ABORDAGENS DE APRENDIZADO ESTATÍSTICO E PROFUNDO PARA OS PROBLEMAS DE ANOTAÇÃO E DECOMPOSIÇÃO DE PEÇAS DE ROUPAS EM FOTOGRAFIAS DE MODA

KEILLER NOGUEIRA

ABORDAGENS DE APRENDIZADO ESTATÍSTICO E PROFUNDO PARA OS PROBLEMAS DE ANOTAÇÃO E DECOMPOSIÇÃO DE PEÇAS DE ROUPAS EM FOTOGRAFIAS DE MODA

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação do Instituto de Ciências Exatas da Universidade Federal de Minas Gerais como requisito parcial para a obtenção do grau de Mestre em Ciência da Computação.

Orientador: Adriano Alonso Veloso
Coorientador: Jefersson Alex dos Santos

Belo Horizonte
23 de fevereiro de 2015

KEILLER NOGUEIRA

STATISTICAL AND DEEP LEARNING ALGORITHMS FOR ANNOTATING AND PARSING CLOTHING ITEMS IN FASHION PHOTOGRAPHS

Dissertation presented to the Graduate Program in Ciência da Computação of the Universidade Federal de Minas Gerais in partial fulfillment of the requirements for the degree of Master in Ciência da Computação.

Advisor: Adriano Alonso Veloso
Co-Advisor: Jefersson Alex dos Santos

Belo Horizonte
February 23, 2015

Nogueira, Keiller

N778a Abordagens de Aprendizado Estatístico e Profundo para os Problemas de Anotação e Decomposição de Peças de Roupas em Fotografias de Moda / Keiller Nogueira. — Belo Horizonte, 2015
xxvi, 78 f. : il. ; 29cm

Dissertação (mestrado) — Universidade Federal de Minas Gerais
Orientador: Adriano Alonso Veloso
Coorientador: Jefersson Alex dos Santos

1. Computação - Teses. 2. Aprendizado do computador - Teses. 3. Anotação de imagens. I. Orientador. II. Coorientador. III. Título.

CDU 519.6*85 (043)

UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

FOLHA DE APROVAÇÃO

Abordagens de aprendizado estatístico e profundo para os problemas de decomposição e anotação de peças de roupas em fotografias de moda

KEILLER NOGUEIRA

Dissertação defendida e aprovada pela banca examinadora constituída pelos Senhores:

PROF. ADRIANO ALONSO VELOSO - Orientador
Departamento de Ciência da Computação - UFMG

PROF. JEFERSSON ALEX DOS SANTOS - Coorientador
Departamento de ciência da Computação - UFMG

PROF. NIVIO ZIVIANI
Departamento de Ciência da Computação - UFMG

PROF. RENATO ANTÔNIO CELSO FERREIRA
Departamento de Ciência da Computação - UFMG

Belo Horizonte, 23 de fevereiro de 2015.

Aos meus pais,
Vander (in memoriam)
e Maria do Carmo.

Acknowledgments

Agradeço...

À toda minha família, principalmente minha mãe e minha irmã, que me apoiaram durante o caminho.

À Carolina, minha namorada, por estar ao meu lado em todos os momentos deste aprendizado. Mesmo à distância, você me auxiliou bastante, às vezes, sem perceber.

Aos meus orientadores, professores Adriano Veloso e Jefersson dos Santos, pela confiança e ensinamentos.

Aos companheiros que contribuíram para realização deste trabalho, Vitor Andrade, Daniel Balbino e Lucas Assunção, que fizeram os momentos fora do laboratório mais agradáveis além de ajudar com novas ideias. Ao Erico e ao Victor Hugo, por em auxiliarem em partes diferentes mas fundamentais no decorrer do trabalho.

À todos aqueles que direta ou indiretamente contribuíram para o desenvolvimento deste trabalho.

À Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), que financiou o este projeto.

Resumo

Esta dissertação apresenta algoritmos eficientes para anotar e decompor peças de roupas a partir de dados provindos de redes sociais, como Facebook e Instagram. Anotação de roupas pode ser informalmente descrito como reconhecer, o mais precisamente possível, cada peça do traje que aparece em uma imagem. A decomposição, por sua vez, procura além de anotar as peças de roupa, também localizá-las na imagem. Tais tarefas tem papel importante em áreas como vigilância, reconhecimento de ações, busca por pessoas, sistemas de recomendação e de comércio eletrônico. Estes problemas trazem desafios interessantes vinculados à visão computacional e ao reconhecimento de padrões como, por exemplo, distinguir roupas visualmente parecidas mas conceitualmente diferentes, ou identificar um padrão para uma peça específica, já que esta pode ter diferentes cores, formas, texturas e aparência. Inicialmente, o problema de anotação de roupas foi analisado considerando métodos estatísticos de aprendizado de máquina. Para isso, uma extensa avaliação das técnicas de extração de características visuais, incluindo descritores locais e globais, foi feita. Em seguida, formulamos a tarefa de anotação como um problema de classificação multi-modal e multi-rótulo, isto é: (i) conteúdo visual e textual (tags relacionadas às imagens) estão disponíveis para os classificadores, (ii) os classificadores precisam predizer um conjunto de rótulos (um conjunto de peças de roupas) e, (iii) a decisão sobre quais os rótulos devem ser atribuídas à imagem ocorre através de uma função, construída a partir de um conjunto de instâncias. Com esta configuração, propomos duas abordagens: (i) a pontual, chamada neste trabalho de MMCA, que usa uma única imagem como entrada para o classificador, e (ii) a pareada, chamada de M3CA, que usa pares de imagens como entrada para seus classificadores. Comparamos ambos os métodos para definir qual o melhor para o problema em questão. Para cada uma, aplicamos um algoritmo de classificação que usa regras de associação para construir modelos de reconhecimento que combina informações visuais e textuais. Também usamos uma estratégia de minimização de entropia para encontrar quais rótulosdevem ser associados a cada imagem. Realizamos uma avaliação sistemática dos métodos propostos usando fotos coletadas de duas grandes mídias sociais relacionadas à moda, `pose.com` e `chictopia.com`. Os resultados mostram que os métodos propostos fornecem melhorias quando comparados a algoritmos popularmente utilizados que variam entre 20% to 30% em termos de acurácia. Em um segundo momento, analisamos o problema de decomposição de imagens utilizando aprendizado profundo. Propomos um modelo de redes de convolução utilizando uma estratégia multi-escala. Mais especificamente, empregamos diferentes níveis de redes onde cada nível processa imagens de dimensões diferentes, ou seja, a cada nível as imagens são decompostas em pedaços menores, possibilitando assim que a rede classifique pequenos detalhes. No primeiro nível, imagens com maiores dimensões são processadas em uma rede mais robusta. As imagens com entropia baixa já adquirem sua classificação neste nível, enquanto as imagens com entropia alta (não classificadas perfeitamente) são subdivididas e passam para o segundo nível. No terceiro patamar, as imagens não classificadas no segundo nível são novamente subdivididas em pedaços ainda menores e, enfim, classificadas. Ao final, teremos as classes de cada pedaço da imagem, e podemos recompô-la. Para avaliar esta abordagem, utilizamos um conjunto de imagens coletadas do site `chictopia.com`, e nossos experimentos mostram que nossa abordagem fornecem resultados promissores.

Palavras-chave: Aprendizado de Máquina, Anotação de Imagens, Decomposição de Imagens, Descritores Visuais, Dicionários Visuais, Aprendizado Profundo, Redes Neuronais.

Abstract

In this work, we present effective algorithms to automatically annotate and parse clothes from social media data, such as Facebook and Instagram. Clothing annotation can be informally stated as recognizing, as accurately as possible, each garment item that appears in a photo. Clothing parsing, in turn, locates and annotates each garment item in a photo. These tasks play important roles in several areas, including surveillance, action recognition, person search, recommender systems and e-commerce. They also pose interesting challenges for existing vision and recognition algorithms, such as distinguishing between similar but conceptually different types of clothes or identifying a pattern of a specific item, since it can have different colors, shapes, textures and appearance. Initially, the clothing annotation problem was analyzed considering statistical methods of machine learning. For this purpose, we perform an extensive evaluation of the visual feature extraction techniques, including global and local descriptors. Then, we formulate the annotation task as a multi-label and multi-modal classification problem (i) both image and textual content (i.e., tags related to the image) are available for learning classifiers, (ii) the classifiers must predict a set of labels (i.e., a set of garment items), and (iii) the decision on which labels to assign to the query photo comes from instances (or *bag* of instances) that are used to build a function, which separates labels that should be assigned to the query photo, from those that should not be assigned. Using this configuration, we propose two approaches: (i) the pointwise one, called MMCA, which uses a single image as input to the classifiers, and (ii) a multi-instance classification, called M3CA, also known as pairwise approach, that uses pair of images as input to the classifiers. We compare both approaches in order to define the best one for the problem. For both of them, we propose a classification algorithm that employs association rules in order to build a recognition model that combines textual and visual information. We also adopt an entropy-minimization strategy in order to find the best set of labels that should be assigned to the query photo. We conduct a systematic evaluation of the proposed algorithms using everyday photos collected from two major fashion-related social media, namely `pose.com` and `chictopia.com`. Our results show that the proposed approaches provide improvements when compared to popular first choice multi-label, multi-modal, multi-instance algorithms that range from 20% to 30% in terms of accuracy. In a second phase, we analyzed the clothing parsing problem using deep learning. We propose a multi-scale convolutional neural network model. Specifically, we use different network levels where each level processes images with different dimensions, i.e., after every level the images are decomposed into smaller patches, allowing the network to capture minimal details. In the first level, larger images are processed in a robust network. Images with low entropy already get their final class in this level, while the others with high entropy (classification still undefined) are splitted into smaller patches and go to the next one. In the third and last level, images without final classification in the second level are again divided into even smaller patches and, finally, classified. At the end, we have a class associated with each patch of the image and we can recompose it. To evaluate this approach, we use a dataset crawled from `chictopia.com`. Our experiments shows that our proposed approach achieves promising results.

Keywords: Machine Learning, Image Annotation, Image Parsing, Descriptor, Visual Dictionary, Neural Networks, Deep Learning.

List of Figures

List of Tables

Abbreviations

ACC *Auto-Correlogram Color*

BC *Borda Counting Method*

BIC *Border/Interior Pixel Classification*

BN *BossaNova*

BoW *Bag-of-words (or bag-of-visual-words)*

BRIEF *Binary Robust Independent Elementary Features*

BRISK *Binary Robust Invariant Scalable Keypoints*

CCV *Color Coherence Vector*

CM *Condorcet Method*

CNN *Convolutional Neural Networks*

DBA *Dirichlet-Bernoulli Alignment*

EOAC *Edge orientation auto-correlogram*

FREAK *Fast Retina KeyPoint*

GCH *Global Color Histogram*

LAC *Lazy Association Classifier*

LAS *Local Activity Spectrum*

LDA *Latent Dirichlet Allocation*

LCH *Local Color Histogram*

LRN *Local Response Normalization*

M-CNN *Multi-scale Convolutional Neural Networks*

M3CA *Multi-label, Multi-modal, Multi-instance Clothing Annotation*

M3LDA *Multi-label, Multi-modal, Multi-instance Latent Dirichlet Allocation*

MDL *Minimum Description Length*

MIML *Multi-instance and multi-label*

MMCA *Multi-label, Multi-modal Clothing Annotation*

MP *Majority Probability*

MV *Majority Voting*

NN *Neural Network*

OA *Overall Accuracy*

ORB *Oriented FAST and Rotated BRIEF*

PCA *Principal Component Analysis*

QCCH *Quantized Compound Change Histogram*

ReLU *Rectified Linear Unit*

SID *Steerable Pyramid Decomposition*

SIFT *Scale-Invariant Feature Transform*

SURF *Speeded Up Robust Features*

Contents

Chapter 1

Introduction

Computer vision, a computer science area, aims at creating methods and algorithms capable of understanding a scene and its characteristics. In this field, there are methods capable of acquiring, processing, analyzing and understanding images and, in general, high-dimensional data from the real world in order to produce information that may be used in decision making process. Visual recognition, a field of computer vision, is responsible for researching and simulating the human vision system. The main goal of this field is the full understanding of any scene. The basis of this task is made by a tripod composed by annotation, segmentation and classification of an image. The annotation task may be described as recognizing the objects of a scene. The segmentation one locates and annotates the objects in the image. These tasks complement each other and may help the image classification task in its duty. Thus, this dissertation is focused on these two tasks applied for fashion images, i.e., clothing parsing and annotation.

Clothing parsing and annotation play important roles in human pose estimation (Yamaguchi et al. [2012]), action recognition, person search (Weber et al. [2011]; Gallagher and Chen [2008]), surveillance (Yang and Yu [2011]), cloth retrieval (Liu et al. [2012]) and have applications in fashion industry (Yamaguchi et al. [2012]). Considering the last one, applications with fashion images gained a lot of visibility with the increase of social networks and the faster spread of information, since these networks allow their members to express themselves in different ways, by creating and sharing content, making, for example, a new trend more successful or not. A particular way of expression being increasingly adopted is to post photos showing their latest looks and clothes. There are even specific networks for this, such as `pose.com` and `chictopia.com`. These social media channels carry a lot of information that, when analyzed, may help retailers and e-commerce systems to capture new trends helping to define new products and sales. To do so, it would be essential to find out the most popular clothes and in which segment they have been used more. Recommendation systems could also use this information to suggest new clothes based on searches already made or in the wardrobe of the users.

Although interesting, to reach suitable results for clothing applications it is necessary to extract all feasible information from the data, and this is only achieved with images entirely prepared, i.e., images fully annotated or segmented. However, only a very small percentage of images collected from social media have been associated with its clothing content (Kalantidis et al. [2013]), and manual methods are too expensive and maybe impracticable given the total amount of images. So, automatic algorithms appear as a very appealing alternative to reduce costs, but with difficult challenges to overcome. One challenge would be to differ similar types of garment items. For example, discerning a shirt from a coat is a very difficult task since both are very similar. Another one is that individual clothing items display many different appearance characteristics. For example, shirts have a wide range of appearances based on cut, color, material and pattern. Occlusions from other humans or objects, viewing angle and heavy clutter in the background further complicates the problem.

As introduced, we are particularly interested in two main tasks: clothing parsing and annotation. The **first part of this dissertation** focuses on image annotation, a task that may be described as assigning short textual descriptors or keywords (called tags) to images. These tags are related to specific garment items, such as shirts, trousers and shoes, and multiple tags may be associated with an arbitrary image. We formulate this task as a supervised classification problem: a process that automatically builds a classifier from a set of previously labeled/annotated examples (i.e., the training-set). Then, given an arbitrary image (i.e., an image in the test-set), the classifier recognizes the labels/tags that are more likely to be associated with it. First, we propose a Multi-modal and Multi-label Clothing Annotation algorithm, or simply MMCA, that uses the pointwise approach, which is the most commonly used strategy (Zhang et al. [2012]). According to Liu [2009], the pointwise approach employs the feature vector of each single image as an instance. In this case, each instance in the training set is composed of the visual and textual features (labels) of an image q, while the test set is only composed by the visual features of an image. Second, we propose a Multi-label, Multi-modal and Multi-instance Clothing Annotation method, or just M3CA, based on the pairwise approach, which is usually defined as an input space that represents instances as being a pair of images, both represented as feature vectors (Liu [2009]). Hence, each data instance, in the training and in the test set, is a pair of images: the query image q and the base image b. Labels associated with base image b are always known in advance in all sets (i.e., base labels) and labels associated with the query image are only known in advance in the training set (i.e., query labels). So, the only difference between the training and the test set is the query labels that are only known in the the former. Thus, for the training set, each instance (q, b) is composed of a set of base and query labels, plus a set of distances between the images q and b, while for the test set, each instance is composed by only the base labels and the visual distances between the images. This combination of visual and textual features (labels) is designed in search of improvements of the annotation results. The visual distances are computed using different image content descriptors (global Huang et al. [1997]; Stehling et al. [2002]; Pass et al. [1996]; Mahmoudi et al. [2003]; Tao and Dickinson [2000]; Swain and Ballard [1991]; Huang and Liu [2007]; Zegarra et al. [2008]; Unser [1986] and local Lowe [2004]; Bay et al. [2008]; Calonder et al. [2010]; Leutenegger et al. [2011]; Rublee et al. [2011]; Alahi et al. [2012]), allowing the proposed method to get the best from each type of descriptor besides creating a more sparse and robust approach. We intend to exploit the similarity between images, since similar ones are likely to share common labels, and thus small distances are expected to increase the membership probabilities associated with the correct labels for the query image q. Finally, MMCA and M3CA approaches are compared, looking for the best approach to our application.

Our classifiers are composed of association rules (Agrawal et al. [1993]), which are essentially local mappings $X \rightarrow y$ relating a combination of features in instance X to a label y. These rules are used collectively, resulting in a membership probability for each label. In order to provide fast learning times, the proposed algorithm extracts rules on a demand-driven basis – instead of learning a single and potentially large classifier which could be applicable to all instances in the test-set, our algorithm builds multiple small classifiers, one for each instance in the test-set. Typical solutions to multi-label classification employ the top-k approach (Veloso et al. [2007]), where a pre-determined threshold k is used to select the labels to be assigned to the query image. That is, only the k labels with the highest membership probabilities are assigned. Instead of relying on this parameter, we propose an entropy-minimization multi-instance approach which finds a different cut point for each instance in the test-set.

The **second part of this dissertation** focuses on image parsing that may be described as a process of partitioning an image into multiple segments (sets of pixels) in order to simplify its representation into something that is more meaningful and easier to analyze. In this case, these sets correspond to specific garment items in the image. We formulate this task using a deep learning strategy. We propose a Multi-scale Convolutional Neural Network model, or simply M-CNN, that creates a hierarchy of networks, where the first level processes a large amount of images with bigger dimension while the last one handles just a small amount of tiles with tiny size. This multi-scale strategy allows the method to capture minimal details of each image contributing to a more robust parsing algorithm. To define which images go from one level to another, a entropy strategy was applied.

The entropy (Alpaydin [2010]), a measure commonly used in information theory, characterizes the (im)purity of an arbitrary collection of examples. In this case, it denotes the purity of a single patch in relation to the number of classes associated to it, i.e., the more classes related to the patch the higher entropy it has (more impure). As introduced, entropy helps our approach to defined which patches are considered classified and which ones are not.

Specifically, the proposed method uses, in this case, three different network levels[1] which process images with different granularities, i.e., after every level the images are decomposed into smaller patches, allowing the network to capture minimal details. In the first level, larger images are processed in a robust network. Images with low entropy already get their final class in this level, while the others with high entropy (classification still undefined) are splitted into smaller patches and go to the next one. Remaining images without classification are again divided into even smaller patches and, finally, classified in the third level. At the end, we have a class associated with each patch of the image and a segmentation mask may be built.

In practice, we may observe the following contributions of this dissertation:

- Novel multi-instance, multi-label, multi-modal clothing annotation algorithms with the aggregation of different types of descriptors.
- Two different methods for clothing annotation that exploits association rules to create the classifiers: the MMCA (which follows a pointwise strategy) and the M3CA (which follows a pairwise strategy) approaches.
- A comparison between all proposed approaches which leads us to define the best one for our annotation task.
- A set of experiments was conducted to evaluate different visual feature representation and to analyze the best configuration for each type in the context of clothing annotation.
- A systematic set of experiments, using a collection of everyday photos crawled from popular fashion-related social networks, reveals that our algorithm improves upon first choice learning algorithms (Nguyen et al. [2013]), by a factor that ranges from 20% to 30% in terms of standard accuracy measures.
- Novel multi-scale clothing parsing algorithm using convolutional neural networks.
- Experiments reveals that the proposed algorithm achieves promising results when compared to popular clothing parsing algorithm.

Some results obtained in this work were published in XXVII Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI). Nogueira et al. [2014] presents preliminary results of the clothing annotation algorithms proposed in this work.

We organized the remainder of this work in six chapters. Chapter 2 presents related work. Chapter 3 presents the background concepts necessary for the understanding of

[1] For this application, only three network levels were used because of the relative small size of the image and the benefit between patch size and processing time.

this work and to make it self-contained. Chapter 4 shows the details of the proposed approaches for the clothing annotation task, as well as the evaluation protocol and experimental results obtained with each approach. The details of the proposed approach for the clothing parsing task, as well as the evaluation protocol and results are presented in Chapter 5. Finally, Chapter 6 concludes and points out future research directions.

Chapter 2

Related Work

This chapter presents a review of the literature surrounding clothing parsing and annotation, as well as some works of image parsing and annotation, since the former are subproblems of the latter. For both tasks, approaches combining supervised machine learning algorithms and visual feature extraction methods are becoming increasingly popular (Zhang et al. [2012]). However, there are approaches (Kuntimad and Ranganath [1999]) that learn the features and the classifiers, all at once.

Main approaches towards automatic image annotation modelled the learning problem as machine translation (Datta et al. [2008]) or correlation (probabilistic) learning tasks (Moran and Lavrenko [2014]). Some approaches (Li et al. [2010]; Vens et al. [2008]) adopt a multi-label model, others use multi-modal strategy to improve results (Putthividhya et al. [2010]; Xie et al. [2015]) and, finally, some works have modelled the problem as a multi-instance problem (Nguyen et al. [2013]). Furthermore, there are works that combine the strategies looking for a better performance (Nguyen et al. [2013]; Nogueira et al. [2014]).

Image parsing has been studied as a step toward general image understanding (Yamaguchi et al. [2012]). Approaches usually are modelled as pixel-based (Yamaguchi et al. [2013]), edge-based (Pal and Pal [1993]) or region(object)-based tasks (Yamaguchi et al. [2012]; Yang et al. [2014]).

The next sections present some relevant approaches related to these strategies, in addition to the advantages and disadvantages of each one. Section 2.1 presents the methods related to clothing annotation, including multi-label, multi-modal and multi-instance strategies. In Section 2.2, approaches for the clothing parsing are presented.

2.1 Clothing Annotation

There has been a great effort in the last few years on the clothing recognition task, with some works focusing on the annotation task. This recent boost, in clothing recognition field, is occurring perhaps influenced by recent advances in pose estimation (Yang and Ramanan [2011]), what caused a lot of works to emerge (Zhaolao et al. [2013]).

Tokumaru et al. [2002] proposed a system, named "Virtual Stylist", which aims to help users to find out outfits that might fit them well. Suh and Bederson [2007] proposed a semi-automatic approach that enables users to efficiently update automatically obtained metadata interactively and incrementally. Shen et al. [2007] introduced the recommendation of outfits for specific occasions based on textual input that defines the occasion and how the user wants to look like. More recently, the work of Vogiatzis et al. [2012] described the recommendation of clothes based on the similarity between users and models appearing in fashion magazines while Kalantidis et al. [2013] presented a scalable approach to automatically suggest relevant clothing products, given a single image without metadata. They, actually, formulate the problem as cross-scenario retrieval where the query is a real-world image, while the products from online shopping catalogues are usually presented in a clean environment.

Next, more works of image and clothing annotation are presented considering different strategies. Section 2.1.1 presents the approaches using multi-label classification. In Section 2.1.2, models that use the multi-modal strategy are presented. Approaches using multi-instance strategy are presented in Section 2.1.3.

2.1.1 Multi-Label Image Annotation

There is a lot of research dealing with single-label classification, where the instances are associated with a single label. However, in many applications, the instances may be associated with a set of labels, which characterizes the problem as a multi-label classification. Typically, in image annotation applications, an image have more than one label associated with it and, classifiers for this task are multi-label ones. According to Tsoumakas and Katakis [2007], multi-label classification algorithms can be categorized into two different groups:

1. problem transformation methods and
2. algorithm adaptation methods.

The first group includes methods that are algorithm independent, i.e., they transform the multi-label problem into one or more single-label problems. Usually, methods from this group tend to use probabilistic models, such as Bayesian or Gaussian ones, to generate adapted algorithms capable to handle and annotate different images. There is a lot of work in this group that includes (Tsoumakas and Katakis [2007]): (i) binary relevance method (Makadia et al. [2008]), which earns a determined number of binary classifiers, one for each different label in the label set, (ii) binary pairwise classification approach (Guillaumin et al. [2009]; Moran and Lavrenko [2014]), which transforms the multi-label dataset into a certain number of binary label datasets, one for each pair of labels, and (iii) label combination or label power-set methods (Read et al. [2008]), which considers each unique set of labels that exists in a multi-label training set as one of the classes of a new single-label classification task.

