find a parameter w, b that satisfies $y(x_n) > 0$ if $t_n = 1$ and $y(x_n) < 0$ if $t_n = -1$ for all the training data. However there exists multiple solutions that satisfy these conditions: We want to find one of the best solutions that gives the smallest generalization error possible. For this purpose, the SVM approach introduces the notion of 'margin' that is the smallest distance between the decision boundary and the samples illustrated in Fig. 3.2. The SVM aims to find the decision boundary that maximizes the margin and probably minimizes the generalization error. As we know, the distance from a point x_n to the decision surface

is shown here:

$$\frac{t_n y(x_n)}{||w||} = \frac{t_n(x\phi(x_n)) + b}{||w||} \tag{3.2}$$

The margin is the smallest distance from the decision boundary. Then finding the max margin can be formulated as follows:

$$\arg\max \frac{1}{||w||} min[t_n(w^T\phi(x_n) + b)] \tag{3.3}$$

Note that rescaling w and b by the same scale won't change the distance from one point to the decision surface. Then we add one constraint to the formula that the points closest to the boundary equal to 1:

$$t_n(w^T\phi(x_n) + b) = 1 \tag{3.4}$$

Therefore all the data points should satisfy the equation below:

$$t_n(w^T\phi(x_n) + b) \geq 1, n = 1, ..., N \tag{3.5}$$

The optimisation problem must be maximized $||w||^-1$, which is equal to a minimization of the inverse form:

$$\arg\min_{w,b} \frac{1}{2}||w||^2 \tag{3.6}$$

In order to solve the optimisation problem, the Lagrange multipliers are introduced.

$$L(x, b, a) = \frac{1}{2}||w||^2 - \sum_{N}^{n=1} \alpha_n t_n(w^T\phi(x_n) + b) - 1 \tag{3.7}$$

By setting the derivatives of $L(w, b, a)$ with respect to w and b equal to zero, we obtained:

$$\begin{aligned} \sum_{n=1}^{N} \alpha_n t_n \phi(x_n) &= w \\ \sum_{n=1}^{N} \alpha_n t_n &= 0 \end{aligned} \tag{3.8}$$

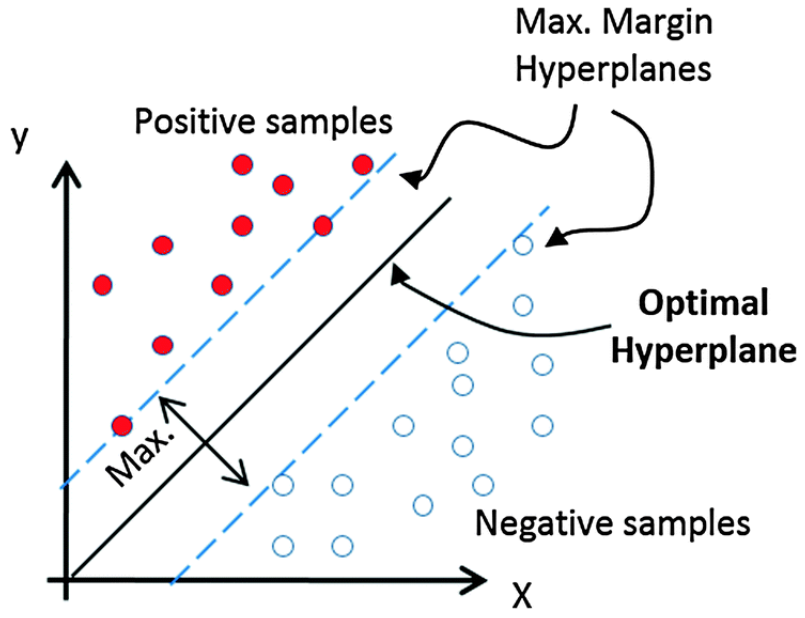


Figure 3.2: Illustration of SVM for 2 class

By eliminating the w, b, Eq. 3.7 becomes the dual problem:

$$L(\alpha) = \sum_{n=1}^{N} \alpha_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} \alpha_n \alpha_m t_n t_m k(x_n, x_m) \tag{3.9}$$

w.r.t the constraints

$$\begin{aligned} \alpha \geq 0, n = 1, ..., N \\ \sum_{n=1}^{N} \alpha_n t_n = 0 \end{aligned} \tag{3.10}$$

where the kernel function is defined by $k(x, x') = \phi(x)^T \phi(x')$. Finally, this quadratic programming problem can be optimized by a quadratic function of α subject to a set of inequality constraints. In order to classify new data points, we replace the w, b in Eq. 3.1 by the Eq. 3.8:

$$y(x) = \sum_{n=1} N \alpha_n t_n k(x, x_n) + b \tag{3.11}$$

In [66], the authors extend the classification problem (OCSVM) to one class classification when the training samples are only containing samples from one class. Then the

origin point is treated as the only sample of the other class. The strategy is to return to a function f which outputs the label $+1$ for a subset of data(most of the data), and -1 elsewhere.

$$y(x) = \min \frac{1}{2}||w||^2 + \frac{1}{vl}\sum_{i=1}^{l}\epsilon_i - \rho \tag{3.12}$$

subject to:

$$(w \cdot \phi(x_i)) \geq \rho - \epsilon_i, \quad i = 1, 2, ..., l, \quad \epsilon_i \geq 0 \tag{3.13}$$

where ϵ_i is the slack variable that penalizes the objective function. v controls the trade off between including more data into the positive and minimizing the $||w||$. The above mentioned problem can be transformed into a dual problem by Lagrangian multipliers and finally solved by the quadratic programs, which is similar to the previous SVM optimization. After solving the problem, the decision function then becomes:

$$f(x) = sign((x \cdot \phi(x)) - \rho) \tag{3.14}$$

3.1.2 Deep neural networks

Recently deep neural networks have achieved remarkable success in object detection. The authors in [42] introduce the very first architecture of convolutional network for digit recognition. The network is trained on MNIST(Mixed National Institute of Standards and Technology dataset) : a large database of handwritten digits containing 60,000 training images and 10,000 testing images. The network is composed of two convolutions layers, two subsampling layers and followed by two full connection layers. The final layer outputs the probability of each digit(0-9).

The network has a good accuracy with the test error rate 7.6%. At that time, due to the limit of hardware computational capacity, it was challenge to apply larger images to the deeper network and with more complex recognition tasks. Nowadays, with the increasing graphical card computational capacity and the decreasing price, the neural

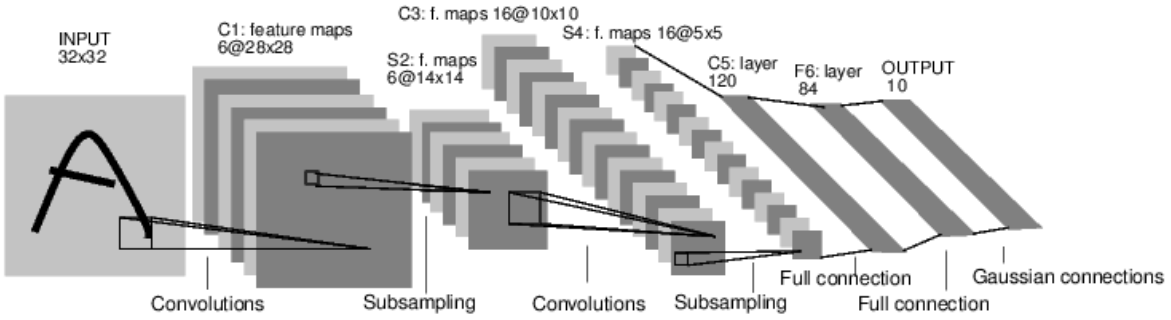


Figure 3.3: Illustration of LENET-5 architecture

network starts to show promising results. Moreover, with the introduction of the large scale Imagenet dataset, the network is fed up with the fact that there are too many images available for the task of object detection. Alexnet [40] proposes a deep convolutional neural network(CNN) to enable the image classification for 1000 classes. As illustrated in Fig. 3.5, the networks contain five convolutional layers and three fully-connected layers. The network outputs a vector in a 1000-dimensional space indicating the classification probability of each class. We present each layer in detail:

The Convolutional layer is a mathematical operator. Given a kernel and an image, the kernel filters across the image. In each computation, the kernel is flipped in an up-down and left-right direction, and multiplied locally.

$$(f * g)(i) = \sum_{j=1}^{m} g(j) \times f(i - j + m/2) \tag{3.15}$$

The convolution can detect the correspondence between the local window and the kernel structure in the signal. As for the applications in the images, the convolution can tackle the task such as edge detection with edge-alike filters, and object detections with object filters, and so on. Instead of using just one filter for detection, the convolution network has divided the detection into hierarchical steps by staking several layers in sequence. As illustrated in Fig. 3.4, the filters in the first layer are the primary filters that extract the local information, such as edge alike filters. While in the higher layers, the filters are more complex and tend to become more object like filters in the semantic level.

The ReLU layer activates the units after the convolutional layer is processed in the

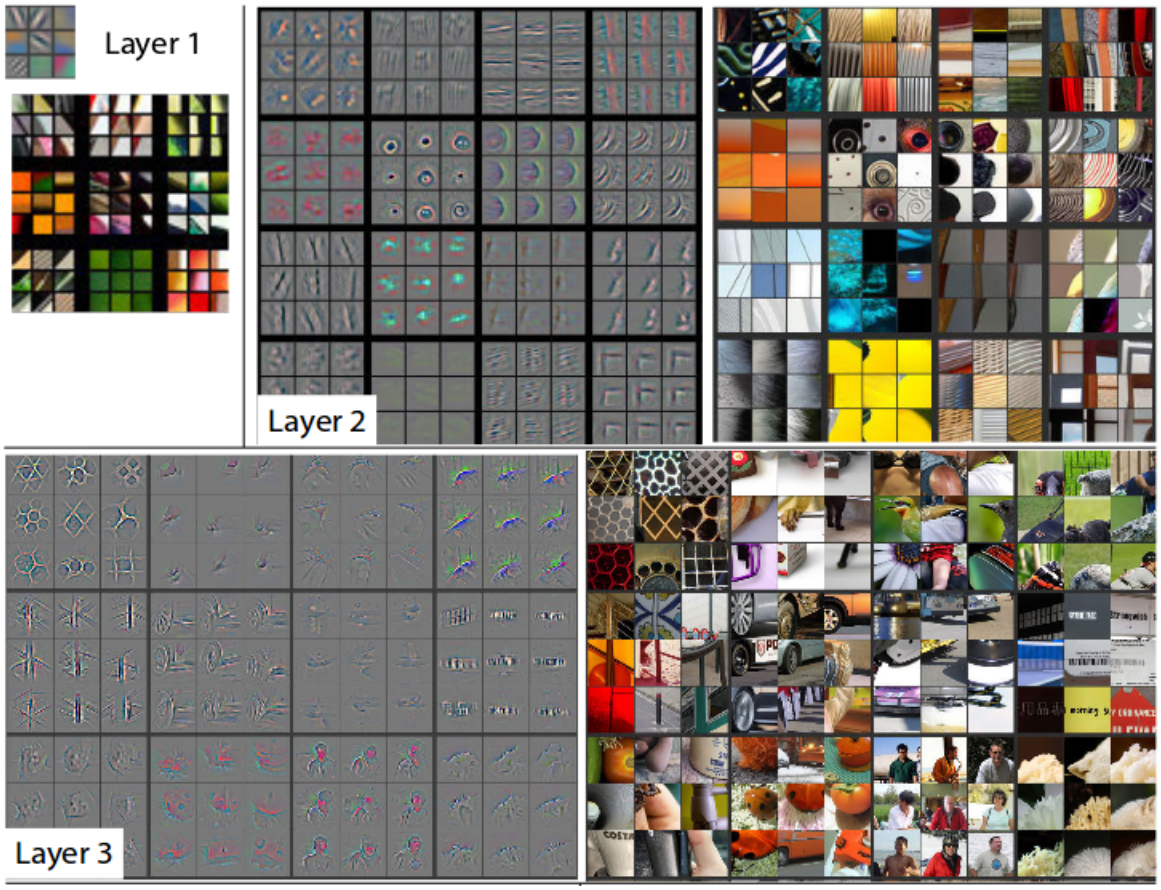


Figure 3.4: Illustration of CNN filters

network. To accelerate the training time, the non-saturating non-linearity operation is commonly used:

$$f(x) = max(0, x) \tag{3.16}$$

The Pooling layer is applied to the pooling operators by taking the maximum or average of the receptive field. The input is thus downsampled which leads to a smoother input and reduces the sensitivity of the filters against noise.

In **the Fully connected layer**, each unit has a complete connection to all the units in the previous layer. This can be interpreted as a multiplication with a bias offset. The last fully connected net has 1000 neurons that produces the probability of each label.

When training the network, **a loss layer** is appended to the final layer. The Loss layer

provides a feedback for the network to correct the weights during training. This feedback can be interpreted by a loss function that computes the deviation from the network output to the expected output, in other words, the ground truth. Here is an example of softmax loss:

$$L_i = -log(\frac{e^{f_{y_i}}}{\sum_j e^{f_{y_i}}}) \tag{3.17}$$

And the overall loss function is the summation of all samples. To update the weight with respect to the loss function, we can use **backpropagation** to compute the gradient for weights and bias W, b. The update rule for weights by stochastic gradient descent:

$$\begin{aligned} v_{i+1} = 0.9 * v_i - 0.0005 * \sigma * w_i - \sigma * \frac{\partial L}{\partial w}|_{w_i} \\ w_{i+1} = w_i + v_{i+1} \end{aligned} \tag{3.18}$$

Where $\frac{\partial L}{\partial w}|_{w_i}$ denotes the derivatives of the cost function with respect to the parameters. And 0.9 denotes the momentums that reduces the training iterations by smoothing out the gradient changes, 0.0005 is the weight decay that is a factor for the regulation. v_i is the momentum variable that considers these factors into derivative loss function in relation to the weights. There are also techniques to improve the network's parameter learning.

Dropout [71] : To prevent network from the overfiting, the neurons are trained randomly with drop out with ρ probability (e.g. 0.5), which means the neurons don't contribute to the forward and backward propagation. This is a simple and efficient technique that prevents units from co-adaptation.

Parameter sharing : If a filter for each position is learned separately, it would be a larger number of parameters to process. Parameter sharing proposes to learn a single filter for one depth slicing. This technique not only reduces large amounts of the parameters' number, but also leads to learning a more pertinent filter.

The parameters in the network are initialized with small random numbers generated by a Gaussian distribution to enable the network's asymmetry.

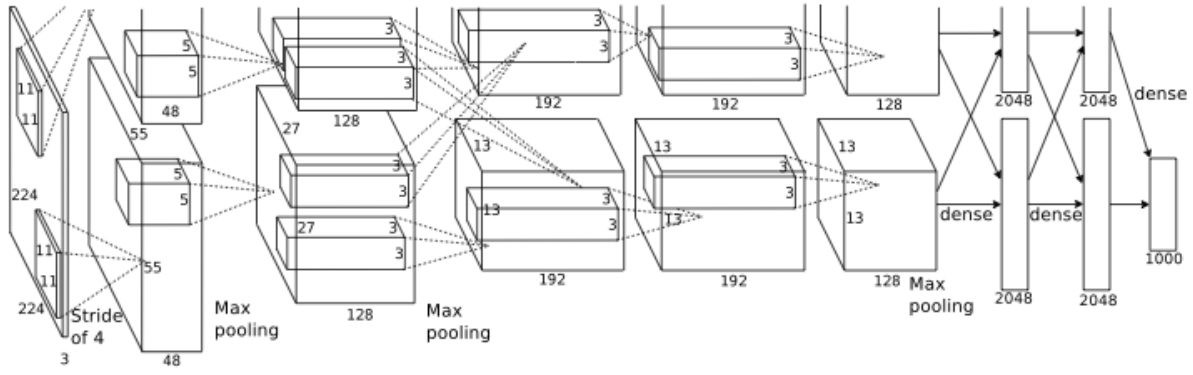


Figure 3.5: Illustration of Alexnet architecture

VGG net [69] proposes a new network which uses a small filter with a size 3×3. This largely reduces the number of parameters for training, thus the network can avoid the problem of degradation.

The 'deep features' are computed by a forward propagation which passes through the network's layer, then receiving the fc7 feature in the fc7 layer, and the same is true for the procedure for the fc6 and pool5 features.The features are proven to have a rich mid-level description [26] and uses the tasks in the image retrieval and image classification.

3.2 Object Detection and classification

3.2.1 Histogram of Oriented Gradients and Object Detection

[13] proposes an HOG (Histograms of oriented gradient) descriptor for person detection(Predict if there is human in image). At first, an image is divided into cells to produce a detection window. A local histogram is computed by the distribution of the gradient orientation in a cell. The HoG descriptor combines the histograms and normalizes by one measure which is computed in all the cells. The HOG feature describes the local edge/shape in the detection window, therefore these features are widely used for the task of detection by feeding it into a support vector machine (SVM) to learn a decision model.

3.2.2 Deformable models

Yet the HOG descriptor captures the local window feature for the rigid model, it's still inadequate to more generic cases. The objects have an infinite amount of potential forms and a large amount of diversity due to the self-deformation and photo taking condition. Furthermore, the authors in [22] introduce the deformable model(DFM). The DFM models consider an object composed of parts for the purpose of capturing the object's partial appearance and deformation. The parts are connected to the root filter by a star model. Then a detection score is given by matching the scores of each filter at their respective locations minus a deformation cost that depends on the relative position of each part with respect to the root filter.

3.2.3 Person detector

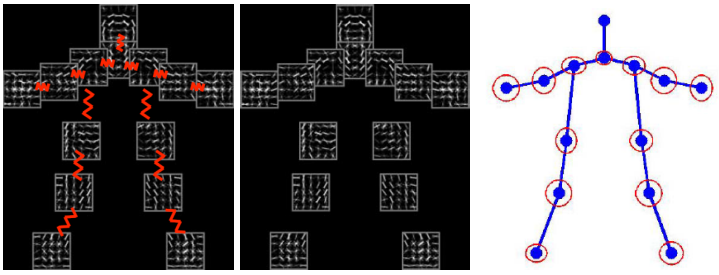
The authors in [83] particularly address the human detection problem with a flexible mixture of parts. They use the tree model to connect the human parts that describes the part to part deformation illustrated in Fig. 3.6, not the above mentioned part-root deformation. They use message-passing to detect deformation from childrens' parts to the parents' parts for person detection. This outputs the position of each body part in a human detection.

For clothing detection and recognition, applying a person detector is a reasonable starting point. As in many other studies (e.g. [52; 36; 79]), we use the person detection with articulated pose estimation algorithm from [83], which has been extensively tested and proven to be very robust in many practical situations. It is based on a deformable model that defines the object as a combination of parts [83]. Let L denote for the image, and p_i for the pixel location of part $ii \in 1, ..., K$ and the type of part t_i. Since the part may include orientations(horizontal arm versus vertical arm) and spans the semantic classes(an open versus closed hand), we define this with a set of spanned types by $t = t_i, ..., t_K$.

First we present a co-occurrence model. The model encodes a compatibility function that chooses the parts of spanned types by a sum of local and pairwise scores:

$$S(t) = \sum_{i \in V} b_i^{t_i} + \sum_{ij \in E} b_{ij}^{t_i, t_j} \tag{3.19}$$

This is where the $b_i^{t_i}$ favours a particular type assignment for part i, and $b_i j^{t_i, t_j}$ favours the particular pair of type assignment for part i and j.



(a) (b) (c)

Figure 3.6: The principle of the model in [83] : (a) Approximation of the deformations by the mixture, (b) Local template, (c) Tree structure.

The detection score is defined as the score to parts minus the parts deformation cost. The mixture model not only encodes object structure but also captures spatial relations between part locations and co-occurrence relations between parts. To introduce the appearance model and deformation model into previous formulation, see below:

$$S(I, p, t) = S(t) + \sum_{i \in V} w_i^{t_i} \cdot \phi(I, p_i) + \sum_{i,j}^{t_i, t_j} \cdot \psi(p_i - p_j) \tag{3.20}$$

$\phi(I, pi)$ denotes a feature vector (e.g. HOG descriptor) extracted from pixel p_i, therefore the first sum computes the local score by placing the template, $w_i^{t_i}$ for part i. The deformation is encoded by a relative location in $psi(p_i - p_j) = [dx, dx^2, dy, dy^2]^T \quad where \quad dx = x_i - x_j \quad and \quad dy = y_i - y_j$. The second term controls the relative placing of part i with respect to part j.

