FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Automatic Interpretation of Promotional Leaflets in Retail for Pricing Strategy

António Maria Aires Pereira Teixeira de Melo

Mestrado Integrado em Engenharia Informática e Computação

Supervisor: Professor Carlos Manuel Milheiro de Oliveira Pinto Soares

Second Supervisor: Telmo Domiciano Pereira Barbosa

September 23, 2019

Automatic Interpretation of Promotional Leaflets in Retail for Pricing Strategy

António Maria Aires Pereira Teixeira de Melo

Mestrado Integrado em Engenharia Informática e Computação

Approved in oral examination by the committee:

Chair: Professor Luís Filipe Teixeira
External Examiner: Professor Cláudio Sá
Supervisor: Professor Carlos Soares
September 23, 2019

Abstract

In retail, the price strategy is characterized by daily competition and constant analysis of competitors. Leaflets are collected manually from physical stores by teams of certified spotters. Promotion leaflets are designed to draw the consumer's attention, therefore they are packed with a considerable amount of condensed information. Even with the existence of leaflets in digital form, they are not easily interpreted by computers making this process very time-consuming and prone to human error.

Currently there is no standard solution for automatic interpretation of leaflets that deals with some critical challenges such as: non-standardized design (different types of lettering that make it harder to extract the text from the images), abbreviations or sentences not following usual grammar rules, the need to find association between information contained in text and images or even ignore images that don't include branding and the indispensability of knowledge of the domain.

In view of the problem described above, the primary objective of this M.Sc. Thesis is to investigate different methods to extract pricing information from leaflets to support the retail pricing strategy. These methods will apply visual inspection techniques based on optical character recognition, image processing, text location/recognition, semantic segmentation and machine learning in leaflet images. For each leaflet, the method outputs a list of product-promotion pairs (types of promotions and respective promotional prices, if present). To evaluate the performance of the implemented methods, will be applied to databases of leaflets which were previously manually annotated, provided by a retailer.

This solution proposes to integrate multiple types of data and domain knowledge to create a cognitive model that enables the understanding and extraction of meaningful information from the promotional leaflets.

Results show that the use of domain logic information, can have significant impact on the results extracted and also that some techniques and strategies work better for different types of layouts created by some companies. Also that, the use of machine learning to identify specific objects that are not easily detected by an OCR engine, can help improve the results of segmentation and information extraction.

Resumo

No comércio retalhista, a estratégia de preços é caraterizada pela competição diária e constante análise da concorrência, que é feita manualmente por uma equipa de profissionais certificados que se dirigem às lojas físicas e recolhem os folhetos promocionais.

Os folhetos promocionais são construidos para chamar a atenção do consumidor, contendo uma elevada quantidade de informação condensada. Mesmo com a existência de folhetos em formato digital, estes não são facilmente interpretados por computadores, tornando o processo muito demorado e sujeito a erros humanos.

Atualmente não existe uma solução semelhante para interpretação automática de folhetos que lide com alguns desafios críticos como: design não padronizado (diferentes tipos de letras que dificultam o reconhecimento em imagens), abreviaturas ou frases que não seguem as regras gramaticais usuais, necessidade de encontrar associações entre informação contida em texto e imagens (ou até ignorar imagens que não incluem marcas) e necessidade de conhecimento do domínio.

Tendo em vista o problema descrito acima, o objetivo principal desta tese de mestrado é desenvolver uma ferramenta para interpretação automática de folhetos promocionais para apoiar a estratégia de preços em retalho, que aplicará técnicas de inspeção visual baseadas em reconhecimento ótico de caracteres, processamento de imagens e localização/reconhecimento de texto, segmentação semântica e "machine learning" em imagens de folhetos.

O resultado esperado será uma lista de pares de promoção/produto (tipos de promoções e respectivos preços promocionais, se presentes). Para avaliar o desempenho dos métodos implementados, o resultado será comparado com bancos de dados retrospetivos de folhetos previamente anotados manualmente ("ground truth"), fornecidos por uma multinacional portuguesa. Além disso, uma ferramenta web será desenvolvida para permitir a visualização e o processamento das informações extraídas.

Esta solução propõe integrar vários tipos de dados e conhecimento de domínio para criar um modelo cognitivo que permita a compreensão e a extração de informações significativas dos folhetos promocionais.

Os métodos desenvolvidos nesta tese mostram que é possível extrair informações dos folhetos com alguma precisão. Os resultados provam que o uso de informações lógicas de domínio, pode ter impacto significativo nos resultados extraídos e também que algumas técnicas e estratégias funcionam melhor para diferentes tipos de "layouts" criados pelas empresas. Além disso, o uso de "machine learning" para identificar objetos específicos que não são facilmente detetados por um mecanismo de OCR, pode ajudar a melhorar os resultados de segmentação e extração de informações.

Acknowledgements

You can never be successful by yourself, so I would like to thank:

Fraunhofer AICOS Portugal for the oportunity to develop my thesis at a company level.

Professor Carlos Soares, my supervisor at FEUP, for an excellent guidance, patience and availability for all my questions.

Telmo Barbosa, my supervisor at Fraunhofer, for all the guidance, suggestions, knowledge exchange, patient during this 6 months and help provided when facing challenges.

And finally, my family, friends and Inês, for all the support and incentive during this time.

António Melo

"If things are not failing, you are not innovating enough"

Elon Musk

Contents

List of Figures

List of Tables

Abbreviations

OCR	Optical Character Recognition
HOG	Histogram of oriented gradient
SIFT	Scale-invariant feature transform
BOV	Bag-of-visual-words
ER	Extremal regions

Chapter 1

Introduction

In retail, the definition of price strategy is characterized by daily competition and constant analysis of its competitors. The fact that multinational companies are now capable of having profit, by selling massive amounts of products with small profit margins, lead to a huge necessity of constant information about the prices and discounts applied by the competition in order not to lose ground in the market. One of most cited examples of disputes between companies, occurred in New York and Washington DC, where two bus services companies engaged in a so called price-war. First one of the companies lowered their initial price from 25$ to 9.95$. The other company respond by lowering even more the price, and so on... This lead to the both companies to eventually match their price to ticket to 5$, resulting in two companies operating below the cost price of the trip. This type of behaviour can have serious damage, not just in the companies performance, but in the market [KJB16].

A more recent example of this price wars between companies is in the airline industry, where low cost airlines are increasingly attracting clients by filling a space on the market that was non existent (low price flight).

In 2012, a Portuguese multinational company gave, for a day, 50% discount in all buys with amounts above 100 euros. Thousands rushed to this supermarkets to benefit for this almost "unrealistic" promotion. This raised a lot of questions from other multinationals that found this campaign a clear sing of unfair competition. At that time, the law could enforce fines between 15 and 30 thousand euros, so the company took advantage of this "loop hole" [San12]. This same company, revealed that in 2014 their profits went down 21% even though their sales went up almost 7% [Ser15]. Companies prefer to lose in profit and make management/organizational efforts that to lose clients to the competition.

This constant competition, brought a clear necessity of good and, most importantly, fast information about the competitors and their prices/promotions.

1.1 Motivation

To fulfill the necessity of having information about the competition and in order to maintain it organized and legal, teams of certified spotters where created with the intent of visiting the competition physical stores, taking note of shelf price products and available leaflets. Most of this leaflets are scanned and then sent to a central location, where another team extracts the information and inserts it in a data base, to be later analyzed and then taken in account when taking decisions about the price strategy to adopt. This process, usually takes more than a full day (24 hours) to be accessible to the administration. This is a lot of time and also very prone to human error, especially in the middle of a price war with the competition.

This created a need for an autonomous computerized process, that would give better and faster results to possibly give the upper hand when dealing with competitors. Once found this demand for faster and more reliable information extraction in leaflets, the idea of a product that would deal with this emerged.

Promotion leaflets are designed to draw the consumer's attention, therefore they are packed with a considerable amount of condensed information, making it more difficult for computers to extract information. Currently there is no off-the-shelf solution for automatic interpretation of leaflets that deals with some critical challenges such as: non-standardized design (different types of lettering that difficult the recognition in images), abbreviations or sentences not following usual grammar rules, the need to find association between information contained in text and images or even ignore images that don't include branding and the indispensability of knowledge of the domain.

1.2 Goals

The primary objective of this M.Sc. Thesis is to investigate different methods for automatic interpretation of promotional leaflets to support the retail pricing strategy, which will apply visual inspection techniques based on optical character recognition, text location/recognition, image processing, semantic segmentation and machine learning to leaflet images.

For each leaflet, the method outputs a list of identified product-promotion pairs (respective promotional prices, if present). To evaluate the performance of the implemented method, the output will be compared against retrospective databases of leaflets previously manually annotated (ground truth), provided by a retailer. Also, a web tool will be developed to allow the visualization and processing of the extracted information.

1.3 Overview

This document is organized into 5 chapters. In Chapter 2, we describe the state of the art of the most relevant areas, techniques and their solutions related with this work. Chapter 3, provides a detailed description of the implementation and different approaches developed. Chapter 4, is

where the results are analyzed and discussed. Finally in Chapter 5, the main conclusions and reflections of this work are presented along with possible future work.

Chapter 2

Literature Review

In this chapter, some concepts and solutions related with this work will be presented. At first, in Section 2.1 a light review of price planning strategies and the impact of publicity in companies performance and clients will be shown. In Section 2.2 OCR (Optical Character Recognition) is discussed. Section 2.3 describes an image processing method called Hough Transform. Scene text location basics and Class-specific Extremal Regions are explained in Section 2.4. After that, in Section 2.5 some semantic segmentation methods will be described and also ways to evaluate their results. Section 2.6 describes some knowledge extraction approaches. At last, in Section 2.7 a short introduction about machine learning is made.

2.1 Price planning and promotional leaflets

Nowadays, a great part of the retail market has moved to online retail, since it provides rich customer support, recommendations, wish lists, and shopping carts. However, paper-based leaflets are still one of the most important advertising methods for retailers [LBDG13]. Retailers made multiple attempts to replace paper based leaflets for electronic versions usually sent by email, but physical material has more emotional processing, making it important for memory and brand association. Millward-Brown [MB09] showed that since processing takes place more in the right retrosplenial cortex, when physical material is present, more emotionally vivid memories are generated [LBDG13].Paper leaflets still remain an effective and thus very important marketing channel [BM04, MB09].

In order to understand how leaflets are designed, Lochtefeld [LBDG13] studied Augmented Reality based strategies for printed leaflets. He analyzed the design of leaflets to better understand how they are constructed. The project used a sample of 16 German paper-based leaflets. Half were from local markets, three from electronic equipment retailers and the rest from stores that sell flowers, furniture and other products. The observations made include the following:

- The layouts and sizes of the leaflets differ from each other, but the same companies share the same size and layout. Finding a pattern in the layout can help in the segmentation phase of the leaflets. If we are presented with a leaflet, we can try to find leaflets with the same layout and use the appropriated techniques.

- Almost 70% of the leaflets used orderly layouts and product images were printed with clear borders. Once more, having clear border between products can enhance the segmentation results.

- 88% are printed in low quality recycled paper. Expecting low quality, we can expect the use preprocessing OCR methods to enhance the quality of the digitization so that the processing performs better.

- In almost all the leaflets, a ratio of 7.02 products were printed on each page. This can help in posprocessing OCR approaches using probabilistic methods. If we find a number of products much smaller than this ratio, we can rerun the Optical character recognition with other methods to obtain better results.

In the same paper, an application was developed, using Augmented Reality, as shown in Figure 2.1 that allowed customers, using their smartphone cameras, to inspect the competition paper leaflets and compare the product prices with theirs. It is important to state that the competitors data is introduced manually. This is possible since most leaflets are available online a day before they are distributed [LBDG13].

Figure 2.1: Augmented Reality price comparison [LBDG13]

2.2 OCR

Optical Character Recognition is a of software that converts scanned images or handwritten notes into a digitized form such that can be manipulated and processed by a machine.

The human brain can easily recognize text/characters in an image. On the other side, machines are not intelligent enough to perceive the information available in the image [IIN16]. The complexity of the problem increased with the variety of languages, fonts, and styles used in documents. OCR can be used by companies and organizations to automatically process an enormous amount of data, for example, forms.