Between these methods, binary pairwise classification approaches detach from the others, since they have been achieving good results. The work of Moran and Lavrenko [2014] is one of these new methods. They address the image annotation problem by formulating a sparse kernel learning framework for the Continuous Relevance Model (CRM) (Lavrenko et al. [2003]) that greedily selects an optimal combination of kernels. Guillaumin et al. [2009], another binary pairwise approach, used log-likelihood to maximize the predictions of different tags in the training set, so they could optimally combine a collection of features that cover different aspects of image content, such as local shape descriptors, or global color histograms. This combination generates also a multi-modal method. Tags of the test images are predicted using weighted nearest-neighbor model.

The second group includes methods that extend specific learning algorithms in order to handle multi-label data directly. Well-known approaches include Adaboost (Li et al. [2010]), decision trees (Vens et al. [2008]), lazy methods (Veloso et al. [2007]; Yamaguchi et al. [2013]) and, more recently, neural networks (Socher et al. [2011]). In this group, neural network methods detach from the others, since they have been achieving excellent results in image annotation, segmentation and classification (Socher et al. [2011]), outperforming traditional algorithms (Gould et al. [2009]), and becoming the current state-of-the-art in these problems. Our approach may be categorized in this group, since we adapted a learning algorithm to predict multiple labels for the data.

2.1.2 Multi-Modal Image Annotation

In addition to multi-label classification, there is the multi-modal fusion that gained a lot of attention recently (Atrey et al. [2010]) due to the benefit it provides. The integration of multiple media data and their associated features creates a new scenario normally referred as multi-modal fusion. Usually, this fusion of multiple modalities can provide complementary information and increase the overall accuracy of the task. There is a lot of feasible fusions, such as audio/video or video/textual, though the most common fusion, when working with images, is the visual/textual one. This fusion takes advantages of: (i) visual features, that come from the images (usually obtained with descriptors) and, (ii) textual ones, which may be simplified by the tags/comments associated with each image. This fusion became really common because given the increasing amount of images that are currently available on the web with poor accuracy annotation, there has been considerable interest in the computer vision community to leverage this data to learn recognition models. According to Atrey et al. [2010], multi-modal fusion algorithms can be categorized into three different groups:

1. feature level or early fusion (Xie et al. [2015]), which combines the features extracted from the input data and then send as input to the classifiers,
2. decision level or late fusion (Guillaumin et al. [2009]; Fergus et al. [2009]; Guillaumin et al. [2010]), which isolates the features to create different combinations of classifiers using some criterion, and
3. hybrid approach (Nguyen et al. [2013]), which is a combination of both feature and decision level strategies, taking advantages of both.

Considering the early fusion method, Xie et al. [2015] use images weakly tagged to improve the image classification performance using statistical approaches. Using the late fusion strategy, Guillaumin et al. [2009] and Fergus et al. [2009] present similar works that combine visual and textual features, where the textual ones are represented by labels/tags associated with images crawled in social networks. Guillaumin et al. [2010] use image tags to improve the performance of the classifiers, but they do not assume their availability for test images. Our approaches follow this strategy by combining visual and textual features from each images, and delivering them to the learning algorithm, which uses this combination to create classifiers. Nguyen et al. [2013] uses the hybrid approach, where the fusion of multi-modalities may be made in both decision level (labels) and feature level (visual/textual) by using different models.

2.1.3 Multi-Instance Image Annotation

In traditional supervised learning, an object is represented by a instance (usually, features) and associated with a class label. In these cases, a instance may be formally represented by $\mathcal{X}$ and $\mathcal{Y}$ represent the set class labels. So, the task is to learn a function $f : \mathcal{X} \rightarrow \mathcal{Y}$ from a given dataset $(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)$ where $x_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$. Although successful, some problems may not fit very well to this model, such as problems where the object may be associated with a multiple number of instances simultaneously, as for example, an image may be represented by a myriad of patches (feature vectors). To deal with this kind of problem, arise the multi-instance learning. In this framework, a object is described by multiple instances. Formally, $\mathcal{X}$ represent the instance space and $\mathcal{Y}$ the set class labels. The task is to learn a function $f : \mathcal{X} \rightarrow \mathcal{Y}$ from a given dataset $(X_1, Y_1), (X_2, Y_2), ..., (X_n, Y_n)$, where $X_i \in \mathcal{X}$ is a set of instances $x_1^{(i)}, x_2^{(i)}, ..., x_{m_i}^{(i)}$, $x_j^{(i)} \in \mathcal{X}(j = 1, 2, ..., m_i)$, and $Y_i \in \mathcal{Y}$ is the set of labels $y_1^{(i)}, y_2^{(i)}, ..., y_{l_i}^{(i)}$, $y_k^{(i)} \in \mathcal{Y}(k = 1, 2, ..., l_i)$. In this case, we are also considering that a object may have more the one label. m_i represent the number of instances in X_i and l_i the number of labels in Y_i.

In short, the multiple-instance learning is a variation of supervised learning, which is the task of learning classifiers from bags of instances (Maron and Lozano-Pérez [1997]) that may contain as many instances as possible. Recently, this kind of approach became very popular for some specific problems because of the good results achieved. Between these works, Tang et al. [2010] proposed a method that exploits a unified learning framework which combines the multiple-instance and single-instance representations for image annotation. Specifically, they use an integrated graph-based semi-supervised learning that associate these types of representations simultaneously. Feng and Xu [2010] proposed an improved Transductive Multi-Instance Multi-Label (TMIML) learning, which aims at taking full advantage of both labeled and unlabeled data to address the annotation problem. Both of these works also use the Corel5K dataset on their experiments.

More recently, Nguyen et al. [2013] proposed a multi-label, multi-modal and multi-instance approach using Latent Dirichlet Allocation (M3LDA). First, they build the gist of a scene using Oliva and Torralba [2006] algorithm and then, they consider each patch of image as an instance, what generated a myriad of items. Each instance may be represented by a bag of prototypes, which are obtained by clustering visual features of the patch (Zhou et al. [2012]). Associating instances and tags, they built a learning algorithm based on Latent Dirichlet Allocation (LDA). With this approach, they can not only annotate the images as a whole but can also annotate its region, if possible.

In this dissertation, we propose clothing annotation techniques, seeking an alternative method associating all concepts aforementioned. As mentioned, our methods combine textual features (labels) with visual ones, in a later fusion mode, looking for improvements of the annotation results. Furthermore, while most works (Nguyen et al. [2013]) treats each region (keypoint) of an image as a instance to create a multi-instance classifier resulting in a myriad of features, our proposed pairwise method creates a classifier by pairing images and calculating the visual distance between them, which makes the approach more sparse and robust. A combination of local and global visual features allows the proposed method to leverage from both of them, exploiting the best of each one in our application. In addition, different from other works (Tang et al. [2010]; Feng and Xu [2010]), instead of using less realistic scenarios, our experiments were on full realistic ones using dataset crawled from the web with tags generated by users from around the globe.

2.2 Clothing Parsing

As presented, there has been a huge interest in the last years on the clothing recognition task. However, many works focus on special clothing classes and applications (Chen et al. [2012]; Cushen and Nixon [2012]) and, only in 2012, that generic clothing recognition has been directly tackled (Yamaguchi et al. [2012]). This interest in clothing recognition is because the important roles it plays in several areas, such as surveillance and person search. For the former area, Yang and Yu [2011] proposed a cloth recognition framework in videos where the camera angles are fixed and background subtraction can be effectively used for human detection. Cushen and Nixon [2012] also proposed a real time upper body cloth segmentation in images where people are wearing a monochromatic clothing and printed/stitched textures. For person search, Gallagher and Chen [2008] proposed a method that uses facial features and clothing features to recognize individuals in images based on others pictures (usually from the same person). According to Singh and Singh [2010], the segmentation process may be organized in three groups:

1. pixel-based (Yamaguchi et al. [2013]), the simplest approach used for segmentation, classifies each pixel individually.

2. edge-based (Pal and Pal [1993]), which uses the edges of the image to create patches that should be used by the learning algorithm.

3. region or object-based (Yamaguchi et al. [2012]; Yang et al. [2014]), that splits the images into regions (or objects) and classify each one separately.

Between the pixel-based clothing parsing, Yamaguchi et al. [2013] proposed a framework that mixtures global and local models of clothing items with human pose and mask estimation (this latter to avoid background effect). More common than pixel-based parsing, the region-based clothing segmentation simplifies the problem by assuming that uniform appearance regions belong to the same item (Gallagher and Chen [2008]) and reduce the problem to the prediction of a labeling over a set of superpixels. In Yamaguchi et al. [2012], a method for clothing parsing is formulated as a labeling problem, with images segmented into superpixels and clothing labels predicted for each segment using a Conditional Random Field (CRF) (Lafferty et al. [2001]) model. Yang et al. [2014] proposes a data-driven framework composed of two phases of inference: first one extracts regions of the images and jointly refines each region over all images while the second phase constructs a multi-image graphical model considering the segmented regions as vertices, and introduces several contexts of clothing configuration (e.g., item location and mutual interactions). The label assignment is solved using the some graph cuts algorithm. Our clothing parsing approach belongs to this group, since we consider little patches to create the classifiers. Next, Section 2.2.1 presents neural network methods to tackle the image parsing task.

2.2.1 Neural Networks

Recently, neural network methods have been employed in a wide range of problems mainly supported by successful cases of tasks which results were improved using this kind of approach (Krizhevsky et al. [2012]).

Although all the attention, just a few applications doing image segmentation have emerged. Between them, Socher et al. [2011] proposed a neural networks capable of recovering a image from recursive structures (objects) obtained by using Comaniciu and Meer [2002] method. This algorithm may be used to parse images, as well as natural language sentences. Schulz and Behnke [2012] proposed a convolutional network architecture for image segmentation. They use several elements, such as multiple output maps, suitable loss functions, supervised pre-training, multi-scale inputs, reused outputs, and pairwise class location filters.

The main difference between these approaches and ours is that we use different network architectures in the same image with different scales, considering these networks as a hierarchical structures. Thus, an image goes through all the levels only if extremely needed, otherwise, it may be segmented without reaching the last level. This process decreases the test time, being propitious for on the fly applications.

Chapter 3

Background Concepts

This chapter presents some methods we use in this work that are essential for a self-contained understanding of it. In the first phase of this work, we exploit the traditional combination of machine learning methods and visual image features. To extract the visual elements of the images, we used feature extraction algorithms (descriptors). These methods can be classified into three levels:

1. low-level feature extraction,

2. mid-level feature extraction and

3. high-level feature extraction.

Descriptors from the first level work at extracting visual properties from the image via pixel-level operations. This level is crucial and needed for all image analysis procedure. The mid-level algorithms aim at combining the set of local features into a global image representation of intermediate complexity. A Bag of Words (BoW), proposed by Sivic and Zisserman [2006], is a good example and also exploited in this work. The last level methods take advantage of semantic informations of the image to reduce the semantic gap. We do not use high-level features, since the reduction of the semantic gap is not the main problem confronted in this work.

Considering these visual features, a myriad of machine learning methods could be exploited to create the classifiers, such as Support Vector Machines (SVM) and association rules. Despite all options, a machine learning method that could support multi-label, multi-modal and multi-instances strategies was preferable. Thus, Lazy Association Classifiers (LAC) (Veloso et al. [2006]) was chosen given its natural adaptation to multi-modal approaches, accepting visual and textual features without too much effort, and also because it permits the scalability of the instances without increasing the processing time, since the number of classes is more relevant to the algorithm than the number of instances. In addition to this, the method easily allows the use of multi-label strategy since its output consists of a ranking with the classes and respectively probabilities.

In the second phase of this work, we exploit the benefits of a Convolutional Neural Network (CNN) to tackle the clothing parsing problem. Neural networks were chosen because of several advantages: (i) it can learn the image features and classifiers (in different layers) at once, (ii) can adjust the learning process, in execution time, based on the accuracy of the network, giving more importance to one layer than other depending on the problem, (iii) it has ability to learn how to do the tasks based on the data given for training or initial experience and (iv) it can create its own organization or representation of the information it receives during learning time.

Next, we present all details related to these frameworks. The low-level descriptors used for the clothing annotation approach are presented in Section 3.1, followed by the mid-level approaches presented in Section 3.2. The learning algorithm, LAC (Veloso et al. [2006]), is presented in Section 3.3. Section 3.4 presents detailed information about the layers used in our neural network.

3.1 Low-level feature extraction

Researchers have been challenged for years to represent images based on their content. Towards scene understanding, the community created many low-level feature algorithms to represent visual elements of an image. Feature representation methods can be categorized as:

1. global and
2. local.

Next, we present the concepts, advantages and disadvantages of each method, and how they evolve through the years. Section 3.1.1 presents the global descriptors used while Section 3.1.2 presents the local ones.

3.1.1 Global Descriptors

The need of translating image properties, like color, texture, and shape, for example, has motivated industry and research communities to keep developing new algorithms for representing images. The search for new methods have been also motivated by the need of compare and match images, enabling creation of new applications that could work over images totally by itself. In the beginning of the decade of 1990, several algorithms were proposed to extract these features from images (Zhang and Lu [2004]). Those techniques have usually relied on computing a representation that encodes global aspects of images, therefore called global descriptors.

These descriptors usually are cheap to obtain since they rely on computing a representation that encodes global aspects of images. This brings the advantages of being simple to compute and to provide a good general idea of the image content. On the other hand, it brings several disadvantages, like deficiency of encode details and low effectiveness in some precise applications, like recognition tasks for cluttered images (Tuytelaars and Mikolajczyk [2007]). Thus, there is a multitude of global descriptors available in the literature (Zhang and Lu [2004]) that can be used to represent visual elements, which are strongly based upon the concept of image descriptors (da Silva Torres and Falcão [2006]). A descriptor expresses perceptual qualities of an image, and is composed by:

1. A feature-vector that encodes image properties, such as color, texture and shape, and,

2. A distance function that returns the similarity between two images as a function of the distances between their corresponding feature-vectors.

Both the feature-vector and the distance function affect how the descriptor encode the perceptual qualities of the images. An image descriptor representation to compute the distance between two input images is presented in Figure 3.1.

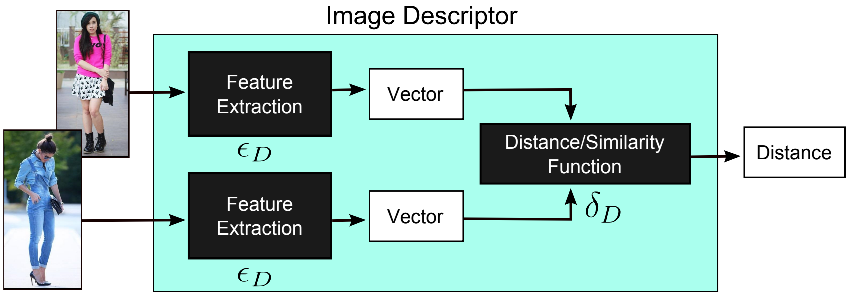


Figure 3.1: An image descriptor representation.

It is known that different descriptors may provide complementary information about images, so the combination of multiple descriptors is likely to provide improved performance when compared with a descriptor in isolation. However, the optimal combination of descriptors is data-dependent, as well as a hard task depending on the problem, since different descriptors may produce different results. We selected 10 global descriptors to be evaluated for the clothing annotation problem, based on extensive experiments performed by dos Santos et al. [2010], dos Santos et al. [2012] and Penatti et al. [2012], which pointed out to some of the most interesting image descriptors in the current computer vision literature. Next, the descriptors we have used in the experiments along with the first phase of this dissertation are presented.

1. Color descriptors:

   a) Auto-Correlogram Color (ACC) (Huang et al. [1997]) maps the spatial information of colors by pixel correlations at different distances, i.e., computes the probability of finding in the image two pixels with color C at distance d from each other.

   b) Border/Interior Pixel Classification (BIC) (Stehling et al. [2002]) creates the feature vector from two histograms: one for for the interior pixels and another for the border ones. When a pixel has the same spectral value in the quantized space as its four neighbors (the ones which are above, below, on the right, and on the left), it is classified as interior. Otherwise, the pixel is classified as border. The two histograms are concatenated and stored into a feature vector.

   c) Color Coherence Vector (CCV) (Pass et al. [1996]), which uses an extraction algorithm that classifies the image pixels into two groups: "coherent" and "incoherent". This classification considers if a pixel belongs or not to a region with similar colors, that is, coherent regions. The two histograms computed after the quantization (one for each group) are merged to compose the feature vector.

   d) Global Color Histogram (GCH) (Swain and Ballard [1991]) uses an extraction method which quantizes the color space in a uniform way and scans the image computing the number of pixels belonging to each color.

   e) Local Color Histogram (LCH) (Swain and Ballard [1991]) splits the image into fixed-size regions and computes a color histogram for each region. The feature vector is composed by a concatenation of the histograms of each region.

2. Texture descriptors:

   a) Quantized Compound Change Histogram (QCCH) (Huang and Liu [2007]) uses the relation between pixels and their neighbors to encode texture information. A square window runs through the image capturing the average gray value in each step. Four variation rates are then computed by taking into consideration the average gray values in four directions: horizontal, vertical, diagonal, and anti-diagonal directions. The average of these four variations is calculated for each window position.

   b) Local Activity Spectrum (LAS) (Tao and Dickinson [2000]) captures texture spatial activity in four different directions separately: horizontal, vertical, diagonal, and anti-diagonal. The four activity measures are computed for a specific pixel by considering the values of neighboring in the four directions.

   c) Steerable Pyramid Decomposition (SID) (Zegarra et al. [2008]) uses a set of filters sensitive to different scales and orientations. The image is recursively decomposed into bands, which have the mean and standard deviation extracted to be used as features values.

d) Unser (Unser [1986]) computes a histogram of sums H_{sum} and a histogram of differences H_{dif}. The former is incremented considering the sum while the latter is incremented taking account the difference between the values of two neighbor pixels. Measures such as energy, contrast, and entropy can be extracted from these histograms.

3. Shape descriptors:

   a) Edge Orientation Auto-Correlogram (EOAC) (Mahmoudi et al. [2003]) classifies the edges based on two aspects: boundary orientation and correlation between neighbor edges. The algorithm has two main steps: (i) image gradient computation, and (ii) edge orientation auto-correlogram calculation. The feature vector is composed of the values from this auto-correlogram.

As introduced, global descriptors have some deficiency when working with object recognition (Tuytelaars and Mikolajczyk [2007]), what motivated the research community to developed new visual extraction algorithms, called local descriptors.

3.1.2 Local Descriptors

Local descriptors were developed in beginning of the decade of 2000, and brought new possibilities to the computer vision community. The success of the local descriptor approach is explained due to the fact that classical global features have difficulty in distinguishing foreground from background objects, and thus are not very effective in recognition tasks for images with complex content (many people and objects)(Tuytelaars and Mikolajczyk [2007]). Furthermore, local descriptors are more powerful to present object properties and are very precise, because small variations in the objects of the image may avoid similar regions to be considered as a match. On the other hand, local descriptors are more expensive to compute and may produce a variable number of feature vectors per image, which makes the comparison between a pair of images more complex and expensive.

A local feature is an image pattern that differs from its immediate neighborhood (Tuytelaars and Mikolajczyk [2007]). It is usually associated with a change of an image property or several properties. Local features may be points, edges or small image patches. According to Tuytelaars [2010], two types of patch-based approaches can be distinguished:

1. Interest Points: such as corners and blobs, which position, scale and shape are computed by a feature detector algorithm, which is computationally expensive.

Interest points focus on “interesting” locations in the image and include various degrees of viewpoint, illumination invariance and perhaps resolution.

2. Dense Sampling: patches of fixed size are placed on a regular grid (possibly repeated over multiple scales) that may have different shapes. It gives a better coverage of the entire object or scene and a fixed number of features per image area. Regions with less contrast contribute equally to the overall image representation.

Examples of each type of extraction can be seen in Figure 3.2.

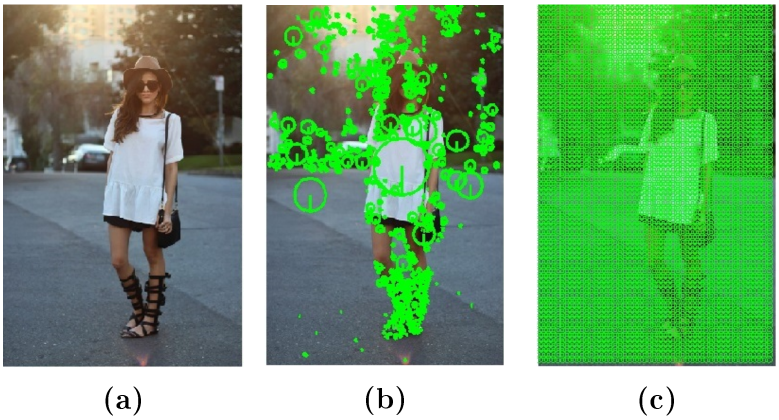


Figure 3.2: Local features extraction: (a) Original image (b) Interest points detection example (c) Dense sampling example

Like the global descriptors, a myriad of local descriptors are available in the literature Li and Allinson [2008] and different descriptors may provide complementary information about images, so the combination of multiple descriptors tends to improved performance. However, the optimal combination of descriptors is data-dependent and unlikely to obtain in advance. To select the best low-level descriptors for the clothing annotation task, we evaluate 6 different feature extraction techniques. This descriptors were chosen based on extensive experiments performed by Bekele et al. [2013]. Next, the local descriptors we have used in the experiments along with the first phase of this work are presented.

1. Scale-Invariant Feature Transform (SIFT) (Lowe [2004]) combines a scale invariant region detector and a descriptor based on the gradient distribution in the detected regions. It is reasonably invariant to changes in illumination, image noise, rotation, scaling, and small changes in viewpoint.

2. Speeded Up Robust Features (SURF) (Bay et al. [2008]) is based on multi-scale space theory and Hessian matrix. It manages to get a smaller feature vector (usually a half of the SIFT vector size), and outperforms the SIFT method in almost every transformation or distortion.

3. Binary Robust Independent Elementary Features (BRIEF) (Calonder et al. [2010]) is created with a set of pairwise intensity comparisons, which are selected randomly from an Gaussian distribution centered at the feature location.

4. Binary Robust Invariant Scalable Keypoints (BRISK) (Leutenegger et al. [2011]) combines a feature location algorithm and a specific keypoint detector in a scale-space pyramid creating a rotation and scale invariant feature vector.

5. Oriented FAST and Rotated BRIEF (ORB) (Rublee et al. [2011]) computes a rotation invariant local orientation through the use of an intensity centroid, which is a weighted averaging of pixel values in the local patch (different from the center of the feature). The orientation is the vector between the feature location and the centroid. The feature vector is composed by less correlated information, i.e., more discriminative.

6. Fast Retina KeyPoint (FREAK) (Alahi et al. [2012]) selects keypoints which the area are evaluated with 43 weighted Gaussians, making it more robust than BRISK. Creates a rotation invariant feature vector just like ORB.

The local descriptors are usually applied to some type of applications like copy detection (Law-To et al. [2007]) or object localization (Sivic et al. [2005]). So, the community has seen that to continue creating applications toward a total scene understanding, new methods have to be created in order to extract the best as possible from the descriptors. Toward this goal, mid-level feature extraction aims at transforming low-level descriptors into a global and richer image representation of intermediate complexity (Boureau et al. [2010]).

3.2 Mid-level Image Representation

A mid-level representation use local features built upon low-level ones creating a new representation for an image, without looking for understanding its high-level features. According to Boureau et al. [2010], in order to get the mid-level representation, the standard processing follows three steps: (i) low-level local feature extraction, (ii) coding, which performs transformation of the descriptors into a representation better adapted to the task and (iii) pooling, which summarizes the coded features. Classification algorithms are then trained on the mid-level vectors obtained.

Next, we present concept details of the mid-level representation and some variation. Section 3.2.1 presents the main principles of Bag of (Visual) Words (BoW) method (Sivic and Zisserman [2006]) while Section 3.2.2 presents BossaNova algorithm (de Avila et al. [2011]), that has the same principles of mid-level representations with some interesting ideas.

3.2.1 Bag of (Visual) Words

A method proposed by Sivic and Zisserman [2006], introduced the idea of representing images in a similar way as representing text documents. Their method became very popular for multimedia retrieval and classification systems because it associates similar objects giving them the same visual representation, which is less expensive to work with since it is smaller than the representation extracted from local descriptors. Like a text document may be decomposed as a set of textual words, an image may be analyzed as a set of local appearances. That way, came the popular name Bag of (Visual) Words (BoW).