To solve the learning problem, suppose that a classifier is in the formulation below:

$$f_\beta(x) = \max_{zinZ(x)} \beta \cdot \phi(x, z) \tag{3.21}$$

where β denotes the model parameters and z is the latent value that is included in the

possible latent value set $Z(x)$. With the labelled examples $D = (< x_1, y_1 >, .., < x_n, y_n >)$, where $y_i \in -1, 1$, we would like minimize the objective function,

$$L_D(\beta) = \frac{1}{2}||\beta||^2 + C\sum_{i=1}^{n} \max(0, 1 - y_i f_\beta(x_i)) \tag{3.22}$$

$\max(0, 1 - y_i f_\beta(x_i))$ is the standard hinge loss. If considered, only the latent value for positive example and the loss function becomes a convex function, and this function can be solve by a latent SVM.

3.2.4 Deep learning based method

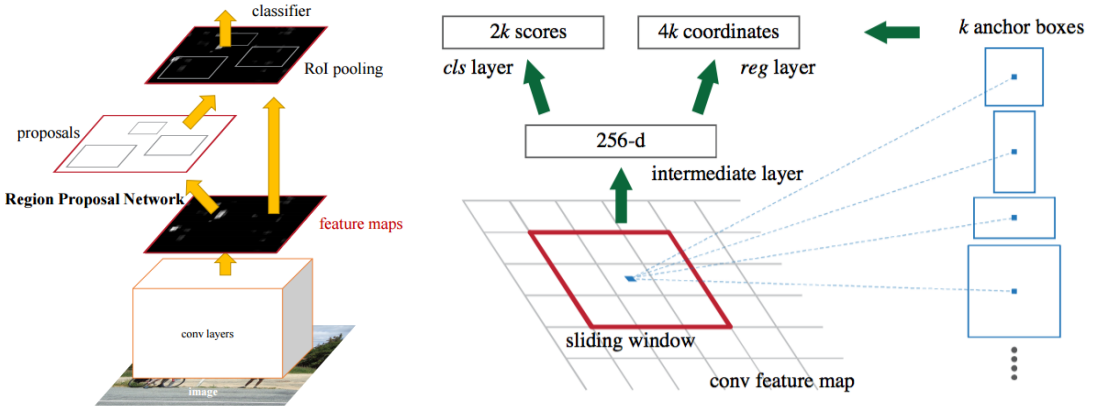
The RCNN network puts forward the detection of the object's location. The researh follows a similar philosophy : a region proposal module to propose regions from the bottom level, and classify the regions by network.

a. RCNN [26] The RCNN starts with a region proposal module [74] to offer the category-independent region proposals from the bottom level. Then, each proposal passes forward to the CNN network and obtains its fc7 feature which deems it to have a good mid-level description for the region. The SVM model is learned on the features to predict the region label. Finally a greedy, non-maximum suppression rejects a region if it overlaps with a higher scoring region larger than a learned threshold.

b. fast RCNN [25] The network first processes the whole image with several convolutional and max pooling layers to produce a convolutional feature map. Next comes the object proposal, which is in a region of interest(RoI), a pooling layer that converts the features inside an ROI leading into a fixed length feature vector with max pooling. Each feature vector is fed into a sequence of fully connected layers that finally branches into two sibling outputs: one that produces softmax probability estimates over K object classes plus a catch-all class and another layer that outputs four real-valued numbers with a bounding box offset.

c. faster R-CNN [62] The faster R-CNN illustrated in Fig. 3.7(a) gives a region proposal network(RPN) that predicts an object bounding box and objectness score. RPN network in Fig. 3.7(b) slides on the convolution feature map and maps to a lower-dimensional

feature containing the class layer and bounding box coordinates. This feature is fed into two sibling fully connected layers: a box-regression layer and a box-classification layer.



(a) (b)

Figure 3.7: Illustration of (a) Faster RCNN, (b) RPN Network(image from [62]).

3.3 Image Segmentation

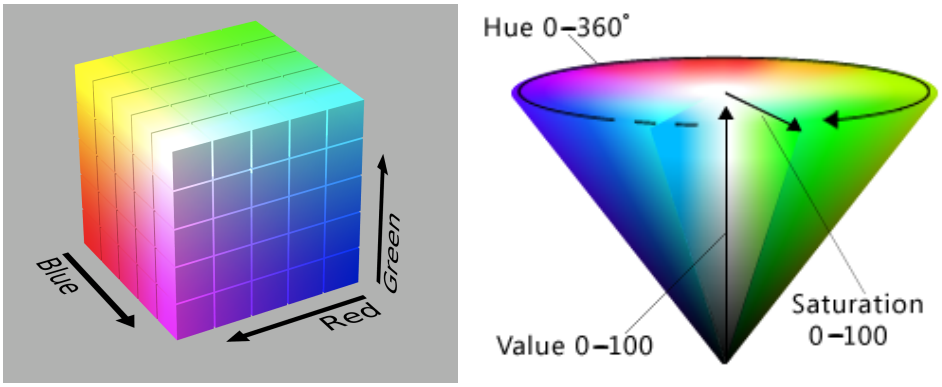
3.3.1 Image Feature

Image features are widely used to describe image content for image segmentation. Here we present the features that have been used in the thesis.

Color histogram

Color histogram is the most common color descriptor. A histogram reflects the item's frequency by calculating the presence of pixel color, thus a color histogram is a global descriptor to describe color distribution in the image. The color value is discretized by bin number N and then added to the discretized bin. Instead of building a histogram for 3 channels separately, A 3d color histogram is built with a dimension of $N_r \times N_g \times N_b$. The color space can be described as a cube, and a pixel value is counted when it falls into the sub cube. The histogram is finally normalized by dividing the total pixel number.

Another Hue Saturation Value(HSV) color space is deemed to be closer to the human perception system. **Hue** channel stands for the perceiving of color varied from 0 to 360,



(a) (b)

Figure 3.8: Illustration of (a) RGB, (b) HSV.

that corresponds to the color variations in red, yellow, green, blue and so on. **Saturation** is the value of the brightness relative to its own color.**Value** denotes the relative brightness to the white.

The drawback in this feature is that the spatial information is lost during computation, therefore, two totally different images can have an identical feature.

Texture feature

Here we present the famous texture feature in [15]. This feature is extracted from the texture information in the Fourier space. The Fourier transformation is defined as follows:

$$F(k,l) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} f(i,j) e^{-i2\pi(\frac{ki}{N} + \frac{kj}{N})} \tag{3.23}$$

After an image is transformed in the Fourier space, the center stands for the DC component(Zero frequency component), the near center area stands for the low frequency component, and the area far from the center stands for the high frequency component. The different motives of texture correspond to different regions in the Fourier space. By dividing the space into N_a angular region and N_d diameter, it leads to a $N_a \times N_d$ regions. And the descriptor is the vector of the summation of the energies in each region.

3.3.2 Traditional methods

Image segmentation approach aims to segment out the desired object. Traditional methods rely on the local context to predict a global distribution. Most methods convert the image segmentation problem into an energy minimization problem, and thus solving the problem by using a global context.

Active contour

Active Contour(AC) aims to converge a closed contour to the desired boundary. In the begining, Snake [37] exhibits an evolvution in the contour with regards to control points. By specifying a proper energy, we can control the snake evolution. The energy is composed of the following: Internal energy to control the continuity; image force to attach the contour to the large gradient and light or dark lines; external constraint force by user input or other higher level information. As the snake is unable to converge with the concave structure. The authors in [77] propose using the Gaussian Vector flow to calculate the driving force in the 'empty' region. But the GVF is still computationally expensive. Level set methods are a tool for surface and shape analysis. The active contour without edges [8] is based on a model that evolves the zero level of a level set function. The authors propose an AC by considering the inside content that can segment objects whose boundaries are not necessarily well-supported by gradient information. The fitting term that measures the fitness of the inside of the pixel value to its mean value, and at the same time, it is similar for the outside region. The fitting term is defined as follows:

$$F_1(C) + F_2(C) = \lambda_1 \int_{inside(C)} |u_0(x,y) - c_1|^2 + + \lambda_2 \int_{outside(C)} |u_0(x,y) - c_2|^2$$

By adding the the regularizing terms: the length and the area, the energy become

below:

$$F(c_1, c_2, C) = \mu * \text{Length}(C) + \nu * \text{Area}(\text{inside}(C)) + \\ +\lambda_1 \int_{inside(C)} |u_0(x,y) - c_1|^2 + \\ +\lambda_2 \int_{outside(C)} |u_0(x,y) - c_2|^2 \tag{3.24}$$

where C is the evolving curve, c_1 and c_2 are the average pixel gray level values $u(x, y)$ inside and respectively outside the contour C. The curvature term is controlled by μ and the fitting terms by λ_1 and λ_2.

To solve this minimization problem, a level set function ϕ is defined as such:

$$\begin{aligned} C = \delta w = (x,y) \in \omega, \phi(x,y) = 0, \\ inside(C) = w = (x,y) \in \omega, \phi(x,y) > 0 \\ outside(C) = w = (x,y) \backslash \omega, \phi(x,y) < 0 \end{aligned} \tag{3.25}$$

With this definition, we replace with C by ϕ :

$$\begin{aligned} F(c_1, c_2, \phi) =& \mu \int_\Omega \delta(\phi(x,y)) |\Delta\phi(x,y)| dxdy \\ &+ \upsilon \int_\Omega H(\phi(x,y)) dxdy + \lambda_1 \int_\Omega |\mu_0(x,y) - c_1|^2 H(\phi(x,y)) dxdy \\ &+ \lambda_2 \int_\Omega |\mu_0(x,y) - c_2|^2 (1 - H(\phi(x,y))) dxdy \end{aligned}$$

The AC minimizes an energy that drives the evolution of the active contour towards the desired boundary. The curvature and the normal vector field can be correctly estimated from the level set function by applying the Euler-Lagrange equation to ϕ:

$$\frac{\delta\phi}{\delta t} = F|\delta U| \tag{3.26}$$

The level set function is derived to make the curve evolve iteratively in time towards the descent direction. The Active contour has been used in 4, 5 in the final step to help converge the contour.

Graph cut

Another interpretation of image segmentation is the attribution of label assignment : 1 to 'object' and 0 to 'background'. The GraphCut formulates the problem by defining the

energy using the unary and pairwise potential thanks to Markov Random Field(MRF). The unary term is the cost of assigning a label to this pixel, and the pairwise term is the cost of assigning the different labels to the neighbouring pixels. The energy representation is as follows:

$$E(\underline{\alpha}, \underline{\theta}, z) = U(\underline{\alpha}, \underline{\theta}, z) + V(\underline{\alpha}, z) \tag{3.27}$$

The segmentation problem is then converted to find a label assignment that minimizes the energy. The minimization is done using a minimum cut [6].

$$\hat{\underline{\alpha}} = \arg\min \underline{\alpha} E(\underline{\alpha}, \underline{\theta}) \tag{3.28}$$

The GrabCut [65] is an iterative segmentation method; users are asked to give the bounding box that contains the foreground or the scribbles denoting the foreground or background. The object's appearance is learned on the labelled pixels with the Gaussian mixture model(GMM) with k component for background and foreground models. Thus the unary term is now computed by the GMM model:

$$U(\underline{\alpha}, k, \underline{\theta}, z) = \sum_n D(\alpha_n, k_n, \underline{\theta}, z_n) \tag{3.29}$$

Shown above is the data term that is the minus log of the sum of the Gaussian probability distribution of k components and the weighing coefficients and finally divided by the size of the set.

$$D(\alpha_n, k_n, \underline{\theta}, z_n) = -\log p(z_n|\alpha_n, k_n, \underline{\theta} - log\pi(\alpha_n, k_n)) \tag{3.30}$$

Therefore we introduce the parameters(the mean $\mu(\alpha, k)$, the covariance $\sum(\alpha, k)$, the weight $\pi(\alpha, k)$) in Eq. 3.30, the complete data model as follows:

$$\begin{aligned} D(\alpha_n, k_n, \underline{\theta}, z_n) = -log\pi(\alpha_n, k_n) + \frac{1}{2}\log det \sum(\alpha_n, k_n) \\ +\frac{1}{2}[z_n - \mu(\alpha_n, k_n)]^{\intercal} \sum(\alpha_n, k_n)^{-1}[z_n - \mu(\alpha_n, k_n)] \end{aligned} \tag{3.31}$$

The smoothness term V encourages the coherence in regions and penalises the neighbouring pixels, having different labels and a subtle contrast. The contrast is computed by

the Euclidean distance in color space.

$$V(\underline{\alpha}, z) = \gamma \sum (m, n) \in C[\alpha_n \neq \alpha_m] exp - \beta ||z_m - z_n||^2 \tag{3.32}$$

where [] is the indicator, producing the number 1 if the condition inside is satisfied, otherwise it produces 0. C is the set of neighbouring pixels. Thus larger distance costs less, and the opposite case takes a larger cost. Another advantage of GrabCut is the iterative segmentation. The GrabCut algorithms take the result of a previous step as an input label to update the GMM model. And it applies the graphcut and then updates the appearance model until convergence is acheived.

A larger neighborhood will include more context into pairwise potential, ideally a fully connected neighbor (A pixel is connected to the rest of pixels in Eq. 3.33), but it then significantly increases the computational time.

$$E(x) = \sum_i \phi_u(x_i) + \sum_{i<j} \phi_p(x_i, x_j) \tag{3.33}$$

Where $x \in L^N$ denotes the labelling, the unary potential ϕ_u is given by a pixel classifier, the pairwise potential ϕ_p is defined by:

$$\phi_p(x_i, x_j) = \mu(x_i, x_j) \sum_{m=1}^{K} w^{(m)} k^{(m)}(f_i, f_j) \tag{3.34}$$

The above mentioned f_i denotes for the pixel feature, and $\mu(x_i, x_j)$ for the label consistency. The GrabCut has been used for comparaison in 4, **??**.

The authors in [39] propose an efficient inference for the fully connected CRF models. It takes on the form of independent product $Q(X) = \prod_i Q_i(X_i)$ is estimated by the mean field approximation by minimizing the KL-divergence from the exact distribution. The distribution is updated iteratively :

$$Q_i(x_i = l) = \frac{1}{Z_i} exp\{-\phi_u(x_i) - \sum_{l' \in L} \mu(l, l') \sum_{m=1}^{K} w^{(m)} \sum_{j \neq i} k^(m)(f_i, f_j) Q_j(l')\} \tag{3.35}$$

The approximated distribution is then breaking into three steps : Message passing, compatibility transform, and local update step. Convolution by down sampled distribution creates an efficient message passing.

Contour detection

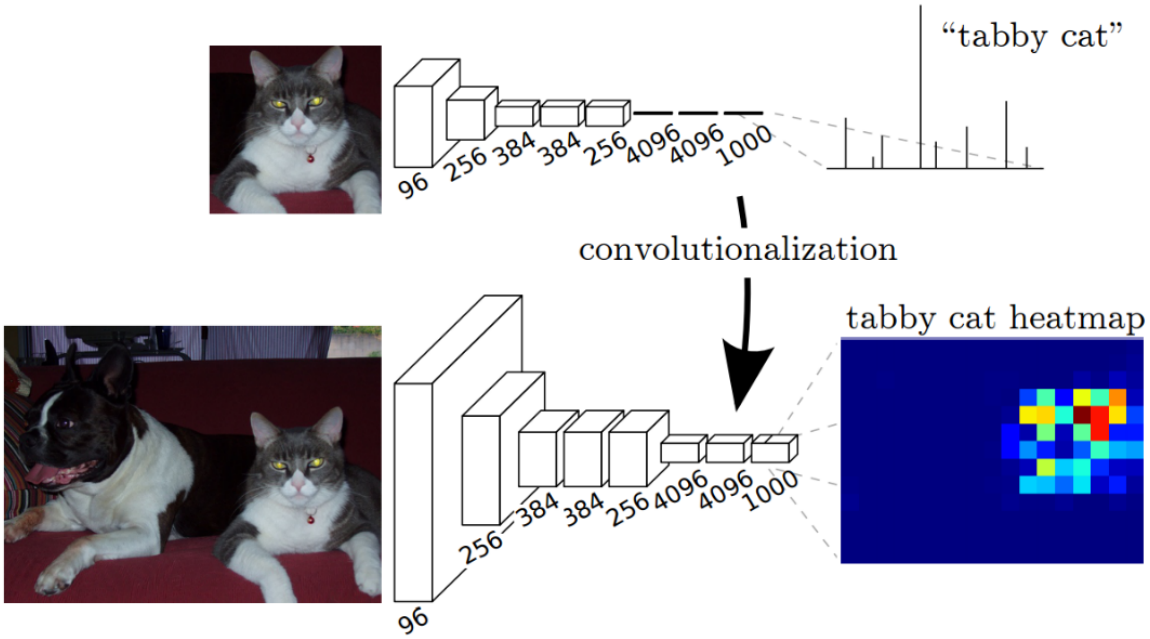
The authors in [2] approach another way of image segmentation using contour detection. To detect the contour, an oriented gradient of histogram is computed on the multiscale cues of Lab channel and texture. A spectral clustering stage is applied by using the cues to partition the regions. The Gaussian directional derivative filter is applied to the eigenvector for a global contour. An over segmentation partition is formed from the previous contour and serves as a seed for an oriented watershed transformation. By iteratively merging the minimum weight contour, an ultrametric contour map(UCM) is formed. A segmentation can be obtained by thresholding from an hierarchical level of the UCM.

3.3.3 Semantic segmentation

The semantic segmentation spreads the problem of segmentation to a larger number of semantic labels. It aims to label each pixel with it's corresponding semantic label and segments out multi areas with semantic meaning in one shot.

Recently, convolutional networks have started to achieve a better performance rate in recognition task and have also been deployed for semantic segmentation. For example, to retain the spatial information for the segmentation in the network, fully convolutional network (FCN) [54], must be converted to fully connected layers for classification into 1×1 fully convolutional layers as illustrated in Fig. 3.10. The VGG net has three fully connected layers that produce a 1000 dimensional vector of the objectness probability. After being replaced by the fully convolutional layers, the network output which is a heatmap(the output for the 10×10 grid). Each grid contains 1000 dimensional probabilities that stand for the objectness probability in this grid. To get an image-sized result, the heatmap should be upsampled with factor f, which is similar to convolution with a stride of $1/f$. Therefore the FCN implements an upsampling through "backward convolution/Deconvolution". The FCN network has been used in the 6 to introduce the high level prediction into formulation.

In order to refine the spatial precision illustrated in Fig. 3.10, the output from higher



Source: http://people.eecs.berkeley.edu/~jonlong/long_shelhamer_fcn.pdf

Figure 3.9: Illustration of conversion from CNN to fully convolutional network

and upsampled intermediate layers are combined together.

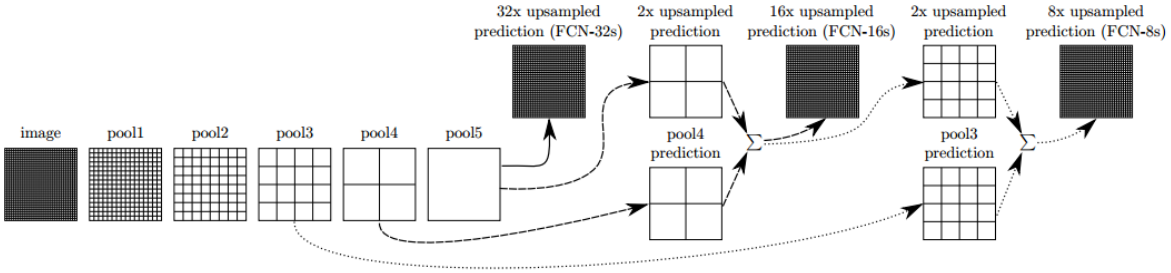


Figure 3. Our DAG nets learn to combine coarse, high layer information with fine, low layer information. Layers are shown as grids that reveal relative spatial coarseness. Only pooling and prediction layers are shown; intermediate convolution layers (including our converted fully connected layers) are omitted. Solid line (FCN-32s): Our single-stream net, described in Section 4.1, upsamples stride 32 predictions back to pixels in a single step. Dashed line (FCN-16s): Combining predictions from both the final layer and the `pool4` layer, at stride 16, lets our net predict finer details, while retaining high-level semantic information. Dotted line (FCN-8s): Additional predictions from `pool3`, at stride 8, provide further precision.