At its early days, OCR was used for mail sorting, bank cheque reading, and signature verification. Other uses of OCR include processing utility bills, passport validation, pen computing, and automated number plate recognition [QA09]. The first implementations were not computers but mechanical devices that would recognize characters (very slowly and with low accuracy). In 1951, M. Sheppard invented a reading robot that could read musical notations as well as words (23 characters in total) on a printed page (one by one)[Sat12]. Later in 1954, J.Rainbow created a machine that was able to read uppercase typewritten English character (one per minute).

Depending on the type of input, OCR systems can be categorized as handwriting or machine printed character recognition, as showed in Figure 2.2. Handwriting can be divided into on-line and off-line systems. On-line OCR systems perform, in real-time, while the user is writing the character. These are less complex since they can capture the temporal or time-based information like speed, velocity, number of strokes made and the direction of writing of strokes. The offline recognition systems run on static data inputs such as bitmaps.

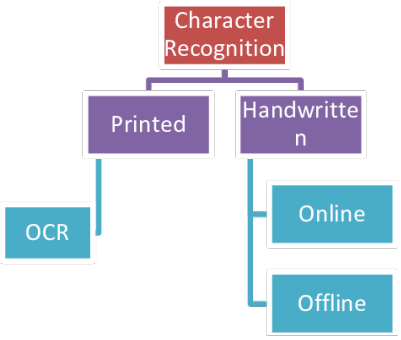

Figure 2.2: Types of character recognition systems [IIN16]

The process of Optical Character Recognition is composed of multiple activities and different phases, such as Image Acquisition, Pre-processing, Character segmentation, Feature extraction, Character Classification, and Post-processing, which will be discussed next.

Image acquisition is the first phase where an image is captured from an external source like a scanner or a camera obtaining a digital image that can be converted into a suitable form that can be easily processed by a computer. This process can involve quantization, such as binarization that involves two levels of an image, as well as compression of the image [Bha12].

Table 2.1: Major Phases of OCR system [IIN16]

Phase	Description	Approaches
Acquisition	The process of acquiring an image	Digitization, binarization, compression
Preprocessing	To enhance the quality of image	Noise removal, Skew removal, thinning, morphological operations
Segmentation	To separate the image into its constituent characters	Implicit Vs Explicit Segmentation
Feature Extraction	To extract features from the image	Geometrical feature such as loops, corner points Statistical features such as moments
Classification	To categorize a character into its particular label	Neural Network, Bayesian, Nearest Neighborhood
Post-processing	To improve the accuracy of OCR results	Contextual approaches, multiple classifiers, dictionary based approaches

Once the image has been collected, different preprocessing steps can be applied to improve the quality of the image, such as noise removal and thresholding. Morphological operations like erosion, dilation opening, and closing can also enhance the quality. Finding the skew in a document can boost considerably the performance of the system, using techniques like projection profiles, Hough transform and nearest neighborhood. Also, finding lines in the document can be part of this phase (based on projections or clustering of the pixels).

After trying to enhance the image to make it as "machine-readable" as possible, the Character segmentation phase starts. In this phase, we try to separate the characters in a way that they can be passed to the recognition engine. In complex situations where characters overlap or a great amount of noise is present, advanced character segmentation techniques are used [GBT14].

With the characters formerly segmented, the characters are then processed to extract different features, where based on these, they will be recognized. These extracted features, such as the image itself, geometric features (loops, strokes) and statistical features (moments), should uniquely identify the characters.

Now that the features are found for each character, we proceed to associate these features to the corresponding character, mapping them to a different category or class. Structural classification techniques are based on features extracted from the structure of the image. On the other hand, Statistical pattern classification uses probabilistic models and other methods like Bayesian classifier, decision tree classifier, neural network classifier, nearest neighborhood classifiers, etc... [CMGS11]

Since the results are never 100 percent correct, Post-processing techniques can be implemented to refine the system. The use of multiple combined classifiers, contextual methods and lexical processing techniques (spell checkers and dictionaries) can also benefit the overall results.

2.2.1 Techniques

In this section, we will present techniques that deal with multiple problems that are not easily solved in Optical Character Recognition systems, such as languages with characters that overlap most of the time, different fonts and sizes in characters, etc...

Moftah Elzobi [EAHAA+14] developed a method that recognizes handwritten Arabic characters, using a segmentation based recognition approach based on Gabor wavelet transform and SVM. Considering the peculiarities of the Arabic characters, the segmentation is divided into two steps. First the PAW's (piece of Arabic word or sub-word [GNP17], as presented in Figure 2.3) overlapping is resolved by finding the connected components and baselines and formulating a set of heuristics. Using the overlapping free image, word segmentation is applied. Once segmented, Gabot filter based features are extracted from each character image that are then used by a SVM classifier (reduces the number of classes). One thousand words were used to train the classifier resulting in a recognition rate between 43 and 93 %, depending on the character.

Figure 2.3: Overlapping Arabic sub-words. [GNP17]

In his paper, Tasnuva Hassan [HK15] presented a method to recognize Bangla Numerals using a local binary pattern texture operator. Initially pre-processing steps are taken in order to improve the image quality such as Gaussian low pass filter, slant correction (KSC algorithm) and images are normalized to a standard size. In the feature extraction phase, three variants of Local Binary Pattern are used (Basic, Uniform and Simplified). From a CMATER database, one thousand of six thousand characters were randomly selected, showing impressive results: 96.7, 96.6 and 96.5 percent for Basic, Uniform and Simplified LBP respectively.

In [LNW15], a commercial OCR software was developed, using Tesseract [Smi07] (open source ORC engine), to train a Tamil model with different fonts and sizes. Computer generated images of Tamil characters were used to train the model. Character segmentation is done by assigning each character a box file. Individual training was done for each each character model, using different images (14031 characters scanned from 20 ancient Tamil books), that varied in font size and type, reaching a 81% accuracy.

Ali Farhat [FAZAQ+16] proposes a method for automatic recognition of Qatar number plates. First, images are converted to binary images and noise removal techniques are applied to remove dust particles and scratches, very common in vehicles number plates. After the pre-processing stages, four types of algorithms are used in the feature extraction stage, to generate encoded text from the segmented characters: Vector crossing, one of the basic methods in OCR, where the number of vectors passing the character is counted and then compared with the set defined for

each character; Zoning, another method that divides the character image in to zones and calculates the density of white and back pixels in each zone and then matches with reference character zones (highly affected by noise and needs more time to process the image); Combination of Vector crossing and Zoning methods taking in consideration both vector found and amount of density pixels, resulting on a 90.43% recognition rate; The idea of the template matching algorithm is to match segmented characters with pre-defined template characters. This matching process is through image correlation, where two images are correlated in order to find the normalized correlation coefficient. This method got the highest rate (99.5%).

It is clear that by alternating the various techniques used in each phase we can achieve the best results [Bha12].

2.2.2 Tesseract

Since we need to recognize the text presented in the leaflet pages a character recognition engine is need. Tesseract is a free open-source OCR engine developed by Google and released in 2006. It's a very popular engine considering it has Unicode(UFT-8) support, and can detect more than 100 languages [RSa, Goo], including Portuguese and it's one of the most accurate open-source OCR engines.

This software can detect very popular fonts like Arial and Time New Tomans, but can encounter difficulties when trying to detect more artistic or exotic fonts. When battling with this font training is required. Training an OCR model takes the following steps [Bar17]:

- Preparation of the training data set: In order to get the best performance out of the model, the character used in training should be as similar as possible to those which the OCR will be used against. It's possible to obtain this samples by extracting them from real images, however this process can be exhausting. A simpler way of having good images to train the model is creating a set of the images of the same character by applying some operations like:
  - Rotation
  - Dilatation and erosion
  - Shearing
  - Adding noise
- Normalization: Essential to this process. Resizes every character to one single size, allowing the OCR to recognize characters of different sizes and, at the same time, reducing the amount of data in character classification.
- Training the OCR model with the data set: After having the training set created the training can start.

The ORC engine used in this thesis(Tesseract) makes use of neural networks and already come trained.

2.3 Hough Transform

When detecting objects in images, classes of shapes defining the objects are constructed. Shape detection is an important challenge for automated image analysis, even for simple geometric patterns like straight lines, circles and ellipses, since it's very computationally intensive [MC15].

While trying to detect and plot the tracks of subatomic particles in bubble chamber photographs, Paul Hough, converted curve detection problem into an efficient peak detection problem in parameter space. The method became popular in computer vision society by Rosenfeld, that gave the first algebraic form for the transform and developed a simple digital implementation of the transform space as an array of counters [MC15].

2.3.1 Hough Line Transform

If a shape can be represented in a mathematical form, an Hough Transform can detect it, even if it's broken or distorted. Taking as example a line, that can be represented as:

$$y = mx + c$$

$$\rho = x\cos\theta + y\sin\theta$$

where ρ is the perpendicular distance from the origin to the line and θ is the angle formed by the this perpendicular line and the horizontal axis (measured in counter-clockwise) [Opec]. A visual representation is presented in Figure 2.4.

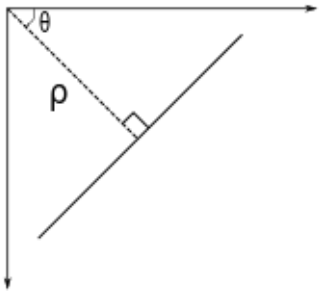

Figure 2.4: Visual representation of a line in parametric form [Opec]

Any line can be represented using (ρ, θ). A two dimensional array or accumulator (used to hold values of the parameters) is initialized to zero. Denoting the rows ρ and the columns θ, the size of the array depends on the accuracy that we want to achieve. If accuracy for the angle is 1, we will have 180 columns and if the maximum distance to the origin is the the diagonal length of the image, we will have 1 pixel accuracy and hence the number of rows equal to that diagonal length of the image [Opec].

Considering a 100x100 image with a vertical line at the middle. First, we take the first point found and since we know its coordinates (x and y) values, in the line equations we put all the

values of θ (depending on the accuracy) and check the values of ρ. For every pair found, we increment the value by on the accumulator cell. On the first iteration, the cell (50,0) will be equal to 1. On the second iteration we do the same and the cell (50, 0) will be incremented one more time, while other cells may or may not. This process is called voting.

At the end, the cell (50,0) will have maximum votes and we know that there is a line at distance 50 from origin and at angle 0 degrees.

2.3.2 Probabilistic Hough Transform

As we saw in the example above, even for a geometric shape as a line, that only needs two parameters, it takes a lot of computation. Probabilistic Hough Transform is an optimization of Hough Transform, since it doesn't take all the points into consideration [Opec], only a random subset of points are used, still being sufficient for shape detection as we can see in Figure 2.5.

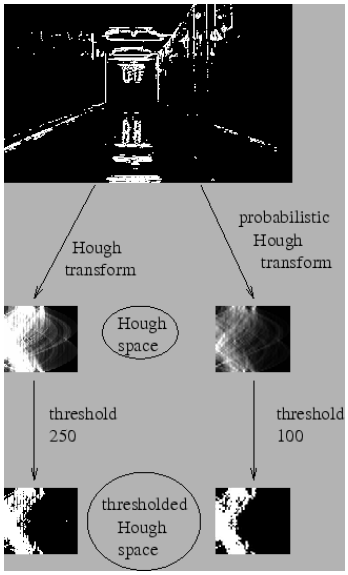

Figure 2.5: Example of Hough transform vs Probabilistic Hough transform [Opec]

2.4 Scene text location

Locating and recognizing text in images is still a problem without a perfect solution in the computer vision community [NM12]. This process can be a very computational expensive task, since any of 2^N subsets can correspond to text (where N is the number of pixels). Text location methods try to solve the problem in two different ways. Some methods are based on a sliding window [CY04, LLL+11, LW02], limiting the search to a subset of image rectangles. This reduction allows that the number of subsets checked reduce to cN, where c is constant where $c < 1$ for one scale and single-rotation methods and $c >= 1$ for methods that try to handle text with different scale, aspect, rotation, etc...