Therefore, images are decomposed into a set of local patches that represent the first step of the mid-level representation. The patches are then assigned to a vocabulary of patches, called visual dictionary, which the scheme is presented in Figure 3.3.

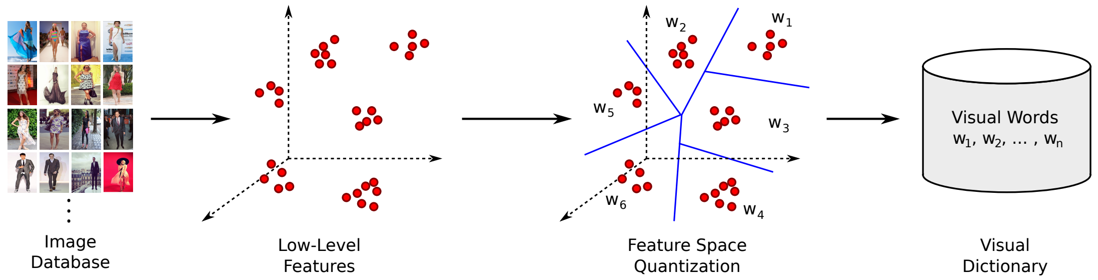


Figure 3.3: Idea behind the generation of the visual dictionary. After extracting local feature vectors from each image, the feature space is quantized and each region corresponds to a visual word. Adapted from dos Santos et al. [2014]

The visual dictionary is the codebook of the available patches that are used to represent image content. The creation of the codebook can also be referenced as a quantization of the space of features generating the codewords. This is usually obtained by unsupervised learning, like k-means, over a sample of descriptors from the training data. Over this new quantized space occurs the coding phase that must consider how the low-level features of an image are distributed according to that new space. This can be performed by simply assigning the image local features to the visual words in the dictionary. The simplest coding in the literature assigns a local descriptor to the closest visual codeword, giving one (and only one) nonzero coefficient. This approach is commonly called hard coding, and may suffer from ambiguity (Philbin et al. [2008]). In order to attenuate the effect of coding errors, one may rely on soft coding (van Gemert et al. [2008]), which is based on a soft assignment to each codeword, weighted by distances/similarities between descriptors and codewords.

BoW exploits the concept of visual dictionary to create a single feature vector for each image as presented in Figure 3.4. The last step of the representation creates a histogram for each image and is usually referenced as the pooling step. This phase compacts all information that belongs to a codeword into a single feature vector. It aims at preserving the information encoded in the coding step or to discard the least important properties, generating a feature histogram for the image. Popular pooling approaches are based on computing the average assignment value of each visual word in the image or considering only the maximum activation value of them.

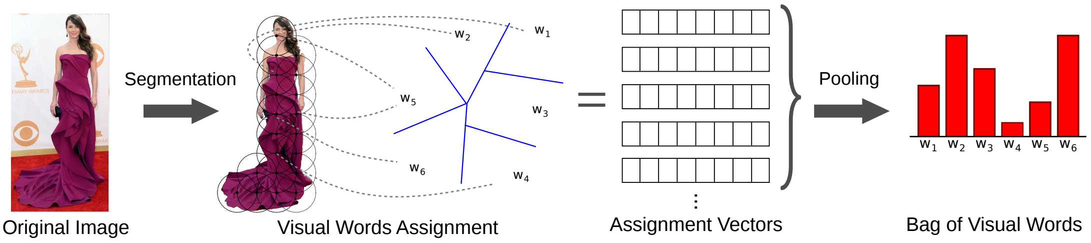


Figure 3.4: Creating a BoW using the visual dictionary for each image. Given an input image, its local feature vectors are computed and then assigned to the visual words in the dictionary. Then, occurs the pooling step creating the histogram. Adapted from dos Santos et al. [2014]

Therefore, the visual dictionary model solves the issue of multiple feature vectors per image computed by local descriptors. Another advantage is that the description is more general, eliminating the problem of very precise representations generated by local descriptors, and making the dictionary-based representations useful in a wider range of applications.

3.2.2 BossaNova Approach

Different coding/pooling functions generate different results and the best functions are problem-dependent. Thus, BossaNova, proposed by de Avila et al. [2011], was chosen to be tested in our application because it may allow a comparison between its different process with a traditional one used on BoW.

BossaNova differentiates from the BoW approach at the coding/pooling stage, resulting in a new representation that better preserves the information from the encoded local descriptors, by using a density-based pooling step.

Their coding function activates the closest codewords to the descriptor, which corresponds to a localized soft coding over the visual codebook. The pooling step estimates the distribution of the descriptors around each codeword, while the BoW estimates the distribution around one or determined number of codewords. The BossaNova pooling process is a non-parametric, density-based estimation of the descriptors distribution generating a histogram of distances between the descriptors found in the image and each codeword. Figure 3.5 presents the whole procedure of the BossaNova approach.

3.3 Lazy Associative Classifiers

Aside the visual features, a multitude of machine learning algorithms could be exploited in an attempt to solve the clothing annotation task. Despite all options, a main require-

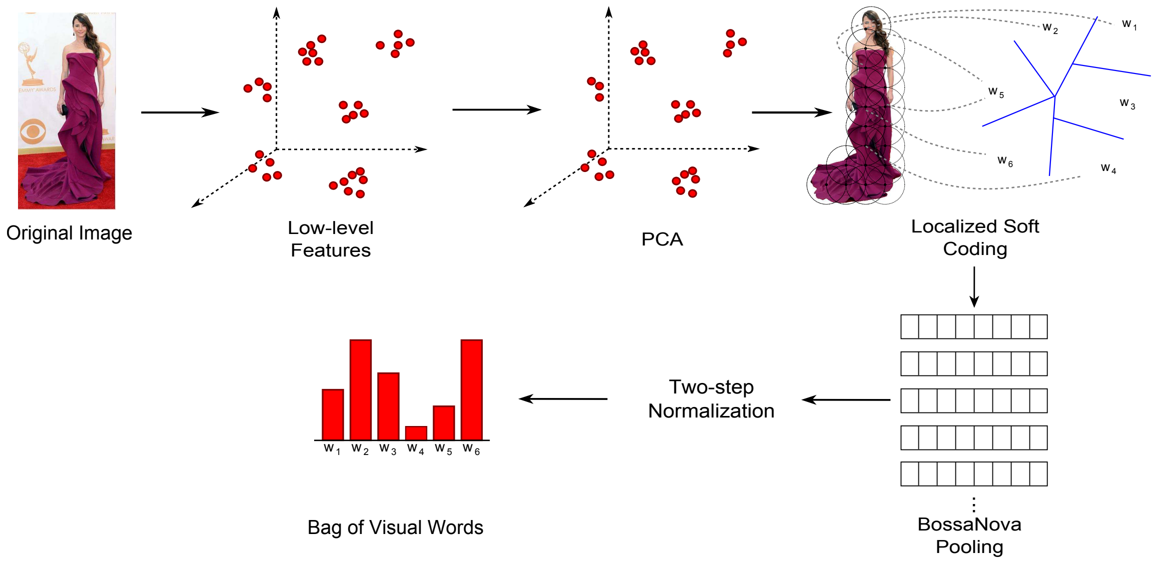


Figure 3.5: The low-level features extracted from the query image have their dimensionality reduced by PCA algorithm. Then, there is a localized soft coding followed by a BossaNova polling. A two-step normalization is made looking for preserve relevant features. Adapted from de Avila et al. [2011]

ment observed was the capacity of handle multi-label, multi-modal and multi-instances strategies. This way, Lazy Association Classifiers (or simply LAC), proposed by Veloso et al. [2006], was chosen given its natural adaptation to multi-modal approaches, accepting visual and textual features without too much effort and also because its output is composed of a ranking with the classes and respective probabilities, allowing the method to be easily adapted to multi-label tasks. Another advantage is that it permits the scalability of the instances without increasing processing time, since the number of classes of an instance is more relevant to the algorithm (in terms of computation) than the number of instances.

LAC uses association rules (Agrawal et al. [1993]) to produce classifiers that, depending on the task, may predict labels of an image or relevance related to a document. These rules are patterns describing implications of the form $X \to y_i$, where X is known as the antecedent of the rule while y_i is the consequent. The antecedent may be any combination of features, depending on the task, while the consequent may be any label or class. The rule does not express a classical logical application where X necessarily entails y_i. Instead it denotes the tendency of observing y_i when X is observed.

Definition 1. *A formal definition of a standard association rule, composed of antecedent X and consequent y_i: $X \xrightarrow{\theta} y_i$*

The strength of the association between the antecedent and the consequent is measured by a statistic θ, which is known as confidence (Agrawal et al. [1993]) and is simply the conditional probability of the consequent given the antecedent.

The algorithm receives as input a labelled training-set $\mathcal{D}$ and a test-set $\mathcal{T}$, with classes/labels unknown. From each instance $X \in \mathcal{D}$ of the training-set, the algorithm extracts a rule-set $\mathcal{R}$ composed of rules used to predict classes/labels $\mathcal{L}_X$ that approximates as accurately as possible $\mathcal{L}_X^*$, which represents the ground-truth of the instance, i.e., the true classes/labels of X. Distances between the instances are discretized (Fayyad and Irani [1993]) and then assigned to distance intervals, in order to allow for the enumeration of the association rules.

Basically, each rule $\{X \rightarrow y_i\} \in \mathcal{R}$ is a vote given for y_i. Thus, after extract the set rules $\mathcal{R}$, given an instance $Z \in \mathcal{T}$, a rule is a valid vote if it is applicable to Z. The way rules are applicable to an instance is described in Definition 2. Using this definition, it is possible to select the rules applicable to a specific instance Z, denoted as $\mathcal{R}_Z$. Hence, only rules in $\mathcal{R}_Z$ are considered as valid votes when predicting classes/labels.

Definition 2. *A rule $\{X \rightarrow y_i\}$ is said to be applicable to instance $Z \in \mathcal{T}$ if all intervals in X are in Z, that is, $X \subseteq (Z)$.*

Further, $\mathcal{R}_Z^{y_i}$ is a subset of $\mathcal{R}_Z$ containing only rules predicting class/label y_i. Votes in $\mathcal{R}_Z^{y_i}$ have different weights, depending on the confidence θ of the corresponding rules. Given an arbitrary image Z, the weighted votes for label y_i are averaged, resulting in the score for y_i, as shown in Equation 3.1.

$$s(Z, y_i) = \frac{\sum \theta(X \rightarrow y_i)}{|\mathcal{R}_Z^{y_i}|} \tag{3.1}$$

where $X \subseteq Z$ and $|\mathcal{R}|$ represents the set size.

The likelihood of an instance Z being associated with class/label y_i is obtained by normalizing the scores, since the sum of the likelihood of all classes/labels of an instance should result one. So, as expressed by $\hat{p}(y_i|Z)$, the Equation 3.2 shows the normalization.

$$\hat{p}(y_i|Z) = \frac{s(Z, y_i)}{\sum\limits_j s(Z, y_j)}. \tag{3.2}$$

At the end, for each instance, LAC generates a ranking with all the classes/labels associated with its likelihood (probability). In this case, higher values of $\hat{p}(y_i|Z)$ indicate that the class/label is likely to be associated with Z. On the other hand, lower values of $\hat{p}(y_i|Z)$ indicate that the class/label is not likely to be associated with Z.

3.4 Convolutional Neural Networks

Artificial Neural Network (NN), an information processing paradigm, is inspired in biological nervous systems, such as the brain. The key element of this paradigm is the novel structure of the information processing system. NN simulations appear to be a recent development but this field was established in 1943 (Mcculloch and Pitts [1943]) with first biological models of the brain. Since then, many important advances have been boosted by the improvements of computers performance, such as the Perceptron (Rosenblatt [1958]) and backpropagation algorithm (Rumelhart et al. [1988]). Recently, this field enjoys a lot of interest mainly supported by almost unlimited computational resources and exciting results in some tasks (Krizhevsky et al. [2012]).

NN is generally presented as systems of interconnected processing units (neurons) which can compute values from inputs leading to a output that may be used on further units. These neurons work in agreement to solve a specific problem, learning by example, i.e., a NN is created for a specific application, such as pattern recognition or data classification, through a learning process. These neurons compose a processing layer which may have different types, such as convolutional, softmax and fully-connected, depending on the operations it realizes over the input. These layers are stacked forming multilayer neural networks. Different networks may be formed using these several types of layers, as Convolutional Neural Network (CNN) (LeCun et al. [1989]), Restricted Boltzmann Machines (RBM) (Salakhutdinov et al. [2007]) and Deep Belief Networks (DBN) (Hinton [2010]).

CNN were proposed to work over images, since they try to take leverage from the natural property of an image, i.e., its stationary state. More specifically, the statistics of one part of the image are the same as any other part. Thus, features learned at one part can also be applied to another region of the image, and the same features can be used in several locations. Although this advantage, CNN can be also used to model Natural Language Processing tasks (Kalchbrenner et al. [2014]).

When compared to other types of networks, CNN present several other advantages: (i) automatically learn local feature extractors, (i) are invariant to small translations and distortions in the input pattern, and (iii) implement the principle of weight sharing which drastically reduces the number of free parameters and thus increases their generalization capacity. Next, more about the processing units and layers used in this work are presented.

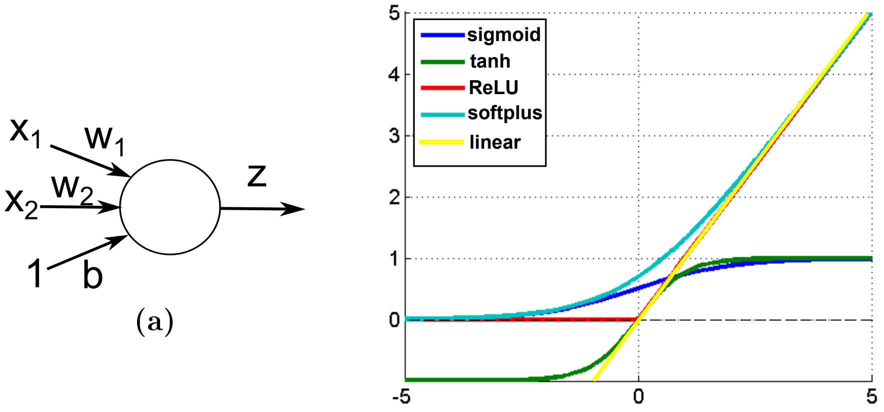
3.4.1 Processing Units

As introduced, NN has been developed trough the years with different models emerging. However, the proposition that neurons are the basis of every network still stands. These artificial neurons try to simulate the biological ones in a limited way. Artificial neurons are basically processing units that use several variables as input and, usually, have one output calculated through the activation function. As presented, an artificial neuron has a weight vector $w = (w_1, w_2, \cdots, w_m)$ and a threshold or bias b. The weights are analogous to the strength of the biological dendritic connections. The bias is considered the value that must be surpassed by the inputs before an artificial neuron become active (i.e., greater than zero). The activation of a neuron is the sum of the inner product of the weight vector with an input vector $x = (x_1, x_2, \cdots, x_m)$ plus the bias. A example of the first and simplest processing unit, called linear neuron, can be seen in Figure 3.6a and its activation function is presented in Equation 3.3. These kind of neurons are simple but computationally limited as can be seen in Figure 3.6b.

$$z = b + \sum_i x_i * w_i \tag{3.3}$$

where z represents the output, b is the bias, x the input data and w are the weights.

Others neurons, more robust, have emerged supported by the activation function of the linear neuron. These innovate by including another step when calculating the function: (i) calculate the output z using the Definition 3.3, and (ii) calculate the final activation of the neuron by using this output in some non-linear function, such as Sigmoid and Hyperbolic Tangent (tanh). Others non-linear functions were created, like Rectifier (Nair and Hinton [2010]) and Softplus (Glorot et al. [2011]) functions. Figure 3.6b presents the activation output of all these functions.



(a)

(b)

Figure 3.6: Artificial Neurons: (a) Example of a neuron: z represents the output, b is the bias, x the input data and w are the weights (b) Possible activation functions.

The processing unit that uses the rectifier as activation function is called Rectified Linear Unit (ReLU) (Nair and Hinton [2010]). This neuron has several advantages when compared to others: (i) works better to avoid saturation during the learning process, (ii) induces the sparsity in the hidden units, and (iii) does not face gradient vanishing problem[1] as with sigmoid and tanh function. Because of these advantages, in this work, our proposed networks are composed of ReLUs. The first step of the activation function of a ReLU is presented in Equation 3.3 (same as a linear neuron) while the second one is introduced in Equation 3.4.

[1] The gradient vanishing problem occurs when the propagated errors become too small and the gradient calculated for the backpropagation step vanishes, making impossible to update the weights of the layers and achieve a good solution.

$$a = \begin{cases} z, if\, z > 0 \\ 0, otherwise \end{cases} \quad \Leftrightarrow \quad a = f(z) = max(0, z) \tag{3.4}$$

The processing units are grouped into layers, which are stacked forming multilayer neural networks. These layers give the foundation to others, such as convolutional and fully connected.

3.4.2 Network Layers

The convolutional layer is composed of processing units responsible to capture the features from the images, where the first layer obtains the low-level features (like edges, lines and corners) while the others get high-level features (like structures, objects and shapes). The process made in this kind of layer can be decomposed into two phases: (i) a fixed-size window runs over the image defining a region of interest, and (ii) Using the pixels inside each window as input to the processing units, the features of this region are extracted, i.e., each pixel is multiplied by its respective weight generating the output of the neuron, just like Equation 3.3. Thus only one output is generated concerning each region defined by the window. This iterative process results in a new image, generally smaller than the original one, with the visual features extracted. Figure 3.7 presents some steps of a convolutional layer capturing features from an image.

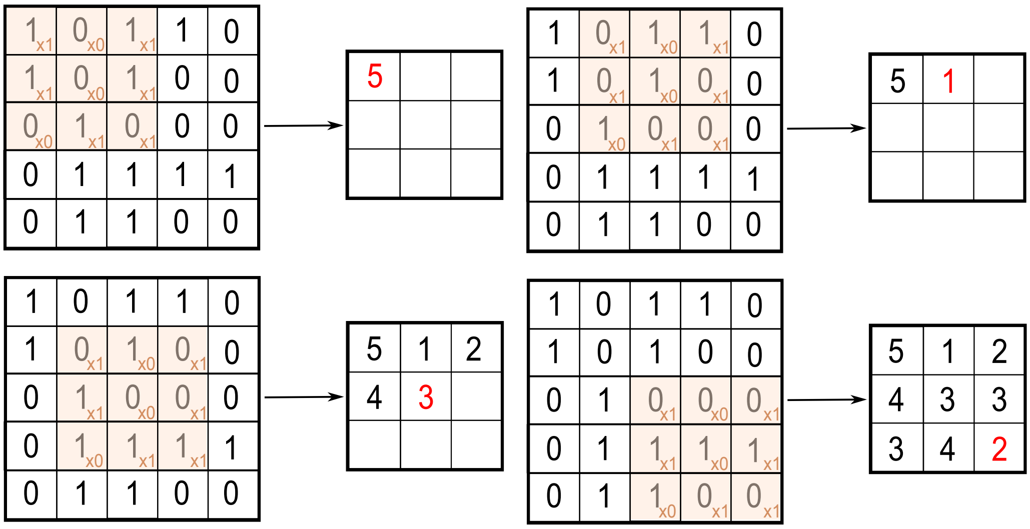


Figure 3.7: Some steps of a 3x3 window of a convolutional layer extracting the features from an image. Figure adapted from Ng et al. [2011a]

A lot of these features are very similar, since each window may have common pixels, generating redundant information. So, usually after each convolutional layer, there are pooling layers that were created in order to reduce the variance of features by computing some operation of a particular feature over a region of the image. Specifically, a fixed-size window runs over the features extracted by the convolutional layer and, at each step, a operation is made to select some features. Usually, two operations may be realized on the pooling layers: the max or mean operation, which selects the maximum or mean value over the feature region, respectively. Figure 3.8 presents an example of a pooling layer using max operation over the features. This process ensures that the same result can be obtained, even when image features have small translations or rotations, being very important for object classification and detection. So, the pooling layer is responsible for sampling the output of the convolutional one preserving the spatial location of the image, as well as selecting the most useful features for the next layers.

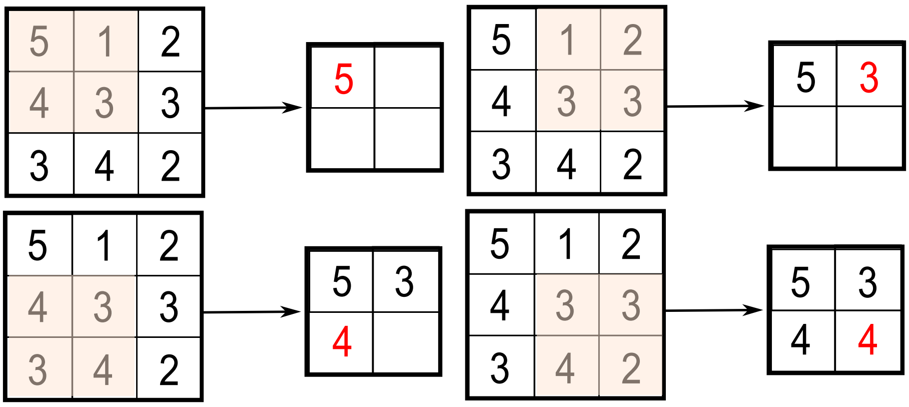


Figure 3.8: A pooling layer selecting the max value between the features inside a window of size 2x2. Figure adapted from Ng et al. [2011b]

After several convolutional and pooling layers, there are the fully-connected ones. This layer, considered as a Multilayer Perceptron Network, is responsible for the high-level reasoning of the network. It takes all neurons in the previous layer and connects it to every single neuron it has. The previous layers can be convolutional, pooling or fully-connected, however the next ones must be fully-connected until the classifier layer, because the spatial notion of the image is lost in a fully-connected layer. Since a fully connected layer occupies most of the parameters, overfitting can easily happen. To prevent overfitting, the dropout method, proposed by Srivastava et al. [2014], was created. This method randomly drops several neuron outputs, which does not contribute to the forward pass and backpropagation anymore. Usually, in the input layer, the probability of dropping a neuron is between 0.5 and 1, while in the hidden layers, a probability of 0.5 is used. This neuron drops are equivalent to decreasing the number of neurons of the network, improving the speed of training and making model combination practical, even for deep neural networks. Although this method creates neural networks with different architectures, those networks share the same weights, permitting model combination and allowing that only one network is needed at test time.

Finally, after all convolution, pooling and fully-connected layers, a classifier layer may be used to calculate the class probability of each instance. The most common classifier layer is the softmax one (Alpaydin [2010]), based on the namesake function, although there are others, such as PKM-SVM (Lazebnik et al. [2006]). The softmax function, or normalized exponential, is a generalization of the multinomial logistic function that generates a K-dimensional vector of real values in the range (0, 1) which represents a categorical probability distribution. Equation 3.5 shows how softmax function predicts the probability for the jth class given a sample vector x.

$$P(y = j|x) = \frac{\exp^{x^T w_j}}{\sum_{k=1}^{K} \exp^{x^T w_k}} \tag{3.5}$$

where j is the current class being evaluated, x is the input vector and w represent the weights.

In addition to all these processing layers, there are also ones responsible to process the data in some special way, such as normalization layers. Several methods to normalize the data may be used, such as local response, local contrast, Gaussian and MinMax normalization. Between these, the Local response normalization (LRN)(Krizhevsky et al. [2012]) is the most useful one when using processing units with unbounded activations (like ReLU), because it permits the local detection of high-frequency features with a big neuron response, while damping responses that are uniformly large in a local neighborhood.

3.4.2.1 Training Neural Networks

To perform some task and achieve satisfactory results, a multilayer neural network, composed with the presented layers, needs to minimize the loss by right classifying the instances. In order to do this, some differentiable cost function, that models the network, is needed. Several functions have been used in NN through the years, such as quadratic loss and logarithm loss (or log loss). The quadratic loss function is more common, for example, when using least squares techniques. It has some interesting properties, like being symmetric, i.e., an error above the target causes the same loss as the same magnitude of error below the target. However, the log loss has become more pervasive because of exciting results achieved in some problems (Krizhevsky et al. [2012]). When a NN implements the softmax function as the classifier layer, log loss regime is used as cost function. Equation 3.6 presents a general log loss function.

$$\mathcal{J}(\theta) = -\sum_{i=1}^{N} \left(y^{(i)} \times \log x^{(i)} + (1 - y^{(i)}) \times \log(1 - x^{(i)}) \right) \tag{3.6}$$

where y represents a possible class, x is a instance and N represents the total number of instances.