Figure 3.10: Illustration of DAG nets

As illutrated in Fig. 3.11 the DeepLab [47] algorithm the image is first fed into the

Deep convolutional network to obtain a coarse score map. Then this map is upsampled by a Bi-linear interpolation and used as the unary term in a dense CRF [39] for post-processing to predict precisely the contour. Unlike FCN, the upsample is improved by using 'a trous algorithm' by introducing zeros to increase the length.

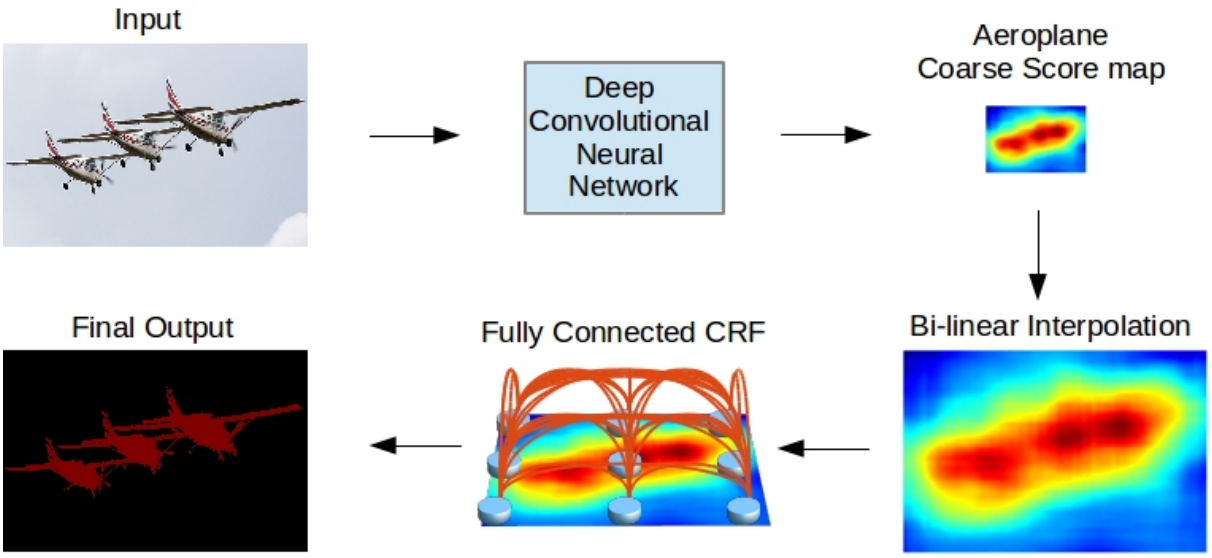


Figure 3.11: Illustration of DAG nets

The authors in [85] propose to formulate the inference steps in [39] by CNN operators. The repeated iterations are formulated in an RNN and now achieve better results on most classes of Pascal VOC [21] database. They present their next paper [53], and the smoothness term is modeled by using a locally convolutional layer. [60] has proposed a symmetric network built with the convolution, deconvolution, pooling and unpooling layers. The unpooling layer adds a switch variable to reconstruct the reduced heatmap from the original activation.

While the networks suffer from the contour localization, domain transform [11] takes the edge map and proposes a layer that recursively applies edge aware filtering across the image to produce the final segmentation scores. In [3], the authors propose the higher order potential to add into the CRF formulation: A detection potential that labels consistency in the detected object's foreground, and a superpixel label consistency energy forces the labels into the superpixel therefore making them identical.

Since the feature in the last layer seems to be too coarse to localize the contour precisely, [29] proposes the use of information gathered in the previous layers. The hypercolumn is the activation mechanism for all the neurons related to this pixel. To include the spatial information and neighbourhood similarity, a coarse grid of classifiers are trained and interpolated between them for the final prediction. For computational efficiency, the network computes each feature map divided by blocks, and then upsamples the scores and outputs the sum for the final segmentation.

RNN(Recurrent Neural Networks) is another powerful network tool, making use of a sequence of information by maintaining a hidden state. The RNN network repeats the same calculation by using a hidden state to store the memory, and pass it onto the next iteration. The memory mechanism can help to minimise the loss. Long Short-Term Memory(LSTM) [31] is a recurrent neural network containing the gates allows for writing, forgetting and updating memory. Globally, LSTM has successfully applied to the data sequence, such as handwriting recognition and text translation. In addition, it has been deployed using in image processing tasks. [31] proposes the grid LSTM to deal with the multidimensional grid. A hidden state and a memory are computed separately in each dimension. Then the hidden vectors are concatenated for the next iteration. The author in [44] combines local and global information into hidden cells illustrated in Fig. 3.12. The network appends the LSTM network to the convolutional feature map to iteratively improve the feature map. In each iteration the local hidden cells are collected by eight LSTM from different directions and the global hidden map is obtained by the max-pooling from nine global cells.

3.3.4 Instance Segmentation

The instance segmentation collectively formulates the semantic image segmentation problem and the object detection problem, and segments out individual object instances. So the instance segmentation can produce a segmentation in each instance. [27] proposes a area-based model by using the Region-of-Interest pooling and aggregating the outputs by using proposals imposed on each pixel. Another attempt in [28] directly maps the predicted bounding box's label to the superpixels and a final classification improves the

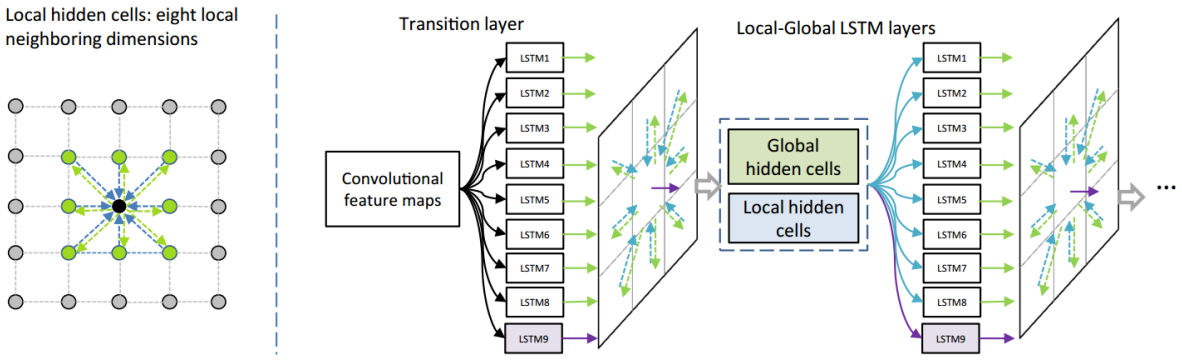


Figure 3.12: Illustration of LG-LSTM

result.

[64] proposes a new end-to-end paradigm that learns to segment out instances sequentially using RNN. They aim to segment out the object one by one by using a spatial memory. Another end-to-end region based model is introduced in [7], a region-to-pixel layer is used to convert region-level prediction to pixel level.

[45] is attempting to improve and refine the region proposal; they propose to use the segmentation result, moving into a recurrent network that combines the proposal refinement sub-network with an instance level segmentation sub-network.

3.4 Fashion application

3.4.1 Clothing retrieval

Many recent research efforts focusing on fashion images deal with a quite different use case, that of meta search engines federating and comparing several online shops. These efforts focus on improving existing search engines to help users find products that match their preferences while preserving the "browsing" aspect [14]. Online shops sometimes provide image tags for common visual attributes, such as color or pattern in Fig. 3.14. Thus some methods are based on the attribute recognition by using image features, and predict the attribute's presence by the binary linear SVMs. But they usually form a proprietary, heterogeneous and non standardized vocabulary, typically too small to characterize the visual diversity of desired clothing [61; 16]. Moreover, in many cases

users look for characteristics expressed through very subjective concepts to describe a style, a brand or a specific design. For this reason, recent research focused on the development of detection, recognition and search of fashion items based on visual characteristics [14; 43]. In [50], the authors approach the street to shop retreival by learning the discrepancies between the two scenario, illustrated in Fig. 3.13. The offline process learn the reconstruction coefficient by minimization of reconstruction error. Then the online retrieval process, the nearest neighbour search in OS dataset by the new feature representation.

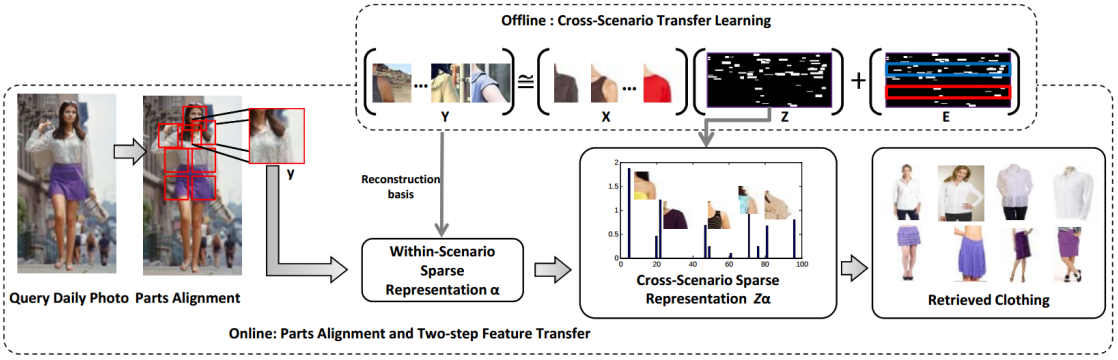


Figure 3.13: Illustration of street-to-shop clothing retrieval system(Image from [50])

In [75], the style retrieval system uses the Siamese CNN network to transform features into a latent space. The outfits can be generated by choosing the nearest item in the style space.

Figure 3.14: Illustration of attributes(Image from [16])

Meanwhile, convolutional networks have also been used in fashion retrieval. Hadi et al. [38] employ three steps for street-to-shop retrieval, including a CNN for feature extraction, and a deep similarity learning network for the street and shop domains. Firstly, a whole image retrieval search for a set of the shop image by the FC6 feature. To filter

out the background influence, an object proposal is deployed on shop image and rank the shop item using FC6 feature. The similarity is then finely predicted by the patch mapping network MatchNet illustrated in Fig. 3.15. MatchNet is composed of two parallel deep convolutional network that extracts features from a street image and a shop image, and combine together both features to pass trough three fully connected layers to computes the similarity. The network is trained independently for each category. To achieve fast retrieval in a large scale dataset, [48] extract hashes-code representations learned by a latent layer in the network.

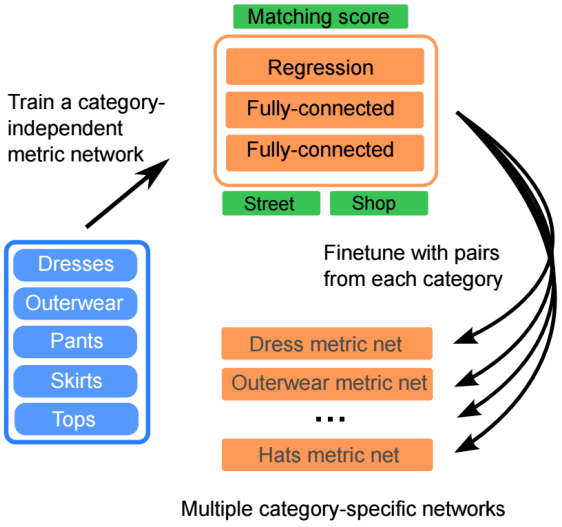


Figure 3.15: Illustration of matchnet in [38]

Another emerging topic is *fashionability*: predict how fashionable a person looks on a photograph. A conditional random field model that reasons about several fashionability factors by using deep network features was put forward in [68]. This prediction can also be used for outfit retrieval to improve the users' outlook.

3.4.2 Clothing attribute selection

One of the approaches models the target item based on attribute selection and high-level classification [15]. For example, in [16] the authors train attribute classifiers on fine-grained clothing styles, formulating image retrieval as a classification problem; they rank

items that contain the same visual attributes as the input, which can be a list of words or an image. A similar idea is explored in [32], where a set of features such as color, texture, SIFT features and object outlines are used to determine similarity scores between pairs of images. In [10], the authors propose to extract low-level features in a pose-adaptive manner and combine complementary features for learning attribute classifiers by employing conditional random fields (CRF) to explore mutual dependencies between the attributes. To narrow the semantic gap between the low-level features of clothing items and the high-level categories, in [50] it is proposed to adopt mid-level clothing attributes (e.g., clothing category, color, pattern) as a bridge. More specifically, the clothing attributes are treated as latent variables in a latent Support Vector Machine (SVM) recommendation model. To address larger fine-grained clothing attributes, [12] introduces a novel, double-path, deep domain adaptation network. This is illustrated in Fig. 3.16. For attribute prediction, they model the data jointly, originating from unconstrained photos, and the images issued from large-scale online shopping stores. An alignment cost function is defined by the post merger of the features from two domains, so as to impose the high level feature, which should therefore show a consistency with labeling. To be able to propose a set of items to specific users, [33] puts forward a functional tensor factorization method to model the interactions between users and fashion items.

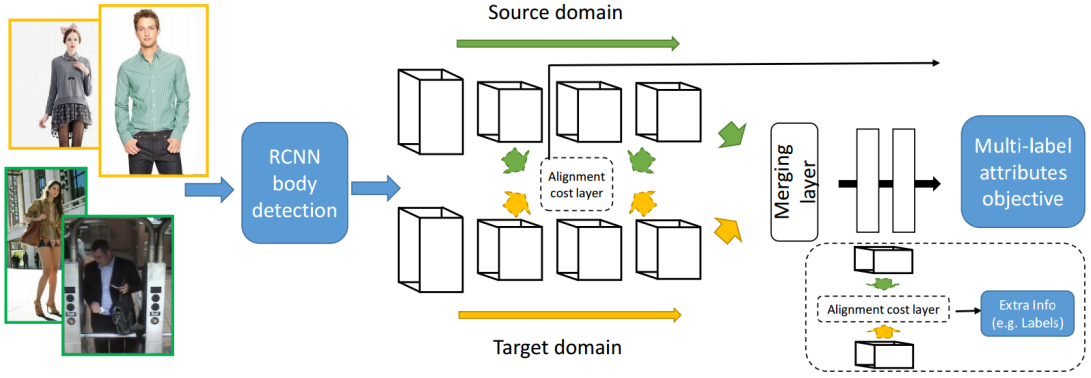


Figure 3.16: Illustration of network in [12]

A second approach consists in using part-based models to compensate for the lack of pose estimation and to model complex interactions in deformable objects [23]. To predict

human occupations, in [70] part-based models are employed to characterize complex details and variable appearances of human clothing on the automatically aligned patches of human body parts, using sparse coding and noise-tolerant capacities. A similar part-based model is proposed in [59], where image patches are described by a mixture of color and texture features. Parts are also employed in [52] to reduce the "feature gap" caused by human pose discrepancy, by employing a graph parsing technique to align human parts for cloth retrieval.

3.4.3 Clothing segmentation

Another approach relies on segmentation and aggregation to select different clothing categories. In [36], articulated pose estimation is followed by an over-segmentation of the relevant human parts. Clustering by appearance then creates a reference frame that facilitates rapid classification without requiring an actual training step. In the training image, the regions are classified by inferring from n nearest neighbors, retrieved by Multi-probe LSH Index. Body joints are incorporated in [34] by estimating their prior distributions and then learning the clothing–joint co-occurrences of different clothing types as part of a conditional random field framework to segment the image into different clothing categories.

A similar idea is proposed in [81], where a CRF is formulated by inter-object and inter-attribute compatibility. A clothing feature is proposed for describing clothing items that combine color histogram, texture histogram, body joint relative location and so on. The labels are obtained with an approximate MAP assignment by using belief propagation. Simo-Serra et al. [67] takes another route to formulate a CRF by taking into account the increase in the interaction with garment's prior knowledge; the pose that the 2D body position takes is coordinated with the limb segments.

The framework in [79] is based on the clothing parsing in Fig. 4.1. The framework firstly retrieve similar images from database by clothing specific feature. Then using three parsing models to tag the image: global prediction by a global clothing model; Local prediction of clothing likelihood learned on-the-fly model from retrieved examples; parse mask predictions from transferred retrieved examples' superpixel to the query image.

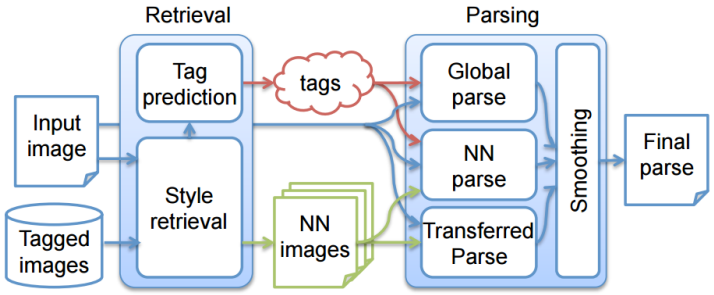


Figure 3.17: Illustration of PaperDoll

Face detection is used in [82] to locate and track human faces in surveillance videos, then clothing is extracted by Voronoi partitioning to select seeds for region growing. For the video applications, [51] use SIFT Flow and superpixel matching to build correspondences across frames and exploit the cross-frame contexts to enhance human pose estimation. Also for human parsing and pose estimation, [18] proposed an unified framework to formulate the problem jointly via a tailored And-Or graph.

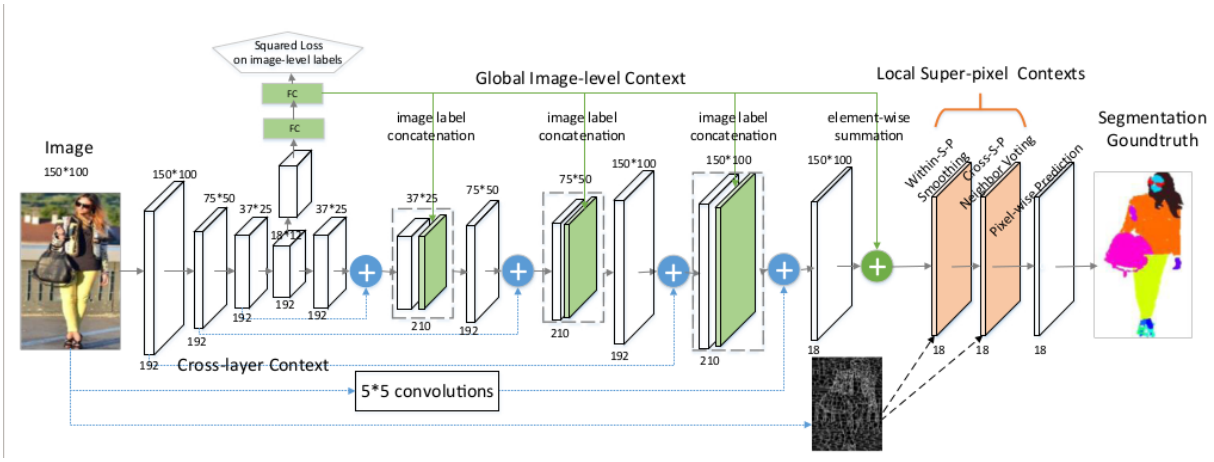


Figure 3.18: Illustration of network human parsing in [46](image from [46])

A network dedicated for human parsing is presented in [46]. As illustrated in Fig. 3.18, the network integrates the multiple types of information, such as inter layer context, global context and superpixel context. The images are first passed through convolutional layers and then gradually upsampled by adding the global image level context, predicted by the last feature map. At the same time, the feature maps are fused to predict the image

edge. Finally, the results are smoothed out by superpixels.

3.5 Dataset

The images are a key factor for the machine learning method, in particular, for the deep learning method, a large and clean dataset can significantly improve the learning results. We present the object detection and fashion datasets used in this thesis.

3.5.1 Object localization and detection dataset

INRIA Person dataset

The database is introduced in [13] especially for the person detection. The images are collected from movie DVDs and personal cameras. The "Motion Training Set 1" contains 2781 human samples, while the "Motion Test Set 1" and "Motion Test Set 2" contain respectively the size of the images are 1704 and 2700. The size of images is normalized to 64×128 pixels.

Imagenet

This dataset is a benchmark for object localization/detection and scene classification. A Large Scale Visual Recognition Challenge (ILSVRC) is held every year. The dataset ILSVRC 2012 is annotated with 1000 object categories, with 150,000 photographs for the validation and the test, and 1.2 million images for training.

Pascal Voc

The Pascal Voc provides a standard dataset for object class recognition evaluation. In the Pascal Voc 2012 there are 11,530 training and validation images with 27,450 regions of interest and 6,929 segmentations; and the images are annotated by only 20 labels.