The second group of methods [EOW10, PHL09, NM11b, NM11a, ZK11] finds individual characters by grouping pixels into regions using connected component analysis. This methods rely on the fact that the pixels that belong to the same character, are likely to have similar properties. The advantage of these methods is that the complexity, in most of the cases, doesn't depend on the properties of the text and also provides segmentation that can be exploited by OCR engines. The obvious disadvantage is the sensitivity to clutter and occlusions that change the connected component structure [NM12].

2.4.1 Class-specific Extremal Regions for scene text detection

This algorithm was developed by Lukás Neumann and Jiri Matas [NM12] and the main idea behind it is similar to Maximally Stable Extremal Regions (MSER). MSER are used as a method of blob detection, that aim to detect regions in digital images that differ in properties, such as brightness or color, when comparing with the surroundings.

Such as MSER, Extremal Regions are selected from the component tree of the image, but this method differs from MSER since the selection is done by a sequential classifier trained for character recognition, removing the requirement for stable Extremal Regions and selecting class-specific regions [Oped].

The component tree may contain a enormous amount of regions even for a very single image as we can see in Figure 2.6, this regions can reach orders of 10^6 for an average 1 Megapixel image.

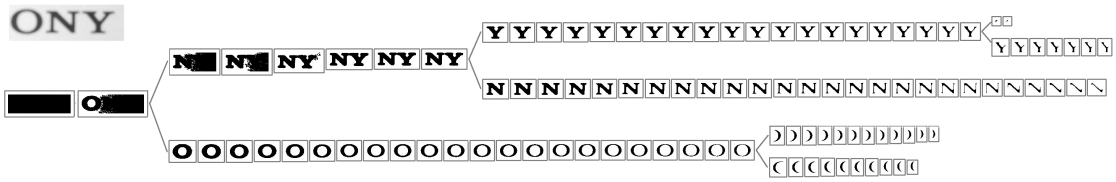

Figure 2.6: Component tree example [Oped]

In order to make an efficient selection of suitable regions, this algorithm makes use of a sequencial classifier with two different stages:

- Stage 1: Computable descriptors like area, perimeter, bounding box and euler number are computed with O(1) complexity for each region found and used as features for a classifier that calculates the probability of a the region being a character. Only regions with probability value above the global minimum defined and with difference between local maximum and local minimum are selected [Oped].
- Stage 2: ER that pass the first stage are then classified as character or not, taking in consideration more computationally expensive features, such as Hole area ratio, convex hull ratio, and the number of outer boundary inflection points.

This filtering process in done in different channel projections of the image to improve the character localization recall. Once this process is done, the character candidates, must be grouped in high-level text blocks (words, text lines,...).

In Figure 2.7, we can see the robustness, against noise and low contrast of character, of the method presented. The "false positive found" is caused by watermarks embed in the image [Oped].

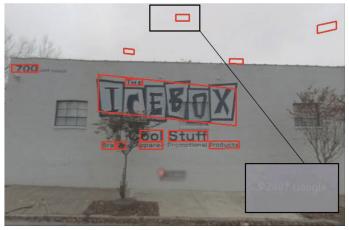

Figure 2.7: Example of algorithm application [Oped]

2.5 Semantic Image segmentation

Image segmentation is a computer vision process that breaks an original image into non-overlapped meaningful regions (objects or parts).

One of the main problems with image segmentation is the definition of this so-called "objects" since it's debatable what makes an object meaningful. Objects can take many forms like dogs, books, wood, rocks or a forest. There are also cases where an object is relevant to another object. It's noticeable that without a well clear definition of the "object" the process of image segmentation becomes more challenging. Once divided, a label has to be given to each particular object/region. This process is called image classification. Representing the content of an image using accurate words it's quite a challenge, but can give an autonomous process a great boost in decision making, for example, autonomous driving in self-driving cars [SKC+17].

Semantic segmentation is also known as image labeling or scene parsing. This process gives labels to every pixel in the image.

To understand the performance of this methods four general parameters have to be defined:

- Accuracy: In Figure 2.8 we can see the qualitative impression of this measure. A per-pixel rate classification is used to compare results. There are some problems associated with this measure, such as classification of large areas than can have sub-classes (wheel of a car, but it's still a car).
- Memory usage: If the algorithm requires a lot of memory to process the images, it is only possible to run it on the latest graphics cards. Making it very hard to run on mobile devices like smartphones or cameras.
- Stability: An important requirement on semantic segmentation is the stability of the process. A change in the input image shouldn't change dramatically the results, for example, a slightly blurred image.
- Speed: Taking the autonomous driving example, the classification of non-street and street pixels needs to be done in the smaller amount of time possible (20 ms [BKTT15]). This time is called latency.

Figure 2.8: Qualitative representation of accuracy [SKC+17]

In most cases, semantic segmentation is done by a classifier that operates on fixed-size feature inputs with a sliding-window approach [HZRS16, LZ16]. So the classifier is trained on fixed sized

images and then fed with rectangular regions of the image (called windows). Neural networks are able to apply this method is a very efficient way [SKC+17].

2.5.1 Techniques

Most of the traditional approaches base themselves in preprocessing methods and features extracted from images. Global features are used to object detection and classification, on the other hand, local features are used for object recognition/identification [SKC+17]:

- Pixel color: RGB, HSV and gray-value are the most common. No single color space has been proved to be more efficient [YBCK10]. RGB is usually selected for its simplicity and for the general support in programming languages.
- Histogram of oriented gradient (HOG): Image is interpreted as a discrete function witch maps a position (x, y) to a color. This method counts occurrences of gradient orientation (directional change in the intensity or color in an image) in a portion of an image.
- Scale-invariant feature transform (SIFT): Combination of key point detector and histogram based gradient representation [SKC+17].
- Bag-of-visual-words (BOV): Very similar to HOG, relies on vector quantization. Features are histograms that count the number of times a certain pattern occurs.
- Poselest: Uses additional manually added key points. Works well with image categories like humans, but not so much with others. Key points have to be chosen for each category.
- Textons: Minimal building block of vision. Deep learning techniques with Convolution Neuronal Networks (CNNs) learn textons within the initial filters.
- Dimensional Reduction: Since pixels can have more than one feature, high resolution images will have millions of features, but not more information. Downsampling high resolution images is a straightforward approach. Principal component analysis (PCA) is another way of solving this problem, by finding a hyperplane on that all feature vectors are project with minimal loss of information.

Unsupervised algorithms used as other source of information to refine segmentation [SKC+17]. While segmented algorithms store information about the classes, non-segmented algorithms like the unsupervised try to detect consistent regions or region boundaries:

- Clustering Algorithms: K-means and mean-shift algorithms are directly applied on the pixels
- Graph-Based Image Segmentation: Interpret pixels as vertices and weight as edges is a measure of dissimilarity like the difference in color.

- Random Walk: Seed points are placed on the image for the various objects within the image. A random walk is calculated to achieve the various seed points from each single pixel [SKC+17].

- Active Contour Models: Algorithms that segment images on solid edges, but also try to find borders witch are smooth.

- Watershed Segmentation: A gray-scale version of the image is taken by the watershed algorithm. Low and high values are catch and used by the algorithm.

2.6 Information extraction

In this section, some general concepts, techniques and methods of Information extraction will be taken in consideration.

A way of improving the accuracy rate of OCR systems, is by introducing knowledge about the language and context information in post-processing phases in OCR [ZZ05]. There multiple post-processing approaches based on language knowledge using lexicon/wordbooks or syntax and semantic rules in order to choose the best characters recognized by the OCR engine.

Besides returning candidate characters, OCR engines also return candidate distance information for each character. This information is used as reliability of the corresponding candidate character to be chosen [ZZ05].

Some statistical approaches proposed [LZH+06] achieved good results in a big part of applications because of its low complexity and easy implementation.

Zhuang [ZZ05], proposed an OCR post-processing approach based on multiknowledge, integrating both language knowledge and candidate distance information. Combining statistical language model and semantic lexicon was used to reduce the size of the search space. The results showed that by integrating this pos-processing techniques accuracy improved from 58.45% to 83.73%, witch represents a 60.84% error reduction.

One of the most used models for statistical language in OCR is the ngram model, that calculates the probability of a word being in a text based on the words already removed by the OCR engine [Goo01].

More recently, many attempts have been done to remove the necessity of having context knowledge and overcoming the lack of knowledge databases.

2.7 Machine learning

In this section, a very brief introduction about machine learning will be made. Some "off-the-shelf" approaches will be analyzed in order to compare with the results of the approaches described so far.

Machine learning is an alternative to the conventional engineering approach for the design of an algorithmic solution [Sim18]. In Figure 2.9 (a), we can see this flow, starting with the acquisition of domain knowledge, where the problem is studied in detail, producing a mathematical model that captures the problem and then an optical algorithm is developed, performing well under the guaranty that the model capture is an accurate representation of the reality [Sim18].

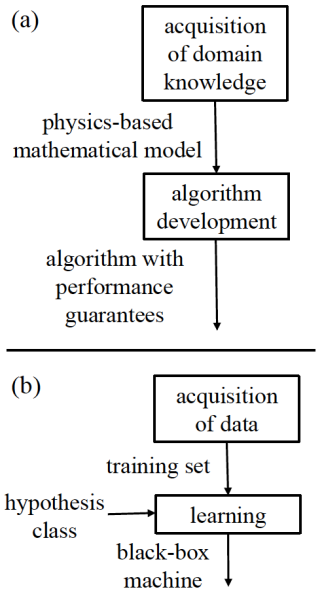

Figure 2.9: (a) Conventional engineering design flow; and (b) baseline machine learning methodology [Sim18]

On the other hand, Machine learning substitutes the step of acquiring domain logic with the potentially easier task of labeling sufficiently large number of positive examples (training set) to train the model. As we can see in Figure 2.9 (b), the training set is then fed to a learning algorithm that produces a trained machine capable of performing the desired task [Sim18].

Machine learning can also integrate domain knowledge during the learning phase, as represented in Figure 2.10.

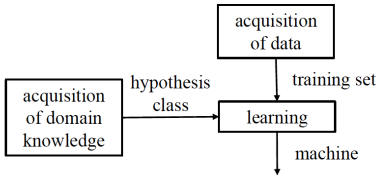

Figure 2.10: Design flow using domain knowledge [Sim18]

Three main classes can also be found in machine learning techniques:

- Supervised learning: training set consists of pairs of input and desired output, an the final goal is learning the mapping between both [Sim18].
- Unsupervised learning: in this case, the training set consists uniquely of unlabelled inputs, without any defined desired output.
- Reinforcement learning: categorized in between supervised and unsupervised. Some form of supervision exists, but is not in the form of desired output. Instead, feedback is given after every selection made by the learning algorithm. Applies to sequential decision making problems.

2.7.1 Haar Feature-based Cascade Classifiers

This is a machine learning approach, developed by Paul Viola and Michael Jones [VJ01], where a cascade function is trained from a big set of positive and negative images, in order to detect objects in images [Opeb].

Taking the well known example of face detection, the algorithm needs a lot of positive images (faces) and negative images (images without faces) to train the classifier. Once having the training set ready, we need to extract the features from it. Three Haar features are used in this method (Figure 2.11), being a single value obtained by subtracting the sum of pixels under the white rectangle from the sum of pixels under the black rectangle [Opeb].

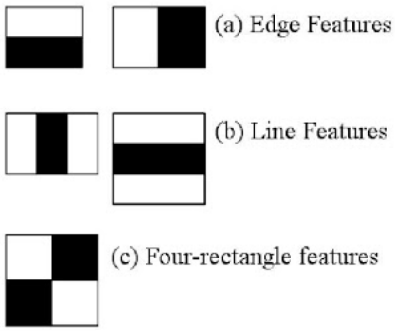

Figure 2.11: Haar features [Opeb]

Now all possible sizes and locations are calculated. This can generate more than 160,000 features for a simple 24x24 window. In order to reduce the number of calculations performed, a concept of integral image was introduced, making it possible, no matter how large the image is, to perform calculations for a given pixel involving just four pixels.