With the cost function defined, the NN can be trained in order to minimize the loss by using some optimization algorithm, such as Stochastic Gradient Descent (SGD), to gradually update the weights and bias in search of the optimal solution. However, to use this approach, the partial derivatives of the cost function, for the weights and bias, are needed. To obtain these derivatives, the backpropagation algorithm is used. Specifically, it must calculate how the error changes as each weight is increased or decreased slightly. The algorithm computes each error derivative by first computing the rate at which the error changes as the activity level of a unit is changed. For classifier layers, this error is calculated considering the predicted and desired output. For other layers, this error is propagated by considering the weights between each pair of layers and the error generated in the most advanced layer.

So, training a NN occurs in two steps: (i) the feed-forward one, that passes the information through all the network layers, from the first until the classifier one, and (ii) the backpropagation one, which calculates the error δ generated by the NN and propagates this error through all the layers, from the classifier until the first one. As presented, this step also uses the errors to calculate the partial derivatives of each layers for the weights and bias. Formally, the training process is presented in Algorithm 1.

Algorithm 1: Training process of a NN.

```
Data: Image
Result: Trained NN
for first layer until last one perform feed-forward pass do
    $z^{(l+1)} \leftarrow W^{(l)} \times a^{(l)} + b^{(l)}$;
    $a^{(l+1)} \leftarrow f(z^{(l+1)})$;
for the classifier layer $n_l$ do
    calculate the error $\delta$
    $\delta^{(n_l)} \leftarrow -(y - a^{(n_l)}) \times f' \left(z^{(n_l)}\right)$;
for each other layer $l = n_l - 1, n_l - 2, n_l - 3, \cdots, 2$ do
    propagate the error through all layers
    $\delta^{(l)} \leftarrow (W^{(l)}\delta^{(l+1)}) \times f' \left(z^{(l)}\right)$;
    calculate the partial derivatives for weights and bias
    $\nabla_{W^{(l)}} \mathcal{J}(\theta) \leftarrow \delta^{(l+1)} a^{(l)}$;
    $\nabla_{b^{(l)}} \mathcal{J}(\theta) \leftarrow \delta^{(l+1)}$;
```

Chapter 4

Clothing Annotation

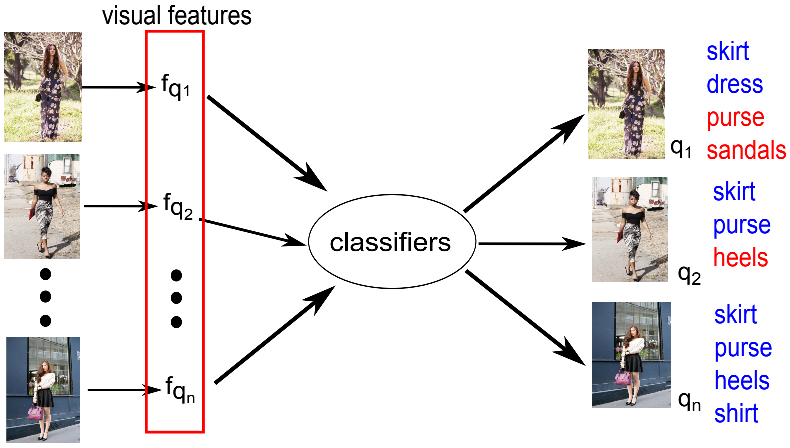


Figure 4.1: Illustration of pointwise approach. Predicted labels in blue represent right labels while red ones represent wrong predictions.

In this chapter, we present our pointwise and pairwise algorithms for automatic clothing annotation, as well as the experimental protocol and results obtained with these approaches. Figure 4.1 shows an overview of the pointwise approach, where an input consists of a single image. Figure 4.2 shows an overview of the pairwise approach, where pairs of images are given as input to the classifiers already trained also with paired images. By doing this, our method calculate the distance between the images, which makes the approach more sparse, when compared with the literature. Both algorithms build classifiers on a demand-driven basis and each classifier returns membership probabilities for each label. The final set of labels to predict comes by minimizing the entropy of such membership probabilities.

It is important to emphasize that images posted in online social networks (in particular those related to clothing) may contain both visual and textual elements, and each modality may be analyzed in a variety of ways. For instance, visual elements can be analyzed based on color, texture, shape, and so on. In turn, textual elements,such as tags or comments, may include terms related to garment items. Specifically, we observed that:

1. Images sharing common garment items are likely to share similar visual elements (e.g., color, texture and shape), and,

2. People tend to use similar tags with images that share common garment items.

These similarities are exploited by both approaches to create classifiers capable of associating similar clothes and then annotate images. Next, the formalism of our proposed methods is presented. Section 4.1 presents the proposed methods for clothing annotation. The experimental protocol used is presented in Section 4.2, while the results obtained are presented in Section 4.3.

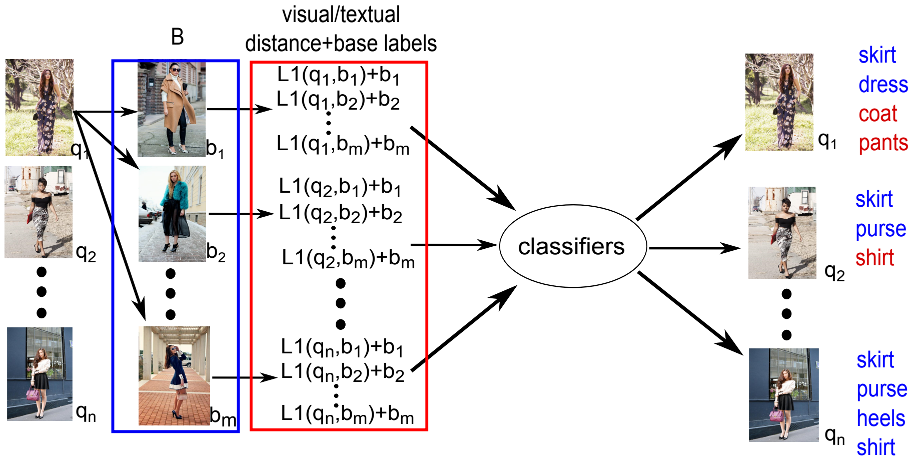


Figure 4.2: Illustration of pairwise approach. In this case, the classifiers are already trained with paired images as well. Predicted labels in blue represent right labels while red ones represent wrong predictions.

4.1 Machine Learning Approches for Clothing Annotation

In this section, we present all the details of the proposed approaches, including algorithms to combine the results of the pointwise approach, as well as methods to select the labels that should be assigned to an image. Section 4.1.1 presents the pointwise approach while the pairwise approach is presented in Section 4.1.2. To simplify its complexity, the algorithm used to define which labels should be assigned to the query image, presented in Section 4.1.1.1, is introduced considering only the pointwise method. However, this algorithm, so-called Minimum Description Length (MDL), is also used in the pairwise approach. We introduce some proposed methods to combine the results of the pointwise approach in Section 4.1.1.2.

4.1.1 Pointwise Approach

Our pointwise algorithm for automatic clothing annotation named Multi-modal/Multi-Label Clothing Annotation, or simply MMCA, is presented in this section. For this approach, it is provided a set of single images as input (Liu [2009]). Each image has its features, i.e., its visual and textual descriptors. Definition 3 formally describes the input of our approach.

Definition 3. *An instance is composed by an image q associated with its labels and visual feature vector. Specifically, an instance is represented as a list $(\tilde{q}) = \{f_1, f_2, \ldots, f_m, v_1, v_2, \ldots, v_n\}$ of feature vector of size m and n labels.*

Our proposed method uses association rules (Agrawal et al. [1993]), as described in Section 3.3. The algorithm receives as input a labelled training-set $\mathcal{D}$ composed of instances, as described in Definition 3. Distances between the instances are discretized (Fayyad and Irani [1993]) and then assigned to distance intervals[1], in order to allow for the enumeration of association rules. The test-set $\mathcal{T}$ also consists of records of the form in Definition 3, except that labels are unknown. From each instance $\tilde{q} \in \mathcal{D}$, the algorithm extracts a rule-set $\mathcal{R}$ composed of garment rules used to predict labels $\mathcal{L}_{\tilde{q}}$, which approximates as accurately as possible $\mathcal{L}^*_{\tilde{q}}$, the ground-truth of the instance $\tilde{q}$. A garment rule is composed of an antecedent and a consequent, as described in Definition 1. In this case, derived from Definition 4, these rules may contain any mixture of visual and textual features in the antecedent and a label l_i (i.e., a garment item) in the consequent.

Definition 4. *A garment rule has the following form:*

$$\overbrace{\{\ f_j \wedge \ldots \wedge f_z \wedge v_t \wedge \ldots \wedge v_u\ \}}^{\textit{Distance intervals}} \xrightarrow{\theta} l_i \begin{cases} \textit{“trousers”}, \\ \textit{“skirt”}, \\ \textit{“handbag”}, \\ \textit{etc.} \end{cases}$$

where $j \geq 1$ and $z \leq m$, and $t \geq 1$ and $u \leq n$.

The operator “$\wedge$” represents that the antecedent of a rule is formed with the simple presence of a determined combination of features and labels. These combinations work like a signature to the rule.

As introduced, we denote as $\mathcal{R}_{\tilde{q}}$ garment rules applicable to instance $\tilde{q}$ according to Definition 2. Further, $\mathcal{R}^{l_i}_{\tilde{q}}$ is the subset of $\mathcal{R}_{\tilde{q}}$ containing only rules predicting label l_i. Votes in $\mathcal{R}^{l_i}_{\tilde{q}}$ have different weights, depending on the confidence and, the weighted votes for label l_i are averaged, resulting in the score $s(\tilde{q}, l_i)$, as shown in Equation 3.1. The likelihood of query image q being associated with label l_i is obtained by normalizing

[1] Hereafter we refer each f_i as the corresponding interval.

the scores, as expressed by $\hat{p}(l_i|\tilde{q})$ shown in Equation 3.2. This normalization occurs to restrict the sum of the likelihood of all labels of an specific instance to exactly one. So, at the end, we have a ranking with the labels and its probability for each instance. In this ranking, higher values of $\hat{p}(l_i|\tilde{q})$ indicate that the label is likely to be associated with q.

4.1.1.1 Minimum Description Length

The minimum description length (MDL) principle is a powerful method of inductive inference, the basis of statistical modeling, pattern recognition and machine learning. It is based on the Razor of Occam and was first proposed by Rissanen [1978]. It holds that the best explanation, given a limited set of observed data, is the one that permits the greatest compression of the data. This strategy was explored before to disambiguate entity names (Davis et al. [2012]) and in our case, was adapted to be used when defining the labels that should be assigned to an image.

So, considering that each instance has some candidate labels (and its probability), the MDL approach is used to find the best cut for these instances, selecting which labels should be assigned for each one of them. This cut is made based on a validation set and it is more robust than the top-k approach typically used on multi-labels problems (Veloso et al. [2007]). More specifically, given an instance $\tilde{q}$ and a set of candidate labels $\mathcal{L}_{\tilde{q}}$ provided by the classifier,[2] we must find a cut point $c_{\tilde{q}}$ which delimits labels that are likely to be associated with the query image from those that are not. In other words, we must find a threshold $c_{\tilde{q}}$, so that only labels in $\mathcal{L}_{\tilde{q}}$ for which $\hat{p}(l_i|\tilde{q}) > c_{\tilde{q}}$ are finally predicted.

Our approach searches for a threshold $c_{\tilde{q}}$ that provides the best entropy cut in the space induced by probabilities $\hat{p}(l_i|\tilde{q})$ $\forall$ $l_i \in \mathcal{L}_{\tilde{q}}$. Figure 4.3 illustrates our approach. In the figure, symbol $\oplus$ indicates that the corresponding label l_i is associated with query image q. Similarly, symbol $\ominus$ indicates that the corresponding label l_i is not associated with query image q. Therefore, in the example, labels $\{l_4, l_5, l_6\}$ are associated with q (i.e., $\oplus$), while labels $\{l_1, l_2, l_3\}$ are not (i.e., $\ominus$). The figure shows three possible cut points for the instance, and the best entropy cut is exactly the one which minimizes the overall entropy in the probability space.

Obviously, there are more difficult cases, for which it is not possible to obtain a perfect separation in the probability space, but our approach is general enough to handle such harder cases. The basic idea is that any value of $c_{\tilde{q}}$ induces two partitions over the space of values for $\hat{p}(l_i|\tilde{q})$, that is, one partition with probabilities that are lower than $c_{\tilde{q}}$, and another partition with probabilities higher than $c_{\tilde{q}}$. Our approach sets $c_{\tilde{q}}$ to the

[2] Labels for which $\hat{p}(l_i|\tilde{q}) > 0$.

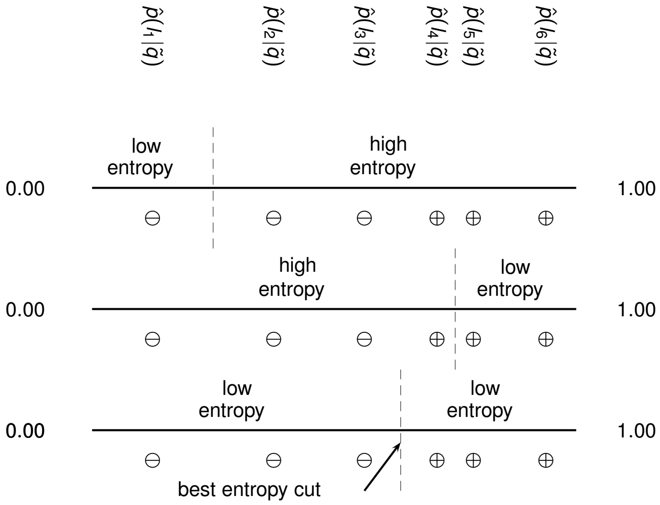


Figure 4.3: Looking for the minimum entropy cut for a specific instance $\tilde{q}$. Figure adapted from Davis et al. [2012]

value that minimizes the average entropy of these two partitions. The idea is formally presented in Definition 5.

Definition 5. *Consider a list $\mathcal{O} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$ where $x_j \in \{\ominus, \oplus\}$ and y_j is a membership probability $\hat{p}(l_i|\tilde{q})$. The list is sorted such that $y_j \leq y_{j+1}$. Also consider c as a candidate value for $c_{\tilde{q}}$. In this case, $\mathcal{O}_c(\leq)$ is a sublist of $\mathcal{O}$ for which the condition $y_j \leq c$ holds for all $(x_j, y_j) \in \mathcal{O}_c(\leq)$. Similarly, $\mathcal{O}_c(>)$ is a sublist of $\mathcal{O}$ for which the condition $y_j > c$ holds for all $(x_j, y_j) \in \mathcal{O}_c(>)$. In other words, both $\mathcal{O}_c(\leq)$ and $\mathcal{O}_c(>)$ are partitions of $\mathcal{O}$ induced by c.*

Firstly, our approach calculates the entropy in $\mathcal{O}$, as shown in Equation 4.1. Then, it calculates the sum of the entropies in each partition induced by c, according to Equation 4.2. Finally, it sets $c_{\tilde{q}}$ to the value of c that minimizes $E(\mathcal{O}) - E(\mathcal{O}_c)$.

$$\begin{aligned} E(\mathcal{O}) = \; & - \left(\frac{\mathrm{N}_{\ominus}(\mathcal{O})}{|\mathcal{O}|} \times \log \frac{\mathrm{N}_{\ominus}(\mathcal{O})}{|\mathcal{O}|} \right) \\ & - \left(\frac{\mathrm{N}_{\oplus}(\mathcal{O})}{|\mathcal{O}|} \times \log \frac{\mathrm{N}_{\oplus}(\mathcal{O})}{|\mathcal{O}|} \right) \end{aligned} \tag{4.1}$$

where $\mathrm{N}_{\ominus}$ gives the number of labels in $\mathcal{L}_{\tilde{q}}$ but not in $\mathcal{L}^*_{\tilde{q}}$, and $\mathrm{N}_{\oplus}$ gives the number of labels in $\mathcal{L}_{\tilde{q}}$ and also in $\mathcal{L}^*_{\tilde{q}}$.

$$\begin{aligned} E(\mathcal{O}_c) = \frac{|\mathcal{O}_c(\leq)|}{|\mathcal{O}|} \; & \times \; E(\mathcal{O}_c(\leq)) + \\ \frac{|\mathcal{O}_c(>)|}{|\mathcal{O}|} \; & \times \; E(\mathcal{O}_c(>)) \end{aligned} \tag{4.2}$$

To use the MDL approach, we employ a validation-set $\mathcal{V}$ composed of several instances $\tilde{q}$, so that both the true labels $\mathcal{L}^*_{\tilde{q}}$ and the predicted labels $\mathcal{L}_{\tilde{q}}$ are previously known for all instances in this set. Our goal is to build a function $\gamma(\mathcal{L}_{\tilde{q}})$ which receives as inputs a set of candidate labels $\mathcal{L}_{\tilde{q}}$ and returns the best entropy cut for these labels, predicting the labels. Thus, the function $\gamma(\mathcal{L}_{\tilde{q}})$ gives the mean of the best entropy cuts associated with instances $\tilde{q} \in \mathcal{V}$ having $\mathcal{L}_{\tilde{q}}$ as candidate labels. Equation 4.3 presents this mean. If there is no instances $\tilde{q} \in \mathcal{V}$ having specifically the candidate labels, then the function returns a mean of best cuts of all instances in the validation set.

$$\gamma(\mathcal{L}_{\tilde{q}}) = \frac{\sum c_{\tilde{q}}^{\mathcal{L}_{\tilde{q}}}}{\mathrm{N}_{\mathcal{L}_{\tilde{q}}}} \tag{4.3}$$

where $c_{\tilde{q}}^{\mathcal{L}_{\tilde{q}}}$ are best entropy cuts associated with the candidate labels $\mathcal{L}_{\tilde{q}}$ and $\mathrm{N}_{\mathcal{L}_{\tilde{q}}}$ is the number of validation instances associated with these labels.

4.1.1.2 Combination Methods Using MMCA

The combination methods proposed in this work join classifiers that use different visual features looking for improvements in the overall accuracy. The proposed algorithms may appear very similar to some ensemble methods in the literature, like bootstrap aggregating or bagging, but they differ from them because: (i) the classifiers are trained with different features (ii) the training set used is always the same for every classifiers (only the features used are different), and (iii) the misclassification of a classifier is never used again.

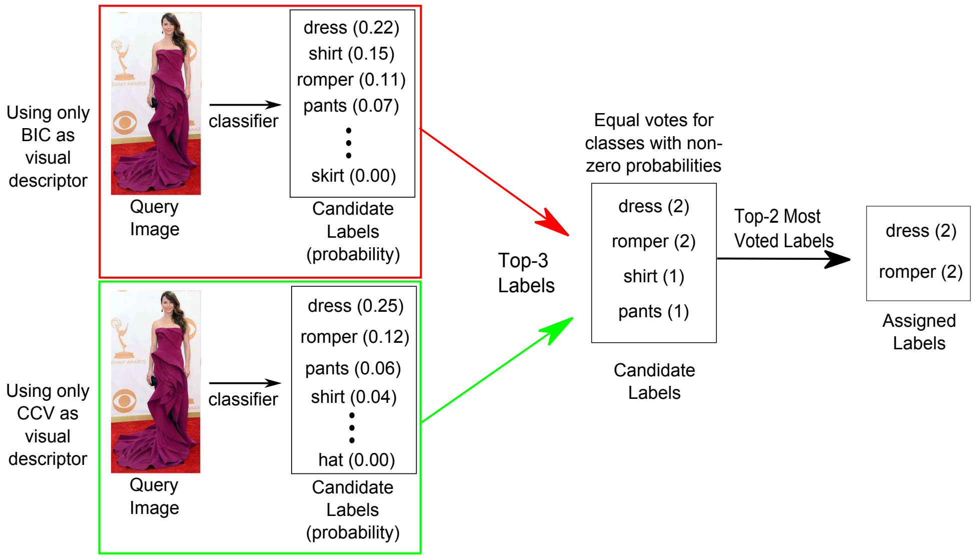


Figure 4.4: Illustration of a proposed combination of the MMCA approach considering only BIC and CCV descriptors: the majority voting consider each class, with probability more than zero, as a vote with equal weight. A top-k defines which labels should be assigned.

First combination method, called Majority Voting (MV), gives each candidate label the same weight when voting. More specifically, for each instance a classifier generates, as presented, a ranking with the labels and its probability. This ranking is pruned using a top-k approach, and then, each remaining label (the ones with higher probability) gives an equal vote, creating a final ranking ordered by the votes. This final ranking is pruned again (also using a top-k method), resulting in the final set of labels that is assigned to the image. Figure 4.4 presents a example of this method considering classifiers trained using BIC and CCV visual descriptors.

The second proposed combination method, called Majority Probability (MP), gives each candidate label a weight (equal its probability) when voting. Specifically, for an instance, the method generates a final ranking by calculating the mean probability of each label considering all the rankings. Then, the final rank is pruned in top-k way. Figure 4.5 presents a example of this method considering classifiers trained using BIC and CCV visual descriptors.

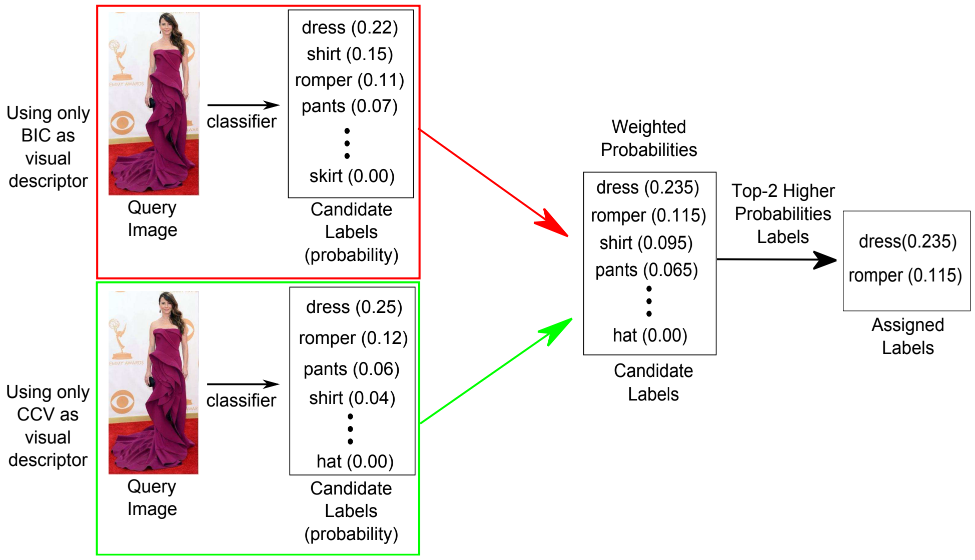


Figure 4.5: Illustration of a proposed combination of the MMCA approach considering only BIC and CCV descriptors: the majority probability calculates the mean of all labels and a top-k is used to define which labels should be assigned.

4.1.2 Pairwise Approach

In this section, we present our pairwise algorithm for automatic clothing annotation named Multi-Modal/Multi-Label/Multi-Instance Clothing Annotation algorithm (or simply M3CA). For this approach (Liu [2009]), pairs of images are provided as input to our classification algorithm. A pair of images is denoted as an instance, as described in Definition 6.

Definition 6. *An instance* $(\widetilde{qb}) = (q, b)$ *is composed by a base image* b *and a query image* q*. Labels associated with the base image* b *are called* base labels *and are always known in advance. Labels associated with the query image* q *must be predicted. An instance is represented by a set of (visual and textual) distances between* q *and* b*, along with the base labels. Specifically, an instance is represented as a list* $(\widetilde{qb}) = (q, b) = \{f_1, f_2, \ldots, f_m, v_1, v_2, \ldots, v_n\}$ *of* m *distance values and* n *base labels.*

It is important to highlight that the L1 distance function[3] was used to calculate the similarity between two images, since it is suitable for generating sparse vectors due to its property in producing results with zero or very small values.