3.5.2 Fashion dataset

Fashionista

In [79], they introduced a novel dataset for training and evaluating clothing extraction algorithms. The images are collected from Chictopia.com and segmented into superpixels. Each superpixel is annotated by the workers from Amazon Mechanical Turk. The Fashionista dataset contains 685 annotated samples with 53 clothing items. The 14 body joint points are annotated for person detector training as well.

Fashion Icon

Fashion Icon in [51] consists of 1,082 images with 18 categories and with a resolution of 600$times$400. Unlike others, the images contain multiple humans, and there are more challenging images with diverse poses.

Color-Fashion Dataset

The Colorful fashion Parsing Database(CFPD) in [49] consists of 2,628 images. To increase the annotation efficiency, the images are first over-segmented into approximately 400 patches. Each patch is labelled with 13 color tags and 23 category tags.

Daily Photos

The DP dataset introduced in [19], contains 2500 high resolution images. The images are annotated at pixelwise level.

ART

The ART dataset is a fusion of the Fashionista, CFPD and Daily photos datasets. The annotation is unified by the merging of 18 labels. The authors introduced a new dataset called Human Parsing in the Wild(HPW) dataset with 1,833 challenging images with the same labels. The dataset is then enriched by including the "Chictopia 10k" dataset, which has 10,000 real-world human pictures with pixel-level annotations.

Clothing Co-Parsing Dataset

The CCP dataset contains high-resolution fashion photos with 59 tags. All images come with image level annotations, and more than 1000 images have pixel-level annotations.

Figure 3.19: Illustration of image annotation by labelme tools by marking the foreground(red) and background(blue), the result overlaps on the image.

Rich Picture Dataset

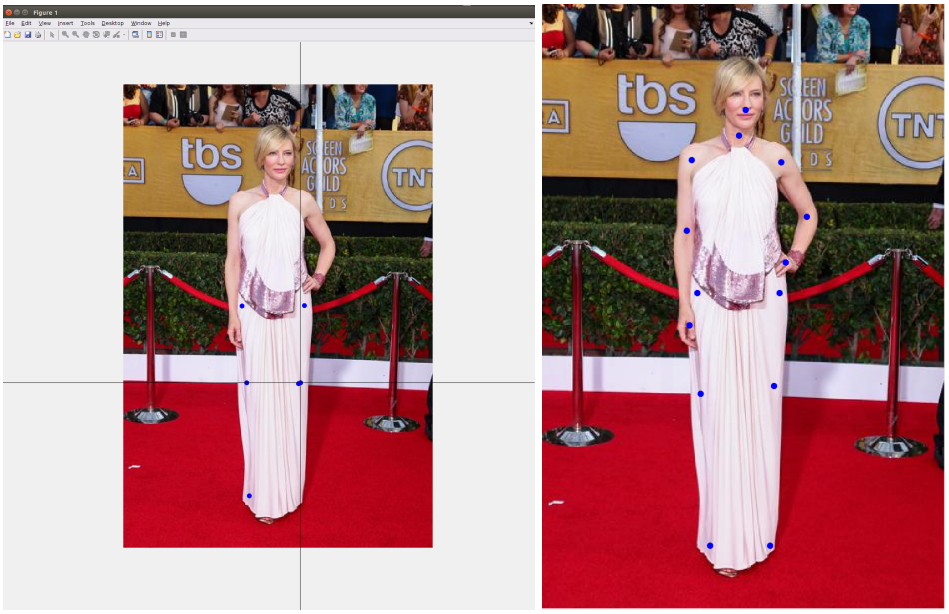
The RichPicture database contains 1000 images with pixel-level annotations. The images are collected by using the request like "fashion/ street/girls" in the google.fr and bing.fr. For our research, we select good quality pictures and only the front of model is shown. We then eliminate the duplicates that have the same image size. The script, which has resizing and compression obstacles, is added, therefore deleting once more the duplicates from the stage containing the element of human intervention.

After communicating with company, we finally decided on ten objects : Boots, Jeans, Shirts, T-shirts, Coats, Vests, Sweaters, Short dresses, Mid-length dresses, Long dresses.

The items chosen have huge selling potential. Since the shoes are generally too small to propose meaningfull results, we then decided to work on boots which are large enough to achieve a good retrieval description. Dresses are a particularly important fashion item, which needs more investment. To achieve a better performance, we then further include more types of dresses into the dataset: Long dresses, Mid-length dresses and Short dresses.

The LabelMe annotation interface is illustrated in 3.19. On the left side of the interface we can choose the foreground/background brush to mark the foreground or background with scribbles. Afterwards, the tool proposes a segmentation that overlaps with an original image. If the segmentation exceeds the boundaries of object, we can mark the space ouside the object with the background brush. Otherwise, we can mark it with the foreground brush. The procedure iterates until a satisfatory result is obtained.

In order to train the human detector, we've developed an annotation tool in Matlab for key points. The interface is shown in Fig. 3.20(a), the users are asked to click on the 14 points in a fixed order. The annotation result is shown in Fig. 3.20(b) and the coordinates are stored.



(a) (b)

Figure 3.20: Annotation of key points by Matlab tools : (a) click on the key points; (b) The 14 key points.

Chapter 4

Classification-Driven Active Contour for Dress Segmentation

4.1 Introduction

In this chapter, our proposal tackles the problem of *precise* segmentation of the object of interest from the background (foreground separation, see Fig. 4.1(b)). It's a difficult problem without user interaction and without using an extensive training database. Indeed, to propose meaningful results in terms of high-level expectations (such as product style or design) we need to achieve a good description of the visual appearance, which is much better if the object is segmented to eliminate the effect of mixing with the background and to include the shape outlined in the description.

Unlike other methods, which have been previously presented, we focus on the precise segmentation for just one object, therefore we've chosen the 'dress' class to start our work. The dress class is a challenging class, in Fig. 6.3, we present some difficult cases that often occur in our use case: (a) A person nearby with the same color of dress, (b) colourful dress, (c) highly textured dress, (d) The bottom of dress roll out the red carpet, (e) semi-transparent dress, (f) other confusing accessories like the scarf.

Facing the above difficulties, the state of art methods fail to produce a satisfying result. As an example, we show in Fig. 4.1(c), the results obtained on the left image by using the state-of-the art method from [79]. Even though this method is capable of multi-label segmentation, the result of the segmentation is inadequate for a fine description of the

dress. Instead, we aim to obtain a finer description like the one in Fig. 4.1(b) (which, incidentally, is produced by the method described in this chapter).

(a) (b) (c)

Figure 4.1: We aim to produce a precise segmentation of the fashion items as in (b). Recent state of the art [79] produces the result in right figure (c), which is insufficient to provide a precise description of the object (dress in this case).

In this chapter, we introduce our first work by using a straightforward way of extraction. Since the dress is worn by humans, a human detector can be applied to first focus on the ROI area. By considering that each box has its special property, we want to explore several prior information obtained to support the extraction: 1. the context of each box: the clothing contents contained in the neighboring boxes are generally similar; 2. the probability map: capture the intra box pixel spatial distribution, i.e. take the box assigned as "left shoulder" for example: the lower right part is more likely to have clothing pixel, the other corner is less likely; 3. the curvature of the box: Reasons for the prior curvature, according to the nature of the box-related human part, i.e. the curvature near the legs is usually small, while the bottom of the dress is relatively larger when the ends roll out. The main idea of our method is to federate the above mentioned information for extraction and then finally segmentation.

To achieve our goal, we combine a person detector with a two stage SVM classification 3.1.1 to achieve a rough estimation of the clothing contour (separation of the object from the background). The result is then used to seed an improved active contour, fine-tuned by prior specific knowledge of the object structure.

(a) (b) (c)

(d) (e) (f)

Figure 4.2: We present some examples to illustrate the diversity of dress items: (a) A person nearby with the same color of dress; (b) Colorful dress; (c) Highly textured dress; (d) The bottom of dress roll out the red carpet; (e) Semi-transparent dress; (f) Other confusing accessories like the scarf.

The novelty of our method is twofold: first, we combine local information in the image with a learning prior to guiding the segmentation (which allows us to predict the contour in the presence of small occlusions, rather than just following local contours) and second, we inject specific knowledge of the object (local curvature) in the energy function that guides the convergence of the active contour, which helps disambiguate the object's contour in the cluttered background. The procedure requires a training database for the person detector and for the foreground detection stage, together with the collection of prior knowledge regarding the object to be segmented. To keep the presentation uncluttered, in this paper we focus on dresses, a challenging class to segment, because of the complexity of the object and the variety of the environment in real images. However, our

procedure can be adapted to any fashion clothing item by following a similar developmental approach.

Evaluation tests performed on a database of 200 manually segmented images show very promising results. We provide examples of successful segmentation, analyze difficult cases, and also evaluate quantitatively each component. Because existing methods, to our knowledge, do not precisely segment fashion items at this moment, we compare our method to the GrabCut [65] foreground extraction method, which is well-known in our community, is frequently used as a baseline case in many research works and has an open source implementation [1].

The rest of the chapter is organized as follows: In Sec. 4.2.1 we give an overview of our proposal, followed by a detailed presentation of each component: person detector in Sec. 5.1.1, SVM-based detection in Sec. 4.2.3 and active contour in Sec. 5.1.4. After the experimental validation is presented in Sec. 6.3, we conclude the paper in Sec. 6.4 by a discussion of the main points and perspectives for further extensions.

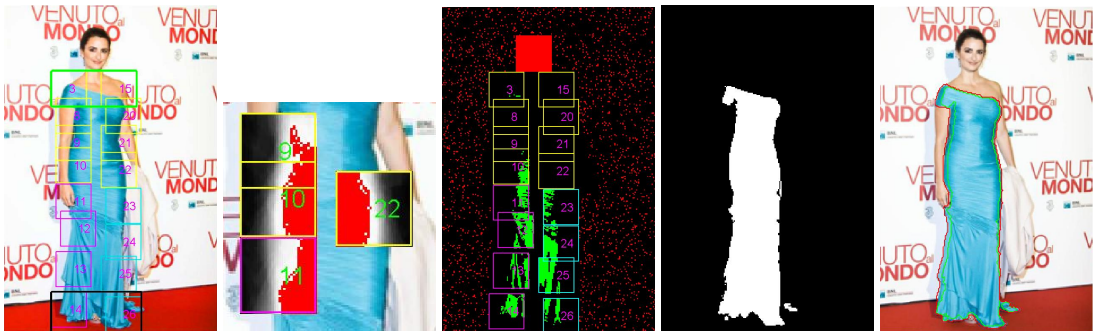
4.2 Our proposal

Detecting dresses in images is a difficult problem, because the object is deformable, can be composed of a large variety of materials, textures and patterns, and shows great differences in style and design inside this class. Also, it can be photographed in very different and complex backgrounds.

4.2.1 Overview of the approach

Since we aim to find as precisely as possible the contour of the dress, a direct approach is deemed to fail because of the aforementioned difficulties. Instead, we adopt a three stage method, each step preparing the following one as follows:
1. **Person detector.** The person detector has been introduced in 3.2.2. We first train a person detector on a manually annotated database to find the regions of the image most likely to contain the contour of the object. We use the articulated human detection model

[1] http://opencv.org/



(a) (b) (c) (d) (e)

Figure 4.3: Different stages of our approach: (a) output of the person detector overlaps on the image with the numbered bounding box, (b) output of the one class SVM, (c) and (d) input and output of the two class SVM, red for negative samples and green for positive samples; (e) final result after first (green) and second (red) active contour steps

with a flexible mixture of parts presented in [83], which works well for both person detection and pose estimation, and has been tested with success in several other fashion-related works (see Sec. 3). The output of the person detector is a set of parts (rectangular boxes) centered on the body joints and oriented correctly.

2. **Coarse foreground detection.** We employ the training data to estimate a probability map that each pixel inside a box belongs to the object. The map is used to seed a one-class SVM estimating the support of the distribution of positive examples (pixels that belong to the object). Then, a two-class SVM is trained to improve the (coarse) detection of pixels belonging to the object, taking as negative examples random background pixels.

3. **Active contour.** The result of the two-class SVM is used as input to a two step active contour procedure that produces the final segmentation. We include several specific terms in the energy function that guides the active contour: the first term uses the results of the learning stage to push the contour towards the SVM separation frontier (i.e., the contour of the object according to the SVM), the second term takes into account the local curvature weighted by the location on the object. This ensures a good balance between the local pixel behavior and the information injected by learning, producing good results in most situations.

4.2.2 Training strategy

To train the person detector, we manually annotate a database of 100 images of dresses. Each person in a training image is annotated with 14 joint points by marking the articulations and main body parts. The 26 key points can be obtained by intersection on the 14 joint points. Another difficulty during annotation is the legs are usually covered by the long dresses, so the lower parts are placed on the edges of the dress rather than on the legs. This not only improves detection accuracy, but also hints to the location of the contours.

The human detector model is presented in 3.2.2. We've kept the tree structure with 27 part model. Because according to the paper, the 27 part model which contains midway points of 14 joint position, has better the performance and better coverage. The t_i denotes for the spanned types for each part. Since our images don't suffer from large deformation, we've choose $t_i = 6$, a reasonable choice : large enough to learn diverse part types and not to increase the computation time.

In RichPicture database, we only have positive examples. Later we collect negative samples from the INRIA person dataset. To deepen our analysis, the data is augmented by sampling randomly from negative training photos from a fixed set of windows. Illustrated in 4.4, the positive samples are augmented 5 times by flipping left right and rotating by -15, -7.5, 7.5, 15 degrees. In our case, the person doesn't have a large amount ofmovement, so it's reasonable to choose a slight rotation. In order to reduce the training times, all the training images are resized to 300 pixels in height by keeping the aspect ratio.

Given an image I, the person detector provides a set of 26 body parts, each part being a square region re-sized to 40×40 pixels. Each part has a symmetric part with respect to the vertical axis and corresponds to a body part or articulation. In ref Fig. 6.4(a) we see the output of the person detector on an unannotated image. Note how boxes slightly overlap at each end most of the time. To reduce the search space, we take advantage of the fact that the dress contour is located inside the boxes. We close the outline by adding the green (upper) and black (bottom) boxes by including the upper and lower regions to

(a) (b) (c)

(d) (e) (f)

Figure 4.4: Each training image is augmented 5 times by following the transformation process: (a) original images; (b) mirrored image by flipping left right; (c, d, e, f) rotation with -15, -7.5, 7.5, 15 degrees.

make the connection between the left and right body part estimations. These new boxes are computed to fit the internal and external hull of the existing ones. All subsequent processing of each image is done only in the region outlined by the boxes.

Probability map. For each location (pixel) in every box, we compute the probability of occurrence of the dress since the clothes have highly spatial correspondence with human body. This probability map indicates for each box where the dress is more likely to be found. We use this map to harvest positive examples for the next stage (SVM classification, see Sec. 4.2.3). To compute the map, we manually segment the dress in all the training images and then, for each pixel position in each box b, we compute the value:

$$p_b(i,j) = \frac{\sum_{k=1}^{n} \delta(I_{kb}(i,j) \in \text{Dress})}{n}$$

where n is the number of training images and $\delta(I_{kb}(i,j) \in \text{Dress})$ is 1 iff pixel (i,j) from box b in image I_k belongs to the dress. The resulting map is shown in ref Fig. 6.4(b), the brighter one stands for a higher probability ratio. This map supports our hypotheses, the dresses are more likely to appear in the inner body.

4.2.3 Coarse foreground detection

In the second stage, for each box we train a two-class SVM 3.1.1 to separate foreground pixels (dress) from the background by using the prior information given by the probability map and the person detector. Each pixel is described by the RGB coordinates concatenated with other local characteristics as described in Sec. 6.3. In a new unseen image we only have the result of the person detector to start with (the 26 part boxes). Using directly the probability map computed earlier to find positive training examples by thresholding does not yield enough good examples to guarantee correct generalization. Instead, we use the 100 most probable pixels in the box (according to the probability map) to train a one-class SVM that computes the support of the positive examples in the input space. We observed experimentally that a number larger than 100 reduces the generalization ability of the resulting classifier. The main source of problems here is that pixel classification is prone to local instability in cluttered scenes. To counter this, we use the *context of each part* to inject confidence into the decision by allowing the neighboring and

symmetric parts to vote. Concretely, for each part, we build four one-class SVMs: one for the part itself, one for the symmetric part w.r.t. the vertical and two for the lower and upper neighboring parts (see Fig. 6.4(b)). A pixel is considered positive is all four one-class SVMs validate it.

$$f(x_i b_j) = \prod_{b_k \in N(b_j)} (f(x_i M_{b_k})) \times f(x_i S_{b_k}) \tag{4.1}$$

Where the prediction for pixel x_i in box b_j, by using the decision function $f(.)$ mapped between 0 and 1 by the logistic function from the model learnt in the neighbor box in $N(b_j)$ and sysmetric box $S(b_j)$.

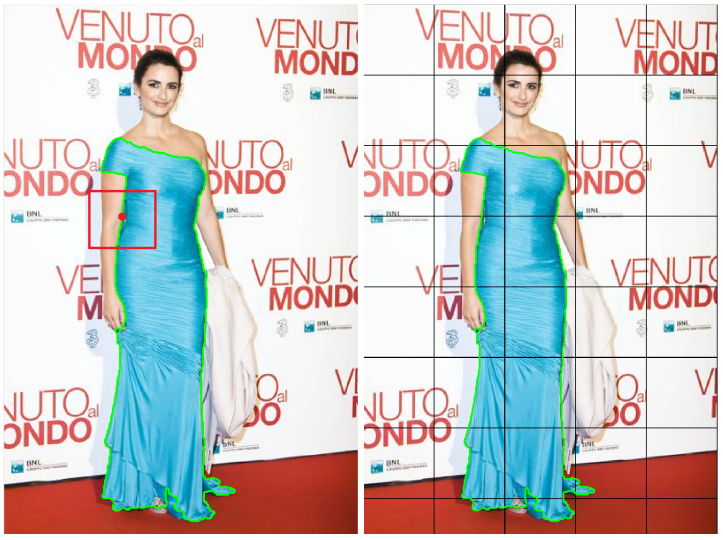
Merely using the output of the previous one-class classification fails to isolate properly the dress because pixels from the background and from the dress may have similar descriptors. We thus take some of the background pixels as negative examples for a two class SVM. More precisely, the pixels predicted by the one-class classifier as belonging to the dress are considered as positive examples. We randomly take as negative examples an equal number of pixels from the background (outside the envelope of all the parts) to obtain a balanced training problem. Since head rarely have dress pixels, we also include the head part in the negative examples and leads to include more skin and hair pixels. In Fig. 6.4(c) we illustrate the training set for the two-class SVM and in Fig. 6.4(d) we show the result of prediction. It can be seen that the cloud of positive predictions outlines the dress quite closely, meaning the learning problem is well posed.

4.2.4 Active Contour

The score of the two-class SVM on a pixel indicates the likelihood of the dress presence. To get the final dress segmentation, we use the active contour (AC) introduced in 3.3.2, a model that can segment objects whose boundaries are not necessarily well-supported by gradient information. The AC minimizes an energy that drives the evolution of the active contour towards the desired boundary.

$$F(c_1, c_2, C) = \mu * \text{Length}(C) + \nu * \text{Area}(\text{inside}(C)) + \\ + \lambda_1 \int_{inside(C)} |u_0(x, y) - c_1|^2 + \\ + \lambda_2 \int_{outside(C)} |u_0(x, y) - c_2|^2 \tag{4.2}$$

where C is the evolving curve, c_1 and c_2 are the average pixel gray level values $u(x, y)$ inside and respectively outside the contour C. The curvature term is controlled by μ and the fitting terms by λ_1 and λ_2.



(a) (b)

Figure 4.5: Illustration of two modes of local mean value computation(a) For each point in the contour, compute on a fixed size window around point, (b) Approximate mean value in image grid for all the inside points.

This model would fail to solve our problem if we applied it directly without modifications. Because this model works well on the simple uniform grey images. So in the following paragraph, we introduce our method which tackles the problem demanded for the images of diverse colors and textures.

To achieve a faster convergence in the final result, a two-step procedure is employed. A fast convergence first converges at a higher speed in the direction of the ROI. Added to this, a finer convergence, which converges slowly toward the desired region with pre-

cision.

A first AC is initialized with the external envelope of the parts produced by the person detector and converges rapidly to an approximation of the desired boundary. It takes as input the binary image produced by the two-class SVM and requires a small μ allowing for strong curvatures.

Then, a second AC is used to converge to the real contour. It is initialized with the contour produced by the previous step and takes the gray level image as an input.