Out of the enormous number of features found, the best are selected using Adaboost. Adaboost applies every features in all the training images, and finds the best threshold which classify the faces to positive. Features with smaller minimum error rate (features that better classified images as faces and not faces) are selected [Opeb]. This reduces the number of features to 6000. But this number is still big and very time consuming. Cascade of Classifiers try to solve this problem.

Instead of having to check every window with every feature, a simpler discard method is used to discard windows that wont have faces. This method uses only a smaller set of features that are easier to calculate. With this, windows that are discarded (don't pass the first set of features) are not checked again. The windows the passed the first method are then checked with features that are harder to calculate to understand if there is a face present or not [Opeb].

Chapter 3

Methodology

In this chapter, a definition of the problem will be described and analyzed. The data sets used and details about the implementations developed during this dissertation will also be explained.

3.1 Problem definition

As stated before, leaflets are designed to call consumer's attention, so most of them have a great amount of text that might not bring useful information to the product/promotion extraction, as we can see in the middle of the Figure 3.1. The presence of noise and text that does not belong to a product/promotion, if not filtered correctly, will generate a lot of false positives. In same cases, pages have a number of product/promotions above normal, making it harder to extract them. Additionally, since companies have different designs for their leaflets, we have to expect different displays, fonts and sizes.

Another challenge in this project is the fact that, most of the times, sentences don't follow usual grammatical rules or they use simple abbreviations as showed in Figure 3.2. In this example, We can see two figures where the first says "Save 25%" and the other just says "More than 35%". For a human is easy to interpret the image, being obvious what the retailer is trying to say with the least possible words (the product has a discount of more than 35%).

In order to show transparency, companies like to show the last price applied for a certain product as we can see in Figure 3.2. Detecting this value is easy for a human eye, but not for an OCR engine.

Another big challenge happens when information is not only in text but also in the images. Figure 3.3 presents a great example. The retailer is applying a 50% promotion to all brands presented in the images. The problem is that the brands are not present as text but in the logos.

Figure 3.1: Noise present in leaflets(Pingo Doce)

3.2 Data sets

In this section we will cover the data sets that were explored in this dissertation. The data set created in Section 3.2.1, comes from a Canadian website that displays weekly leaflets allowing users to click on the product and check the prices. The second data set (Section 3.2.2), was provided by a multinational company, containing annotations about products in leaflets.

3.2.1 Flyertown

The motivation for this project was a challenge from a multinational portuguese company, which provided some annotated data. Additionally, we identified some other sources of potentially useful data. Flyertown [Fly19] is a website that manually annotates leaflets, and allows users to click on the leaflet and inspect the products. The website allows users to see the leaflet and interact with it by clicking on the product and a pop up appears with the product description and price as seen in Figure **??**.

Unfortunately, for reasons that can be understood, collecting the detailed information from this site is very difficult to automate and, thus, we ended up not being able to use it.

3.2.2 Portuguese leaflets

Obtaining data from national retailers was also not easy. We used two small sets of data: 1. 4 leaflets from different retailers which we collected and manually annotated; 2. 30 leaflets from 3 retailers, provided by a multinational portuguese company. The latter was sampled from a larger set of leaflets that also included some annotations. However, significant data preparation was still necessary, which lead us to work on a small sample. For every company in the competition the data set had different numbers of annotations as showed in Table 3.1.

Figure 3.2: Grammar and abbreviation examples(Pingo Doce)

After inspecting the data set provided a great number of problems were quickly noticed:

- The first semester annotations of Jumbo (9355) and Lidl (7424) and some in other semesters don't have a page reference. With the time left, they had to be discarded.
- Since Sonae uses this as business intelligence, there was a relationship between the products found in the leaflets and their own (Sonae products were also on the xlsx file).
- Some leaflets were not provided.
- Sometimes, leaflets have different numbers but are the same (because of different dates of promotions).
- The product description is never exactly what is in the image. For example, if in a leaflet page a cheese promotion was present, instead of writing the complete portuguese word ("Queijo"), a simplification was made ("Qj."). Since we want to evaluate the words extracted this came quite a challenge.
- Product discount(%) sometimes is not what is in the image, but the real percentage (based on the previous annotations made by the company).
- Some products are not annotated.
- Some products didn't have a leaflet number associated.
- Some products are not on the right page.
- Cases like "50% discount in all BRAND product", all the products of that brand are there and not just the promotion.

After an inspection of the data set, a list of all the Sonae sub brands was created and used it to develop a script that deletes them from the data set.

Figure 3.3: Example of text present in an image(Pingo Doce)

Even though almost every product had a leaflet number associated, there was no direct relationship between the leaflet number on excel and the leaflets provided by Sonae, so a translator was developed, based on the title and the dates provided on the excel to find them in the folder (some leaflets had to be renamed since they were completely off).

With all the problems described above, we had to selected and fix manually the data. Since the data set has a great amount of leaflets and to ensure variety, we tried, as possible, to select 2 random pages of different types of leaflets and was able to extract the numbers presented in Table 3.2.

The first set of data was used for development while the second was used for testing purposes.

Table 3.1: Sonae data set - unparsed data

Company	Number of rows	Product promotions
Pingo Doce	176060	???????
Jumbo	9355	380
Lidl	7424	13064

3.3 Implementation

In this section, we provide details about the approaches chosen to implement in order to solve the problem described in Section 3.1. In the first approach 3.3.1, we will apply an OCR engine in the full leaflet page. In Section 3.3.2, we take advantage of the fact that sometimes product/promotions are inside boxes by finding lines and applying OCR inside those boxes. A more general approach is explored in Section 3.3.3, where areas with text are first located and then the text is extracted from them. To understand the importance of having information about the context, in Section 3.3.4 we explore the impact of using a brand/type of promotion database on other methods. Finally, given the recent success of machine learning approaches in processing images, in Section 3.3.5 we explore Haar Feature-based Cascade Classifiers to detect prices and promotion tags.

3.3.1 Baseline

This approach is a very basic method, that will be used as a baseline when comparing with other developed methods.

The objective of this approach is when receiving a leaflet page, apply off-the-shelf OCR in each page and try to extract information about the promotions. A visual representation of this approach can be seen on Figure 3.4.

Approach 1: OCR in full image and simple text compilation

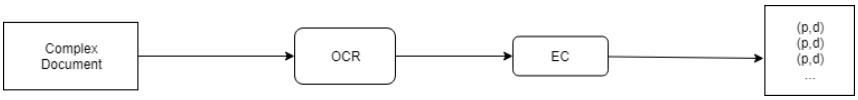

Figure 3.4: Approach 1 - Baseline

3.3.1.1 Method description

In order to develop a reasonable method, it was necessary to understand the structure of what would be return by Tesseract when given a full page. We used the image_to_string method in pytesseract which has the parameters showed in Table 3.3.

When running the method in Figure 3.5, the method returned the following string:

Table 3.2: Data set Sonae - parsed data

Company	Leaflets	Pages	Promotions
Pingo Doce	10	20	433
Lidl	10	20	124
Jumbo	10	40	456
Total	30	80	1013

```
"Pilhas PANASONIC\nAA/ AAA 6 + 4 Gr\u00e1tis\n\nUnid.\n\nh\n\n:.\n\nV\u00e1rias
    Desde\nUnid.\n\n'\u00ed*\nF\nS\n\n&\n\nA\n=\n\nEscova MICHELIN Flat Blade\n\n \
    n\n \n\nEstendal Ch\u00e3o\nVILEDA\n\nX-Legs Universal\nBm\n\nDesde Unid.\n\n \
    n\nSalamandra a Lenha\nCh\u00e3 SBA\n\n \n\nCapa A4 AMBAR Capa A4 c/ Recarga
    AMBAR\nLombada Larga Marshmallow Marshmallow\nSortido Unid. Sortido Unid.\n\
    nCOMPRE COM\n\n \n\nCaixa Vidro Herm\u00e9tica\nFRIGOVERRE sSortida\n\n \n\
    nTrem Cozinha 7 Pe\u00e7as + Faqueiro\n24 Pe\u00e7as GUIMAR\u00c3ES\n\n \n \n \
    n\n- ET\n\nTrolley ABS 4 Rodas AIRPORT\nPreto/ Azul/ Creme 55 x 40 x 20 cm\n44
    ,99\u20ac 65 x 44 x 24 cm\n\n49,99\u20ac 75 x 48 x 28 cm\n\nUnid.\n\n \n\nLivro
    \n\nA Arte da Guerra - Sun Tzu/\n\nO Principezinho - Antoine de Saint-Exup\
    u00e9ry/\nMensagem - Fernando Pessoa/\n\nAlicg no Pa\u00eds das Maravilhas -
    Lewis Carroll\nUnid.\n\nLewis Carroll\ne\nM\n=\n\n* Para pagamentos com o Cart\
    u00e3o Jumbo mais.\n\nTAEG do cart\u00e3o de cr\u00e9dito: 15,2%\n\nExemplo
    para um limite de cr\u00e9dito de 1.500\u20ac pago em 12 meses. TAN: 14,69%.
    Ades\u00e3o ao Cart\u00e3o sujeita a aprova\u00e7\u00e3o.\n0% DE ]u ROS*
    Anuchan atua na qualidade de Intermedi\u00e1rio de Cr\u00e9dito a t\u00edtulo
    acess\u00f3rio com exclusividade. Informe-se junto do Oney Bank Portugal.\n\na\
    n1 . T Fs\n' \u00c1 '?/ Mensagem\n' \u00e9 S\n! S\n; -\nA Arte/da Guetra *\n:
    Sta a N o . *\n1 \u2014 gm\n\u00f5 L\u00aa = o\n;\n1\n1\n'\n1"
```

A lot of text is recognized, however, to retrieve the promotions, additional parsing is needed. Consider, for instance, the first part of the string:

```
"Pilhas PANASONIC\nAA/ AAA 6 + 4 Gr\u00e1tis\n\nUnid."
```

This is actually the description of the first product but we have a lot of "\n" in the middle. Since "\n" means a new line, the line before its close and it's probably the same product, but two new lines probably mean that they are further apart and are different products.

As we can see this is not always true, but it's a start. So, after retrieving the text found, the string was splited every time two new lines were found, generating something like:

```
['Pilhas PANASONIC\nAA/ AAA 6 + 4 Gr tis',
'Unid.', 'h', ':.',
'V\ rias  Desde\nUnid.',
"'  *\nF\nS",
'&', 'A\n=', 'Escova MICHELIN Flat Blade', ' ', ' ', 'Estendal Ch o\nVILEDA',
'X-Legs Universal\nBm', 'Desde Unid.', ' ', 'Salamandra a Lenha\nCh  SBA', ' ',
```

Table 3.3: Pytesseract - image to string parameters

Parameters	Description
image	Object, PIL Image/NumPy array of the image to be processed by Tesseract
lang	String, Tesseract language code string

```
7 'Capa A4 AMBAR Capa A4 c/ Recarga AMBAR\nLombada Larga Marshmallow Marshmallow\
      nSortido Unid. Sortido Unid.',
8 'COMPRE COM', ' ', 'Caixa Vidro Herm tica\nFRIGOVERRE sSortida', ' ',
9 'Trem Cozinha 7 Pe as + Faqueiro\n24 Pe as  GUIMAR ES', ' \n \n ', '- ET',
10 'Trolley ABS 4 Rodas AIRPORT\nPreto/ Azul/ Creme 55 x 40 x 20 cm\n44,99    65 x 44
      x 24 cm',
11 '49,99    75 x 48 x 28 cm', 'Unid.', ' ', 'Livro', 'A Arte da Guerra - Sun Tzu/',
12 'O Principezinho - Antoine de Saint-Exup ry/\nMensagem - Fernando Pessoa/'...]
```

Empty strings and strings that are too small after being stripped (i.e removal of all empty spaces) are discarded. After that we start to extract the information about the promotion. The first string found is split by the end line:

```
1 ['Pilhas PANASONIC', 'AA/ AAA 6 + 4 Gr tis']
```

The output is then:

```
1 {
2     "brand": "PILHAS PANASONIC",
3     "description": "PILHAS PANASONIC AA/ AAA 6 + 4 GR\u00c1TIS"
4     "type_of_promotion": null,
5     "promotion": null,
6     "price": null,
7     "last_price": null
8 }
```

There is a high probability of the brand being in the first line. The brand is almost correctly identified in this example, having one more word (an additional word was included - "PILHAS"). The description includes all the text found without new lines. If a "%" is found in that line, the type of promotion is set to 'DISCOUNT (%)' and the one/two digits preceding it are extracted as the value of the promotion. When a "€" is found in that line, the price is set to the digits preceding it. If two prices are found the first is the price being applied and the second the last price. This is not necessarily the best approach since the last price can appear first.