Our proposed multi-instance clothing annotation algorithm also uses association rules (Agrawal et al. [1993]) to produce classifiers that predict garment items associated with an arbitrary image, as presented in Section 3.3. The algorithm receives as input a labelled training-set $\mathcal{D}$ composed of records of the form $< q, B >$, where q is a query image and B is a bag of base images. The bag B is partitioned into multiple instances of the form $(\widetilde{qb}, \mathcal{L}_q^*) = ((q, b), \mathcal{L}_q^*)$, where $b \in B$ and $\mathcal{L}_q^*$ is a set of labels associated with the query image q (i.e., the garment items appearing in image q). Hence, in this case, it is known in advance the base and query labels, in addition to the feature distances between q and b. The test-set $\mathcal{T}$ also consists of records of the form $< q, B >$. Again, the bag B is partitioned into multiple instances $(\widetilde{qb}, ?) = ((q, b), ?)$. In this case, however, only the distances between images q and b and the base labels are known, whereas labels $\mathcal{L}_q^*$ are unknown.

Just like the pointwise approach, the algorithm extracts a rule-set $\mathcal{R}$ composed of garment rules from each instance $\widetilde{qb} \in \mathcal{D}$. As described in Definition 4, a garment rule is composed of an antecedent, with any mixture of visual and textual features, and a consequent, with a label l_i. These rules are used to predict labels $\mathcal{L}_q$, which approximates as accurately as possible $\mathcal{L}_q^*$ (the ground-truth for query image q).

From $\mathcal{R}$, we extract $\mathcal{R}_{\widetilde{qb}} = \mathcal{R}_{(q,b)}$ that is a set of garment rules applicable to instance $\widetilde{qb}$, according to Definition 2. Thus, only rules in $\mathcal{R}_{\widetilde{qb}} = \mathcal{R}_{(q,b)}$ are considered as valid votes when predicting the labels for image q. The subset $\mathcal{R}_{\widetilde{qb}}^{l_i}$ of $\mathcal{R}_{\widetilde{qb}}$ contains only rules predicting label l_i. Like Equation 3.1, a score for each label $s(\widetilde{qb}, l_i)$ is calculated using the confidence as weight. The likelihood of query image q being associated with label l_i is obtained by normalizing the scores, as expressed by $\hat{p}(l_i|\widetilde{qb})$, shown in Equation 3.2. Just like the pointwise approach, higher values of $\hat{p}(l_i|\widetilde{qb})$ indicate lower distances between images q and b, and labels associated with b are also likely to be associated with q.

Analogously to the MMCA aproach, the M3CA algorithm also needs to build the function $\gamma(\mathcal{L}_{(\widetilde{qb})})$ to select the labels that should be assigned to the query image q. However, instead of using the function to directly predict the labels, as the MMCA approach, the M3CA needs to aggregate different instances related to a same query image q to, finally, predict the labels using the MDL algorithm.

More specifically, a query image q may appear within several (i.e., n) instances $(\widetilde{qb}_i) = (q, b_i) \in \mathcal{T}$. For each instance $(\widetilde{qb}_i) = (q, b_i) \in \mathcal{T}$ a specific set of labels $\mathcal{L}_{(\widetilde{qb}_i)}$ is associated with q. The final set of labels to be predicted is given as $\mathcal{L}_q = \{\mathcal{L}_{(\widetilde{qb}_1)} \cup \mathcal{L}_{(\widetilde{qb}_2)} \cup \ldots \cup \mathcal{L}_{(\widetilde{qb}_n)}\}$. After aggregating the labels, we use the best entropy cut to predict

[3] L1 distance function calculates the difference between two feature vectors by summing the absolute value of each keyword: $L1 = \sum_{i=1}^{N} |p_i - q_i|$

Table 4.1: Datasets.

	pose.com	chictopia.com
Number of photos	2,306	1,579
Number of tags	7,501	5,093
Tags per photo	3.25	3.23

the labels that are associated with q.

4.2 Experimental Protocol

In this section, we present the experimental setup we used in the first phase of this dissertation. For the clothing annotation task, we carried out experiments in two scenarios:

1. Ideal scenario: used to evaluate the visual descriptors and their best configuration (for example, the size of the visual dictionary). To achieve this, we created a small dataset composed of 100 images, and to avoid the effects of the background, we performed a manual segmentation of each image. In this scenario, we have single-class classification (only one class should be assigned to each image) with 10 images per class. Therefore, we use only the MMCA approach considering that the label with higher probability is assigned to the image.

2. Realistic scenario: used to analyze the proposed algorithms (MMCA and M3CA) and the baseline. In this scenario, we used two datasets crawled from social networks. Each image may have more the one label (tag), that represents the garment items, which makes the scenario a multi-label classification. The segmentation was made automatically using a pose estimation algorithm, proposed by Yang and Ramanan [2011].

In this section, we distinguish some differences between the scenarios. Section 4.2.1 presents some statistics of the datasets used. The visual and textual features used are presented in Section 4.2.2. Section 4.2.3 presents the baselines used in this work. The experimental protocol used are presented in Section 4.2.4. Finally, Section 4.2.5 presents the measures used to evaluate the experiments.

4.2.1 Datasets

As presented, the ideal scenario was designed to study the impact of the visual features over the overall accuracy in an attempt to avoid any external or unadvised error. This scenario consists of a dataset of 100 images (10 classes with 10 images per class) crawled from instagram.com between October 11 and November 10, 2013.

We chose to work with the realistic scenario to evaluate the performance of our method in a more real situation. Thus, we have crawled images and associated tags from two fashion-related social networks, namely `pose.com` and `chictopia.com`. Basic information about the resulting datasets is shown in Table 4.1. The `pose.com` dataset was crawled from January 15, 2014 to January 25, 2014. This resulted in more than three thousand images. The `chictopia.com` dataset was crawled from January 25, 2014 to February 5, 2014 resulting in more than two thousands images. At the end, the whole dataset for our realistic scenario is composed of approximately five thousands images. Combining labels from both datasets leads us to a set of 31 discrete possibilities, including “trousers”, “glasses”, “shirts”, “shoes”, and “sneakers”.

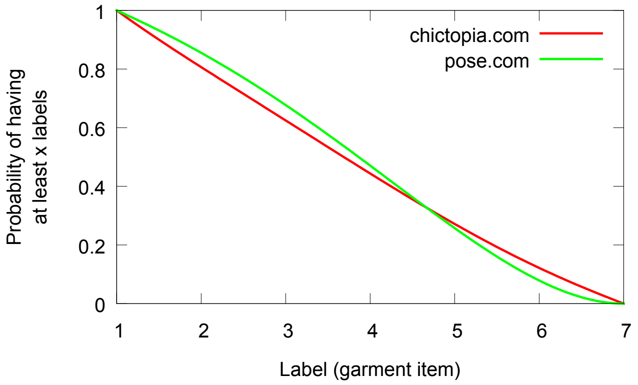
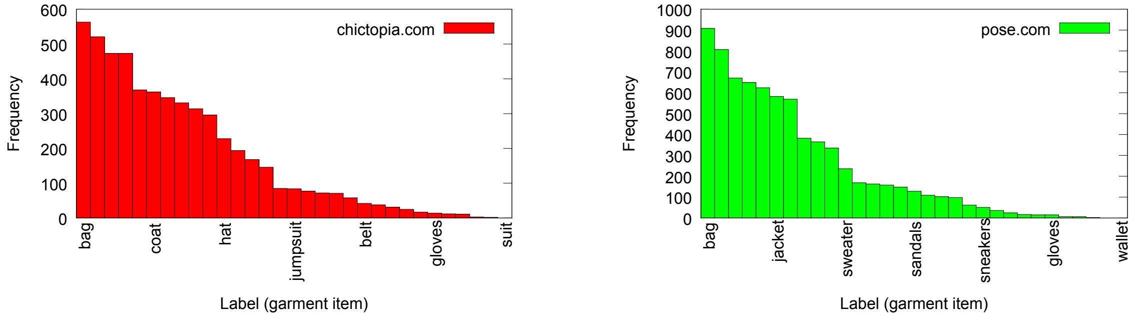


Figure 4.6: Cumulative distribution function of labels for both datasets.

Figure 4.7 shows the frequency of each label. As expected, some labels occur frequently (e.g., “shirt”, “jeans”, and “coat”), while others occur only few times (e.g., “tie”, “stockings”, and “romper”). Figure 4.6 shows the cumulative distribution function for labels in `chictopia.com` and `pose.com`. The probability for an arbitrary image having at least x labels decreases almost linearly in both cases.



(a) (b)

Figure 4.7: Frequency distribution related to the dataset: (a)-(b) for Chictopia and Pose, respectively.

When working with visual image descriptors a pose estimation and image segmentation is needed since the background may make features more noise. Thus, in order to avoid the effect of background pixels over the description of the image, we have created a mask to separate the relevant pixels. For the ideal scenario, the mask creation was made manually for each image. Figure 4.8 shows the original image and relevant pixels.

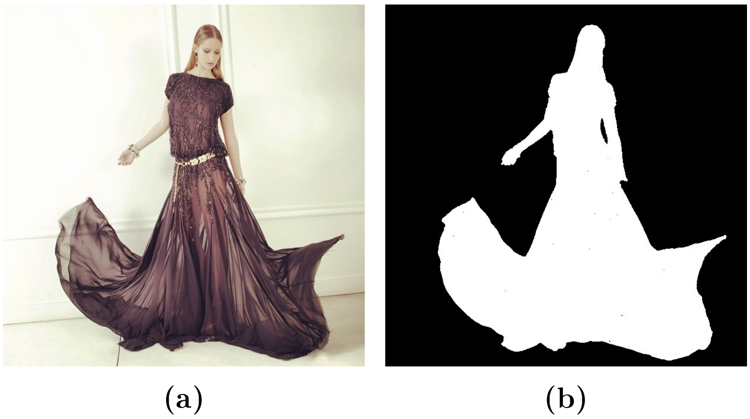


Figure 4.8: Example of manual segmentation: (a) Original image (b) Pixels of interest in white.

For the realistic scenario, we ran a human pose estimation algorithm (Yang and Ramanan [2011]) and then, create a mask based on the skeleton generated. Specifically, using the skeleton that estimates the human pose, we employ a factor of proportionality in order to enlarge each line of this estimation, thus encompassing the entire pose. So, we separate the pixels and obtain the final set of relevant pixels (i.e., non-background pixels).

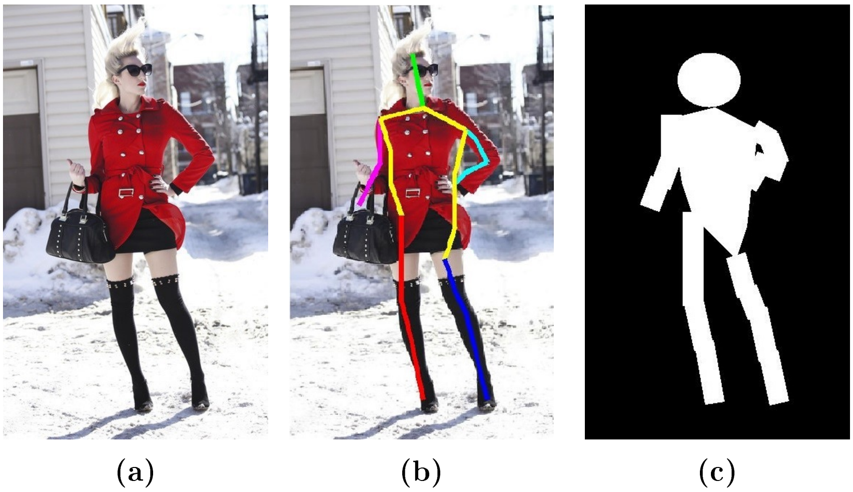


Figure 4.9: Example of automatic segmentation: (a) Original image (b) Skeleton generated by Yang and Ramanan [2011] (c) Pixels of interest in white.

To use this mask, an adaptation was made in the global descriptor algorithms while they extract the features. For BoW and BossaNova approaches, this mask was used as an intermediate step to select only relevant points before creating the visual dictionary, since was used a dense sampling. Figure 4.9 shows the original image, a pose estimation skeleton (Yang and Ramanan [2011]) and relevant pixels.

We discarded images according to the proportion of background/non-background pixels. More specifically, we discarded all images for which the proportion of relevant pixels (non-background pixels) is lower than a fixed threshold α_{min}. We evaluate the impact of this parameter over the results in Chapter 4.3. Table 4.2 shows the number of remaining images (i.e., the final dataset) for different values of α_{min}.

Table 4.2: Images with enough relevant pixels.

α_{min}	pose.com	chictopia.com
0.05	1,308	1,257
0.10	969	937
0.15	578	421

4.2.2 Features

As introduced, visual and textual features are exploited by our methods in order to create more efficient classifiers. Considering the visual features, there is a myriad of visual image descriptors available in the literature (Zhang and Lu [2004]) and choosing the most appropriate descriptors for a determined problem is a hard task, since different descriptors may produce different results. One contribution of this work is to define the most interesting descriptors to solve the clothing annotation problem. We evaluated 10 global and 6 local feature extraction techniques, as presented in Section 3.1. The global descriptors were evaluated using the raw features extracted while the local ones were transformed into some mid-level representation and then evaluated. As introduced in Section 3.2, each local descriptor were evaluated using two different mid-level representations: BoW (Sivic and Zisserman [2006]) and BossaNova (de Avila et al. [2011]). For the former approach, a hard assignment was chosen at the coding stage. This occur in order to try a more sparse histogram, that tends to be easier to learn with. At the pooling stage, a max method was chosen. For BossaNova technique, after creating the visual features a normalization is made to get a more sparse histogram. As presented in Section 3.2.2, localized soft assignment was chosen at the coding stage while a BossaNova pooling was chosen at the pooling step. It is important to highlight that, for all this techniques, we used a dense sampling to get the patches, and then, we extract the features.

The textual features are represented by the tags associated with an image, which may bring useful information of photos that, associated with visual features, may help creating a more robust application for image annotation. For the realistic scenario, we created a vocabulary containing relevant terms related to different garments items using the tags crawled with the images. After filtering out all terms not in the vocabulary, the remaining textual content is described with TF-IDF vectors. The TF-IDF transformation weights each term according to its discriminative capacity. Textual similarity between two images is assessed using the standard cosine and BM25 measures (Baeza-Yates and Ribeiro-Neto [2011]).

It is important to emphasize here that the focus of this work is to automatic annotate clothes based on the visual features. We use the textual ones looking only for improvements of the results and to create a more robust method. Thus, we did a extensive evaluation of the visual descriptors, leaving the evaluation of the textual features as a future work.

4.2.3 Baseline

To evaluate the methods proposed in this work, we considered the M3LDA algorithm, proposed by Nguyen et al. [2013] as the baseline. This algorithm is a representative of the state-of-the-art in multi-label, multi-modal and multi-instance classification. It uses Latent Dirichlet Allocation (LDA) to create a rank with the most likely labels of each test image. It provides superior mean Average Precision (mAP) numbers when compared against popular algorithms such as two MIML models RankLoss (Briggs et al. [2012]), DBA (Yang et al. [2009]), and two annotation models that allow region annotation, TM (Duygulu et al. [2002]) and Corr-LDA (Blei and Jordan [2003]).

4.2.4 Cross Validation

For both scenarios, we conducted k-fold cross-validation in order to evaluate the algorithms. According to this protocol, a dataset is randomly split into k mutually exclusive subset (folds) of almost the same size. For the ideal scenario, the $k-1$ subsets are chosen as training set, and the remaining one is the test set. To work with all the dataset, the cross-validation process is repeated k times, and each time a subset is chosen to be the test set (without repetition). For the realistic scenario, $k - 2$ subsets are chosen as training set, one fold is used as test-set, and the remaining one is the validation-set (i.e., in order to build the MDL function). The last subset is only used in the latter scenario, because in the former we predict only one class per image, so there is no need of the MDL function. The process is repeated k times, and each time a subset is chosen to be the validation set while other subset is chosen to be the test one (without repetition), working with all dataset. At the end, the cross-validation estimate the arithmetic mean of all runs and the standard deviation between each one. The results reported are the average of the five runs.

Table 4.3 presents the cross-validation used in different scenarios. For the ideal scenario, where we only use the MMCA approach, we made the experiments with with cross-validation without the validation-set, since this scenario is composed of single-class classification. For the realistic scenario, we used cross-validation with the validation-set when working with both approach, since this is a multi-label scenario, when we need to build the MDL function to define which labels should be assigned.

Table 4.3: Cross-validation in different scenarios.

	Ideal Scenario (single-class)	**Realistic Scenario (multi-label)**
MMCA	✓	✓ (MDL)
M3CA	×	✓ (MDL)

4.2.5 Evaluation Measures

To evaluate the experiments in the ideal scenario, we used the overall accuracy. For the realistic scenario, which may be categorized into a multi-label classification, we used the Jaccard distance as evaluation measure. Specifically, given the correct set of labels $\mathcal{L}_q^*$ and the predicted set of labels $\mathcal{L}_q$ for each query image q in the test-set $\mathcal{T}$, the Jaccard distance J is given as shown in Equation 4.4.

$$J = \frac{\sum \frac{|\{\mathcal{L}_q^* \cap \mathcal{L}_q\}|}{|\{\mathcal{L}_q^* \cup \mathcal{L}_q\}|}}{\mathrm{N}_q} \tag{4.4}$$

where N_q is the number of distinct query images in $\mathcal{T}$.

4.3 Results and Discussion

In this section, we present the experimental results to evaluate: (i) visual features, and (ii) proposed methods. When evaluating the visual features, we build the experiments in order to investigate how clothing annotation is impacted by different types of visual features. The second set of experiments, to evaluate the proposed methods, were devised to investigate: (i) the most suitable approaches for the clothing annotation task, (ii) how each method is impacted by the proportion of relevant pixels, and (iii) how the proposed algorithms perform relatively to the baseline.

For investigating the presented items, we tested and varied some parameters to achieve more robust results. Concerning global descriptors, we have considered only the size of the association rule used on the classifier as a parameter. For BoW (Sivic and Zisserman [2006]), the parameters observed were the size of the rule as well as the size of the visual dictionary $\mathcal{K}$ (the number of keywords generated by the mid-level representation). Regarding BossaNova approach (de Avila et al. [2011]), in addition to the parameters evaluated for the BoW, we have observed the number of bins β used in the quantization step to encode the distances from one local descriptor to clusters. The default values for the size of the dictionary and the number of bins β were selected using a parameter evaluation made by de Avila et al. [2011].

In Section 4.3.1, we present the experimental evaluation of the feature descriptors and then the evaluation of the proposed approaches. For each evaluation process, we computed the mean processing time[4], in seconds, and the standard deviation based on five executions of each procedure. In Section 4.3.2, we present a comparison among the methods proposed for clothing annotation and the baseline.

4.3.1 Visual Features Evaluation

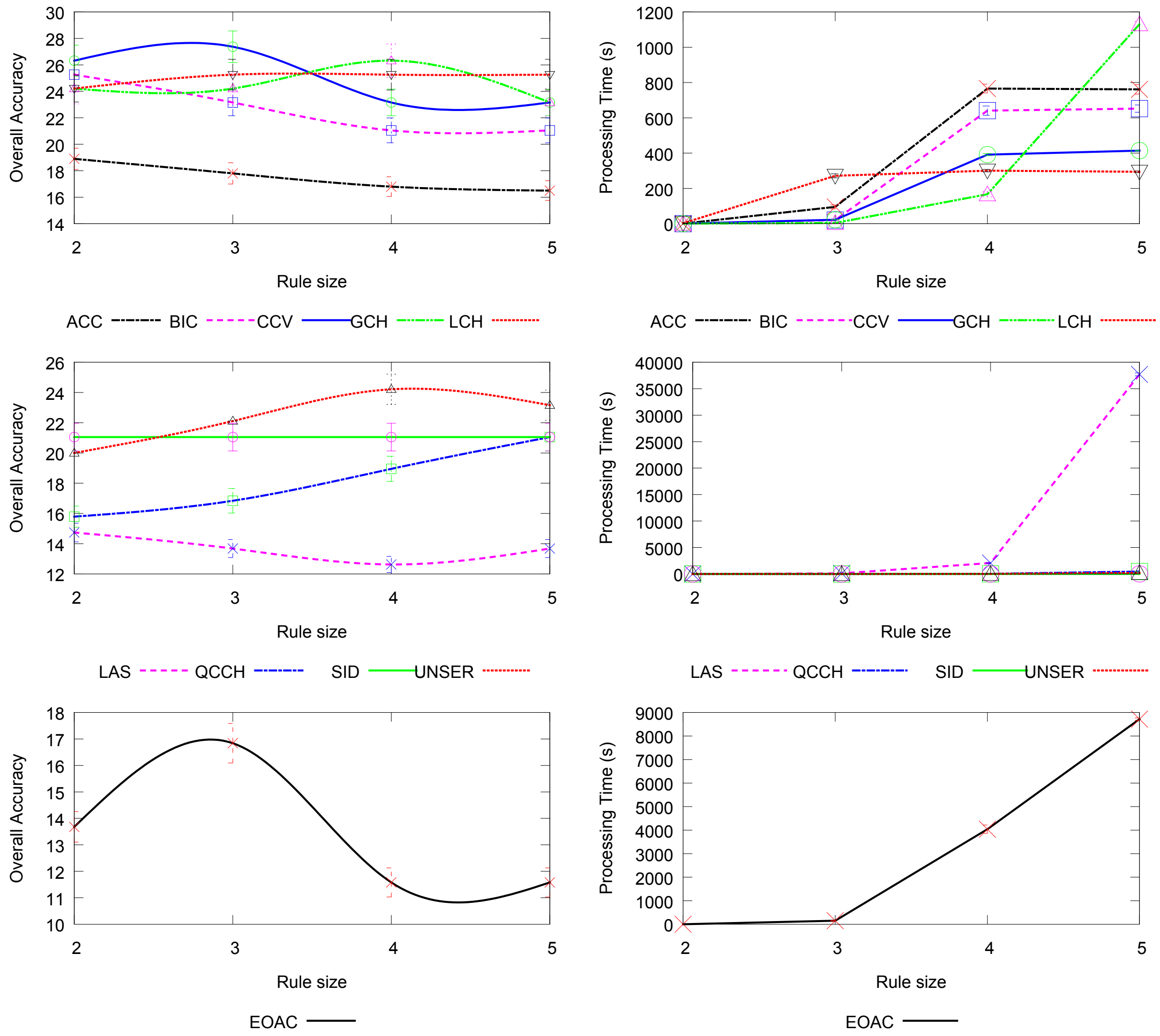


Figure 4.10: The overall accuracy (left) and the processing time (right), in seconds, obtained using global descriptors. First row shows accuracy numbers for color descriptors. Second one shows the accuracy for texture descriptors. Last row shows accuracy numbers for shape descriptor.

In this section, we present the experimental results carried out for evaluating visual descriptors. As introduced in Section 4.2, we use the overall accuracy and an ideal scenario for these experiments.

Figure 4.10 shows the overall accuracy for the global descriptors followed by the mean processing time, in seconds, for each descriptor. Each plot groups descriptors of the same type: color, texture and shape. Between the global descriptors presented in

[4] The processing time computed is only the time spent by the classification algorithm.

Section 3.1.1, the best ones yield overall accuracy around 25%, which includes BIC, CCV, GCH and LCH descriptors. ACC, EOAC, and LAS achieved lower accuracy (around 15%) and are good candidates to be discarded on our next experiments.