The parameter of mu is still configurable and is directly related to the local curvature. This step uses prior information of the object: the curvatures in specific regions are expected to be different. In some images when the occlusion issue occurs, a large μ value in those regions leads to a smoother result. On the contrary, for example the shoulder, the lower part of the dress and the elbow, may manifest strong curvature, a small μ in these regions forces the edge to correctly follow the contour. For the other parts, we take a medium value for μ.

The mean values c_1 and c_2 are computed usually on the entire image. Because of the large variability of the background in real images, this values is meaningless in a local context in our case and we replace them by the average values calculated in a local window of size 40×40 pixels around each contour pixel as illustrate in 4.5(a).The computation is heavy since the computation is repeated on each points of contour. Therefore we can approximate this mean value by dividing the grids in each image, and compute a mean value locally for each grid in 4.5(b). This act keeps the local mean and reduce significantly the computation.

In the previous step, a grossly estimation of clothing distribution is obtained. To reinforce the role of the SVM-based classification on the position of the AC, we include a new term in the energy function (Eq. 5.1) that pushes the contour towards the SVM separation frontier:

$$F_{svm}(C) = \eta \int_{on(C)} |f_{svm}(x, y)|^2 \tag{4.3}$$

where f_{svm} is the two-class SVM decision function mapped between 0 and 1 by the logistic function $1/(1 + e^{-|\cdot|})$. Theoretically, by using this term, the contour separates the

region in the image and respect the decisions boundary at the same time. Since the contour evolves in the local context, this term reinforce the SVM prediction to produce a better convergence with respect to the global prediction.

4.3 Experimental results

At the time when we started to work, we found no databases dedicated to evaluating cloth extraction/segmentation from natural images that could be used with our framework, i.e. which has enough number of dresses (our object of interest) to make training feasible. The closest we found is Fashionista [78; 80] which is build to test accuracy of multi-label assignment to pixels, but their method (Paper Doll Parsing) trained on this database did not perform very well for extracting long dresses (see Fig. 4.1 and Fig. 4.6 for some examples).

(a) (b) (c)

Figure 4.6: Paper Doll [80] is inadequate for the precise segmentation needed by our use case: (a) original image, (b) our method obtains a satisfy result, (c) Paper Doll fails to segment the dress and miss detection.

Instead, we evaluate our method on a database of 200 manually segmented dress images, half of which are used for training and the other half for testing. This is enough for preliminary testing of the method, but of course a larger database is needed for full validation, also including other fashion objects. We plan to do this in an extension of the present work.

For the SVM prediction stage, we describe each pixel by its RGB values concatenated

with the x and y derivatives. To correctly separate the dress from the background, especially when the two have similar colors, we further concatenate with the frequency distribution of the power spectral density computed in a patch of size 8×8 pixels around the pixel. This is a well-known texture descriptor [15]. Concerning the SVM, we used the LibSVM implementation [9] with $C = 100$ and the Gaussian kernel with scale $\gamma = 0.1$, parameters obtained by cross-validation on the training data.

In regard to the computational efficiency of our method, the time needed to extract the object contour from an image of size 800×600 pixels in of the order of 5 seconds on an average PC. This could likely be improved by a factor of 5 to 10 by parallel computation and code optimization. The However, in our application scenario this is not needed, because the extraction is not real-time, so we did not pursue further this direction.

Quantitative evaluation: We compare the segmentation produced by our method to the one provided by a human. As performance measures we use the average rates of true positive (TP), true negative (TN), false positive (FP) and false negative (FN) pixels, traditionally used for classifiers. To show the impact of each component, in Table 6.2 we compare the segmentation obtained by the original AC [8], the classification obtained by the SVM with and without the original AC and our method (SVM classification followed by the two step active contour with energy function enhanced by curvature compensation and SVM regularization terms). We see that both the active contour alone and the SVM alone produce few false positives, but a very high rate of FN, i.e. tend to miss-classify dress regions as background. The full method achieves a more adequate behaviour in most cases: rather low error rates because of the curvature and learning compensation terms in the active contour stage.

Method	TP	TN	FP	FN
SVM	65.22	95.42	4.58	34.78
Original AC	79.41	93.64	6.36	20.59
Original AC + SVM	84.2	89.91	10.09	15.80
Full Method	87.06	90.3	9.7	12.94
GrabCut	93.72	52.29	47.71	6.28

Table 4.1: Evaluation of different configurations by statistical measure for our method and comparison with GrabCut.

Figure 4.7: More results on different types of background: (a, c, e, g, i, k) original images, (b, d, f, h, j, l) associated segmentation result.

In a second set of experiments we compare our method with GrabCut [65], a well-known method for foreground extraction (see Table 6.2). To outline the object for GrabCut we used the external envelope of the parts identified by the person detector. GrabCut has a good rate of true positives (i.e. good classification of dress regions) but a too high FP rate, i.e. tendency to classify background as dress. This is likely due to the fact that GrabCut performs better for extracting objects on an uniform background, while out database

contains many cases where the background is complex or dress and background are visually similar (see Fig 6.6).

We also evaluate the segmentation by means of the Jaccard (Intersection-Over-Union) score, score more frequently used in papers dealing with image segmentation:

$$S = \frac{\text{Surface}(Y \cap Y')}{\text{Surface}(Y \cup Y')}$$

Also for this measure, our method (79.7%) largely outperform GrabCut (64.86%).

(a) (b) (c)

(d) (e) (f)

Figure 4.8: Qualitative evaluation: (a, b, c) original images; (d, e, f) associated segmentation results.

Qualitative evaluation. In this part we illustrate the preceding conclusions with some examples taken from the test database. A first example of successful segmentation was already shown in Fig. 6.4(e). In Fig. 6.6 we present some other difficult examples of successful segmentation: (a,b) semi-transparent dress against skin color and (c,d) black dress against cluttered background with people dressed in black. In Fig. 6.6(e, f) we see a case of less successful segmentation: a red dress on a red carpet. Here, the pixel descriptors are not sufficiently discriminant to separate the foreground. More examples are presented in

Fig. 4.7 to illustrate the behavior of the method with respect to different types of background.

(a) Original image (b) Our method (c) GrabCut

Figure 4.9: Visual comparison with GrabCut : Our method segment perfectly the dress, while the result of GrabCut includes background and skin pixels.

In Fig. 4.9 we show a visual comparison with GrabCut: as hinted by the quantitative results, GrabCut includes many background pixels and skin pixels in the foreground. This probably due to lack of context and occurs on all the database, explaining the high FP rate in Table 6.2.

4.4 Conclusion

In this chapter, we presented a novel method for dress segmentation that injects specific knowledge about the object into a three stage detection model combining learning and active contours. Each component is also been tested separately and proves to be efficient. By comparing with the GrabCut, the method shows promising result. For the future work, we'll enrich the dataset by inclusion of more training images, both for the person detector and for the probability map, together with the use of more sophisticated pixel descriptors should allow to further improve the results. In an extension of this work, we intend to evaluate and adapt the method for other types of clothing items and, by replacing the person detector with a more general detector, like deformable parts model [23], to other fashion objects. The results in this chapter have been published in [1].

Chapter 5

A Gloabal-Local approach to fashion items extraction

In the previous chapter, with precision, we successfully segment dress class in the image by using information gathered early. In this chapter we extend our previous segmentation to a broader selection of items. The importance of precise segmentation has been stated above.

The previous method needs the use of prior information to define the parameters. Though parameters can be found by cross-validation, it's still time-consuming and the best configuration can't always be found. In this chapter, we expect to propose a method to automatically analyze clothing-specific prior information gathered from training the image and projecting the information, helping to segment new images. Since clothes have a large spatial dependence, among the prior information gained, we've chosen this information to project the clothes occupation, which could be easily inferred by the coordinates of human joints.

In order to adapt to diverse deformation, we've prototyped the templates into eight groups and then select the most similar templates in each cluster to better guide segmentation. Since a good segmentation will attach well to the edge, the quality of the segmentation is evaluated by the contour fitness. At last, the best segmentation is selected by this criteria.

We propose a Global-Local approach, based on the idea that a local search is likely to converge with a better fit if the initial state is harmonious with the expected global

appearance of the object. And to select a final segmentation by a global measuring.

Our method is validated on the Fashionista database [79] [1]and on a new database of manually segmented images that we specifically built to test fashion objects extraction and that we make available to the community. In ref Fig. 6.3, we illustrate our new database Rich Picture. This database collect from everyday usecase that include all occasions and varied from all kinds of clothing categories. Our method compares favorably with the well-known Paper Doll [79] clothing parsing and with the recent GrabCut on One Cut [73] generic foreground extraction method. We provide examples of successful segmentation, analyze difficult cases and also quantitatively evaluate each component.

In Sec. 6.2 we describe our proposal, followed by a detailed presentation of each component. After the experimental validation in Sec. 6.3, we conclude the paper with Sec. 6.4 by a discussion of the main points and extension perspectives.

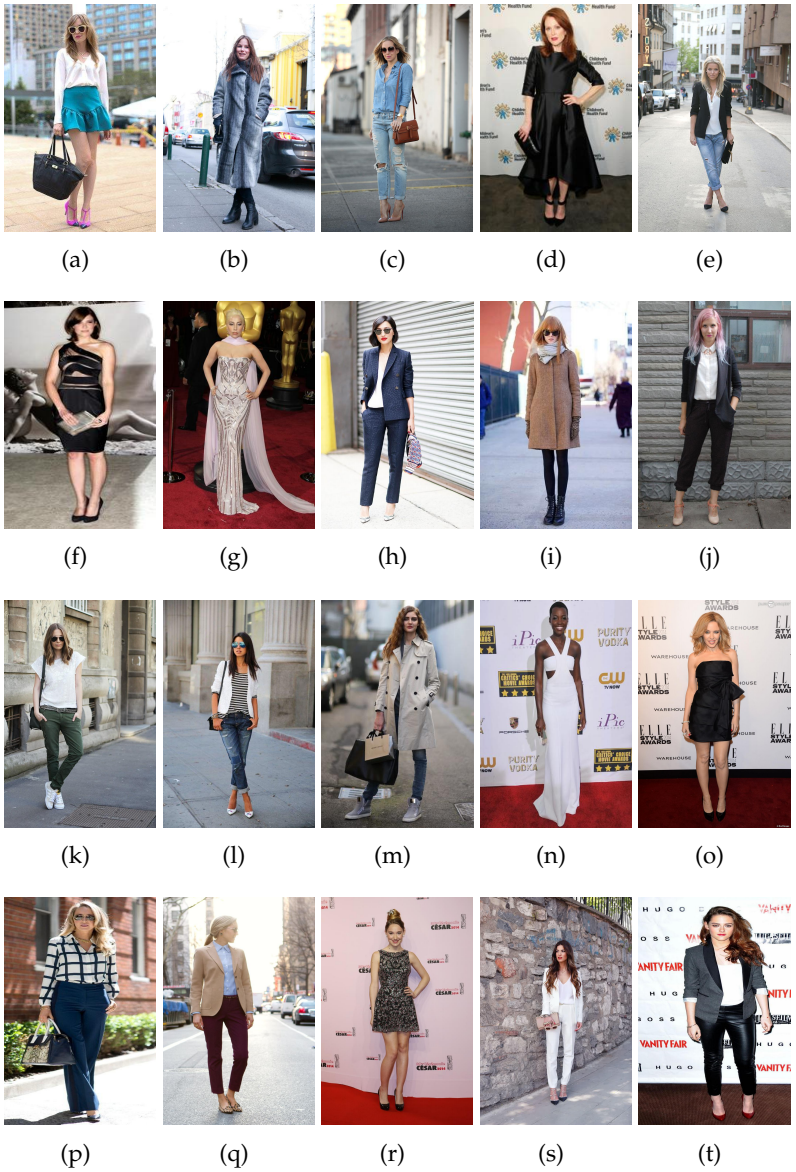
5.1 Our proposal

Detecting clothes in images is a difficult problem because the objects are deformable, have large intra-class diversity and may appear against complex backgrounds. To extract objects under these difficult conditions and without user intervention, methods solely relying on optimizing a local criterion (or pixel classification based on local features) are unlikely to perform well. Some knowledge about the global shape of the class of objects to be extracted is necessary to help a local analysis converge to a correct object boundary. In this paper we use this intuition to develop a framework that takes into account the local/global duality to select the most likely object segmentation.

We investigate here fashion items that are worn by a person. This covers practically most of the situations encountered by users of fashion and/or news web sites, while making possible the use of a person detector to restrict the search regions in the image and to serve as reference for alignment operations.

First, we prepare a set of images containing the object of interest and we manually segment them. These initial object masks (called templates in the following) provide the

[1] http://vision.is.tohoku.ac.jp/~kyamagu/research/paperdoll/



(a) (b) (c) (d) (e)

(f) (g) (h) (i) (j)

(k) (l) (m) (n) (o)

(p) (q) (r) (s) (t)

Figure 5.1: We present some examples to illustrate the diversity of the database RichPicture.

prior knowledge used by the algorithm. Of course, a given manual segmentation will not match exactly the object in an unknown image. We use each segmentation (after a suitable alignment) as a template to initiate an active contour (AC) procedure that will converge closer to the true boundaries of the real object in the current image. We then extract the object with a suitable GrabCut procedure to provide the final segmentation. Thus, at the end we have as many candidate segmentations as hand-made templates. In the final step we choose the best of them according to a criterion that optimizes the coherence of the proposed segmentation with the edges extracted from the image. In the following subsections we detail each of these stages (see also ref Fig. 6.4 for an illustration).

(a) (b) (c) (d) (e) (f) (g)

Figure 5.2: Different stages of our approach: (a) original image, (b) a template overlap with segmentation, (c) output of the person detector, (d) result after the alignment step, (e) result after the active contour step, (f) the GrabCut band , (g) result after the GrabCut step.

To summarize, the main contributions of this chapter are: we introduce a new framework for the extraction of fashion items in web images that combines local and global object characteristics, framework supported by a new active contour that optimizes the gap with respect to the global segmentation model, and by a new measure of fit of the proposed segmentation to the real distribution of the contours. In addition, we prepare a new benchmark database and make it available to the community.

5.1.1 Person detector

For clothing extraction, it is reasonable to first apply a person detector. As in many other studies (*e.g.* [52], [36], [80]), we use the person detector with articulated pose estimation algorithm from 3.2.3 that was extensively tested and proved to be very robust in several other fashion-related works (see Sec. 3). It is based on a deformable model

that sees the object as a combination of parts [83]. The detection score is defined as the fit to parts minus the parts deformation cost. The mixture model not only encodes object structure but also captures spatial relations between part locations and co-occurrence relations between parts. The output of the detector is a set of parts (rectangular boxes) centered on the body joints and oriented correctly. The boxes are used as reference points for alignment by translation and re-scaling in several stages of our proposal (see below).

To train the person detector, we manually annotate a set of 800 images. Each person is annotated with 14 joint points by marking the articulations and main body parts. When the legs are covered by long dresses, the lower parts are placed on the edges of the dress rather than on the legs. This not only improves detection accuracy, but also hints to the location of the contours. ref Fig. 6.4(c) shows the output of the person detector on an unannotated image. Boxes usually slightly cover the limbs and body joints.

5.1.2 Template selection

As we have seen, each initial template can provide a candidate segmentation for a new, unknown image. However, this is redundant and may slow down unnecessarily the procedure. Since we focus on the fashion items that are worn by a person, the number of different poses in which an object may be found is relatively small, and many initial templates are thus quite similar. Intuitively, templates that are alike in shape should also produce similar segmentation masks. To reduce their number, the initial templates are clustered into similar-shape clusters by using the K-Medoid procedure [1]. We employ 8 clusters for each object class, which is a reasonable choice in our case because the number of person poses is not very large. Each resulting cluster is a configuration of deformable objects that share a similarity in pose, viewpoint and clothing shape. The dissimilarity of two object masks is defined by the complement of the Jaccard index:

$$\mathrm{d}(S_1, S_2) = 1 - \frac{\mathrm{Surface}(S_1 \cap S_2)}{\mathrm{Surface}(S_1 \cup S_2)}$$

where S_1 and S_2 are the binary masks of two objects.

Each cluster represents a segmentation configuration and its prototype is used in the next stages of the procedure. However, we do not simply choose the medoid as the

prototype of the cluster, but rather the element in the cluster that is visually closest to the corresponding box parts produced by the person detector on the unknown image. To do so, we apply the object detector on both the unknown image and the template image and we compare the boxes that contain the object in the template with the corresponding ones in the unknown image by using the Euclidean distance. To represent the content of the boxes we first considered HOG features [13] (to favor similar shape content) but finally settled for Caffe features [35] that provide better results. This suggests that mid-level features give better clues to identifying the correct pose of an object compared to local pure shape features. Shape is relevant for comparing the boundaries of two objects but less so when comparing what is inside those boundaries.

Specifically, we use the AlexNet model in 3.2.4 within the Caffe framework [35]. The network was pre-trained on 1.2 million high-resolution images from ImageNet, classified into 1000 classes. To fine-tune the network to our image domain, we replace the last layer by a layer of ten outputs (the number of classes considered here) and then retrain the network on our training database with back-propagation to fine-tune the weights of all the layers. After the fine-tuning, the feature we employ is the vector of responses for layer fc7 (second to last layer) obtained by forward propagation.

To illustrate this step we show in Fig. 5.3 the medoids (centers) of the 8 clusters obtained for three classes of our benchmark database. We notice the diversity in poses, scale and topology. For example, some coats are segmented into several disjoint parts, some have openings and some jeans are covered by a vest. This diversity encodes the clothing deformation for the next segmentation task.

5.1.3 Template Alignment

The output of the previous stage is a set of segmentation templates (8 in our case) for each object class. They will be used one by one to initiate an active contour process. But they first need to be aligned into the unknown image at the right site and with the correct angle and scale. A first attempt is by simply placing the template by reference points in shoulder or hips and adjusting the scale just by the people height. But this method isn't invariant to pose transformation since the key point is not often well detected. We thus

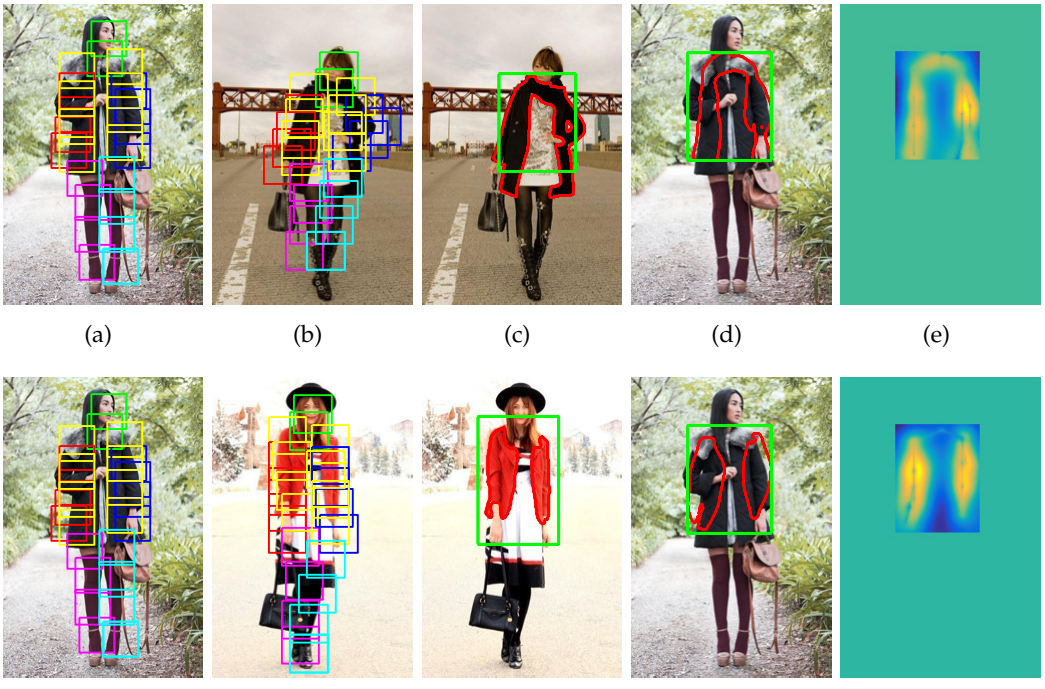
Figure 5.3: Medoids of the 8 clusters of template overlap with the segmentations for three classes: jeans (top), long dress (middle) and coat (bottom).

propose an SVM alignment technique based on the observation that the person detector places the boxes centered on the body joints. Thus, the line joining adjacent boxes represents the body limbs. Since the clothing's spatial distribution highly depends on the pose of human body, and thus on limb placement, we use the vector of distances from a pixel to the limbs as a feature vector to learn a pixel-level SVM classifier that predicts if a pixel belongs to the object.