Figure 3.5: Leaflet example

3.3.2 Hough Line Transform

As stated in Section 2.1, 70% of the leaflets analyzed had clear border layouts. Additionally, in a considerable amount of cases, products are inside boxes/squares as shown in Figure 3.6. As we saw in the baseline method, there was difficulties while parsing the string to generate promotions, since the OCR is applied to the full page. It is reasonable to assume that these boxes will most of the times contain information about a single product. So, if we can identify these boxes in the leaflet, the OCR will more easily extract the information for the corresponding product.

This approach is an image processing approach that tries to finds lines in the leaflet page and constructs boxes to be later analyzed by the OCR engine, as showed in Figure 3.7. As explained in section 2.3.1, Hough Line Transform finds lines in images, but first the image needs to be processed by an edge detector.

3.3.2.1 Canny Edge detection

Canny Edge detection is a popular edge detection developed by John F. Canny in 1986 [Opea]. The algorithm takes the following stages:

1. Noise reduction: The method is susceptible to the noise presented in the image, so the first step removes the noise with 5 x 5 Gaussian filter, returning a smoother image.

Figure 3.6: Pingo Doce - Example

2. Finding Intensity Gradient of the Image: The image is then filtered with a Sobel kernel in both horizontal and vertical direction to get the first derivative in both directions. From this, we can find edge gradient and direction for each pixel with the following formulas:

$$Edge\_Gradient(G) = \sqrt{Gx^2 + Gy^2}$$

$$Angle(\theta) = \tan^{-1}(\frac{Gx}{Gy})$$

Gradient direction is always perpendicular to edges. It is rounded to one of four angles representing vertical, horizontal and two diagonal directions.

3. Non-maximum Suppression: The next step is to perform a full scan of the image removing unwanted pixels that might not contain edges. hen, the pixels with local maxima in the direction of the gradient (i.e. in the corresponding neighborhood) [Opea].

   Taking the example in Figure 3.8, point A is on an edge and the gradient direction is orthogonal to the edge. Since points B and C are in gradient directions, A is checked in order to see if it forms a local maximum. If so, is considered to the next stage otherwise put to 0.

4. Hysteresis Thresholding: This stage decides which candidate edges are really edges. For this, we need two threshold values: minVal and maxVal. Pixels with gradients above the maxVal are sure to be edges and values bellow minVal are discarded. Values between the

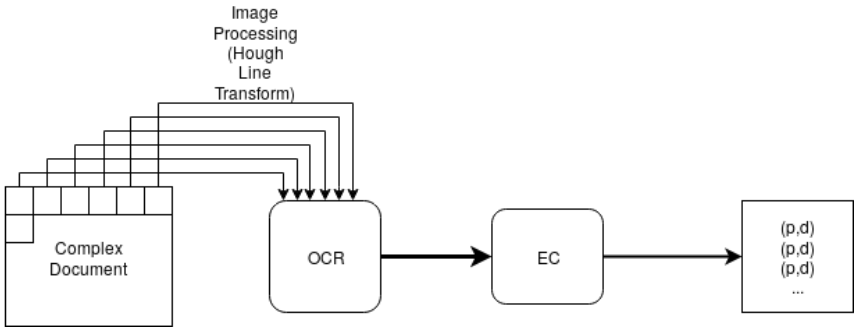

Figure 3.7: Approach 2 - Hough Line Transform

thresholds are classified based on their connectivity. If they are connected with any "sure-edge" they are considered to be part of the edges. Otherwise, they are not considered.

Looking at Figure 3.9, edge A is above the maxVal so it's considered a sure-edge. Edge C is bellow the maxVal, but is connected to A, that as we saw is a sure-edge, so it's connected to sure-edge, being considered a valid edge. Edge B is not connected to an edge, so is removed.

The values of maxValue and minVal need to selected correctly in order to have the results expected.

It is expected that lower parameter values will identify more points as edges while higher values will make the method less sensitive. Some experiments were carried out to understand the effect of those parameters in our problem. We used the OpenCV method called Canny [Opea] that uses the Canny Edge detection algorithm. The parameters are described in Table 3.4. In Figure 3.17 we can see the method applied with threshold values of 50/100, 200/250 and 300/500 respectively (with default aperture size and L2 gradient flag). This result confirms the hypothesis above.

Table 3.4: OpenCV - Canny method parameters

Parameters	**Description**
image	8-bit input image
threshold1	first threshold for the hysteresis procedure
threshold2	second threshold for the hysteresis procedure
apertureSize	aperture size for the Sobel operator (default = 3)
L2gradient	a flag, indicating whether a more accurate L2 norm =sqrt((dI/dx)2+(dI/dy)^2) should be used to calculate the image gradient magnitude (L2gradient=true), or whether the default L1 norm =|dI/dx|+|dI/dy| is enough (L2gradient=false)

3.3.2.2 Detection of boxes

After applying the edge detector, we are ready to find the lines in the page. In this method, a optimization of Hough Transform method will be used to detect lines in the leaflet pages, as

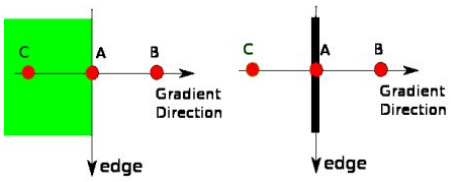

Figure 3.8: Canny - Non-maximum suppresion [Opea]

described in section 2.3.2. OpenCV HoughLinesP method uses the parameters in Table 3.5.

Table 3.5: OpenCV HoughLinesP parameters

Parameters	**Description**
image	8-bit, single-channel binary source image
rho	Distance resolution of the accumulator in pixels
theta	Angle resolution of the accumulator in radians
threshold	Accumulator threshold parameter. Only those lines are returned that get enough votes (>threshold)
minLineLength	Minimum line length. Line segments shorter than that are rejected
maxLineGap	Maximum allowed gap between points on the same line to link them

When running this method on a leaflet page, a list of lines found will be returned. In Figure 3.17, we can see an example, obtained with the parameters in Table 3.6, in a page (a) and, in green, the lines returned (b). Canny edge detection is previously applied with values 50 and 200, for minValue and maxValue, respectively.

Table 3.6: Test run parameters

Parameters	rho	theta	threshold	minLineLength	maxLineGap
Values	1	$\pi/180$	80	30	10

The method returns between 6000 to 8500 lines, depending on the page. Since we need to find the boxes, only vertical and horizontal lines need to be considered. Sometimes the lines found don't have exactly 0 or 90 degrees, so a interval of values is given for horizontal and vertical lines. Line with angles between 88 and 92 degrees are considered vertical and lines with angles between 0 and 2 are considered horizontal.

Now that we only have horizontal and vertical lines, we need to find the pairs of lines that intercept. These will define the box that determines the content that will be analyse by the OCR engine to find a promotion using the same information extraction techniques used in the baseline.

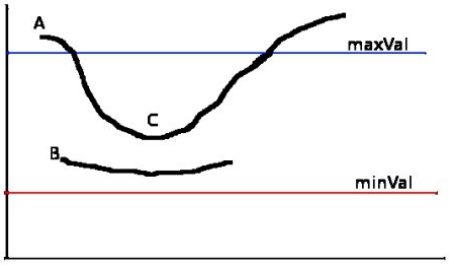

Figure 3.9: Canny - Hysteresis Thresholding [Opea]

3.3.3 Scene text location

The method presented in Section 3.3.2.2, only works when the promotions that are inside of boxes. Since this not always the case, a more general method needs to be explored. As described in Section 2.4, it's possible to locate text in images. In this approach we will use OpenCV implementation of Class-specific Extremal Regions for scene text detection (Section 2.4.1). OCR will be applied to the detected region. If text is really present, a pair product/promotion will be created. A diagram representing this approach can be found in Figure 3.12.

First we need to extract the channels out of the image to be process individually, using the OpenCV computeNMChannels method, that receives an image as input. Then we append negative channels to detect ER (bright regions over dark background).

Then, we extract Class Specific Extremal Regions from every channel. Using a pre-trained model, we load the classifiers, using OpenCV loadClassifierNM1 and loadClassifierNM2 methods. The first classifier will be used for the first stage. It filters the characters using features calculated quickly ($\mathcal{O}(1)$). A second filter is applied to the ones that pass the first one. It uses the second classifier, which is based on more computational expensive features. More details can be found in Section 2.4.1.

We use the results from the filter to find the Extreme Region using the OpenCV `detectRegions` method. We then group them to form words, paragraphs, etc... using the OpenCV method erGrouping. This method returns the box coordinates of the text found.

After extracting the locations of the texts (represented as boxes), we need to apply the OCR engine to retrieve the text and create the product/promotions.

In Table A.1, we can see the methods used in this approach their parameters and default values. In Figure 3.13, we can see the results of this approach when applied to a leaflet page (a) using the default values (b). The text representing the products/promotions is not being found. This occurs because the default value for the minimum area is too high. The product/promotions characters in the image have an average size of 35x35, a lower values has to be used. Reducing this value allows the algorithm to check smaller areas and find the characters as we can see in Figure 3.13 (c). In the example detailed above no pre-processing method was performed on the original image.

After extracting the locations of the texts (represented as boxes), we need to apply the OCR engine to retrieve the text and create the product/promotions.

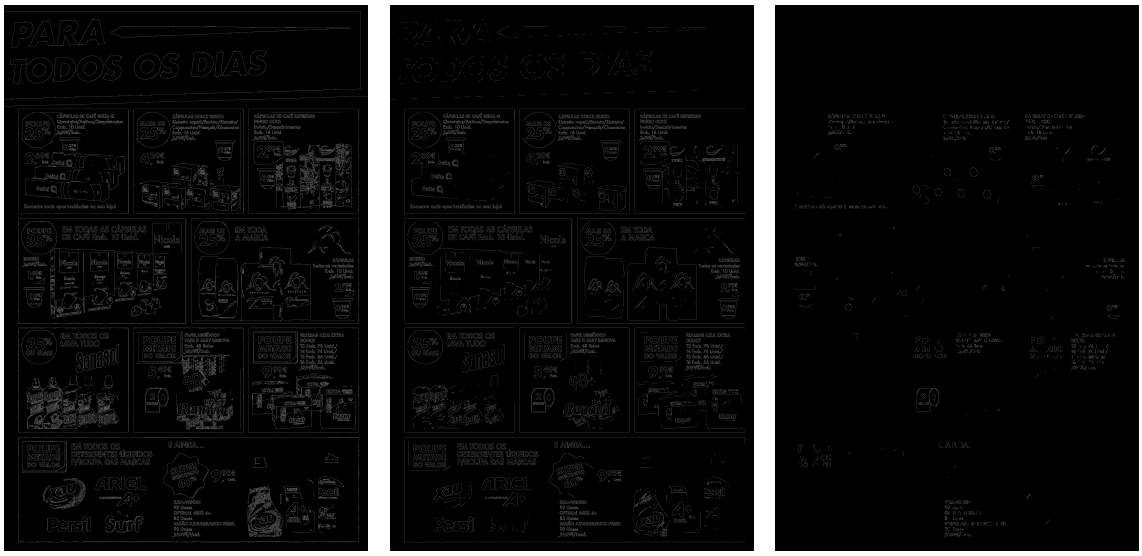

Figure 3.10: Canny - test in leaflet page

3.3.4 Domain logic

In this approach, we will test the advantages off having domain logic information. Using a data set provided by Fraunhofer with information about all the products of a multinational company better results can be expected when trying to extract brands or type of promotions. We assume that a database containing information about the products is available. Using this data, we can check if any brand is inside of the text that is extracted from the leaflets. If this is the case, we assume that that will be the brand for the product/promotion found.