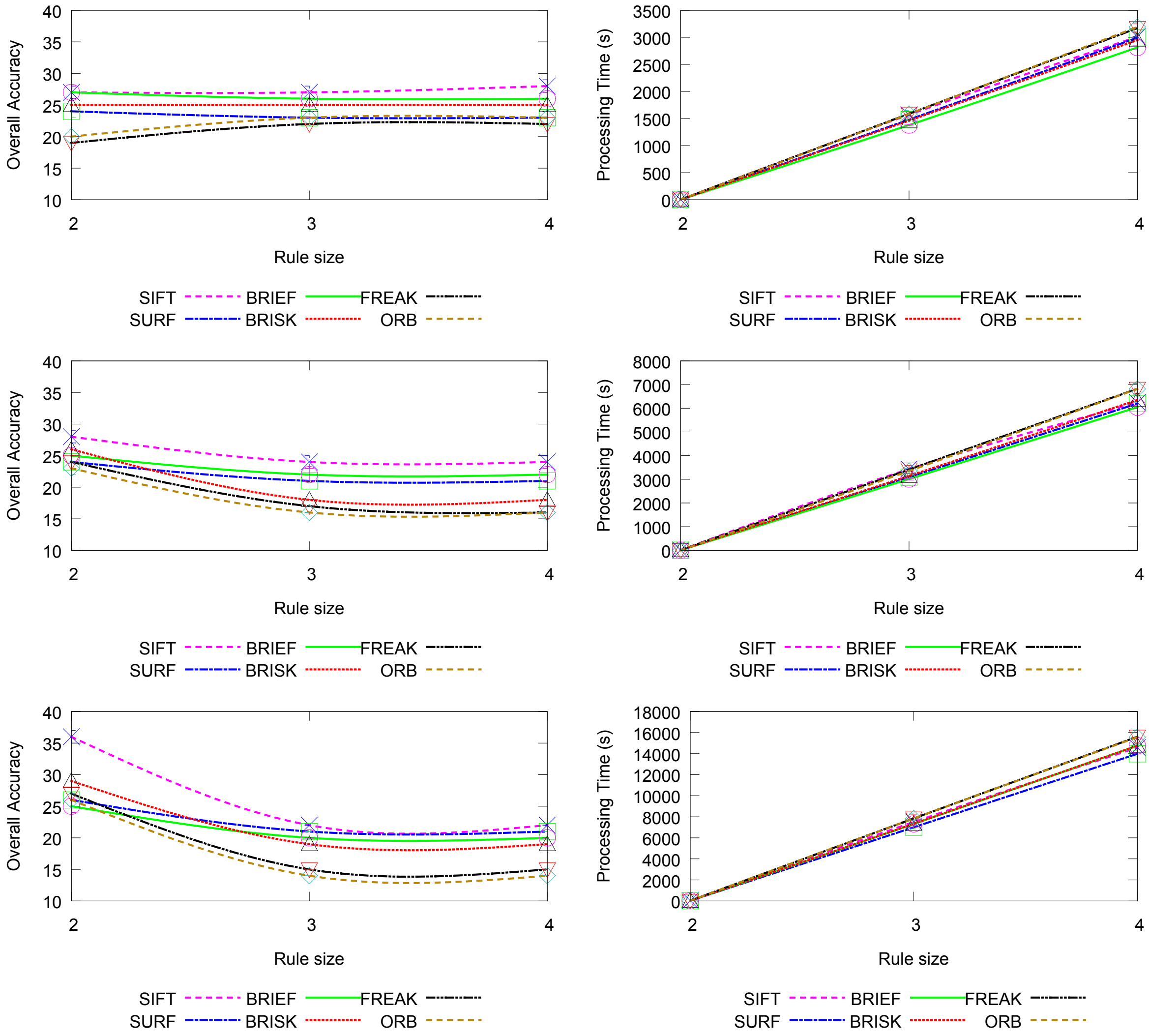


Figure 4.11: The overall accuracy (left) and the processing time (right), in seconds, obtained with BoW using hard assignment coding and max pooling in different local descriptors. First row shows the results with $\mathcal{K} = 1024$. Second one shows the results with $\mathcal{K} = 2048$, and the third shows the overall accuracy with $\mathcal{K} = 4096$.

Figure 4.11 shows the overall accuracy for the BoW using different types of local descriptors and the mean processing time of each one. The plot represents the results varying the size of the visual dictionary (or feature vector) $\mathcal{K}$, which was chosen based on a parameter study made by de Avila et al. [2011]. Through the plot, it is possible to see that SIFT descriptors yields a good accuracy with any configuration of $\mathcal{K}$. It is also possible to observe that when $\mathcal{K} = 1024$ the proposed approach spends much less time if compare with the others.

Figure 4.12 shows the overall accuracy for the BoW using SIFT descriptor and its processing time. The plot represents the results varying the size of the visual dictionary (or histogram of a image) $\mathcal{K}$. According to the plot, one can see that $\mathcal{K} = 1024$ yields

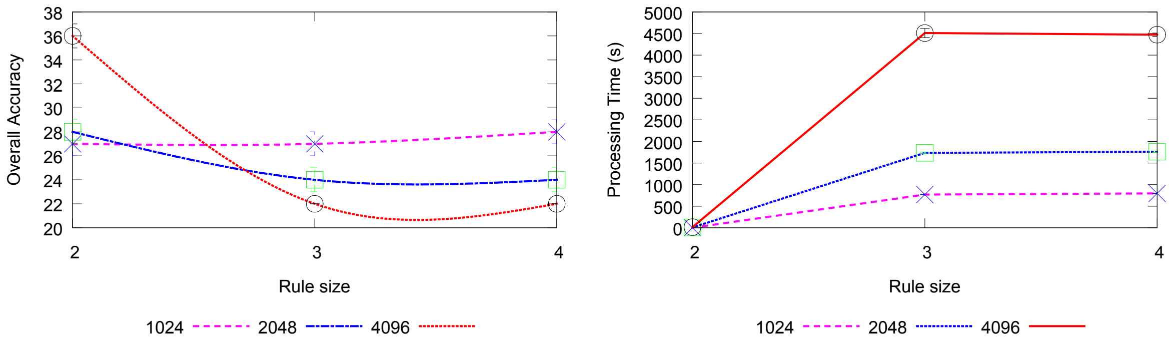


Figure 4.12: The overall accuracy (left) and the processing time (right), in seconds, obtained with BoW+SIFT, using hard assignment coding and max pooling.

a good accuracy (27%) if compared with the others. It is also the most stable, with is virtually the same accuracy for all values of rule size. However, the best results, around 36%, were achieved with $\mathcal{K} = 2048$ and rule size 2, but with the increasing of the rule size, the accuracy tends to decrease.

Figure 4.13 shows the overall accuracy of BossaNova using different types of local descriptors, followed by the mean processing time of each one. The plot represents the results varying the size of the visual dictionary (or histogram of a image) $\mathcal{K}$, which was chosen based on a parameter study made by de Avila et al. [2011], and preserving the number of bins used in the quantization step in $\beta = 2$.

Note that, according these results, SIFT descriptor is the most consistent one, since it yields good results independent of the configuration of $\mathcal{K}$. The ORB descriptors, for example, yields good results when $\mathcal{K} = 1024$ and $\mathcal{K} = 4096$, but it decreases with $\mathcal{K} = 2048$. Another example is the SURF descriptor, that yields good results when $\mathcal{K} = 2048$ and $\mathcal{K} = 4096$, but not so good with $\mathcal{K} = 1024$. It is also possible to observe that when $\mathcal{K} = 1024$ the processing time spends much less time if compare with the others.

Figure 4.14 shows the accuracy for the BossaNova approach using only SIFT descriptor, with different dictionary sizes $\mathcal{K}$ and the numbers of bins β used in the quantization step. The values were defined based on a parameter evaluation study conducted by de Avila et al. [2011]. For the parameter β, three different values were evaluated: 2, 3 and 4. However, the results were very similar for all these values. This happens due to a normalization made by the BossaNova approach while creating the histogram, since with the increase β the numbers of codewords with high value tends to decrease and the normalization tries to maintain only the codewords with higher value. Thus, we report only the results for $\beta = 2$. Through the plots, it is possible to see that $\mathcal{K} = 1024$ yields the best results.

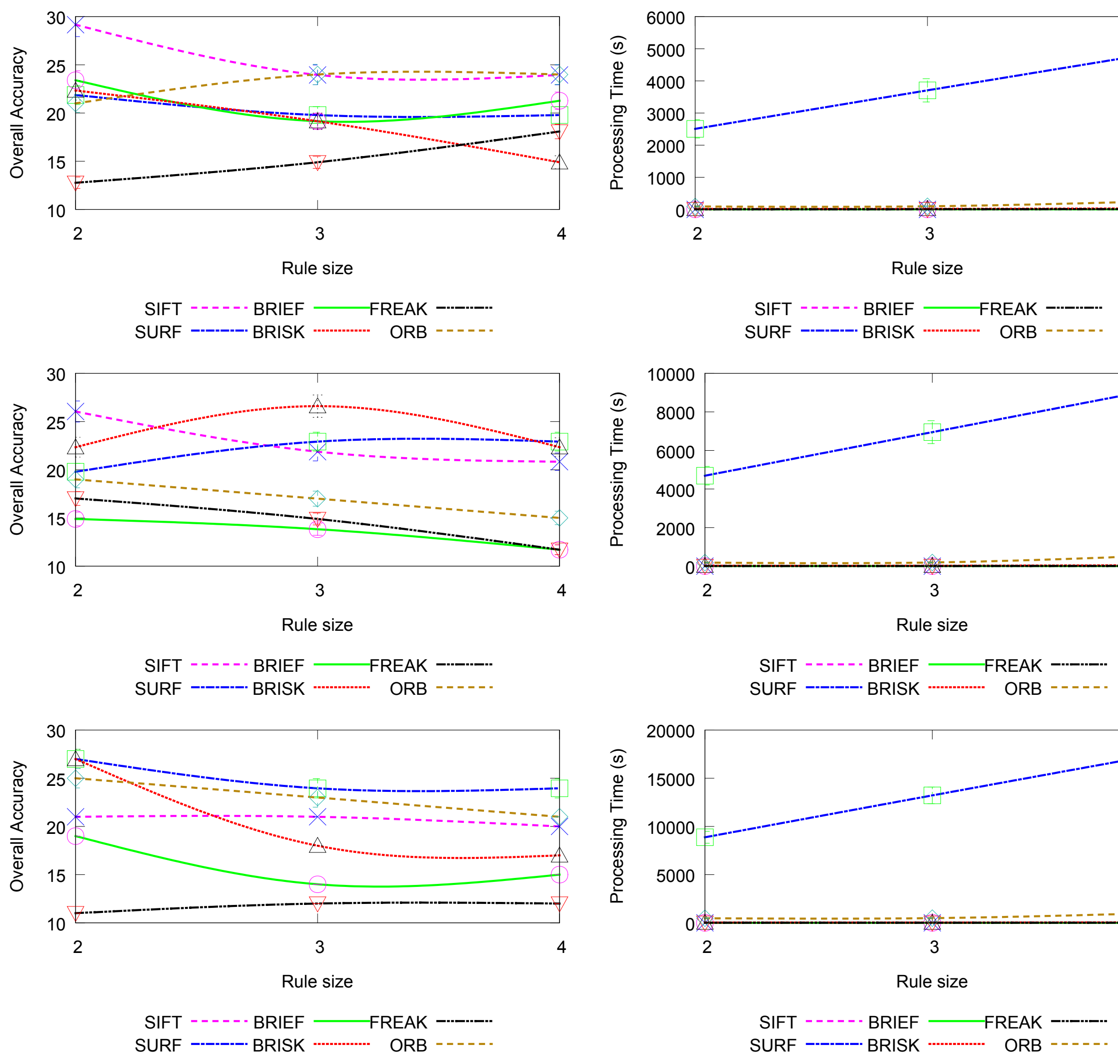


Figure 4.13: The overall accuracy and the processing time, in seconds, obtained with BossaNova using different local descriptors. First row shows the results with $\mathcal{K} = 1024$. Second one shows the results with $\mathcal{K} = 2048$, and the third shows the overall accuracy with $\mathcal{K} = 4096$. For all results, the numbers of bins β used in the quantization step was fixed in 2.

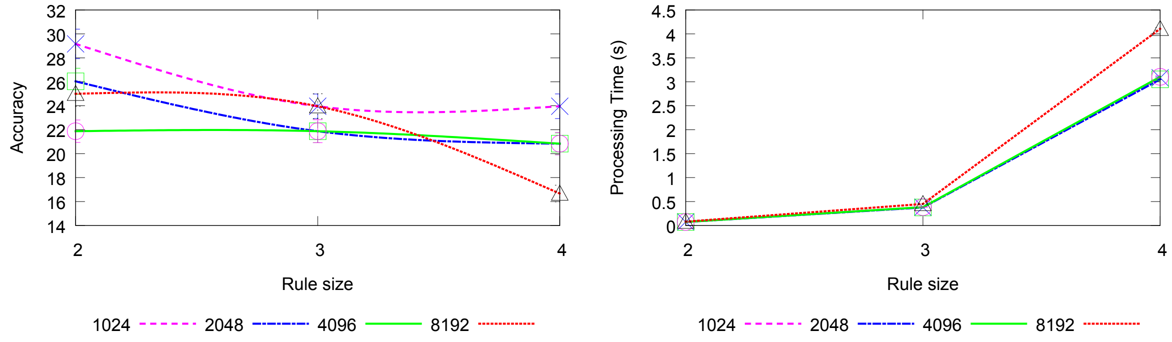


Figure 4.14: The overall accuracy (left) and processing time (right) obtained for BossaNova using SIFT descriptor. The experiments were conducted using number of bins β with three different values: 2, 3 and 4. However, the results were very similar for all the values.

4.3.2 Proposed Methods Evaluation

For the experiments in realistic scenario, we have selected seven descriptors based on the experimental results presented in Section 4.3.1. The selected ones are those that yield at least 21% of accuracy: BIC, CCV, GCH, LCH, QCCH, SID and UNSER. For BoW and BossaNova approaches, we chose only the best local descriptor: the SIFT one. In addition to this, we chose to create visual dictionaries with $\mathcal{K} = 1024$ since they achieve best and stable results.

It is also important to emphasize that based on the results with the ideal scenario, we could observe that smaller the size rule smaller the processing time and, in most cases, the accuracy tends to be very close for all variations of size rule. This allows us to conclude that the lowest value of size rule tends to be the best choice, since we capture the best benefit, i.e., we achieve good results in less processing time if compare with others size rules. Thus, all experiments in this section were made using size rule 2.

Next, the results of the MMCA approach using each one of the visual descriptors are presented in Section 4.3.2.1. The results of combinations of the outputs of the MMCA method are presented in Section 4.3.2.2. Finally, a comparison between the proposed methods and the baseline are presented in Section 4.3.2.3.

4.3.2.1 MMCA Evaluation with Different Visual Descriptors

Figure 4.15 shows all the results for the evaluation of the methods. For each realistic scenario, we ran all feature descriptors using the MMCA approach and the results are shown in terms of Jaccard distance and standard deviation between the folds. It is possible to observe that, for most cases, SID descriptor is the best one amongst all of them. The BossaNova (BN) approach (using SIFT descriptor) is in second place in some cases, however, in general, mid-level approaches were not so effective, differing from the results observed from in the ideal scenario, where mid-level approaches were better than the global descriptors. This can be explained by the fact that, how the keypoints of the mid-level strategies were extracted using dense sampling, when using a perfect segmentation mask, as in the ideal scenario, there is no noise or wrong codewords generated. However, if the mask do not perfectly adjust, wrong codewords may be created, interfering in the final result.

4.3.2.2 Visual Descriptors Combination with MMCA

Figure 4.16 shows a comparison between the different combinations of the MMCA approach. In addition to the two combinations proposed in Section 4.1.1.2, we use two traditional ones: Condorcet Method (CM) and Borda Counting (BC). All combinations were evaluated using top-5, top-6 and top-7 approach. As expected, The combination of MMCA results yields better accuracy than the MMCA approach. The results were very

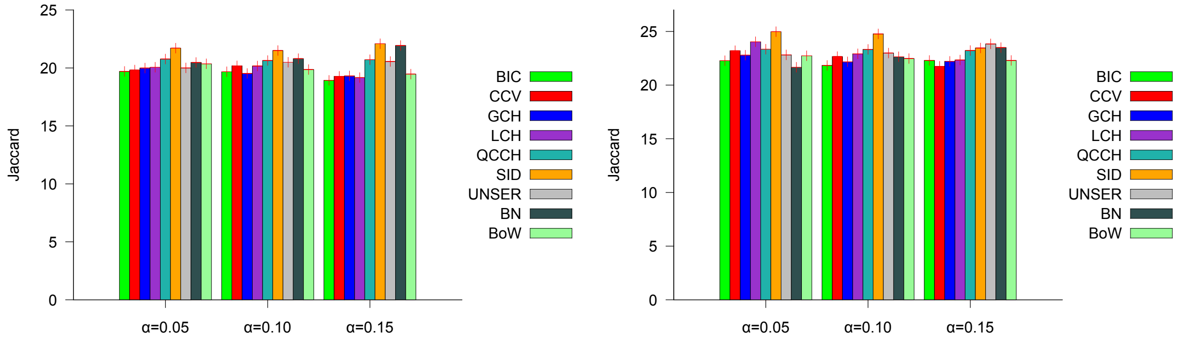


Figure 4.15: Results for the MMCA method for Chictopia (left) and Pose (right).

similar, with Borda Counting being better, in most cases, for the Chictopia dataset, and Majority Probability being better, in most cases, for the Pose dataset.

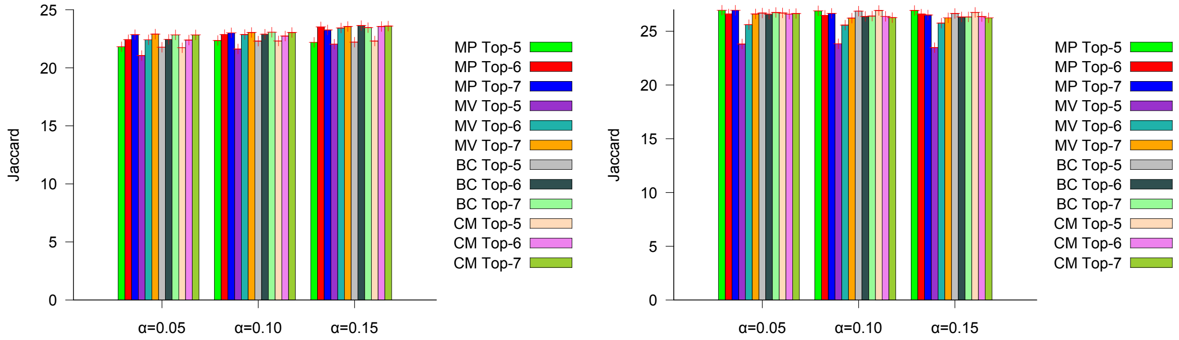


Figure 4.16: The results of the combination of outputs of the MMCA for Chictopia (left) and Pose (right). Four combinations methods were compared: Majority Voting (MV), Majority Probability (MP), Condorcet Method (CM) and Borda Counting (BC).

4.3.2.3 Comparison with the Proposed Methods with the Baseline

Figure 4.17 shows a comparison between the proposed M3CA and the baseline (M3LDA). We also included the best results yielded using MMCA and the best results using some combination algorithm. As expected, accuracy increases with the number of features available. For both dataset, M3CA provides accuracy improvements that vary from 20% (M3LDA top-3) to 30% (M3LDA top-7). Through the figure, it is also possible to see that with the increasing of the mask α, the accuracy tends to increase. This reveal that small mask discard important visual features that may be used by the learning algorithm.

The combination of MMCA results yields better accuracy than the M3CA approach, however, the MMCA approach, without combination, was not capable to achieve accuracy close to the M3CA method. Despite of achieving best results, the accuracy of the M3CA are almost as good as the combinations, but with much less processing time spent, since to get the combination, we need to get all results from each descriptor.

Table 4.4 presents some examples of annotation of our proposed M3CA and the original annotations. First example shows a case when the algorithm could distinguish

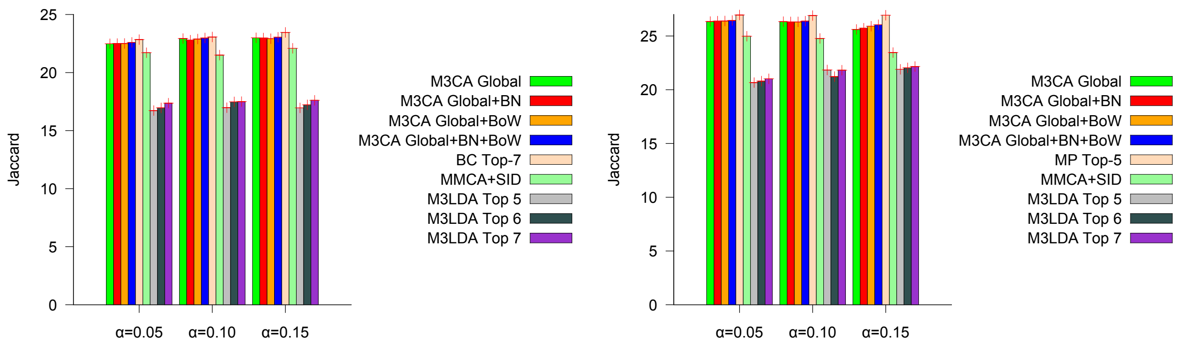
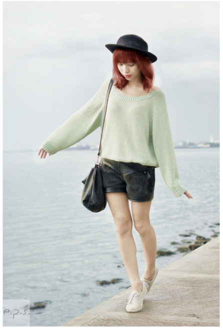
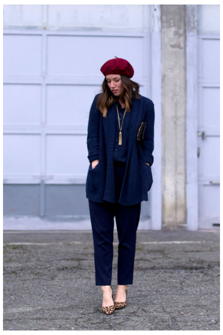
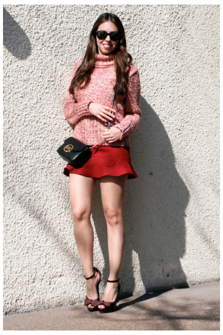


Figure 4.17: The results of the M3CA and the baseline for Chictopia (left) and Pose (right). We also considered the best MMCA using SID, and the best combination algorithm for each dataset.

Table 4.4: Example of output of our proposed M3CA compared with the original annotations. In the third image, the predicted annotations are identical to the original ones.

Images			
Original Annotation	bag, hat, shorts, sneakers, sweater	bag, hat, heels, pants, shirt, sweater	bag, skirt, shoes, sweater
Automatic Annotation	shorts (0.12), sweater (0,09), shoes (0,08)	hat (0.10), romper (0.09)	skirt (0.11), bag (0.09), shoes, (0.08) sweater (0.08)

between several garment items but could not separate the sneakers from the ground, since both are very similar. Second example shows when the method could not distinguish the clothes, since all the garment items have the same color. Thus, the algorithm considered all the clothes (pants, shirt, sweater) as a single cloth, and predicted a romper. The last case is a perfect match of the predicted labels and the original ones. This case is only achieved when the function generated by the MDL suggested the perfect cut, predicting only right labels.

Chapter 5

Clothing Parsing

In this chapter, we present the proposed method for clothing parsing, called Multi-scale Convolutional Neural Network, or simply M-CNN. Section 5.1 shows the proposed method. The experimental protocol used is introduced in Section 5.2, while the results obtained are presented in Section 5.3.

5.1 Deep Learning Approches For Clothing Parsing

Figure 5.1 shows an overview of the proposed method, that has three levels with different network architecture in each one. These network levels process images with different dimensions, i.e., the first level is responsible to classify larger images while the last level processes smaller ones. Our proposed method, based on Convolutional Neural Networks (CNN), works using some kind of hierarchical model, that classifies images with different sizes in different levels of the hierarchy. More specifically, the query image is splitted into small tiles (or patches) with fixed size, which is, in this case, $64 \times 64 \times 3$ (this last dimension corresponds to the format model of the image, which is RBG). These tiles are evaluated and only the ones with significant information (not background) are selected. The first level network processes these tiles and, the ones considered as classified (low entropy) do not need more processing, unlike the ones with high entropy (several classes with high probability associated). These are again splitted into smaller tiles, with size $32 \times 32 \times 3$, and evaluated. The selected tiles are processed by the second level network. Once more, tiles considered as classified stop being processed after this layer and, the unclassified ones are splitted into even smaller patches, with size $16 \times 16 \times 3$. The last level network is responsible to finally classify the remaining tiles. At the end, a class is associated to each tile and a new segmented image may be recomposed.