The samples for learning is used by the pixels in the bounding box where the clothing appears in the template. The positive samples are the pixels on the manual segmentation, while the negative sample take the rest pixels. Learning is performed on the template image and prediction on the unknown image. Pixels predicted as positives form the mask whose envelope serves as initialization for the active contour step. The SVM uses a Gaussian kernel with a scale parameter $\sigma = 1$ found through experiments. To reduce the computation time, the learning and prediction is just process on a small region. The region is delimited by the envelop of the boxes that appeared the segmented region of the template image.

As illustrated in the Fig. 5.4, the same unknown image Fig. 5.4(a, f) is in alignment

with two templates Fig. 5.4(b, g). The predicted spatial distribution is shown in Fig. 5.4(e, j), the brighter color indicates stronger projection. In addition, this prediction overlaps with the unknown image in 5.4(d, i). We can conclude from the illustration that the spatial distribution is projected on new images; the form of opened jacket is well conserved. Even the proper priority information is retained, for example, the detail in the template Fig. 5.4(c) has a coat with collar around the neck. Afterward this prior information is again projected on the unknown image Fig. 5.4(e), whereas the prediction result on Fig. 5.4(j) doesn't contain this information.



(a) (b) (c) (d) (e)

(f) (g) (h) (i) (j)

Figure 5.4: SVM template alignment : (a,f) images and outputs of person detector, (b,g) templates and outputs of person detector, (c, h) template overlapped with the segmentation of coat, (d,i) the predicted coat occurrence, (e, j) the predicted SVM score.

5.1.4 Active Contour

Once the template is embedded in the image, we use it to initialize an active contour (AC) that should converge to the boundaries of the object. The result is highly dependent on the initial contour, but usually one of the 8 segmentation templates leads to a final

contour that is quite close to the true boundary. The AC is initialized with the aligned segmentation contour produced by the previous step and has as input the gray-level image. We use the AC introduced in [8] because it can segment objects whose boundaries are not necessarily well-supported by gradient information. The AC minimizes an energy defined by contour length, area inside the contour and a fitting term:

$$F(c_1, c_2, C) = \mu \cdot \text{Length}(C) + \nu \cdot \text{Area}(\text{in}(C)) + \lambda_1 \int_{in(C)} |u(x,y) - c_1|^2 + \\ + \lambda_2 \int_{out(C)} |u(x,y) - c_2|^2 \tag{5.1}$$

where C is the current contour, c_1 and c_2 are the average pixel gray-level values $u(x, y)$ inside and respectively outside the contour C. The curvature term is controlled by μ and the fitting terms by λ_1 and λ_2.

A simple AC can not segment the image with prior informations, as it is discussed in previous chapter. The model should be adjusted to the use case.

Firstly, the averages c_1 and c_2 are usually computed on the entire image. Because of the large variability of the background in real images, these values can be meaningless locally. Consequently, in our case we replace them by averages computed in a local window of size 40×40 pixels around each contour pixel.

To reinforce the influence of the global shape of the template on the position of the AC, we include a new term in the energy function (Eq. 5.1) that moderates the tendency to converge too far away from the template:

$$F_t(C) = \eta \int_{on(C)} D_m(x, y) \tag{5.2}$$

$D_m(x, y)$ is the distance between pixel (x, y) and the template. By minimizing this term, the contour will evolve near the template's contour region, where the distance function indicates a smaller value. By including this term, the contour converges with those image regions that best separates the inside from the outside and, at the same time, not spreading too far away from the template contour. This potential preserves the prior

information, which is imposed by the template and simultaneously adapts to the nearby contour.

5.1.5 Segmentation

The contours obtained in the previous step suffer from two implicit problems: (1) only the grey-level information is used by the AC process, and (2) possible alignment errors may affect the result. For solving the problem, it's preferable to add the color information. As for the second problem, it's mostly caused by the essence of alignment constraint, but the alignment algorithm will give a gross location. To correct the alignment, the segmentation can be refined by using the more certain region to introduce more context. To compensate for these problems, an "exclusion band" illustrated in Fig. 6.4(f) between the red contour of constant thickness is defined around the contour produced by the previous step, then the inside region is labelled as "certain foreground" and the outside area as "certain background". A GrabCut algorithm [65] is then initialized by these labels to obtain the final result. GrabCut takes into account the global information of color in the image and will correct the alignment errors within the limits of the defined band.

5.1.6 Object Selection

After obtaining the segmentation proposals initialised from each template, we need to select a single segmentation as the final result. For the selection, we propose a score based on a global measure of fit to the image:

$$F(C) = \frac{\int_{on(C)} D_e(x,y)ds}{\int_{on(C)} ds} \tag{5.3}$$

where $D_e(x,y)$ is the distance from the current pixel to the closest edge detected by [17] and C is the boundary of the segmentation proposal. This score measures the average distance from the segmentation boundary to the closest edges in the image. A small value indicates a good fit to the image. See Table 5.1 for an illustration of this step, the second row shows the image overlapping with the result. We can assume from the figure that the quality of segmentation result is correlated with our proposed measure.

Because poor results are manifesting in the undesired region, when segmentation often pass through the contour, leading to a high score. We can conclude that this measurement is a reasonable choice, in this case a better contour fit implicates a better segmentation(a smaller score).

Edges	2.5133	2.3492	**1.5187**	2.8724	2.8082	2.8112	1.9229	2.7667

Table 5.1: Segmentation selection from the results based on the 8 templates of the class, using the corresponding fit values. The test image is given top left, with the extracted edges shown bottom left. The best score is the smallest (outlined in boldface).

5.2 Experimental results

To assess the performance of the proposed method, we perform two sets of experiments. In the first set, our method is compared to a recent improvement of GrabCut [65] that is the standard approach in generic object extraction, on a novel fashion item benchmark we built. The second set of experiments compares our proposal to the recent PaperDoll [79] fashion item annotation method on the Fashionista database [80]. [2]. The parameters for our method are optimised by cross validation on the training set(devided in two parts: 80% for training and 20% for validation respectively).

[2]http://vision.is.tohoku.ac.jp/~kyamagu/research/paperdoll/

Figure 5.5: Qualitative evaluation: (a, c, e, g, i, k, m, o, q, s, u, w) are original images, (b, d, f, h, j, l, n, p, r, t, v, x) are associated segmentation results. Segmentation for each item: (a, c) for boots, (e, g, m) for dresses, (i, k) for shirts, (o, q, s) for jeans, (u, w) for pull.

5.2.1 RichPicture Database

Since, to our knowledge, at this time there is no public benckmark specifically designed for clothing extraction from fashion images, we introduce a novel dataset called RichPicture, consisting of 1000 images from Google.com and Bing.com. It has 100 images for each of the following fashion items: Boots, Coat, Jeans, Shirt, T-Shirt, Short Dress, Mid Dress, Long Dress, Vest and Sweater. Each target object in each class is manually segmented. To train the person detector (see Sec. 5.1.1), images are also annotated by 14 key points. This database will be made available with the paper and open to external contributions. We shall further extend it with new classes and more images per class.

5.2.2 Comparison with GrabCut in One Cut

In this set of experiments, we compare our proposal to GrabCut in one cut [73], a recent improvement on the well-known GrabCut [65] foreground extraction algorithm, which is frequently used as a baseline method in the literature. GrabCut in One Cut was shown in [73] to have higher effectiveness, is less resource demanding and has an open implementation. These reasons makes it a good candidate as a benchmark baseline. For the purpose of this evaluation, we split each class of our database in 80 images for training (template selection) and 20 images for test.

The segmentation produced by the algorithms is tested against the ground truth obtained by manual segmentation. As performance measure we employ the Jaccard index, traditionally used for segmentation evaluation, and averaged over all the testing images of a class. To outline the object for One Cut we use the external envelope of the relevant parts (the ones that contain parts of the object) identified by the person detector. Table 6.1 shows a class by class synthesis of the results (best results are in boldface).

It can be seen that the proposed method performs significantly better on all the classes except “Shirt” where the scores are equal. While both segmentation methods are automatic (do not require interaction), these results speak in favor of including specific knowledge into the algorithm (by the use of segmentation templates in our case).

Table 5.2: Comparison with the One Cut algorithm. The comparison measure is the Jaccard index.

Class	Boots	Coat	Mid dress	Jeans	Shirt
Our method	**0,54**	**0,74**	**0,84**	**0,78**	**0,77**
One Cut	0,26	0,31	0,54	0,71	**0,77**
Class	T-shirt	Short dress	Long dress	Vest	Pull
Our method	**0,67**	**0,80**	**0,74**	**0,65**	**0,74**
One Cut	0,45	0,47	0,57	0,35	0,36

5.2.3 Comparison with Paper Doll

To our knowledge, there is no published method concerning fashion retrieval that aims to precisely extract entire fashion items from arbitrary images. The closest we could find is the Paper Doll framework, cited above, that in fact attributes label scores to a set of blobs in the image. By taking the union of all the blobs that correspond to a same clothing class, one can extract objects of that class. The authors of Paper Doll also introduced the Fashionista database, used to test annotation algorithms, which we use for this evaluation. Table 6.2 presents the synthesis of the results of Paper Doll *vs.* One Cut *vs.* our method. The object classes we selected for tests are those that correspond to fashion items that are worn by persons (compatible with our method).

For our method, training and template selection are performed on the same part of the database that Paper Doll employed for training. As seen from Table 6.2, on most object classes we compare favorably to Paper Doll. For objects like "Boots", our method needs a more dedicated alignment process, since the object is very small compared to the frame given by the person detector that serves as alignment reference. For objects of the "Jeans" class, the problem also comes from the alignment stage, because the boxes proposed by the person detector are not very well positioned when the legs are crossed. It is necessary to increase the number of training examples with this specific pose.

Table 5.3: Comparison with Paper Doll and One Cut on Fashionista. The comparison measure is the Jaccard index.

Class	Vest	Jeans	Shirt	Boots	Coat	Dress	Skirt	Sweater
Our method	**0,32**	0,72	**0,35**	0,35	**0,56**	**0,52**	**0,62**	**0,52**
Paper Doll	0,19	**0,74**	0,24	**0,44**	0,28	**0,52**	0,52	0,07
One Cut	0,23	0,62	0,29	0,01	0,23	0,33	0,32	0,25

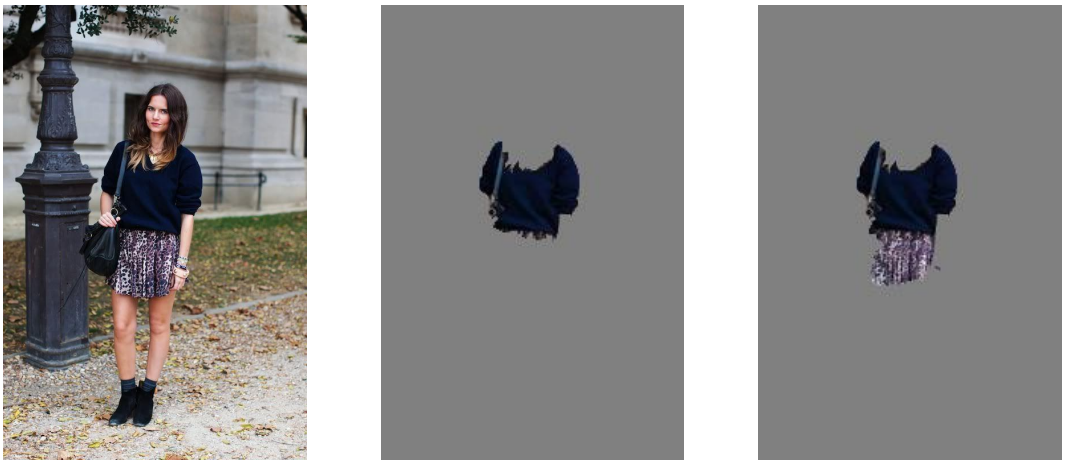


Figure 5.6: Comparison with OneCut: original image (left), our method (middle) and OneCut (right)

5.2.4 Qualitative evaluation

We illustrate here the results of the proposed method with some examples taken from the our test database. First, Table 5.1 shows the final segmentation selection stage, based on the values of fit associated to the results obtained from each of the 8 templates of the class. Visually, the object segmentation in the test image is close to the template.

A first example of successful segmentation was shown in Fig. 6.4(g). In Fig. 6.6 we present other difficult but successful segmentations: (a, c) for small object extraction, (e, g, i) for clothes against confusing or cluttered background, and (k, m, o) for deformed clothes. Fig. 6.6 also shows examples where the segmentation is not perfect: in (r) the extracted object includes some hair and in (t) also part of the leggings. These inclusions probably occur here because the energy term we introduced in the active contour encourages the contour to stay close to the global shape of the segmentation template.

A visual comparison with One Cut is shown in Fig. 6.5. As hinted by the quantitative results, One Cut includes larger parts of external objects, mainly due to the lack of prior shape information. This occurs on most of the images in the database, explaining the significantly lower performance of One Cut in Table 6.1 and in Table 6.2.

5.3 Conclusion

We proposed a novel framework for extracting deformable clothing objects from web images. Our proposal combines a three stage global-local approach that injects specific knowledge about the object by using segmentation templates to guide an active contour process. Comparisons with GrabCut in One Cut and with Paper Doll show that the proposed approach is promising and performs favorably compared to generic or more dedicated object extractors. The method can easily be extended to new object classes at relatively low cost, *i.e.* by manually segmenting objects from these classes. We intend to continue adding new object classes to the RichPicture database. Also, a better alignment solution should benefit the proposed method, as well as the annotation of more images for training the person detector with other class-specific poses. The results in this chapter have been published in [2].

Chapter 6

Fully convolutional network with superpixel parsing for fashion Web image segmentation

6.1 Introduction

6.1.1 Outline of our approach

In this chapter we tackle the same problem: to *precisely* segment the object of interest, then we extend the single-label segmentation to multi-label segmentation, see Fig. 6.2. Compared with the single object segmentation, the semantic segmentation can take into account the label occurrence and the label prediction. However this is a difficult problem, it aggregates the tasks of object detection, object localization, and object segmentation.

As the deep learning has shown the success in a variety of fields, such as Fully Convolutional Network 3.3.3 (denoted by FCN in the following) has achieved the state of art of semantic segmentation. In general, the FCN produces a good object location with a rough segmentation result. Likewise most networks, the FCN still suffers from contour localization. Since the low level information has contour structure that has been abandoned in pooling layers, we propose a post processing to take advantage of the rough segmentation from a higher level and therefore correcting the contour localization by using the lower level information.

Among the lower level information (such as gradient, texture, superpixel, contour),

Figure 6.1: To illustrate the fitness of contour by the superpixels, the red contour of superpixels is overlapped on the original image.

superpixel seems to almost include all strengths of the above information. We've shown in Fig. 6.1, the superpixels are overlapping in the image, and we can note that generally the superpixels attach well to the image's edge and segment the image into small uniform regions. We can expect the superpixels to mainly improve the contour recovery.

The superpixels may suffer from bad segmentation, and thus inside some superpixels they could have the real contour pass through. To correct some defects, we can use pixelwise smoothness term to further enforce the contour strength between pixels.

To take the informations into account(high level prediction, superpixels structure, pixelwise smoothness), we extend the output of a FCN by optimizing an objective function that iterates over all possible pixel-level labellings. The objective function considers the local adequacy with the class (label) under test, the global mid-scale structure of higher level units (superpixels), as well as the global smoothness of the labelling.

Illustrated the CFPD database in Fig. 6.3, the images are the everyday usecase in all occasions that dispose all kinds of combinaisons of clothing and all kinds clothing categories from all kinds of point of view. We test our method on the fashion image database CFPD [51] and we compare to the unmodified but fine-tuned FCN introduced in [55] that was shown to achieve state-of-the-art results on the Pascal VOC benchmark. We also compare to Co-parsing [51] and to the Paper-Doll framework [79]. We provide examples

(a) Original image

(b) Desired output (our result)

Figure 6.2: Our goal is to produce a precise semantic segmentation (extraction) of the fashion items as in (b) which has been obtained by our method. Each color signifies a semantic clothing label.

of successful segmentation, analyze difficult cases and evaluate each component of our framework. The rest of the chapter is organized as follows. In Sec. 6.2 we give an outline of our proposal, followed by a detailed description of each component. An experimental validation including both quantitative and qualitative results is then presented in Sec. 6.3. We conclude the chapter in Sec. 6.4 and provide perspectives for future work.

6.2 Our proposal

Clothes segmentation is a difficult problem that hasn't been solved yet. Because the objects are deformable, have large intraclass diversity and may appear against complex backgrounds. To extract objects under such difficult conditions and without user intervention, methods solely relying on optimizing a local criterion (or pixel classification based on local features) are unlikely to perform well. Some knowledge about the global objectness to be extracted is necessary to help a local analysis of a correct object boundary. In this chapter we use this intuition to develop a framework that takes into account the higher/lower level dual information to segment the image.

As baseline we employ the output of the fully convolutional network [55] that was

Figure 6.3: We present some examples to illustrate the diversity of the database CFPD [51].

shown to perform very well on the Pascal VOC benchmark. For every pixel in a test image, the FCN provides objectness scores for all the labels (object classes) in the training set. The segmentation result is obtained by Softmax that each pixel takes the label that has the highest score.

However, convolutional networks tends to produce a smooth output to preserve the network's generality. This is possibly due to the nature of deconvolution. Thus, the resulting scores can predict the presence and rough position of objects but are less well suited to detect the exact outline of the objects. To address the localization challenge, some approaches use information from several convolution layers to better estimate the object boundaries [55; 20]. Other approaches employ local representations, transforming the localization into a local optimization task [57].

In this chapter we pursue an alternative path, by inserting before the softmax layer a fully connected conditional random field model. This idea has emerged recently [47; 4; 84]. The novelty of our proposal is that our model takes advantage of the middle-scale structure of the image by using superpixel co-localization. This model can perfectly bridge the low level structure and the high level prediction. As we shall see in Fig. 6.2(b), this method helps the network recover object boundaries at a higher level of detail.

Let X be the test image, of size $W \times H$ pixels. For simplicity, we consider the pixels of the image are enumerated in linear fashion, from 1 to $N = W \times H$. The goal is to associate to each pixel one of the C classes (denoted by the labels $a_1, \ldots, a_C$) predicted by our model. Just before the output softmax layer, the FCN produces a vector of L scores for each pixel of the image. The FCN scores for the entire image are given by $F(X)$, the element of $F(X)$ corresponding to pixel i being the L-dimensional vector f_i. A labeling L of the test image consists in attaching a label l_i to each pixel $i \in 1 \ldots N$. The goal is then to maximize a likelihood function $P(L|X, F(X))$ over the space of all labelings L, or equivalently to minimize the corresponding energy:

$$\arg\min_L E(X) = \arg\min_L(-\log(P(L|X, F(X)))) \tag{6.1}$$

The function $E(X)$ contains several terms aimed at preserving the predictive power of the FCN while improving the localization of the contours:

$$E(X) = \sum_{i=1}^{N} \lambda_{l_i}^{(f)} \theta_i^{(f)}(x_i) + \sum_{i=1}^{N} \lambda_{l_i}^{(r)} \theta_i^{(r)}(x_i) + \sum_{i,j=1,i>j}^{N} \theta_{ij}(x_i, x_j) \tag{6.2}$$

Sums are over all the pixels of the image X. The first term, containing $\theta_i^{(f)}(x_i)$, encodes the degree of agreement between the produced output and the FCN scores, which is denoted by the superscript (f) in the equation. If only this term were present, the output would be the same as the one provided by the unmodified FCN. To allow for more flexibility, the terms in the sum are weighted by the parameters $\lambda_{l_i}^{(f)}$, where l_i is the label assigned by L to the pixel i. The best values for these parameters, including $\lambda_{l_i}^{(r)}$ from the second term, are found by using a Nelder-Mead simplex method as described in [41]. The

second term in Eq. 6.2 encodes the agreement between the proposed labelling and the visual descriptions of the middle-scale image content units (superpixels). Finally, the third term is a smoothness measure based on the fact that nearby pixels with similar low-level features are likely to belong to the same class. We now provide a detailed description of each of these terms. See also Fig. 6.4 for an illustration of the results of different stages of our work chain.