The approach is summarized in Figure 3.14. In our experiments, we have used a database provided by a multinational portuguese company.

3.3.5 Haar Feature-based Cascade Classifiers

n this project, our focus was on trying to use very simple methods on the problem of extracting information from leaflets. However, we need references to assess the quality of the results obtained with those methods. The first method presented, which simply consists in applying an OCR to the whole leaflet, is a very simple baseline. It is used to assess if the remaining methods are actually doing something useful or not. However, we also need an alternative, proven method to assess if the methods proposed are competitive or not. We selected Haar Feature-based Cascade Classifiers, which is an effective object detection method.

If the exact location of the labels in the image was known, OCR with specific configurations could be applied in order to extract the prices/promotions correctly. For example, combining this with the scene text location presented before, we could associate the price/promotion tag with the closest text location.

First, price and promotion tags that need to be found were annotated using LabelImg. As a proof of concept, we decided to test this approach with a part of the data set (Pingo Doce). For that

Figure 3.11: Canny - test in leaflet page

subset, 301 price and 275 promotion tags were annotated. Since this machine learning approach requires training, for each type of label we want to find, we had to label the tags correctly, . The results are summarized in Table 3.7.

A script was developed to crop this images out of the original image and save them in the correct folders. This images represent the positive images (images that we want to find).

In order to train a classifier that detects the tags, OpenCV Cascade Classifier, as stated in Section 2.7.1, will be used and also a open source Docker image [RSb] that installs OpenCV in Ubuntu 16.04 and runs the training.

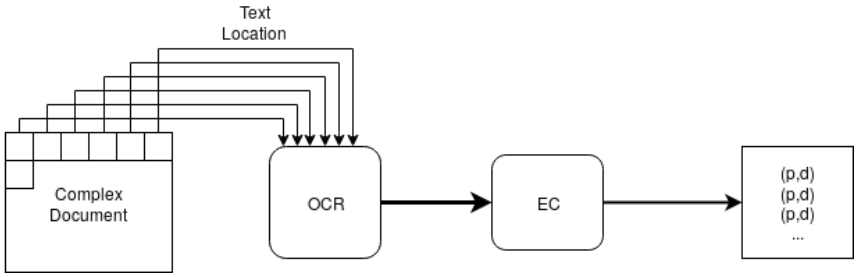

Figure 3.12: Research Question 3 - Scene text location

We used all the price images labeled as positive images and as negative images a set of random generated images cropped from leaflet pages (with the same size of the window). The parameters used can be seen in Table 3.8.

```
docker run \
  --detach \
  --name haar-training \
  --volume /Users/antoniomelo/Documents/Fraunhofer/training/positive_images:/
      positive_images  \
  --volume /Users/antoniomelo/Documents/Fraunhofer/training/negative_images:/
      negative_images \
  --volume /Users/antoniomelo/Documents/Fraunhofer/training/classifier:/classifier
      \
haar-training
```

We trained models separately for different labels. Some additional improvements were implemented [RSc]:

- Negative images should work as background images, that don't contain images that we want to find. This images need to be bigger that the positive images. We used a data set [RSc] with 3000 random negative images.
- numPos and numNeg need to be smaller than the actual number of images

Running the training again with parameters presented in Table 3.9, for almost four days, showed much better results. Since, not all type of tags had enough annotations, training was done for type 1 of price tags and type 1 of promotion tags.

In an ideal situation and in order to perform better, the negatives images shouldn't be random images but random cropped images of leaflets. This would possibily reduce the number of false positives and but consequence increase precision.

Figure 3.13: Example of approach run in a leaflet page (a) with default values (b) and with corrections (c)

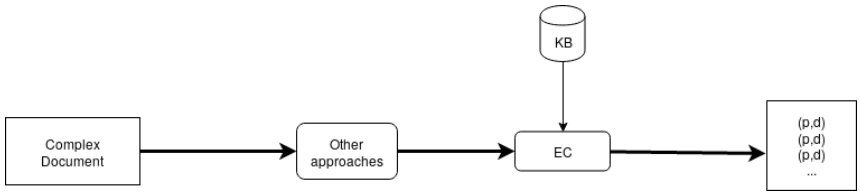

Figure 3.14: Approach 5 - Previous approaches + Domain knowledge

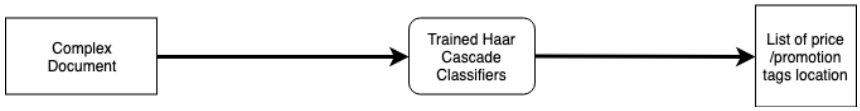

Figure 3.15: Research Question 6 - Haar Feature-based Cascade Classifiers

Table 3.7: Pingo doce price & promotion annotation

Tag	Type Number	Description	Amount
price	1	Yellow background	193
price	2	Yellow background + red strip	29
price	3	Red background + yellow strip	64
price	4	Yellow background + extra info	11
price	5	Yellow background + black strip	1
price	6	Red background + yellow strip + black extra info	3
promotion	1	Red circle	123
promotion	2	Red circle + yellow/black info	1
promotion	3	Red circle + black info	64
promotion	4	Red/Black rectangle	10
promotion	5	Red rectangle	15
promotion	6	Two red rectangles	9
promotion	7	Red circle + yellow strip	32
promotion	8	Two red rectangles + black info	1
promotion	9	Green circle	2

Table 3.8: Test run 1 - parameters values

	numStages	minHitRate	maxFalseAlarmRate	numPos	numNeg	width	height	mode	precalcValBufSize	precalcIdxSize
Values	20	0.999	0.5	301	294	80	40	ALL	256	256

Figure 3.16: Test run 1

Table 3.9: Test run 2 - parameter values

	numStages	minHitRate	maxFalseAlarmRate	numPos	numNeg	width	height	mode	precalcValBufSize	precalcIdxSize
Values	12	0.999	0.5	1000	2000	80	40	ALL	256	256

Figure 3.17: Canny - test in leaflet page

Chapter 4

Results and Discussion

This chapter is fully dedicated to analyze and compare the results of the solutions developed in order to answer the research questions initially proposed. In Section 4.1, the metrics used to evaluate the results and their meanings are explained. In Section 4.2, results of all the methods developed in this dissertation are analyzed.

4.1 Metrics

In order to evaluate the results of the methods developed in this thesis, we will use the following metrics: precision, recall, accuracy and f1. These metrics are based on four values, computed by comparing the predictions made by a model to the truth:: true positives, true negatives, false positives and false negatives (explained bellow)

These values depend on the target of the model. Taking the example of this project, 6 fields may be associated with a promotion: brand, description, type of promotion, promotion, price and last price. Of course not all the values have to exist in a single promotion. For instance, we can have a product with a price and without a promotion or a type of promotion and no promotion value ("Take 3 and pay 2"). For every field, we will calculate the values of all the metrics mentioned above, with exception of brand and description fields as explained below.

Let us explain these concepts with an example: price. True positives happen when the it detects a value, per example, 1 and the actual value for that promotion is also 1. True negatives occur when the model indicates there is no price and, in fact, there isn't one. When the model detects a price but there actually isn't one, we have a false positive. Finally, when we fail to find the price in a promotion, but it actually exists, we have a false negative.

Evaluation measures are then based on these values. Continuing with the example above, in this case, accuracy (the simplest measure) is the ratio of correctly predicted prices over the total predictions. Precision is the ratio of correctly predicted positive prices to the total predicted positive prices. It addresses the question: from all prices detected how many are actually prices.

Recall is the ratio of correctly predicted positive prices over the number of prices the are actually in the leaflet. This shows from all the prices present in the leaflet how many did we find. F1 is the weighted average of precision and recall.

The metrics have the following formulas:

$$Precision = \frac{True\_Positives}{True\_Positives + False\_Positives}$$

$$Recall = \frac{True\_Positives}{True\_Positives + False_Negatives}$$

$$Accuracy = \frac{True\_Positives + True\_Negatives}{True\_Positives + False\_Positives + False\_Negatives + True\_Negatives}$$

$$F1 = \frac{2 * Recall * Precision}{Recall + Precision}$$

The definitions show that the accuracy metric cannot be applied to the brand and description fields, as these fields do not have true negatives (i.e. the model didn't find a brand and it does not exist on the page). Given that our method always makes a guess, there is always a brand or a description associated with an example.

Measuring true positive is also a challenge. If the datahad information about the location of the promotions (e.g., promotion is contained in the box with coordinates x1, y1, x2, y2; the following promotion is present for the product X, with price Y), the evaluation could be more strict. Since we only know that for a certain page, we have X promotions and their field values, we have to be more flexible about the assessment of true positives. We used string distance to compare strings, such as the value of the brand field. So, if we find something like "NIVEF" and the actual value is "NIVEA", instead of discarding the result, the string distance calculation is calculated instead of a binary evaluation as correct or incorrect. In this case the result would be 0.8, since one character is not correct out of 5.

Since the companies presented in the data, have different types of design layouts, all the metrics are also calculated for each company, to understand if any method works better for a specific company/layout.

All the approaches explained in section 3.3, use threads in order to enhance the performance of the results.

4.1.1 Hardware

All the test done on this thesis were performed on the machine with the specs in Table 4.1.

Table 4.1: Macbook specs

Name	MacBook Pro 13-inch 2018
CPU	2,3 GHz Intel Core i5
RAM	8 GB 2133 MHz LPDDR3
GPU	Intel Iris Plus Graphics 655 1536 MB

4.2 Results

In this Section, the results of each implemented method will be analyzed and discussed.

4.2.1 Baseline

In this subsection we will analyze the results of the baseline method described in 3.3.1. Its important to state that no pre-processing methods were applied in the images (leaflet pages) used in this approach.

This method has no parameters so it just runs once. The method took 134 seconds to run, averaging 1,7 seconds for leaflet page. The results of this run can be seen on Table 4.2.

Table 4.2: Baseline - precision, recall, accuracy and f1

Fields	Precision	Recall	Accuracy	F1
Brand	0.091	0.223	None	0.116
Description	0.247	0.451	None	0.294
Type of Promotion	0.003	0.004	0.167	0.003
Promotion	0.002	0.004	0.177	0.003
Price	0.004	0.004	0.013	0.004
Last Price	0.0	0.0	0.171	0.0

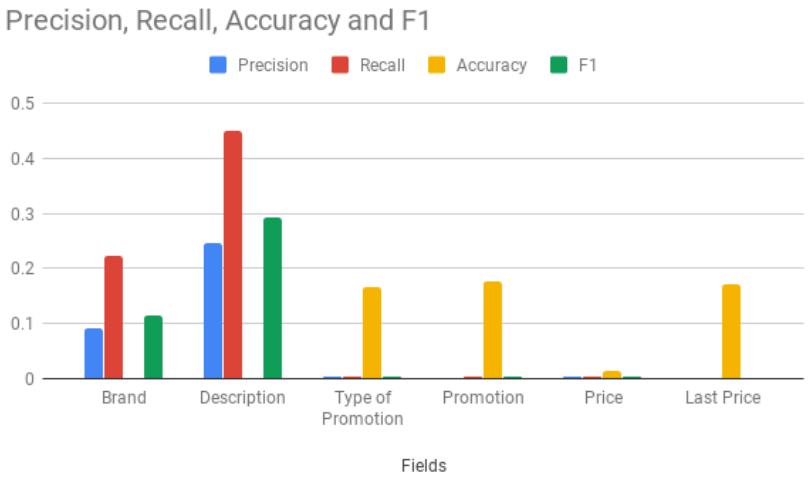

Figure 4.1: Baseline - Global

Type of promotion and last price have recall results close to zero. The accuracy for these fields is positive because of true negatives. Price is the field that shows the worst results. Other fields like brand, show that assuming that the position is an indication of field is not a very good method, since this does not happen all the time. At last the most generic field description, shows very reasonable results.

Jumbo had better results than the other brands using this approach since it's the company with the simplest design (less products per page, simple fonts, etc...). This comparison can be seen in Figure 4.2. It's also important to state, that zero precision is not always a bad sign. In the Jumbo leaflets, there was no type of promotions or promotions at all, just the price of the products.