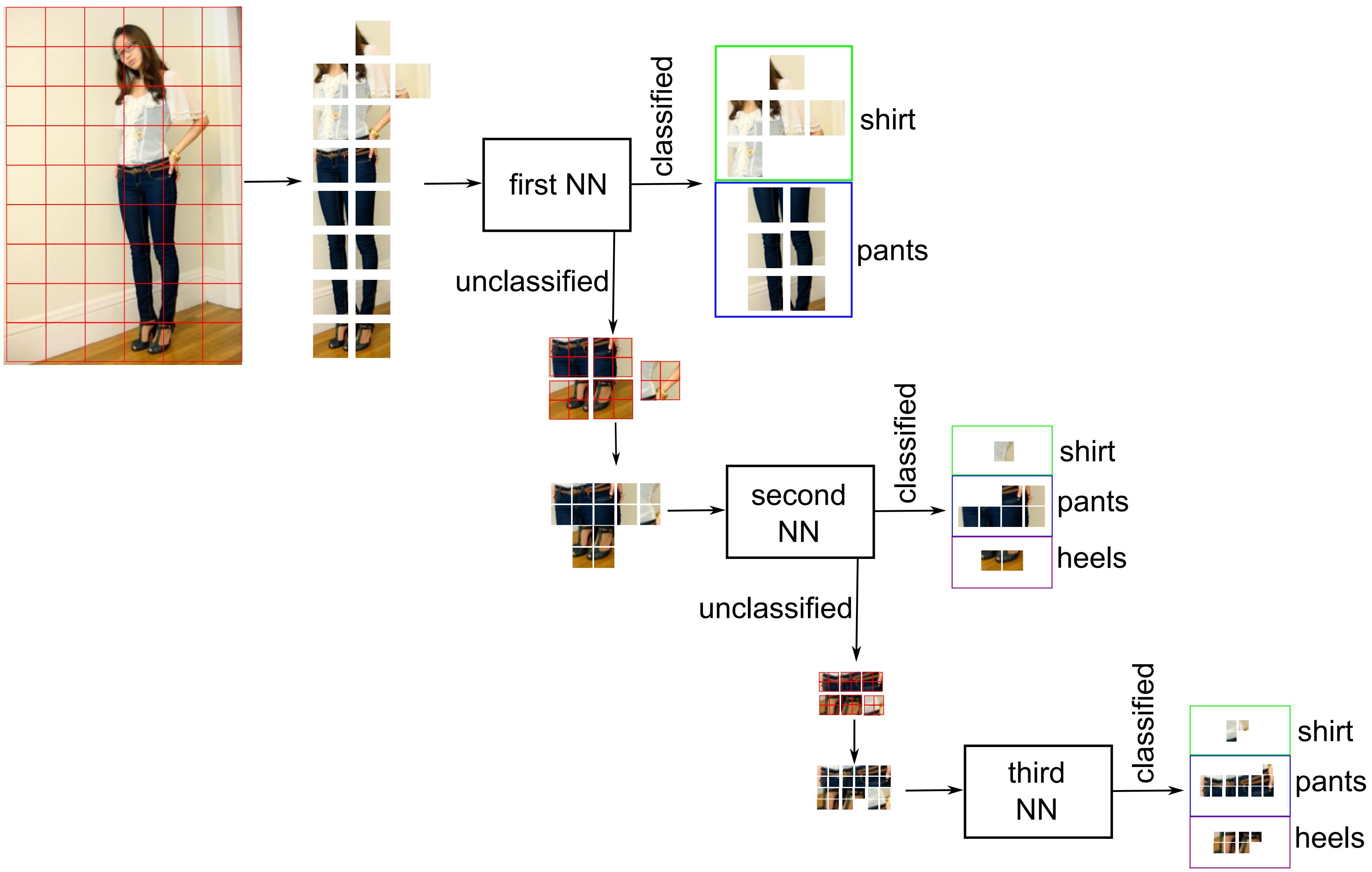


Figure 5.1: Overview of the M-CNN approach. The original image is splitted into little tiles that are candidates to be classified in the first level network. The unclassified tiles are splitted again and goes for the second network. The same occurs on the last level of our architecture.

It is important to highlight that the proposed method could have as any levels as needed. However we used only three in this case, because the smallest clothing item has total amount of pixels similar (248 pixels of a bracelet) to the size of the last level capacity ($16 \times 16 = 256$ pixels). So, based on this, an additional level would have only parts of the items and could never receive a entire one.

A tile is selected and processed based on its significant information. This is performed by evaluating (using the groundtruth mask[1]) the number of pixels not in the background. If the total amount of relevant pixels achieves a certain threshold (in this case, 10% of the total size of the tile) then the tile is selected. This is done, in order to avoid the method to classify a tile with almost no significant information with a class, just based on some pixels. To evaluate if a tile is considered classified or not, the entropy (Alpaydin [2010]) measure was applied. Commonly used in information theory, this measure characterizes the (im)purity of an random collection of instances. In this case, it denotes the purity of a single tile in relation to the number of classes associated to it, i.e., the more classes with high probability related to the patch the higher entropy it has (more impure). Thus, entropy helps our approach to defined which patches are considered classified and which ones are not. Therefore, an entropy threshold is defined to categorize the tiles between classified and unclassified. How the output of each level of the NN is a ranking with the classes and respective probabilities, a value of entropy for each tile may be calculated and, if this value is higher than the threshold, the tile is unclassified, otherwise, it is classified.

Before describing the architecture of each network, we must emphasize that the rectifier function was used as activation function (so the neurons are called ReLUs) for every processing unit, the softmax function was used as classification layer in all networks, thus the cost function is a log loss one. All details about these frameworks, as well as the equations, are presented in Section 3.4. After modelling the networks, we used Convolutional Architecture for Fast Feature Embedding (Caffe) (Jia et al. [2014]), a deep learning framework, to create and experiment them. This framework is more suitable due to its simplicity and support to parallel programming using CUDA®, a NVidia® parallel programming based on graphics processing units.

A drawback of deep learning strategy is the large number of parameters, which are, in this case, five different ones: (i) learning rate, a parameter that determines how much an updating iteration influences the current value of the weights, (ii) weight decay, a regularization that is an additional term in the weight update rule that penalizes large weights to prevent overfitting, (iii) step size, which defines the number of iterations until the learning is divided by a constant value (gamma) equals to 0.1, (iv) momentum, a parameter that is used to prevent the system from converging to a local minimum or

[1] The groundtruth mask is an image with every pixel classified with its corresponding class. Usually, this mask is built using human effort.

saddle point, and (v) maximum iterations, that represents the total number of iterations of the neural network. Select the best value for each parameter is totally empirical in this case. This requires a high number of experiments and a well-structured protocol.

In this case, the final architectures and its parameters were adjusted considering a full set of experiments guided by Bengio [2012]. We started the setup experiments with a small networks and, after each step [2], new layers, with different number of processing units, were being attached until a plateau was reached, i.e., until there is no change in the loss and accuracy of the networks. At the end, initial architectures for each level were obtained. After defining these architectures, the best set of parameters was selected based on convergence velocity versus the numbers of iterations needed. During this step, a myriad of parameters combinations, for each level, were experimented and, for the best ones, new architectures, close to the initial one, were also experimented. The networks with best results were used in this work and are presented next.

The first level network is composed of six layers. Figure 5.2 presents an overview of the network. The first convolutional layer filters the $64 \times 64 \times 3$ input image with 128 processing units of size $4 \times 4 \times 3$ with a stride[3] of 2 pixels. After, a max pooling layer with $2 \times 2 \times 3$ window is applied followed by a LRN layer. The second convolutional layer takes as input the output of the first LRN layer and filters it with 512 processing units of size $4 \times 4 \times 3$ with a stride of 1 pixel. Another max pooling layer with $2 \times 2 \times 3$ window is applied followed by other LRN layer. The third convolutional layer filters the output of the second max pooling with 512 units of size $2 \times 2 \times 3$ with a stride of 1 pixel. After this layer, a max pooling with $2 \times 2 \times 3$ window is applied, but no normalization. The fully-connected layers have 1024 neurons each. The classifier layer has 31 neurons that correspond to the number of classes of this problem. After the experiments performed to evaluated the parameters, in this level, the best values for the learning rate, weight decay, step size, momentum and max iterations were 0.01, 0.001, 10000, 0.9 and 100000, respectively.

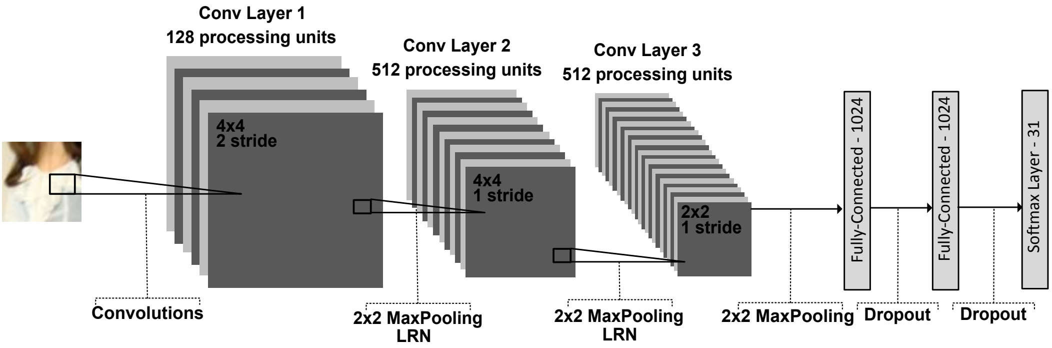


Figure 5.2: First Level Network: more robust network to classify larger tiles.

The second level network is also composed of six layers. The layers are the same of

[2] Experiment considering the five folds in each level.
[3] This is the distance between the centers of each window step.

those from the first level network, changing only the number of neurons, the size of the window and the stride. An overview of the second level network is presented in Figure 5.3. The first convolutional layer filters the $32 \times 32 \times 3$ input image with 96 processing units of size $4 \times 4 \times 3$ with a stride of 1 pixels. After, a max pooling layer with $2 \times 2 \times 3$ window is applied followed by a LRN layer. The second convolutional layer uses with 256 neurons of size $4 \times 4 \times 3$ with a stride of 1 pixel. Another max pooling layer with $2 \times 2 \times 3$ window is applied followed by other LRN layer. The third convolutional layer uses 256 units of size $2 \times 2 \times 3$ with a stride of 1 pixel. After this layer, a max pooling with 2x2x3 window is applied. The fully-connected layers have 512 neurons each. As the first level network, the classifier layer has 31 neurons. In this level, the best values for the evaluated parameters learning rate, weight decay, step size, momentum and max iterations were 0.01, 0.0001, 50000, 0.8 and 200000, respectively.

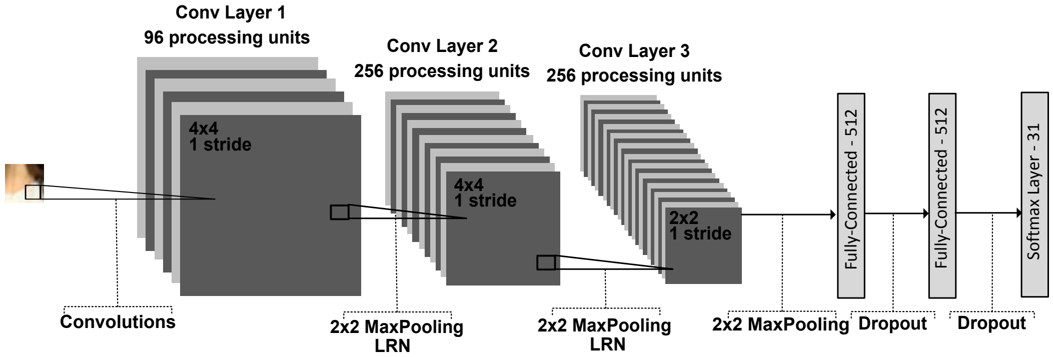


Figure 5.3: Second Level Network

The third and last level network is composed of only four layers. Figure 5.4 presents an overview of the third level network. The first convolutional layer filters the $16 \times 16 \times 3$ input image with 60 processing units of size $4 \times 4 \times 3$ with a stride of 2 pixels. After, a max pooling layer with $2 \times 2 \times 3$ window is applied followed by a LRN layer. The second convolutional layer uses with 128 neurons of size $4 \times 4 \times 3$ with a stride of 1 pixel. After this layer, a max pooling with $2 \times 2 \times 3$ window is applied. The only fully-connected layer has 512 neurons. As the first and second level network, the classifier layer has 31 neurons. After the experiments performed to evaluated the parameters, in this level, the best values for the learning rate, weight decay, step size, momentum and max iterations were 0.001, 0.0001, 50000, 0.8 and 300000, respectively.

5.2 Experimental Protocol

In this section, we present the experimental setup we used in the second phase of this dissertation. Section 5.2.1 presents some statistics of the dataset used. Section 5.2.2 shows the baseline used in this work. The experimental protocol used is presented in Section 5.2.3. Finally, Section 5.2.4 presents the measures used to evaluate the experiments.

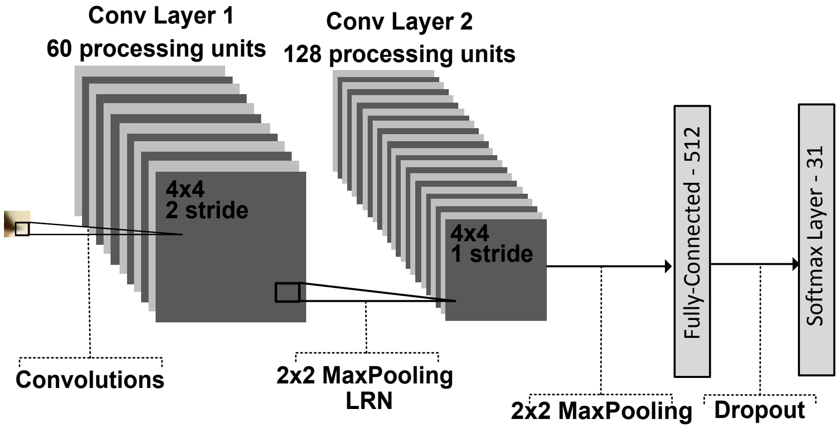


Figure 5.4: Third Level Network: smaller network since the tiles have only 16x16 pixels.

5.2.1 Datasets

We used a segmented dataset, created by Yamaguchi et al. [2012], to evaluate the proposed method. This dataset, called Fashionista, is composed of images collected from the fashion-related social network, `chictopia.com`, together with the related tags. Then, 685 photos with good visibility of the full body and covering a variety of clothing items were selected to be segmented. For this carefully selected subset, two Amazon Mechanical Turk jobs were used to gather annotations. The first Turk job gathered ground truth pose annotations while the second one gathered ground truth clothing labels on super-pixel regions. All annotations are verified and corrected if necessary to obtain high quality annotations. In this ground truth dataset, 31 different clothing items were observed, of which 24 items have at least 50 image regions. Figure 5.5 shows the frequency of each label. As expected, some labels occur frequently (e.g., “shirt”, “jeans”, and “coat”), while others occur only few times (e.g., “intimate”, “stockings”, and “romper”).

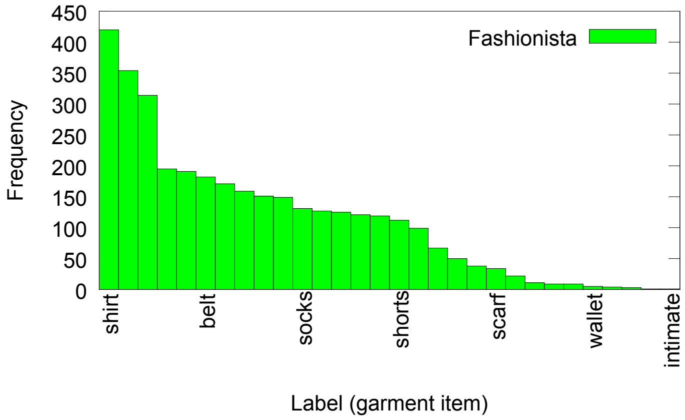


Figure 5.5: Frequency distribution of labels.

5.2.2 Baseline

The proposed method was evaluated considering the pointwise approach, also proposed in this work and described in Section 4.1.1, as baseline. In this case, fixed-size tiles from all clothing items were extracted considering the parsed dataset. Specifically, 16×16 tiles were extracted from all parsed segments of the Fashionista dataset and delivered to be classified. Thus, there is no tile with background pixels (since only tiles completely inside the segment are considered) so when the classifier predicts a right label for a tile it successful classify all pixels inside the tile.

Considering all the evaluation made in Section 4.3, we modelled this task using mid-level representation. So, based on Section 4.3.1, this MMCA approach was created using BoW strategy (which achieved good results just like BN one) with dictionary of $\mathcal{K} = 1024$ and SIFT as the local feature extraction technique. LAC was used as the machine learning technique with rules of size 2, which achieved best results in experiments presented in Section 4.3.1.

5.2.3 Cross Validation

Just like the clothing annotation phase, we conducted experiments using a k-fold cross-validation strategy. The main difference is the validation subset, not needed anymore. Thus, the dataset is randomly split into k mutually exclusive subset (folds) of almost the same size. The $k-1$ subsets are chosen as training set, and the remaining one is the test set. To work with all the dataset, the cross-validation process is repeated k times, and each time a subset is chosen to be the test set (without repetition).

5.2.4 Evaluation Measures

We used the overall accuracy over the pixels to evaluate the proposed approach. Just like the normal overall accuracy, presented in Section 4.2.5, this measure calculates the proportion of cases with right classification over the total population. However, in this case, the population are pixels. This measure is commonly used when working with segmentation problems (Yamaguchi et al. [2013, 2012]), and can be seen as the rate of positive classification of the method.

5.3 Results and Discussion

In this section, we present the experimental results to evaluate the proposed method. All computational experiments presented were performed on a 64 bits Intel® i7® 4,960X machine with 3.6GHz of clock and 64 GB of RAM memory. A GeForce® GTX Titan Black with 6GB of internal memory and 2,880 CUDA Cores was used as graphics processing units, under a 6.5 CUDA version. Ubuntu 14.04.1 LTS (kernel 3.13.0-39-generic) was used as operating system.

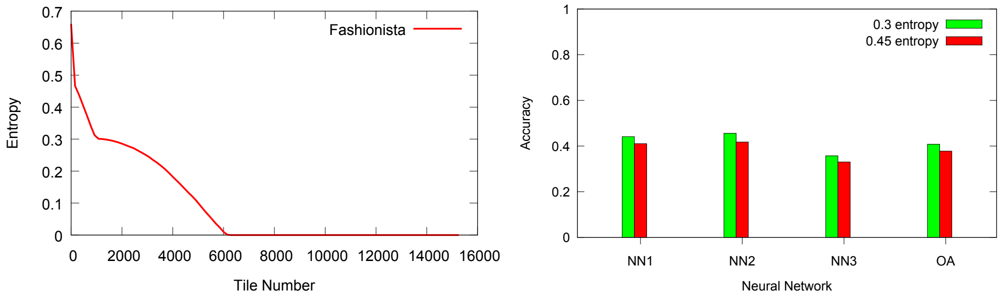


Figure 5.6: Entropy distribution of tiles in the first level of our proposed method (left) and the accuracy with different entropy threshold (right).

To evaluate the method, we need first to define a threshold for the entropy, that defines which tiles are considered classified and which are not. With this threshold defined, the output of the each NN, a ranking with all 31 possible classes with respective probability, may be used to calculate the entropy value of each test tile, allowing it to be categorized as classified or not. To define this threshold, we used a plot relating the tiles of the first level (with size 64 × 64 × 3) and the number of classes (and probabilities) of each tile. More specifically, all pixels from each tile are processed and separated into a class amongst the 31 possibilities. This results in a ranking for each tile relating the classes that composed it and the probability of each class, respectively. A plot with this ranking can be seen in Figure 5.6. Through the image, it is possible to notice two corners (one with entropy value equals 0.30 and another around 0.45) that symbolize points of interest. These two corners were evaluated in our proposed method to defined the best value for the entropy threshold. As presented in Figure 5.6, the entropy value 0.3 achieves best results for our proposed method.

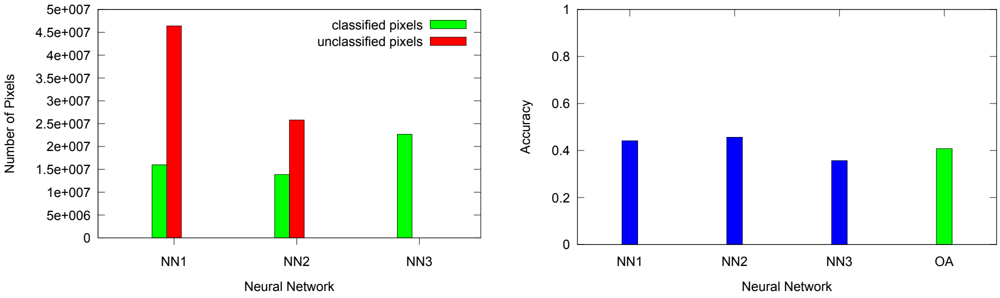


Figure 5.7: The number of pixels classified (left) and the accuracy over the pixels (right) for each level of the proposed M-CNN. In green, the overall accuracy over pixels of the method.

After defining the entropy threshold, the proposed method can be evaluated. Figure 5.7 shows the number of classified pixels, as well as the accuracy for each level and overall accuracy of the proposed algorithm. The first and second layers present similar accuracy, although they have different input size (the former has 64 × 64 × 3 while the latter has $32 \times 32 \times 3$ as input size). These similarity may also be seen when comparing the number of pixels classified. This was expected, since the networks are very similar and tend to learning approximate features. The last level, which has very different network architecture, has the smallest input size, which is $16 \times 16 \times 3$. Mandatory, this level should classify all the remain tiles, as presented in Figure 5.1. This causes a decrease of the accuracy when compared to the others levels, maybe because some features could not be learned well given the small size of the dataset.

A trade-off between the accuracy and the processing time reaches all neural network systems (Krizhevsky et al. [2012]) and it is not different in the proposed approach. The time of the neural networks to classify the tiles varied according to the number of train instances, as can be seen in Figure 5.8. Thus, the first level performed the classification in less processing time, taking around three to four hours to realize all the procedure. The second level has more tiles and, obviously, takes more processing time than the first one. It takes around ten hours to finally classify all the tiles. The last level, the one with the biggest amount of tiles, takes, at least, twelve hours to finish the classification process. At the end, the total processing time to complete the whole procedure of the proposed method turns around a day. It may look like expensive to train the proposed method, however, it is expected when using neural network to have high processing time (Kattan et al. [2009]).

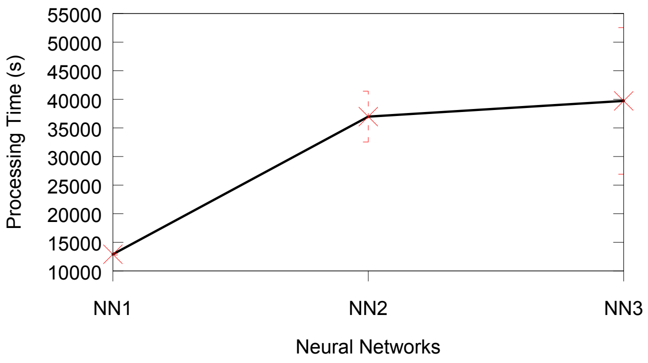


Figure 5.8: Classification processing time, in seconds, of each level of our method.

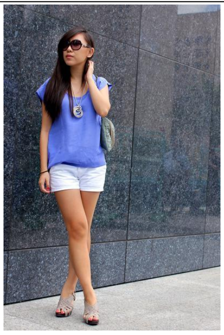
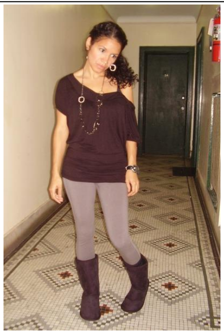
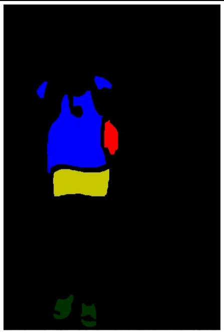
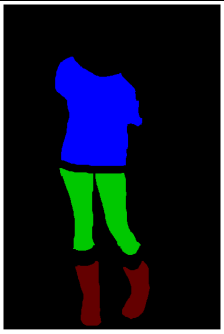
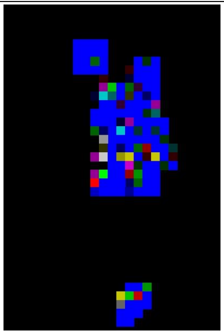
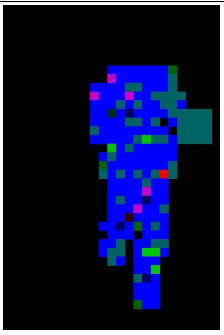
Table 5.1 presents the final results based on the overall accuracy. The pointwise approach were trained using the parameters as described in Section 5.2.2. It is possible to see that the pointwise approach for clothing parsing achieve better result than this same method for clothing annotation, since for the former, there is no effect of the background. However, the M-CNN approach achieved much better results than the pointwise one, verifying that the proposed method is very promising.

Table 5.2 presents some images with the proposed segmentation and the ground truth parsing. Figure 5.9 presents the classes associated with the colors. Through the images is possible to see that the method is a little biased, since a lot of tiles were classified

Table 5.1: Clothing Parsing results.