6.2.1 Convolutional Term.

The FCN [55] takes as input an image of arbitrary size and produces an output of the same size. The classifier firstly transforms fully connected layers into convolutional layers to output a spatial classification map. To make a dense prediction, a deconvolutional layer unsamples the coarse outputs to pixelwise outputs. It employs a skip architecture by combining the final prediction layer with lower layers with finer strides. The network is initialized by using a pre-trained model learnt on Pascal VOC [21] and then fine-tuned to the dataset employed here following a procedure that is similar to the one described in [5]. The output contains score predictions for each class and each pixel. The output of FCN-8s is shown in Fig. 6.4(c), the items are coarsely segmented and annotated by the correct label. We want to take the advantage of the high precision of label prediction. Therefore the first term in Eq. 6.2 is given by the FCN pixel prediction:

$$\theta_i^{(f)}(x_i|l_i) = -\log f(i, l_i) \tag{6.3}$$

where $f(i, l_i)$ is the FCN output for pixel i and label l_i. By simply minimizing this term, this has the same function as the softmax of the FCN that takes the label of maximal score. Therefore this term can preserve the FCN prediction.

6.2.2 Superpixel generation.

Despite numerous work on the extraction of the superpixels, we use the well-known method from [24], which is also employed by other work on fashion segmentation [51]. The authors addresses the problem of segmenting an image into the prediction for the

boundary evidence D between regions. The predicate is computed by the differences between the inter-region disparity and the inside region disparity. A boundary is more evident if the dissimilarity between the region is larger than dissimilarity within the region.

The difference between region is defined by the minimum boundary strength that connects two regions, and is formulated as follow:

$$Dif(C_1, C_2) = \min_{v_i \in C_1, v_j \in C_2, (v_i, v_j) \in E} w((v_i, v_j)) \tag{6.4}$$

While the internal difference is denoted by the maximum boundary inside the region, and it's defined as below:

$$Int(C) = \max_{e \in MST(C,E)} w(e) \tag{6.5}$$

As claimed previously, the region predicate are calculated by the difference. If the dissimilarity between the regions are larger than the dissimilarity within the region by a threshold, then predict the contour evidence as true:

$$D(C_1, C_2) = \begin{cases} true & if Dif(C_1, C_2) \geq MInt(C_1, C_2) \\ false & \text{otherwise} \end{cases} \tag{6.6}$$

where the minimum internal difference:

$$MInt(C_1, C_2) = \min(Int(C_1) + \rho(C_1), Int(C_2) + \rho(C_2)) \tag{6.7}$$

where $\rho(C) = k/|C|$, After introducing this term, the small regions will be attributed a stronger evidence for boundary, and thus eliminate the isolated small region. For the segmentation algorithm, the two regions are merged according to the predicate evidence in Eq. 6.6.

6.2.3 Region/superpixel prediction.

The second term in Eq. 6.2 encodes the level of agreement between the neighbouring labels and the label of the current pixel. The image is first over-segmented in superpixels,

following the idea that all pixels in a superpixel should be attributed the same label, since the superpixels clustered the small uniform pixel. This is reasonable because objects are in general delimited by physical contours, and are thus obtained as disjoint unions of superpixels. This procedure should thus improve the contour localization, which was one of the weaknesses of the original FCN.

This term proceed the label prediction of the superpixel where the pixel lies in. Superpixels have richer informations, such as color and texture, the prediction by superpixel will certainly be more accurate. To this end, we predict the label by aggregating the prediction from all superpixels. This prediction is weighted by the number of superpixels having this label, this can avoid the larger object from getting larger probability.

For each superpixel we compute a single label, obtained by the Softmax procedure applied on the average of FCN class scores of the pixels in the region, see Fig? 6.4(d). Given a label l_i, let $N(l_i)$ be the set of all superpixels having this label. For a given superpixel S and for each possible label l_i we compute the agreement between the superpixels in $N(l_i)$ and S according to:

$$\phi(l_i|S) = \frac{1}{|N(l_i)|} \sum_{s \in N(l_i)} \frac{1}{1 + |h_S - h_s|^2} \cdot \frac{1}{1 + |p_S - p_s|^2} \tag{6.8}$$

where h_s denotes the low-level feature (see Sec. 6.3) of superpixel s, p_s is the barycenter of this superpixel and $|\cdot|$ the L_2 norm. The first component in Eq. 6.8 is larger for superpixels having similar low-level descriptions, implementing the idea that visually similar superpixels must share the same label. However, this behavior is weighted by the second term that is larger for superpixels that are far away in the image. This filters the effect of similar superpixels that are far from S. The 1 is included in the numerator to protect against numerical instabilities.

If S_i is the superpixel to whom pixel i belongs and l_i is its candidate label, then the second term in Eq. 6.2 is computed using $\theta_i^{(r)}(x_i) = -\log \phi(l_i|S_i)$. To limit even further the influence of superpixels that are too far away from the candidate, we use the superpixels situated inside a circle of a given radius around the candidate. The best radius is likely to depend on the scale of the objects; in our case, by cross-validation over 0.1 to 1

by a step of 0.1, we found a radius of 0.2 of the image size.

6.2.4 Smoothness term.

Compared with the pure output of FCN in Fig. 6.4(c), the output of superpixel parsing Fig. 6.4(d) has a better contour localization, for example the bag is perfectly segmented. However there are a few superpixels badly predicted, for example at the border of coat. This is probably due to bad segmentation that the contour may pass through the superpixels. To correct this imperfection, we include the pixel wise smoothness to correct prediction locally.

The third term in Eq. 6.2 implements a smoothing condition: two pixels are more likely to share the same label if they have similar low-level visual features and are not very distant in the image, corresponding to the idea that objects are localized units in an image. We found that the following formulation, proposed in [39], works well for our purpose:

$$\theta_{ij}(x_i, x_j) = -\log g_{ij}((x_i, x_j))$$

where

$$\theta_{ij}(x_i, x_j) = (1 - \delta_{ij})(w_1 \exp(-\frac{|p_i - p_j|^2}{2\delta_\alpha^2} - \frac{|h_i - h_j|^2}{2\delta_\beta^2}) + w_2 \exp(-\frac{|p_i - p_j|}{2\delta_\gamma^2}))$$

where p_i is the position in the image of pixel i, h_i its visual description (see Sec. 6.3) and δ_{ij} the Kronecker delta. The first part is an appearance kernel inspired by the observation that nearby pixels that are visually similar are likely to be in the same class. The second part is a smoothness kernel that helps removing small isolated regions. The values of $w_{1,2}$ and $\delta_{\alpha,\beta,\gamma}$ are obtained by cross-validation.

6.2.5 Training.

Due to the limit of images available for training, instead of directly learning from the FCN model, we fine tune a FCN model which is pre-trained on the Pascal VOC dataset [21] to learn relevant filter. And then we modify the network structure by changing the output number in network's layer parameters and fine-tuned on the specific clothing dataset considered here. To adapt the weight slightly in each iteration, the fine-tuning

(a) (b) (c) (d) (e) (f)

Figure 6.4: Original image (a), ground truth (b) and the stages of our approach: (c) softmax FCN, (d) softmax with superpixels labeled by mean value of FCN, (e) superpixel parsing by region prediction, (f) final result by solving the maximum probability defined by the FCN output, region prediction and pixel smoothness.

is performed with a lower learning rate of 10^{-14} and a high momentum of 0.99. The network is successively fine tuned for FCN-32s, FCN-16s, and FCN-8s. For FCN-32s we employ 200K iterations, then 100K iterations for FCN-16s and FCN-8s.

The CRF parameters λ_f and λ_r are obtained by optimization of the F_1 score on the validation set. Given the size of the search space ($2 \times L$), the Nelder-Mead simplex method [41] is employed to find the parameter that produces the smallest minus mean F_1 score.

6.3 Experimental results

The proposed method is evaluated on the Colorful Fashion Parsing Data (CFPD), put forward in [51]. This clothing dataset consists of 2,682 images with 23 class labels. We employed the same training and test partitions suggested in [51]. We constitute a validation set consisting of 100 randomly selected images from the initial training set. As visual features for describing the superpixels, after many attempts, we use the concatenation of normalized RGB and HSV 3D histograms, each having 10 bins in single channel.

To assess the performance of the proposed method, we perform two sets of experiments. First, a class-by-class comparison is performed on CFPD with the recent deep network FCN [55] method for semantic segmentation. Then, the global performance of our proposal is compared to one of the Co-parsing [51] fashion item annotation method.

We eventually provide a qualitative evaluation of our proposal.

6.3.1 Class-by-class comparison.

The FCN [55] is used as a baseline method in the recent work on object segmentation. The FCN has significantly improved state-of-the-art results in semantic segmentation and has an open implementation. This makes it a good candidate as a baseline and supports a class-by-class comparison. For the purpose of this evaluation, we also fine-tuned the FCN on our training images and employ the `argmax` output of the FCN-8s net.

In Table 6.1 we show a class-by-class comparison between the proposed method and the FCN (best results are in boldface). As perfomance measure we employ the average F_1 score over pixels, further averaged over all the testing images of each class. It can be seen that the proposed method performs significantly better on all the classes. While both segmentation methods are automatic (do not require any interaction on test images), these results speak in favor of better taking into account local image information into the algorithm. In our case this is achieved by parsing superpixels.

Table 6.1: Class-by-class comparison with the FCN [55] using the average F_1 score on CFPD database.

Class	background	T-shirt	Bag	Belt	Blazer	Blouse	Coat	Dress
FCN	95.38	24.30	29.02	10.09	15.44	23.76	13.32	35.08
Our Method	**96.67**	**28.73**	**34.83**	**12.87**	**18.80**	**26.94**	**16.48**	**39.44**
Class	Face	Hair	Hat	Jeans	Legging	Pants	Scarf	Shoe
FCN	46.03	45.11	17.02	26.99	20.77	24.94	11.34	35.40
Our Method	**49.71**	**54.82**	**21.59**	**31.38**	**24.61**	**28.22**	**13.95**	**43.35**
Class	Shorts	Skin	Skirt	Socks	Stocking	Sunglass	Sweater	–
FCN	38.07	43.48	41.90	9.48	28.17	2.43	11.65	–
Our Method	**46.35**	**51.35**	**48.23**	**10.91**	**34.79**	**2.75**	**13.39**	–

6.3.2 Global comparison.

We also have to compare our proposal to existing methods that were specifically developed for labelling or extracting fashion items from images. Two prominent frameworks are Paper Doll [79] and Co-parsing [51]. To validate our proposed new term, the algorithm without the superpixel term (second term in Eq. 6.2) was also tested (denoted

"CRF w.o. superpixels" in Table 6.2). The authors of Paper Doll had introduced the Fashionista database of 685 images that was used to test annotation algorithms. However, this database only contains 456 training images, which is quite small for fine-tuning the FCN, and the classes are not exactly the same, so we did not evaluate on Fashionista but only on CFPD.

Table 6.2 presents a synthesis of the results obtained on CFPD by Paper Doll [79], Co-parsing [51], FCN [55], CRF w.o. superpixels and by the proposed method. Note that the results for Paper Doll come from [51]. Several performance measures are shown: the accuracy, the foreground accuracy, the average precision, the average recall and the average F_1 score, traditionally used for fashion segmentation evaluation. The measures are averaged over pixels, over all the testing images of each class and over classes. As seen from Table 6.2, the proposed method compares favorably to the other methods according to all the performance measures considered.

Table 6.2: Global comparison of Paper Doll [79], Co-parsing [51], FCN [55] and our method on CFPD database.

Mesure	Accuracy	FG. Accuracy	Avg. Precision	Avg. Recall	Avg. F_1
Paper Doll	82.79	44.08	49.20	32.00	32.66
Co-Parsing	84.7	52.49	42.31	42.31	41.42
FCN	86.09	50.62	47.06	51.13	40.29
CRF w.o. superpixels	86.77	49.87	50.21	49.64	40.45
Our method	**88.69**	**55.69**	**53.40**	**56.93**	**45.91**

6.3.3 Qualitative evaluation.

In Fig. 6.6 we present some difficult but quite successful segmentations: (a, d) for clothes against confusing or cluttered background, (g, j) for deformed clothes (opened jacket) and (m, p) for small object extraction (shoes). Some parts of our results show that the ground truth is not perfect and an automatic segmentation method can do better. Fig. 6.6 also shows examples where the proposed method is not perfect: when comparing to the ground truth, in (s) it failed to detect the sunglasses and in (v) it failed to detect the belt and the skin (neck). This reflects the lower F_1 score in Table 6.2 for sunglasses and belt. Small objects are quite difficult to extract and may require a specific setting,

including *e.g.* a higher penalty during training.

A visual comparison with the results of the FCN is also shown in Fig. 6.5. As hinted by the quantitative results, the FCN leads to an excessive smoothing and the segmented clothes include larger parts of external objects. This occurs on most of the images in the database, explaining the poor performance of FCN in Table 6.1 and Table 6.2.

Figure 6.5: Qualitative comparison with fine tuned FCN: original image (first), ground truth (second), our full method (third) and fine tuned FCN (last).

6.4 Conclusion

To extract clothing objects from web images, we propose to exploit both the output of a Fully Convolutional Neural Network (FCN) used for semantic segmentation and the superpixels obtained from local visual information. By bridging the high-level prediction provided by the deep network and the mid-level image description, the proposed method significantly improves contour localization. The proposed approach is validated by comparisons with the (fine-tuned) FCN alone and to the Co-parsing [51] method, arguably the current state-of-the-art in fashion item extraction.

The results can probably be further improved by the use of more training data and of refined visual features for the superpixels. To better extract small objects we intend to relate the penalty to the relative size of the objects. Also, we believe that some confusion is inevitable between specific classes, *e.g.* legging *vs.* pants or blouse *vs.* sweater,

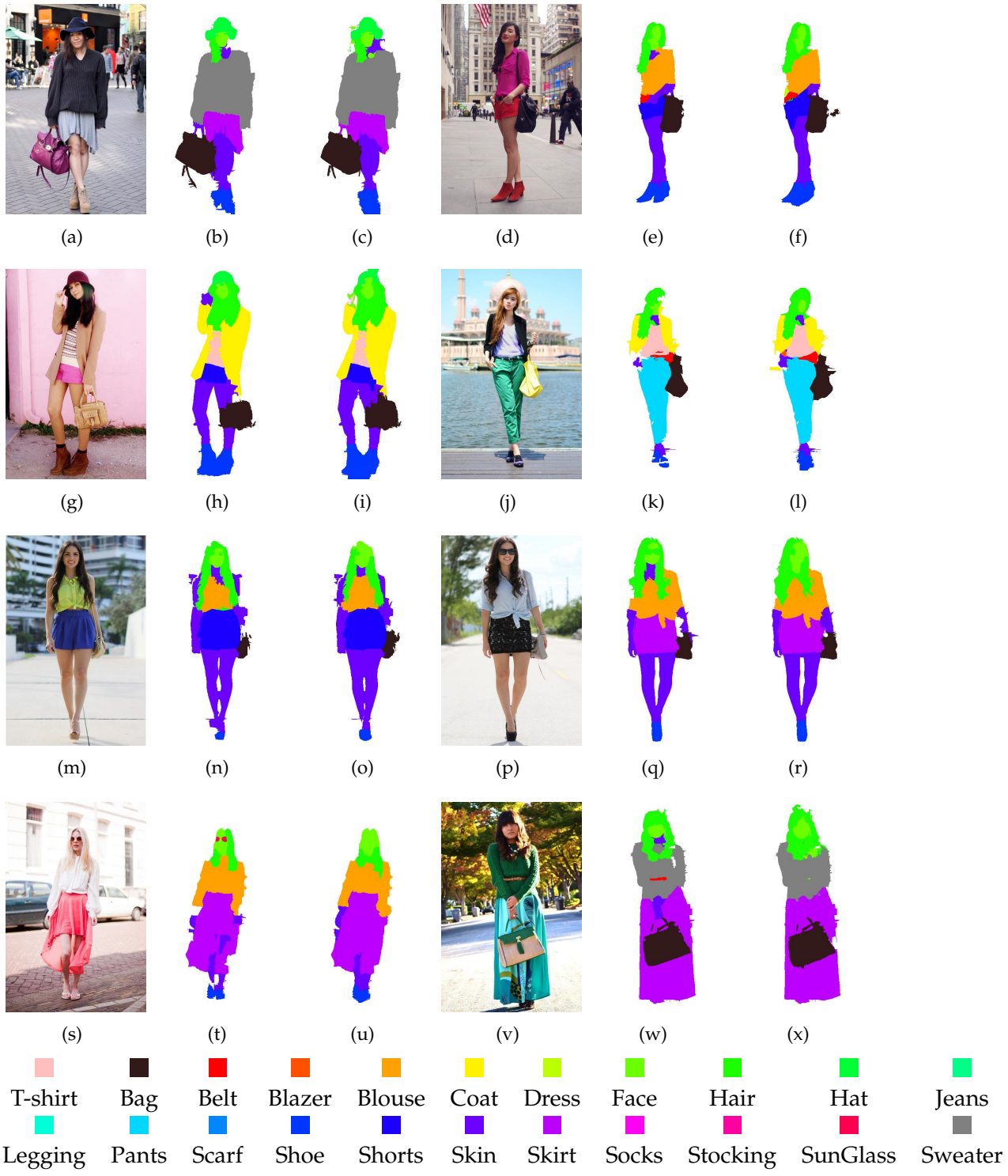


Figure 6.6: Qualitative evaluation: original images (a, d, g, j, m, p, s, v), ground truth (b,e, h, k, n, q, t, w) and associated segmentation results (c, f, i, l, o, r, u, x). The first three rows show the good results,while the last row for bad results. The labels are presented by colors. The white color stands for the background. And the other color's correspondence is illustrated at the bottom of figures by the color squares with the related clothing semantic labels underneath.

but this may not be a problem for subsequent similarity-based clothing retrieval if object segmentation is correct. The proposed method can easily be extended to other classes at relatively low cost, *i.e.* by manually annotating objects from these new classes to train the FCN and the CRF. The results in this chapter have been published in [3].

Chapter 7

Conclusion

The semantic segmentation is the key component in the fashion retrieval system to ensure a meaningful proposal. Although there are many publications on this subject, it's still problematic obtaining a satisfying result. This is because there are numerous difficulties : complex backgrounds, a varying degree of deformed clothing and unconstrained photo taken conditions. Throughout our work, we not only aim to segment as many labels as possible, but also to tackle the problem of precise segmentation leading to a better contour localization. In this thesis, we solve the clothing extraction problem by using three methods. We started by doing a dress segmentation to segment precisely only one object by using information gathered in the beginning of process. Then we extend the work for ten labels by using template-based segmentation. Both methods are easy to be applied on new fashion items. Finally we extend the work for the semantic segmentation that segments multi labels in the same image. The third method can be appliable for new fashion items and other generic objects. Our methods compares favorablely with the state of the art methods.

In this chapter, we first summarize our contributions and give our insights for the future work to improve the semantic segmentation. We then discuss about some possibles ways for the fashion retrieval.

7.1 Summary of contributions

RichPicture Database. Up to now, the RichPicture database is the only database that has pixel wise manual segmentation with higher precision. Most databases use a superpixels based annotation tool, therefore the annotation can be inaccurate due to the bad-segmented superpixels. We annotate the images manually by the LabelMe interactive segmentation tool by using the background and foreground brush.

Dress segmentation. In Chapter 2, we present a novel method for dress segmentation by using clothing-specific knowledge for extraction. Since the clothing object is very deformed, a global model can hardly adapt to all the different kinds of clothing. One of our contributions is the procedure to learn the clothing model which uses the context of the human part. In order to achieve a better contour localization, the learned clothing model and the local contour curvature are injected into the energy function of the active contour. Our method outperforms GrabCut compared to the pixel wise statistic score and the Jaccard score. This method can be quickly extended to new object classes.

Template based segmentation. We proposed in Chapter 5 a global-local approach that guides the segmentation by using object templates, eliminating thus need for object-specific knowledge. The templates are selected at a global image level (large scale) but are used to guide the active contour evolution in a local context. A new contour fitness measure is proposed to select the best candidate. Our method shows promising results compared to OneCut and PaperDoll. This method can be applied to other types of objects by simply including training images.

Superpixel parsing. We proposed a model in Chapter 6 that includes the high level prediction of FCN output and the mid level structure of superpixels to better recover the boundaries of the object. This algorithm outperforms by a margin of more that 10% the fine tuned original FCN Network and the Co-Parsing method [51] in terms of global results but also in a class-by-class comparison on the CFPD database [51]. Our algorithm is generic, thus not limited to the fashion segmentation.

7.2 Perspectives for future research

Here we give several insights about how to improve the results for the item extraction and new design for the fashion retrieval system.