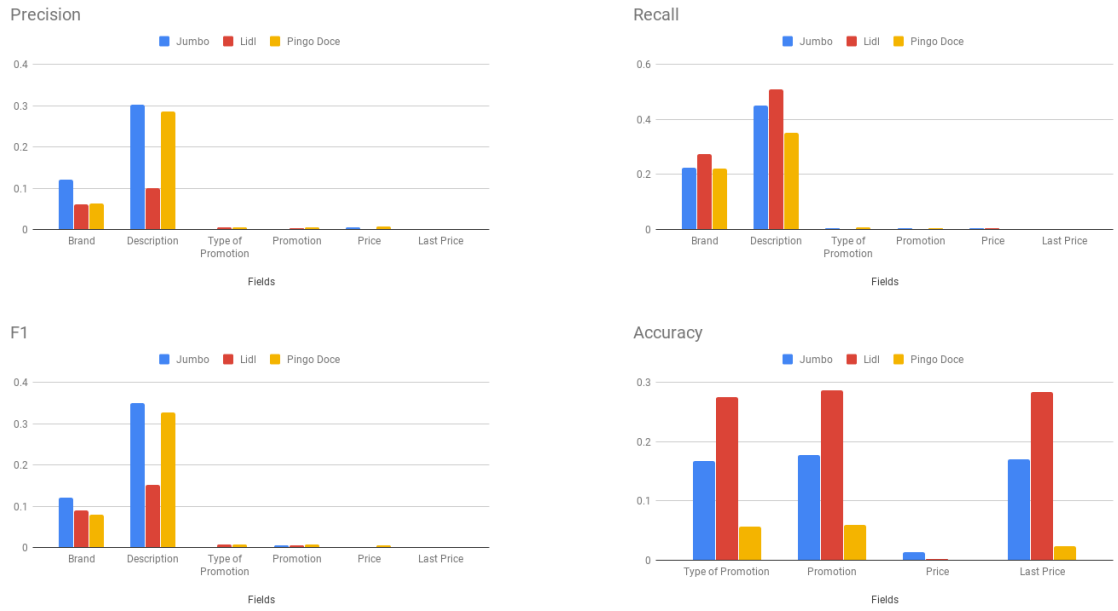

Figure 4.2: Baseline results - Companies analyze

4.2.2 Hough Line Transform

This subsection will be used to analyze the results of Hough Line Transform approach. As stated before, a Canny edge detector is used to detected edges in the original image and then a Hough Line transform to detect the lines.

In Section 3.3.2.1, a preliminary test was made to understand the effects of choosing the correct values for the Canny Edge detector, but here we perform a more extensive test. We carried out a systematic variation at the threshold values to confirm what was observed in Section 3.3.2.1. The values of `minLineLegth` and `maxLineGap` were locked at 125 and 20, respectively. More than 30 intervals we tested. Since f1, combines the precision and recall, we found the parameter that gave the best f1 results for each field in the product/promotion structure and also for accuracy. These results can be seen in Tables 4.3 and 4.4.

Table 4.3: Threshold systematic variation - best results precision, recall and f1

Fields	Precision	Recall	F1	minValue	maxValue	threshold
Brand	0.06	0.035	0.034	0	70	35
Description	0.193	0.112	0.114	60	90	75
Type of Promotion	0.008	0.006	0.006	0	90	45
Promotion	0.002	0.001	0.002	0	50	25
Price	0.018	0.009	0.009	60	90	75
Last Price	0.001	0.001	0.001	90	180	135

Table 4.4: Threshold systematic variation - best results accuracy

Fields	Accuracy	minValue	maxValue	threshold
Type of Promotion	0.052	0	70	35
Promotion	0.059	0	70	35
Price	0.018	90	180	135
Last Price	0.067	0	70	35

As predicted, lower intervals show better results than higher intervals.

After that, a systematic variation was done to understand the impact of minLength and maxLineGap parameters. Values for minValue, maxValue and treshold are fixed at 50, 100 and 75 respectively. More than 30 tests were done to understand their impact and the best results can be found in Table 4.5 and 4.6.

In order to extract the best results, test need to be performed using the best value of each systematic variation parameter. Result for a specific parameter can be found in Table 4.7.

Companies like Jumbo and Lidl, do not have their products inside boxes, making the results for this method not useful. In the case of Pingo Doce, for which leaflets have boxes surrounding the promotions, , results were superior from 3 to 8% when comparing with other companies.

Since this approach tries to cover a specific case of design layout, results are not better than the baseline method, since it's a more general method.

Table 4.5: Minimum Line Length & Maximum Line gap systematic variation - best results precision, recall and f1

Fields	Precision	Recall	F1	minLineLength	maxLineGap
Brand	0.076	0.061	0.051	140	35
Description	0.15	0.269	0.156	100	45
Type of Promotion	0.015	0.004	0.006	180	45
Promotion	0.006	0.002	0.003	100	35
Price	0.027	0.008	0.011	160	35
Last Price	0.001	0	0.001	180	15

Table 4.6: Minimum Line Length & Maximum Line gap systematic variation - best results accuracy

Fields	Accuracy	minLineLength	maxLineGap
Type of Promotion	0.085	120	45
Promotion	0.088	120	35
Price	0.02	120	35
Last Price	0.087	120	15

Table 4.7: Results for parameters - 60, 90, 75, 100 and 45

Fields	Precision	Recall	F1	Accuracy
Brand	0.037	0.106	0.045	None
Description	0.141	0.269	0.15	None
Type of Promotion	0.003	0.005	0.0078	0.004
Promotion	0.001	0.002	0.002	0.082
Price	0.007	0.008	0.014	0.007
Last Price	0	0	0	0.079

4.2.3 Scene text location

4.2.3.1 Systematic variation

Since this method has a lot of parameters, we wanted to do a systematic variation of each parameter to understand their effects on the approach developed. Our approach consisted in locking all parameter (using their default values) except one where the value is changed within a interval. More than 80 tests were perform but the results remained unchanged. For every field in product/promotions found, precision, recall, accuracy and f1 were 0.These results indicate that the default values are not suitable for our problem. They also indicate that parameters are not independent. That is, the performance of the method depends on setting multiple parameters simultaneously. Therefore, manual tuning of the parameters was carried out. The default custom test has the parameters represented in Table 4.8 and the results displayed in Table 4.9. he main observations we made can be summarised as:

- thresholdDelta: Results showed that increasing the number of threshold steps gets better results until a certain point. Up to 12 steps the computation time decreases significantly as

Table 4.8: Default custom test parameters

	FilterNM1						FilterNM2	Grouping	
	thresholdDelta	minArea	maxArea	minProbability	nonMaxSuppression	minProbabilityDiff	minProbability	orientation	minProbability
Value	16	5e-06	0.13	0.4	TRUE	0.1	0.3	ANY	0.5

Table 4.9: Results for parameters in Table 4.8

Fields	Precision	Recall	F1	Accuracy
Brand	0.027	0.275	0.046	None
Description	0.089	0.626	0.147	None

we can see in Figure 4.3 (a). The best F1 scores are also obtained for higher values of this parameter (Figure 4.3 (b)).

- maxArea: As observed in Section 2.4 characters have are proportionally small comparing with the leaflet page, so there is no need to check large areas. These results confirm this. Besides computational time that increases since we need to check more areas, F1 scores also decrease as we can see in Figure 4.4.

- minProbabilityNM1: Results indicate that relaxing the probability of a character being a real character can improve the F1 results slightly as seen in Figure 4.5.

- nonMaxSuppression: Results show that when this parameter is set to true, computational time decreases by 62% and f1 results improve from values between 3 to 8 % as we can see in Figure 4.6.

- minProbabilityDiff: As expected, low values reduce the F1 score. Up to 0.5 the results are the same, but the computational cost is better at 0.3 than 0.1 as we can see in Figure 4.7.

- minProbabilityNM2: The best results were with value of 0.5, reducing also the computational time as presented in Figure 4.5.

- orientation: As expected results are better when every orientation is checked rather than only looking for text. F1 results improve from 3 to 9%. Unexpectedly it was actually 43% faster to check all directions, as we can see in figure 4.9.

- minProbability: The results indicate that lower values for minProbGrouping perform better even if they are slower. This results can be seen in Figure 4.10.