Method	Pixel Accuracy (%)
Pointwise+BoW+SIFT+Rule Size 2	24.45
M-CNN	**40.79**

Table 5.2: Example of output of our proposed M-CNN compared with the original parsing.

Images		
Original Parsing		
Predicted Parsing		

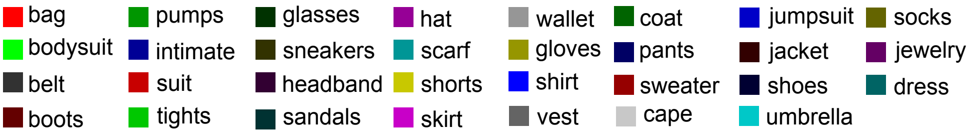


Figure 5.9: Color and the respective class.

as shirt (blue). Perhaps, it is motivated by the distribution of classes of the dataset, since most images (around 550) have this specific garment item. To avoid this problem, a large dataset would be useful.

Chapter 6

Final Remarks and Future Work

This work presented algorithms for clothing parsing and annotation. In the first phase, we tackle the clothing annotation task, proposing a pointwise and a pairwise approach. The first one, called MMCA, takes advantages from the multi-modal method resulting in a robust multi-label classification. The latter one, called M3CA, takes advantage of being multi-instance/multi-modal resulting also in a multi-label classification. It also exploits the benefits from different types of visual features. We also proposed two methods of combination of the pointwise results.

The novelty of this work relies on a multi-instance method that is capable of using the information from an image creating a sparse classifier. The performed experiments show the benefit of it, since it yields good results with less processing time when compared with a state-of-the-art algorithm (Nguyen et al. [2013]).

Considering the descriptors, the SID descriptor is the best one amongst all of them. The BossaNova approach (using SIFT descriptor) is in second place in some cases, however, in general, mid-level approaches were not so effective for the realistic scenario, differing from the results observed from in the ideal scenario. This can be explained by the fact that, if the mask does not perfectly adjust, wrong codewords may be created, interfering in the final result of the mid-level strategies. Because SID descriptors analyze the image as a whole, this problem is softened.

In three cases, the best result for the investigated problem were achieved using combinations for the output of the MMCA approach: Majority Probability yields best results for Pose dataset in two cases, while Majority Voting achieves best results for Chictopia in one configuration. For the remain cases, Borda Counting (BC) achieves the best results in the last two configurations of the Chictopia dataset while Condorcet Method (CM) achieves the best results in the last case for the Pose Dataset. However, in the cases where the proposed combination methods loses, it generally stays close to the best results, which makes this an advantage, since they are easier to implement than the traditional ones.

Though the combination of MMCA results yields better accuracy than the M3CA approach, the MMCA approach, without combination, was not capable to achieve accuracy similar to the M3CA method. Despite of achieving better results, the accuracy of the M3CA are almost as good as the combinations, but with much less processing time spent, since to get the combination, we need to get all results from each descriptor.

Although M3CA is designed for clothing annotation, it is possible to be applied to others tasks. In the future, we plan to adapt the proposed M3CA for different applications and evaluate the method with other learning techniques.

In the second phase, we propose a deep learning algorithm, based on convolutional neural networks, to solve the clothing parsing problem. After some research in the literature, we observed a lack in problems to solve the image segmentation task using NN. So we proposed a multi-scale algorithm, called M-CNN, in a attempt to solve this task. We model the problem as some kind of hierarchical strategy, that classify images with different sizes in different levels of the hierarchy.

Specifically, our method has three different network levels that process images with different dimensions, i.e., after every level the images are decomposed into smaller tiles, allowing the network to capture minimal details. In the first level, larger images are processed in a robust network. Some images are considered as classified, depending on the entropy value. Unclassified images are splitted into smaller patches and go to the next level. Remaining images without classification in this level are again divided into even smaller patches and, finally, classified in the third level. At the end, we have a class associated with each patch of the image and we can recompose it.

The experiments showed that the proposed method presents exciting results, outperforming the baseline. Although the relevant results, the method needs some improvements, such as taking advantage of the contextual information which is a future work.

As introduced, the proposed method could have as many levels as needed. So, another future work is to test the proposed method with different number of levels and maybe using different methods, such as a fuzzy one, to classify the final tiles at the last level (this way, the last network would not be forced to classify the remaining tiles). Use a convolution strategy instead of a fixed-size grid when creating tiles is another suitable strategy that should be tested, since the former is more robust and could deliver to the learning tiles with more representativeness. We also plan to use a larger dataset to validate and improve the results of the proposed algorithm. Compare the method with other baselines, such as Yamaguchi et al. [2012], and adapt the algorithm for other applications, such as general image segmentation, are other future works.

Bibliography

Agrawal, R., Imielinski, T., and Swami, A. N. (1993). Mining association rules between sets of items in large databases. In *Proceedings of the 1993 ACM International Conference on Management of Data, SIGMOD 1993*, pages 207--216.

Alahi, A., Ortiz, R., and Vandergheynst, P. (2012). FREAK: fast retina keypoint. In *2012 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2012*, pages 510--517. IEEE Computer Society.

Alpaydin, E. (2010). *Introduction to Machine Learning.* MIT Press, 2nd edition.

Atrey, P. K., Hossain, M. A., El-Saddik, A., and Kankanhalli, M. S. (2010). Multimodal fusion for multimedia analysis: a survey. *Multimedia System*, 16(6):345--379.

Baeza-Yates, R. A. and Ribeiro-Neto, B. A. (2011). *Modern Information Retrieval - the concepts and technology behind search, Second edition.* Pearson Education Ltd., Harlow, England, 2nd edition.

Bay, H., Ess, A., Tuytelaars, T., and Gool, L. J. V. (2008). Speeded-up robust features (SURF). *Computer Vision and Image Understanding*, 110(3):346--359.

Bekele, D., Teutsch, M., and Schuchert, T. (2013). Evaluation of binary keypoint descriptors. In *IEEE International Conference on Image Processing, ICIP 2013*, pages 3652--3656.

Bengio, Y. (2012). Practical recommendations for gradient-based training of deep architectures. In *Neural Networks: Tricks of the Trade*, pages 437--478. Springer.

Blei, D. M. and Jordan, M. I. (2003). Modeling annotated data. In *Proceedings of the 26th Annual International ACM Conference on Research and Development in Information Retrieval, SIGIR 2003*, pages 127--134.

Boureau, Y., Bach, F., LeCun, Y., and Ponce, J. (2010). Learning mid-level features for recognition. In *The 23rd IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2010*, pages 2559--2566.

Briggs, F., Fern, X. Z., and Raich, R. (2012). Rank-loss support instance machines for miml instance annotation. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD 2012, pages 534--542. ACM.

Calonder, M., Lepetit, V., Strecha, C., and Fua, P. (2010). BRIEF: binary robust independent elementary features. In *11th European Conference on Computer Vision, ECCV 2010*, pages 778--792.

Chen, H., Gallagher, A. C., and Girod, B. (2012). Describing clothing by semantic attributes. In *12th European Conference on Computer Vision, ECCV 2012*, pages 609--623.

Comaniciu, D. and Meer, P. (2002). Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603--619.

Cushen, G. A. and Nixon, M. S. (2012). Real-time semantic clothing segmentation. In *8th International Symposium in Advances in Visual Computing, ISVC 2012*, pages 272--281.

da Silva Torres, R. and Falcão, A. X. (2006). Content-based image retrieval: Theory and applications. *RITA*, 13(2):161--185.

Datta, R., Joshi, D., Li, J., and Wang, J. Z. (2008). Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys*, 40(2).

Davis, A., Veloso, A., da Silva, A. S., Laender, A. H. F., and Jr., W. M. (2012). Named entity disambiguation in streaming data. In *The 50th Annual Meeting of the Association for Computational Linguistics, ACL 2012*, pages 815--824.

de Avila, S. E. F., Thome, N., Cord, M., Valle, E., and de Albuquerque Araújo, A. (2011). BOSSA: extended bow formalism for image classification. In *18th IEEE International Conference on Image Processing, ICIP 2011*, pages 2909--2912.

dos Santos, J. A., Faria, F. A., da Silva Torres, R., Rocha, A., Gosselin, P. H., Philipp-Foliguet, S., and Falcão, A. X. (2012). Descriptor correlation analysis for remote sensing image multi-scale classification. In *Proceedings of the 21st International Conference on Pattern Recognition, ICPR 2012*, pages 3078--3081.

dos Santos, J. A., Penatti, O. A., Gosselin, P.-H., Falcão, A. X., Philipp-Foliguet, S., and da S. Torres., R. (2014). Efficient and effective hierarchical feature propagation. In *Journal of Selected Topics on Earth Observations and Remote Sensing*. IEEE Computer Society.

dos Santos, J. A., Penatti, O. A. B., and da Silva Torres, R. (2010). Evaluating the potential of texture and color descriptors for remote sensing image retrieval and classification. In *Proceedings of the 5th International Conference on Computer Vision Theory and Applications, VISAPP 2010*, pages 203--208.

Duygulu, P., Barnard, K., de Freitas, J. F. G., and Forsyth, D. A. (2002). Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In *7th European Conference on Computer Vision, ECCV 2002*, pages 97--112.

Fayyad, U. M. and Irani, K. B. (1993). Multi-interval discretization of continuous-valued attributes for classification learning. In *Proceedings of the 13th International Joint Conference on Artificial Intelligence, IJCAI 1993*, pages 1022--1029.

Feng, S. and Xu, D. (2010). Transductive multi-instance multi-label learning algorithm with application to automatic image annotation. *Expert Systems with Applications*, 37(1):661--670.

Fergus, R., Weiss, Y., and Torralba, A. (2009). Semi-supervised learning in gigantic image collections. In *Advances in Neural Information Processing Systems 22: 23rd Annual Conference on Neural Information Processing Systems, NIPS 2009*, pages 522--530.

Gallagher, A. C. and Chen, T. (2008). Clothing cosegmentation for recognizing people. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2008*.

Glorot, X., Bordes, A., and Bengio, Y. (2011). Deep sparse rectifier neural networks. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2011*, pages 315--323.

Gould, S., Fulton, R., and Koller, D. (2009). Decomposing a scene into geometric and semantically consistent regions. In *IEEE 12th International Conference on Computer Vision, ICCV 2009*, pages 1--8.

Guillaumin, M., Mensink, T., Verbeek, J. J., and Schmid, C. (2009). Tagprop: Discriminative metric learning in nearest neighbor models for image auto-annotation. In *IEEE 12th International Conference on Computer Vision, ICCV 2009*, pages 309--316.

Guillaumin, M., Verbeek, J. J., and Schmid, C. (2010). Multimodal semi-supervised learning for image classification. In *IEEE 23rd Conference on Computer Vision and Pattern Recognition, CVPR 2010*, pages 902--909.

Hinton, G. E. (2010). Deep belief nets. In *Encyclopedia of Machine Learning*, pages 267--269.

Huang, C. and Liu, Q. (2007). An orientation independent texture descriptor for image retireval. In *The 5th IEEE International Conference on Computer and Computational Sciences, ICCCS 2007*, pages 772–776.

Huang, J., Kumar, R., Mitra, M., Zhu, W., and Zabih, R. (1997). Image indexing using color correlograms. In *IEEE 10th Conference on Computer Vision and Pattern Recognition, (CVPR 1997*, pages 762--768.

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., and Darrell, T. (2014). Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*.

Kalantidis, Y., Kennedy, L., and Li, L. (2013). Getting the look: clothing recognition and segmentation for automatic product suggestions in everyday photos. In *ACM International Conference on Multimedia Retrieval, ICMR 2013*, pages 105--112.

Kalchbrenner, N., Grefenstette, E., and Blunsom, P. (2014). A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 655--665.

Kattan, A., Abdullah, R., and Salam, R. A. (2009). Reducing feed-forward neural network processing time utilizing matrix multiplication algorithms on heterogeneous distributed systems. In *Proceedings of the 2009 First International Conference on Computational Intelligence, Communication Systems and Networks, CICSYN 2009*, pages 431--435.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems, NIPS 2012*, pages 1106--1114.

Kuntimad, G. and Ranganath, H. S. (1999). Perfect image segmentation using pulse coupled neural networks. *IEEE Transactions on Neural Networks*, 10(3):591--598.

Lafferty, J. D., McCallum, A., and Pereira, F. C. N. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning, ICML 2001*, pages 282--289.

Lavrenko, V., Manmatha, R., and Jeon, J. (2003). A model for learning the semantics of pictures. In *Advances in Neural Information Processing Systems 16: 17th Annual Conference on Neural Information Processing Systems, NIPS 2003*, pages 553--560.

Law-To, J., Chen, L., Joly, A., Laptev, I., Buisson, O., Gouet-Brunet, V., Boujemaa, N., and Stentiford, F. (2007). Video copy detection: a comparative study. In *Proceedings* *of the 6th ACM International Conference on Image and Video Retrieval, CIVR 2007*, pages 371--378.

Lazebnik, S., Schmid, C., and Ponce, J. (2006). Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2006*, pages 2169--2178.

LeCun, Y., Boser, B. E., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W. E., and Jackel, L. D. (1989). Handwritten digit recognition with a back-propagation network. In *Advances in Neural Information Processing Systems 2: 1st Annual Conference on Neural Information Processing Systems, NIPS 1989*, pages 396--404.

Leutenegger, S., Chli, M., and Siegwart, R. (2011). BRISK: binary robust invariant scalable keypoints. In *IEEE International Conference on Computer Vision, ICCV 2011*, pages 2548--2555.

Li, J. and Allinson, N. M. (2008). A comprehensive review of current local features for computer vision. *Neurocomputing*, 71(10-12):1771--1787.

Li, R., Lu, J., Zhang, Y., and Zhao, T. (2010). Dynamic adaboost learning with feature selection based on parallel genetic algorithm for image annotation. *Knowledge-Based Systems*, 23(3):195--201.

Liu, S., Song, Z., Liu, G., Xu, C., Lu, H., and Yan, S. (2012). Street-to-shop: Cross-scenario clothing retrieval via parts alignment and auxiliary set. In *2012 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2012*, pages 3330--3337.

Liu, T. (2009). Learning to rank for information retrieval. *Foundations and Trends in Information Retrieval*, 3(3):225--331.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91--110.

Mahmoudi, F., Shanbehzadeh, J., Eftekhari-Moghadam, A., and Soltanian-Zadeh, H. (2003). Image retrieval based on shape similarity by edge orientation autocorrelogram. *Pattern Recognition*, 36(8):1725--1736.

Makadia, A., Pavlovic, V., and Kumar, S. (2008). A new baseline for image annotation. In *10th European Conference on ComputerVision, ECCV 2008*, pages 316--329. Springer-Verlag.

Maron, O. and Lozano-Pérez, T. (1997). A framework for multiple-instance learning. In *Advances in Neural Information Processing Systems 10: 9th Annual Conference on Neural Information Processing Systems, NIPS 1997*, pages 570--576.

Mcculloch, W. and Pitts, W. (1943). A logical calculus of ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5(4):115--133.

Moran, S. and Lavrenko, V. (2014). Sparse kernel learning for image annotation. In *ACM International Conference on Multimedia Retrieval, ICMR 2014*, page 113.

Nair, V. and Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th ACM International Conference on Machine Learning, ICML 2010*, pages 807--814.

Ng, A., Ngiam, J., Foo, C. Y., Mai, Y., and Suen, C. (2011a). Feature extraction using convolution. `http://ufldl.stanford.edu/wiki/index.php/Feature_extraction_using_convolution/`. Accessed: 2015-02-10.

Ng, A., Ngiam, J., Foo, C. Y., Mai, Y., and Suen, C. (2011b). Pooling. `http://ufldl.stanford.edu/wiki/index.php/Pooling/`. Accessed: 2015-02-10.

Nguyen, C., Zhan, D., and Zhou, Z. (2013). Multi-modal image annotation with multi-instance multi-label LDA. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence, IJCAI 2013*.

Nogueira, K., Veloso, A. A., and dos Santos, J. A. (2014). Learning to annotate clothes in everyday photos: Multi-modal, multi-label, multi-instance approach. In *27th Conference on Graphics, Patterns and Images, SIBGRAPI 2014*, pages 327--334. IEEE Computer Society.

Oliva, A. and Torralba, A. (2006). Building the gist of a scene: The role of global image features in recognition. *Visual Perception, Progress in Brain Research*, 155:23--36.

Pal, N. R. and Pal, S. K. (1993). A review on image segmentation techniques. *Pattern Recognition*, 26(9):1277--1294.

Pass, G., Zabih, R., and Miller, J. (1996). Comparing images using color coherence vectors. In *Proceedings of the 4th ACM International Conference on Multimedia, ICM 1996*, pages 65--73.

Penatti, O. A. B., Valle, E., and da Silva Torres, R. (2012). Comparative study of global color and texture descriptors for web image retrieval. *Journal of Visual Communication and Image Representation*, 23(2):359--380.

Philbin, J., Chum, O., Isard, M., Sivic, J., and Zisserman, A. (2008). Lost in quantization: Improving particular object retrieval in large scale image databases. In *2008 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2008*.

Putthividhya, D., Attias, H. T., and Nagarajan, S. S. (2010). Topic regression multimodal latent dirichlet allocation for image annotation. In *The 23rd IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2010*, pages 3408--3415.

Read, J., Pfahringer, B., and Holmes, G. (2008). Multi-label classification using ensembles of pruned sets. In *Proceedings of the 8th IEEE International Conference on Data Mining, ICDM 2008*, pages 995--1000.

Rissanen, J. (1978). Modeling by shortest data description. *Automatica*, 14(5):465--471.

Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386.

Rublee, E., Rabaud, V., Konolige, K., and Bradski, G. R. (2011). ORB: an efficient alternative to SIFT or SURF. In *IEEE International Conference on Computer Vision, ICCV 2011*, pages 2564--2571.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1988). Learning representations by back-propagating errors. *Cognitive modeling*, 5.

Salakhutdinov, R., Mnih, A., and Hinton, G. E. (2007). Restricted boltzmann machines for collaborative filtering. In *Proceedings of the 24th ACM International Conference on Machine Learning, ICML 2007*, pages 791--798.

Schulz, H. and Behnke, S. (2012). Learning object-class segmentation with convolutional neural networks. In *20th European Symposium on Artificial Neural Networks, ESANN 2012*.

Shen, E. Y., Lieberman, H., and Lam, F. (2007). What am I gonna wear?: scenario-oriented recommendation. In *Proceedings of the 2007 International Conference on Intelligent User Interfaces, IUI 2007*, pages 365--368.

Singh, K. K. and Singh, A. (2010). A study of image segmentation algorithms for different types of images. *International Journal of Computer Science*, 7(5):414--417.

Sivic, J., Russell, B. C., Efros, A. A., Zisserman, A., and Freeman, W. T. (2005). Discovering objects and their localization in images. In *10th IEEE International Conference on Computer Vision, ICCV 2005*, pages 370--377.

Sivic, J. and Zisserman, A. (2006). Video google: Efficient visual search of videos. In *Toward Category-Level Object Recognition*, pages 127--144.

Socher, R., Lin, C. C., Ng, A. Y., and Manning, C. D. (2011). Parsing natural scenes and natural language with recursive neural networks. In *Proceedings of the 28th ACM International Conference on Machine Learning, ICML 2011*, pages 129--136.

Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929--1958.

Stehling, R. O., Nascimento, M. A., and Falcão, A. X. (2002). A compact and efficient image retrieval approach based on border/interior pixel classification. In *Proceedings of the 2002 ACM International Conference on Information and Knowledge Management, CIKM 2002*, pages 102--109.

Suh, B. and Bederson, B. B. (2007). Semi-automatic photo annotation strategies using event based clustering and clothing based person recognition. *Interacting with Computers*, 19(4):524--544.

Swain, M. J. and Ballard, D. H. (1991). Color indexing. *International Journal of Computer Vision*, 7(1):11--32.

Tang, J., Li, H., Qi, G., and Chua, T. (2010). Image annotation by graph-based inference with integrated multiple/single instance representations. *IEEE Transactions on Multimedia*, 12(2):131--141.

Tao, B. and Dickinson, B. W. (2000). Texture recognition and image retrieval using gradient indexing. *Journal of Visual Communication and Image Representation*, 11(3):327--342.

Tokumaru, M., Fujibayashi, T., Muranaka, N., and Imanishi, S. (2002). Virtual stylist project - dress up support system considering user's subjectivity. In *Proceedings of the 1st International Conference on Fuzzy Systems and Knowledge Discovery: Computational Intelligence for the E-Age, FSDK 2002*, pages 207--211.

Tsoumakas, G. and Katakis, I. (2007). Multi-label classification: An overview. *International Journal of Data Warehouse and Mining*, 3(3):1--13.

Tuytelaars, T. (2010). Dense interest points. In *The 23rd IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2010*, pages 2281--2288.

Tuytelaars, T. and Mikolajczyk, K. (2007). Local invariant feature detectors: A survey. *Foundations and Trends in Computer Graphics and Vision*, 3(3):177--280.

Unser, M. (1986). Sum and difference histograms for texture classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(1):118--125.

van Gemert, J., Geusebroek, J., Veenman, C. J., and Smeulders, A. W. M. (2008). Kernel codebooks for scene categorization. In *10th European Conference on Computer Vision, ECCV 2008*, pages 696--709.

Veloso, A., Jr., W. M., Gonçalves, M. A., and Zaki, M. J. (2007). Multi-label lazy associative classification. In *11th European Conference on Principles and Practice of Knowledge Discovery in Databases, PKDD 2007*, pages 605--612.

Veloso, A., Jr., W. M., and Zaki, M. J. (2006). Lazy associative classification. In *Proceedings of the 6th IEEE International Conference on Data Mining, ICDM 2006*, pages 645--654.

Vens, C., Struyf, J., Schietgat, L., Dzeroski, S., and Blockeel, H. (2008). Decision trees for hierarchical multi-label classification. *Machine Learning*, 73(2):185--214.

Vogiatzis, D., Pierrakos, D., Paliouras, G., Jenkyn-Jones, S., and Possen, B. J. H. H. A. (2012). Expert and community based style advice. *Expert Systems with Applications*, 39(12):10647--10655.

Weber, M., Bäuml, M., and Stiefelhagen, R. (2011). Part-based clothing segmentation for person retrieval. In *8th IEEE International Conference on Advanced Video and Signal-Based Surveillance, AVSS 2011*, pages 361--366.

Xie, L., Pan, P., and Lu, Y. (2015). Markov random field based fusion for supervised and semi-supervised multi-modal image classification. *Multimedia Tools and Applications*, pages 613–634.

Yamaguchi, K., Kiapour, M. H., and Berg, T. L. (2013). Paper doll parsing: Retrieving similar styles to parse clothing items. In *IEEE International Conference on Computer Vision, ICCV 2013*, pages 3519--3526.

Yamaguchi, K., Kiapour, M. H., Ortiz, L. E., and Berg, T. L. (2012). Parsing clothing in fashion photographs. In *2012 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2012*, pages 3570--3577.

Yang, M. and Yu, K. (2011). Real-time clothing recognition in surveillance videos. In *IEEE International Conference on Image Processing, ICIP 2011*, pages 2937–2940.

Yang, S., Zha, H., and Hu, B. (2009). Dirichlet-bernoulli alignment: A generative model for multi-class multi-label multi-instance corpora. In *Advances in Neural Information Processing Systems 22: 23rd Annual Conference on Neural Information Processing Systems, NIPS 2009*, pages 2143--2150.

Yang, W., Luo, P., and Lin, L. (2014). Clothing co-parsing by joint image segmentation and labeling. In *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014*, pages 3182--3189.

Yang, Y. and Ramanan, D. (2011). Articulated pose estimation with flexible mixtures-of-parts. In *2011 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2011*, pages 1385--1392.

Zegarra, J., Leite, N., and Torres, R. (2008). Wavelet-based feature extraction for fingerprint image retrieval. *Journal of Computational and Applied Mathematics.*

Zhang, D., Islam, M. M., and Lu, G. (2012). A review on automatic image annotation techniques. *Pattern Recognition*, 45(1):346--362.

Zhang, D. and Lu, G. (2004). Review of shape representation and description techniques. *Pattern Recognition*, 37(1):1--19.

Zhaolao, L., Zhou, M., Wang, X., Fu, Y., and Tan, X. (2013). Semantic annotation method of clothing image. In *15th International Conference on Human-Computer Interaction, HCI 2013*, pages 289--298.

Zhou, Z., Zhang, M., Huang, S., and Li, Y. (2012). Multi-instance multi-label learning. *Artificial Intelligence*, 176(1):2291--2320.