A larger database. A large image database is always a core factor for the machine learning methods. By including more images with more labels, the algorithm correctly analyzes the richer information available. It would be a good idea to re-evaluate our algorithms on a larger fashion database to validate the algorithm in the larger context, when such a database will become available. Another interesting experiment is the evaluation of our work on other databases or the generic database called PascalVOC, however care must be taken for those parts that are specific for fashion items.

The intermediate layer. The intermediate layers in the FCN network contain rich information. By extracting information from intermediate layers, several works have shown very promising results [60; 54]. During the forward propagation, the pooling layers have been reduced in image size, resulting in a smaller heat map. However, the intermediate layers characterize the rich mid level structure of the image and this information has been lost in the pooling layers. In order to recover the boundaries with precision, this information needs to be recovered in the subsequent steps of the analysis. In addition, the segmentation network can be built on top of other networks, such as GoogLeNet [72], ResNet [30] which are now the state of the art for recognition. Another effort can be dedicated to improving the network, so that the architecture is closer to human perception and by improving the learning technique to boost the performance. For example, RNN and LSTM networks are largely used in the field of natural language processing. A promising idea would be to exploit these networks for segmentation by making use of the memory mechanism.

The small object extraction. Small objects have large business potential (ear rings, hand bags, watches, etc.), but the existing methods have poor performance rate for these objects. To address this problem, the post processing step should be improved to specifically segment these objects on the regions of interest and in higher resolution if available.

The fashion retrieval. A fashion retrieval system can significantly benefit from our extraction algorithm. Our algorithm provides important information about the objects, for example the item label, the Region of Interest (ROI), and the clean segmented region. The item label can first reduce the search area into a specific category of products. Also, it enables the intra-category specific search, i.e. different categories have different aspects of similarity (for example, the similarity of t-shirts are mainly relied on the visual content since they have the same shape). Moreover, the ROI and the clean segmented regions will help to describe the visual content by eliminating the influence of background.

Furthermore, when searching fashion items there are several retrieval criterion as well, such as style retrieval, and personal preferences. The style retrieval aims to associate the style concept with the image features in order to propose style alike products. This can be done through the feature transformation from a clothing feature space to a style space. In the style space the similar product should be closely related in style regardless of the product category.

And personal retrieval should be able to receive the feedback and train the retrieval system iteratively with the the feedback from the user [63; 76]. In general, this similarity can be learn by giving the user's browsing order. The previous viewed photo and the next viewed photo can be organized in relevant pairs, and the rest into unrelated pairs. A model can be learnt on the fly that maximizes the distance between the unrelated pairs and minimizes the distance for related pairs.

Bibliography

[1] On statistical data analysis based on the l1-norm and related methods. New York, NY, USA, 1987. Elsevier Science Inc.

[2] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(5):898–916, May 2011.

[3] Anurag Arnab, Sadeep Jayasumana, Shuai Zheng, and Philip H. S. Torr. Higher order potentials in end-to-end trainable conditional random fields. *CoRR*, abs/1511.08119, 2015.

[4] Sean Bell, Paul Upchurch, Noah Snavely, and Kavita Bala. Material recognition in the wild with the materials in context database. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3479 – 3487, 2015.

[5] Yoshua Bengio. Deep learning of representations for unsupervised and transfer learning. In *JMLR W&CP: Proc. Unsupervised and Transfer Learning challenge and workshop*, volume 27, pages 17–36, 2012.

[6] Yuri Y. Boykov and Marie-Pierre Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images. *International Journal of Computer Vision*, 2001.

[7] Holger Caesar, Jasper R. R. Uijlings, and Vittorio Ferrari. Region-based semantic segmentation with end-to-end training. *CoRR*, abs/1607.07671, 2016.

[8] T. F. Chan and L. A. Vese. Active contours without edges. *Trans. Img. Proc.*, 10(2):266–277, February 2001.

[9] Chih-Chung Chang and Chih-Jen Lin. Libsvm: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.*, 2(3):27:1–27:27, May 2011.

[10] Huizhong Chen, Andrew Gallagher, and Bernd Girod. Describing clothing by semantic attributes. In *Proceedings of the 12th European Conference on Computer Vision - Volume Part III*, ECCV'12, pages 609–623, Berlin, Heidelberg, 2012.

[11] Liang-Chieh Chen, Jonathan T. Barron, George Papandreou, Kevin Murphy, and Alan L. Yuille. Semantic image segmentation with task-specific edge detection using cnns and a discriminatively trained domain transform. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[12] Qiang Chen, Junshi Huang, Rogerio Feris, Lisa M. Brown, Jian Dong, and Shuicheng Yan. Deep domain adaptation for describing people based on fine-grained clothing attributes. In *CVPR*, pages 5315–5324, Boston, MA, 2015. IEEE Computer Society.

[13] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1 - Volume 01*, CVPR '05, pages 886–893, Washington, DC, USA, 2005. IEEE Computer Society.

[14] Ritendra Datta, Dhiraj Joshi, Jia Li, and James Z. Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Comput. Surv.*, 40(2):5:1–5:60, May 2008.

[15] Thomas Deselaers, Daniel Keysers, and Hermann Ney. Features for image retrieval: An experimental comparison. *Inf. Retr.*, 11(2):77–107, April 2008.

[16] Wei Di, Catherine Wah, Anurag Bhardwaj, Robinson Piramuthu, and Neel Sundaresan. Style finder: Fine-grained clothing style detection and retrieval. In *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, CVPRW '13, pages 8–13, Washington, DC, USA, 2013. IEEE Computer Society.

[17] Piotr Dollár and C. Lawrence Zitnick. Fast edge detection using structured forests. *PAMI*, 2015.

[18] Jian Dong, Qiang Chen, Xiaohui Shen, Jianchao Yang, and Shuicheng Yan. Towards unified human parsing and pose estimation. In *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '14, pages 843–850, Washington, DC, USA, 2014.

[19] Jian Dong, Qiang Chen, Wei Xia, Zhongyang Huang, and Shuicheng Yan. A deformable mixture parsing model with parselets. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2013.

[20] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. volume abs/1411.4734, 2014.

[21] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The Pascal visual object classes (VOC) challenge. *International Journal of Computer Vision*, 88(2):303–338, jun 2010.

[22] Pedro F. Felzenszwalb, Ross B. Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(9):1627–1645, September 2010.

[23] Pedro F. Felzenszwalb, Ross B. Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(9):1627–1645, September 2010.

[24] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Efficient graph-based image segmentation. *Int. J. Comput. Vision*, 59(2):167–181, September 2004.

[25] Ross Girshick. Fast r-cnn. In *International Conference on Computer Vision (ICCV)*, 2015.

[26] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Computer Vision and Pattern Recognition*, 2014.

[27] Ross B. Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *CoRR*, abs/1311.2524, 2013.

[28] Bharath Hariharan, Pablo Arbelaez, Ross B. Girshick, and Jitendra Malik. Simultaneous detection and segmentation. *CoRR*, abs/1407.1808, 2014.

[29] Bharath Hariharan, Pablo Andrés Arbeláez, Ross B. Girshick, and Jitendra Malik. Hypercolumns for object segmentation and fine-grained localization. *CoRR*, abs/1411.5752, 2014.

[30] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[31] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November 1997.

[32] Esther Hsu, Christie Paz, and Shizhe Shen. Clothing image retrieval for smarter shopping (Stanford project), 2011.

[33] Yang Hu, Xi Yi, and Larry S. Davis. Collaborative fashion recommendation: A functional tensor factorization approach. In *Proceedings of the 23rd ACM International Conference on Multimedia*, MM '15, pages 129–138, New York, NY, USA, 2015. ACM.

[34] Nataraj Jammalamadaka, Ayush Minocha, Digvijay Singh, and C. V. Jawahar. Parsing clothes in unrestricted images. In *British Machine Vision Conference, BMVC 2013, Bristol, UK, September 9-13, 2013*, 2013.

[35] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. pages 675–678, 2014.

[36] Yannis Kalantidis, Lyndon Kennedy, and Li-Jia Li. Getting the look: Clothing recognition and segmentation for automatic product suggestions in everyday photos. In *Proceedings of the 3rd ACM Conference on International Conference on Multimedia Retrieval*, ICMR '13, pages 105–112, New York, NY, USA, 2013. ACM.

[37] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. *INTERNATIONAL JOURNAL OF COMPUTER VISION*, 1(4):321–331, 1988.

[38] M. Hadi Kiapour, Xufeng Han, Svetlana Lazebnik, Alexander C. Berg, and Tamara L. Berg. Where to buy it: Matching street clothing photos in online shops. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV '15, pages 3343–3351, Washington, DC, USA, 2015. IEEE Computer Society.

[39] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *Advances in Neural Information Processing Systems 24*, pages 109–117. Curran Associates, Inc., 2011.

[40] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[41] Jeffrey C. Lagarias, James A. Reeds, Margaret H. Wright, and Paul E. Wright. Convergence properties of the nelder–mead simplex method in low dimensions. *SIAM J. on Optimization*, 9(1):112–147, May 1998.

[42] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Comput.*, 1(4):541–551, December 1989.

[43] Michael S. Lew, Nicu Sebe, Chabane Djeraba, and Ramesh Jain. Content-based multimedia information retrieval: State of the art and challenges. *ACM Trans. Multimedia Comput. Commun. Appl.*, 2(1):1–19, February 2006.

[44] Xiaodan Liang, Xiaohui Shen, Donglai Xiang, Jiashi Feng, Liang Lin, and Shuicheng Yan. Semantic object parsing with local-global long short-term memory. *CoRR*, abs/1511.04510, 2015.

[45] Xiaodan Liang, Yunchao Wei, Xiaohui Shen, Zequn Jie, Jiashi Feng, Liang Lin, and Shuicheng Yan. Reversible recursive instance-level object segmentation. *CoRR*, abs/1511.04517, 2015.

[46] Xiaodan Liang, Chunyan Xu, Xiaohui Shen, Jianchao Yang, Si Liu, Jinhui Tang, Liang Lin, and Shuicheng Yan. Human parsing with contextualized convolutional neural

network. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[47] Chen Liang-Chieh, George Papandreou, Iasonas Kokkinos, kevin murphy, and Alan Yuille. Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs. In *International Conference on Learning Representations*, San Diego, United States, May 2015.

[48] Kevin Lin, Huei-Fang Yang, Kuan-Hsien Liu, Jen-Hao Hsiao, and Chu-Song Chen. Rapid clothing retrieval via deep learning of binary codes and hierarchical search. In *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*, pages 499–502, New York, USA, 2015.

[49] Si Liu, Jiashi Feng, Csaba Domokos, Hui Xu, Junshi Huang, Zhenzhen Hu, and Shuicheng Yan. Fashion parsing with weak color-category labels. *IEEE Trans. Multimedia*, 16:253–265, 2016.

[50] Si Liu, Jiashi Feng, Zheng Song, Tianzhu Zhang, Hanqing Lu, Changsheng Xu, and Shuicheng Yan. Hi, magic closet, tell me what to wear! In *Proceedings of the 20th ACM International Conference on Multimedia*, MM '12, pages 619–628, New York, NY, USA, 2012. ACM.

[51] Si Liu, Xiaodan Liang, Luoqi Liu, Ke Lu, Liang Lin, and Shuicheng Yan. Fashion parsing with video context. In *Proceedings of the 22Nd ACM International Conference on Multimedia*, MM '14, pages 467–476, New York, NY, USA, 2014. ACM.

[52] Si Liu, Zheng Song, Meng Wang, Changsheng Xu, Hanqing Lu, and Shuicheng Yan. Street-to-shop: Cross-scenario clothing retrieval via parts alignment and auxiliary set. In *Proceedings of the 20th ACM International Conference on Multimedia*, MM '12, pages 1335–1336, New York, NY, USA, 2012. ACM.

[53] Ziwei Liu, Xiaoxiao Li, Ping Luo, Chen Change Loy, and Xiaoou Tang. Semantic image segmentation via deep parsing network. *CoRR*, abs/1509.02634, 2015.

[54] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *CoRR*, abs/1411.4038, 2014.

[55] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. volume abs/1411.4038, 2014.

[56] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, November 2004.

[57] Mohammadreza Mostajabi, Payman Yadollahpour, and Gregory Shakhnarovich. Feedforward semantic segmentation with zoom-out features. volume abs/1412.0774, 2014.

[58] Arsalan Mousavian and Jana Kosecka. Deep convolutional features for image based retrieval and scene categorization. *Computer Science*, 2015.

[59] Tam V. Nguyen, Si Liu, Bingbing Ni, Jun Tan, Yong Rui, and Shuicheng Yan. Sense beauty via face, dressing, and/or voice. In *Proceedings of the 20th ACM International Conference on Multimedia*, MM '12, pages 239–248, New York, NY, USA, 2012. ACM.

[60] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. *CoRR*, abs/1505.04366, 2015.

[61] Miriam Redi. *Novel methods for semantic and aesthetic multimedia retrieval*. PhD thesis, Univ. Nice, Sophia Antipolis, 2013.

[62] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2015.

[63] J J Rocchio. *Relevance feedback in information retrieval*. 1971.

[64] Bernardino Romera-Paredes and Philip H. S. Torr. Recurrent instance segmentation. *CoRR*, abs/1511.08250, 2015.

[65] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. "grabcut": Interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph.*, 23(3):309–314, August 2004.

[66] Bernhard Schölkopf, John C. Platt, John C. Shawe-Taylor, Alex J. Smola, and Robert C. Williamson. Estimating the support of a high-dimensional distribution. *Neural Comput.*, 13(7):1443–1471, July 2001.

[67] Edgar Simo-Serra, Sanja Fidler, Francesc Moreno-Noguer, and Raquel Urtasun. A high performance CRF model for clothes parsing. In *Computer Vision - ACCV 2014 - 12th Asian Conference on Computer Vision, Singapore*, pages 64–81, 2014.

[68] Edgar Simo-Serra, Sanja Fidler, Francesc Moreno-Noguer, and Raquel Urtasun. Neuroaesthetics in fashion: Modeling the perception of fashionability. In *CVPR*, 2015.

[69] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[70] Zheng Song, Meng Wang, Xian sheng Hua, and Shuicheng Yan. Predicting occupation via human clothing and contexts. In *Proceedings of the 2011 International Conference on Computer Vision*, ICCV '11, pages 1084–1091, Washington, DC, USA, 2011. IEEE Computer Society.

[71] 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, January 2014.

[72] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. *CoRR*, abs/1409.4842, 2014.

[73] Meng Tang, Lena Gorelick, Olga Veksler, and Yuri Boykov. Grabcut in one cut. In *Proceedings of the 2013 IEEE International Conference on Computer Vision*, ICCV '13, pages 1769–1776, Washington, DC, USA, 2013. IEEE Computer Society.

[74] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders. Selective search for object recognition. *International Journal of Computer Vision*, 104(2):154–171, 2013.

[75] Andreas Veit*, Balazs Kovacs*, Sean Bell, Julian McAuley, Kavita Bala, and Serge Belongie. Learning visual clothing style with heterogeneous dyadic co-occurrences. In *International Conference on Computer Vision (ICCV)*, Santiago, Chile, 2015. *The first two authors contributed equally.

[76] Sean Zhou Xiang and Thomas S. Huang. Relevance feedback in image retrieval: A comprehensive review. *Multimedia Systems*, 8(6):536–544, 2003.

[77] Chenyang Xu and Jerry L. Prince. Gradient vector flow: A new external force for snakes. In *Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)*, CVPR '97, pages 66–, Washington, DC, USA, 1997. IEEE Computer Society.

[78] Kota Yamaguchi. Fashionista image database, http://vision.is.tohoku.ac.jp/~kyamagu/research/clothing_parsing/, Last checked: September 2015.

[79] Kota Yamaguchi, Kiapour Hadi, E.Ortiz Luis, and L. Berg Tamara. Retrieving similar styles to parse clothing. *IEEE TPAMI*, 37:1028–1040, 2015.

[80] Kota Yamaguchi, M. Hadi Kiapour, and Tamara L. Berg. Paper doll parsing: Retrieving similar styles to parse clothing items. In *Proceedings of the 2013 IEEE International Conference on Computer Vision*, ICCV '13, pages 3519–3526, Washington, DC, USA, 2013. IEEE Computer Society.

[81] Kota Yamaguchi, Takayuki Okatani, Kyoko Sudo, Kazuhiko Murasaki, and Yukinobu Taniguchi. Mix and match: Joint model for clothing and attribute recognition. In *Proceedings of the British Machine Vision Conference (BMVC)*, pages 51.1–51.12. BMVA Press, September 2015.

[82] Ming Yang and Kai Yu. Real-time clothing recognition in surveillance videos. In *Proceedings of the 2011 International Conference on Computer Vision*, ICIP '11, pages 2937–2940. IEEE, 2011.

[83] Yi Yang and Deva Ramanan. Articulated human detection with flexible mixtures of parts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(12):2878–2890, December 2013.

[84] Ning Zhang, Jeff Donahue, Ross B. Girshick, and Trevor Darrell. Part-based r-cnns for fine-grained category detection. volume abs/1407.3867, 2014.

[85] Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip H. S. Torr. Conditional random fields as recurrent neural networks. volume abs/1502.03240, 2015.

Appendix A

List of Publications

[1] Lixuan Yang, Helena Rodriguez, Michel Crucianu, and Marin Ferecatu. Classification-driven active contour for dress segmentation. In *Proceedings of the 11th Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2016) - Volume 4: VISAPP, Rome, Italy, February 27-29, 2016.*, pages 22–29, 2016.

[2] Lixuan Yang, Helena Rodriguez, Michel Crucianu, and Marin Ferecatu. A global-local approach to extracting deformable fashion items from web images. In *Advances in Multimedia Information Processing - PCM 2016 - 17th Pacific-Rim Conference on Multimedia, Xi'an, China, September 15-16, 2016, Proceedings, Part II*, pages 1–12, 2016.

[3] Lixuan Yang, Helena Rodriguez, Michel Crucianu, and Marin Ferecatu. Fully convolutional network with superpixel parsing for fashion web image segmentation. In *MultiMedia Modeling - 23rd International Conference, MMM 2017, Reykjavik, Iceland, January 4-6, 2017, Proceedings, Part I*, pages 139–151, 2017.

le cnam

Lixuan YANG

Structuring of image databases for the suggestion of products for online advertising

shopedia

Résumé :
Le sujet de cette thèse est l'extraction et la segmentation des vêtements dans les images fixes en utilisant des techniques de vision par ordinateur et apprentissage statistique, pour la recommandation de manière non intrusive aux utilisateurs des produits similaires provenant d'une base de données de produits. Nous proposons tout d'abord un extracteur d'objets dédié à la segmentation des vêtements qui combine des informations spécifiques locales avec un apprentissage préalable. Un détecteur de personnes localise les sites de l'image ou se trouve l'objet. Ensuite, un processus d'apprentissage intra-image en deux étapes est développé pour séparer les pixels de l'objet du fond. Dans une deuxième étape, nous proposons ensuite un framework pour l'extraction des vêtements qui utilise une procédure d'ajustement globale et locale à trois étapes. Dans notre dernier travail, nous étendons la sortie d'un réseau de neurones FCN (Fully Convolutional Network) pour l'inférence du contexte à partir d'unités locales de contenu (superpixels). De plus, nous proposons une nouvelle base de données, appelée RichPicture, constituée de 1000 images annotées manuellement pour l'extraction de vêtements à partir des images de mode. Nos propositions sont validées sur plusieurs bases de données et se comparent favorablement à plusieurs méthodes état de l'art en ce moment.
Mots clés :
Segmentation des vêtements, Contour Actif, Réseau de neurones, Apprentissage profond

Abstract :
The topic of the thesis is the extraction and segmentation of clothing items from still images using techniques from computer vision, machine learning and image description, in view of suggesting non intrusively to the users similar items from a database of retail products. We firstly propose a dedicated object extractor for dress segmentation by combining local information with a prior learning. We then propose a new framework for extracting general deformable clothing items by using a three stage global-local fitting procedure. In our latest work, we extend the output of a Fully Convolution Neural Network to infer context from local units(superpixels). To achieve this we optimize an energy function, that combines the large scale structure of the image with the local low-level visual descriptions of superpixels, over the space of all possible pixel labellings. In addition, we introduce a novel dataset called RichPicture, consisting of 1000 images for clothing extraction from fashion images. The methods are validated on the public database and compares favorably to the other methods according to all the performance measures considered.
Keywords :
Clothing Segmentation, Active Contour, Fully convolution network, Deep learning