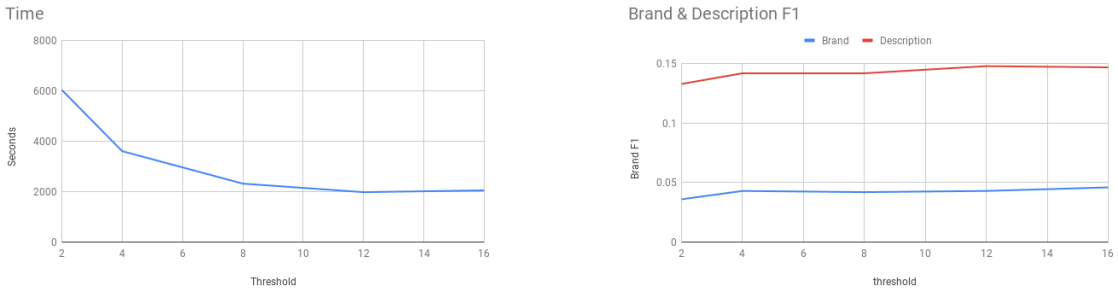

Figure 4.3: Threshold systematic variation results

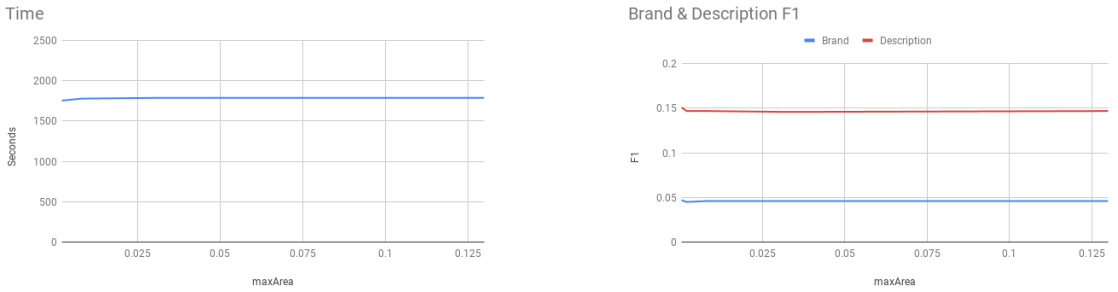

Figure 4.4: MaxArea systematic variation results

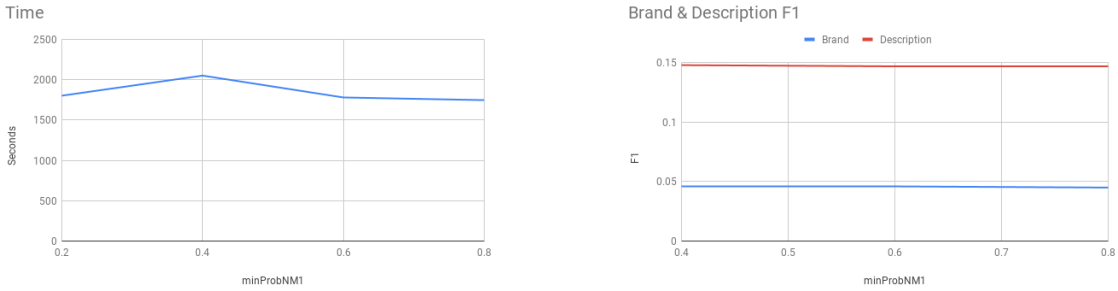

Figure 4.5: minProbNM1 systematic variation results

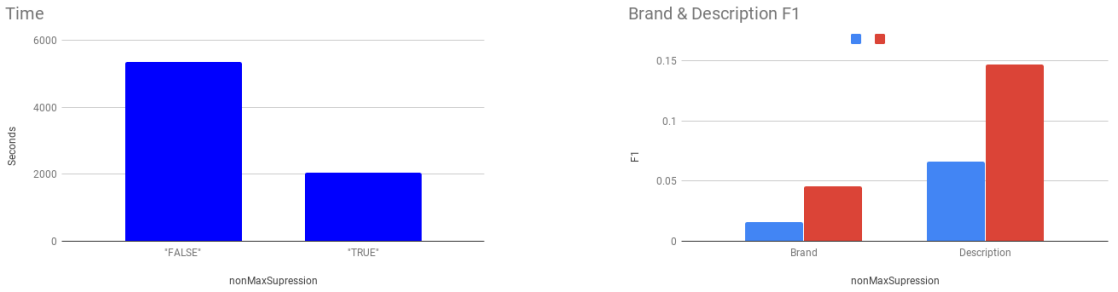

Figure 4.6: NonMaxSupression systematic variation results

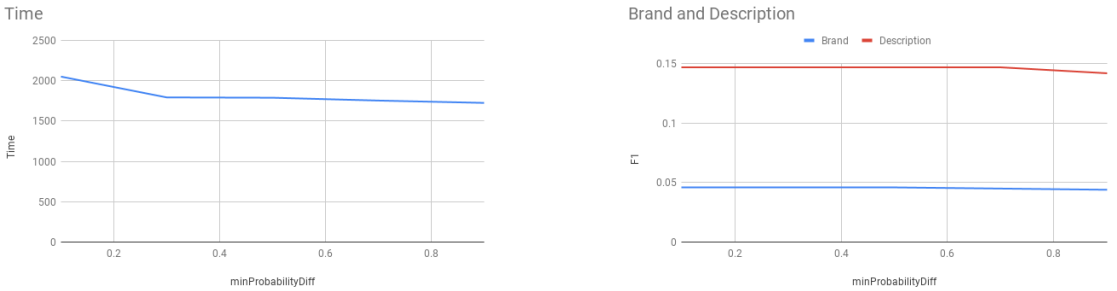

Figure 4.7: NonMaxSupression systematic variation results

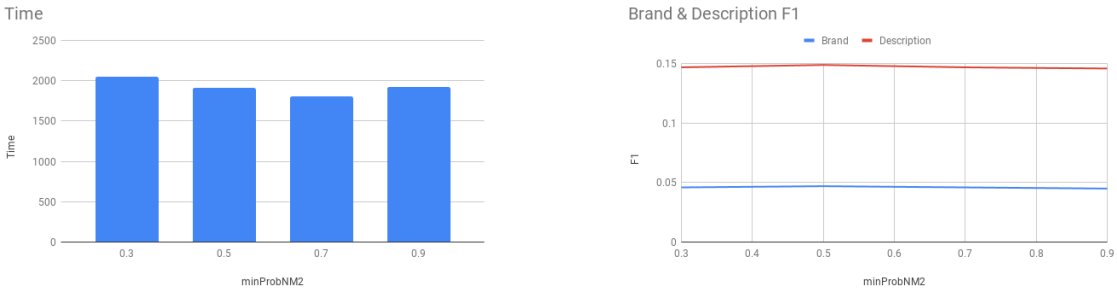

Figure 4.8: MinProbNM2 systematic variation results

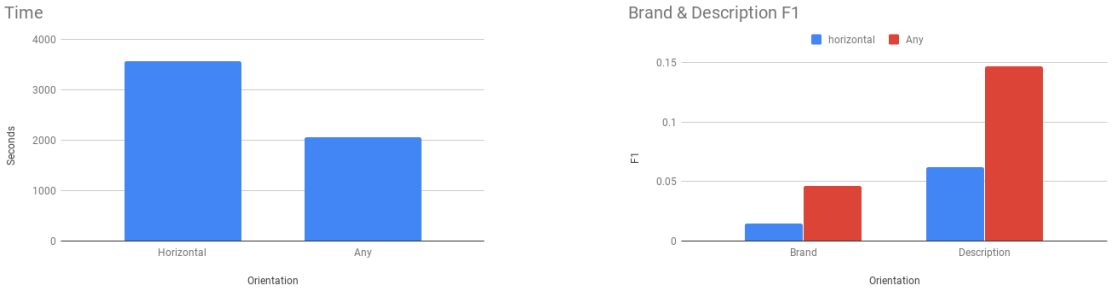

Figure 4.9: Orientation systematic variation results

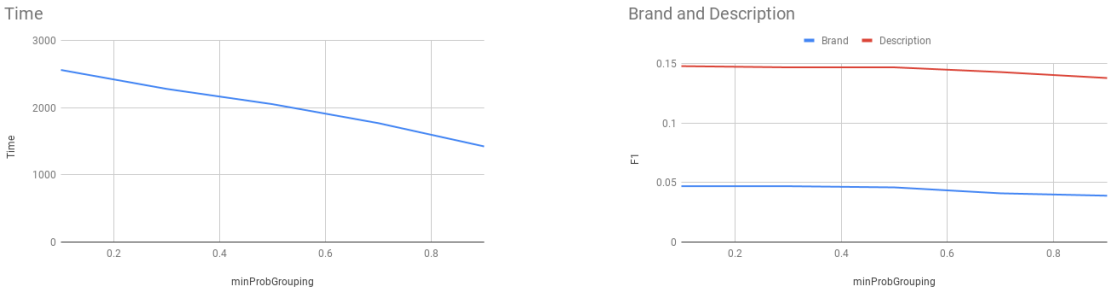

Figure 4.10: minProbGrouping systematic variation results

4.2.3.2 Best results

Using the parameters with the best results of each custom made systematic variation, presented in Table 4.10, we found the results displayed at Table 4.11. This increased the recall of the method

Table 4.10: Best parameters of systematic variation

	FilterNM1						FilterNM2	Grouping	
	thresholdDelta	minArea	maxArea	minProbability	nonMaxSuppression	minProbabilityDiff	minProbability	orientation	minProbability
Value	12	5e-06	0.00051	0.2	TRUE	0.3	0.5	ANY	0.1

from values between 10 and 15 percent, with an additional computational cost of 35% without important variations in other measures.

Table 4.11: Recall comparison results - Default and best

	Default	**Best**	**Difference**
Brand	0.0275	0.367	9.2%
Description	0.626	0.767	14.1%

4.2.4 Domain knowledge

Here, we analyze the impact of using a data base that contains all the brands and type of promotions. As before, string distance is used to match brands and descriptions extracted by our methods with information in the database.

First we discuss the impact of extending the baseline approach with this method. When creating the product/promotion found in the leaflet page, we check in the text found if any known brand name is present, if so we predict that that brand is related to the promotion that we are creating. The same thing happens for the type of promotion. If we find "Take 2 pay 1" inside the extracted text, we predict that is the type of promotion being applied. Results for brand field improved 5% for precision, recall and f1 as we can see in Figure 4.11. The type of promotion field didn't have significant improvements.

On method results described on Section 4.2.3 brand results didn't improve so well as results in baseline method. Precision, recall and F1 only increased 1, 3 and 2 percent respectively.

4.2.5 Haar Cascade Classifier & Scene text location

The last method we tested was object location, in order to perform segmentation of the leaflet pages. Having the location of the price and promotion tags, could have direct impact in the results, since now we would know that the text extracted from an OCR engine on that location would be directly linked with a price/promotion.

A preliminary test was performed, to understand if this tags could be easily located in a leaflet with a large amount of information. As described in Section 3.3.5, one price and one promotion tag were trained from a subset of the leaflets (Pingo Doce) in order to understand the power of this method.

Even though the experiments are very limited, results were very good. Price and promotion tags were being found with good precision and even better recall. Taking the example of the price tag, all the tags trained were being found, and almost all of the false positives were price tags with

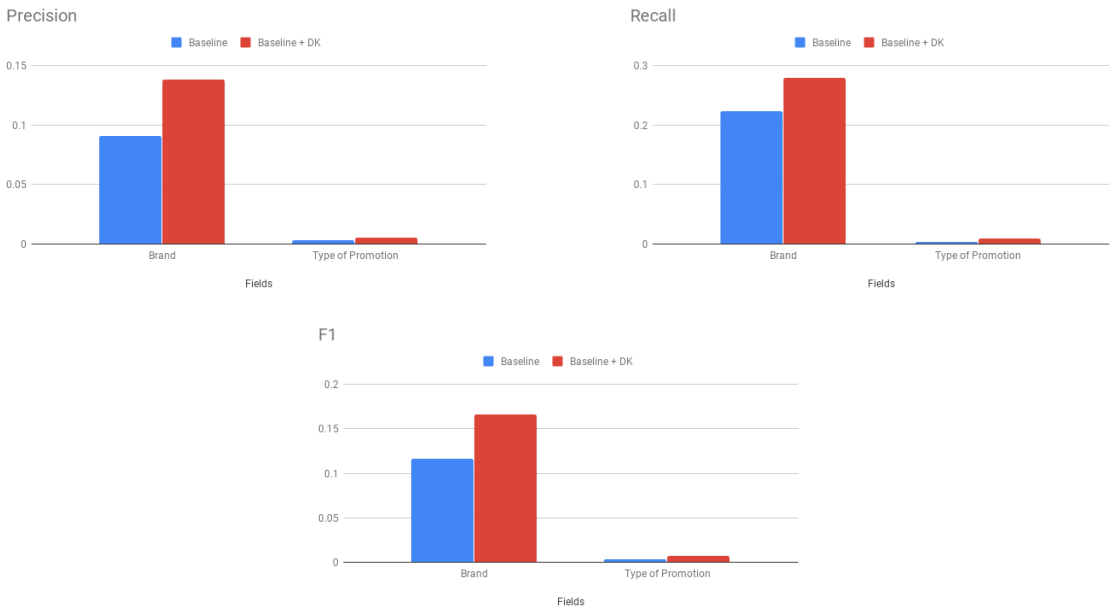

Figure 4.11: Baseline results for brand & type of promotion using domain knowledge

very similar design (in this case the price type 2 on Table 3.7) or the same tag but on a bigger scale.

Table 4.12: Haar Cascade training results

	Precision	**Recall**	**F1**	**Time Training**
Prices	0.800	0.915	0.853	4 days
Promotions	0.628	0.742	0.680	2 days

Chapter 5

Conclusions and Future work

In this chapter an overview of the project is done. The appreciation of the results are discussed in Section 5.1 as well as possible future work and improvements, in Section 5.2.

The primary objective of this M.Sc. Thesis was to investigate different methods to extract pricing information from leaflets to support the retail pricing strategy. These methods apply visual inspection techniques based on optical character recognition, image processing, text location/recognition, semantic segmentation and machine learning in leaflet images. For each leaflet, the method outputs a list of product-promotion pairs (types of promotions and respective promotional prices, if present).

5.1 Results

The Baseline method (Section 3.3.1), being the more general method, showed better results in more simpler designs. As predicted the Hough Line Transform (Section 3.3.2) method performed better in designs where product where inside boxes. Scene text location (Section 3.3.3) showed better results than the baseline method for fields like brand and description. We could also prove that the use of databases with knowledge (Section 3.3.4) about the domain can improve the results of predicting the brand of the product. Haar Cascade Classifiers (Section 3.3.5), showed that is possible to find tags of prices and promotions (previously trained) with great accuracy and precision.

One of the biggest difficulties found while developing the methods was evaluating the results and, more importantly the intermediate steps of each approach. For each of the approaches we are only evaluating the final result (the products/promotions found), but evaluation of the intermediate steps should be performed in the algorithms to ensure better results. Taking the example of the method developed in Section 3.3.2. In this approach, before constructing the pairs of product/promotions, lines are being found in order to construct the boxes and this step is not being evaluated. Having annotated, for each page, the location of this boxes, would probably lead to better results.

The same happens for the method developed in Section 3.3.3. The systematic variation performed only helps realize the impact of each parameter on the final result, but the actual location of the text in the pages is only being partially evaluated. First, a list of all the characters or words in the leaflet page should be created, in order to evaluate which values work better for detecting all the text. After having the parameters that work better, better extraction techniques should be used to understand what is useful to create the pairs.

In Chapter 4, we can see that the results, especially precision, are not the most accurate. This happens because a lot of false positives are being found, for example, the sentence on a leaflet "IN COMPANY X; LOWER PRICES ARE WORTH MORE" doesn't have any any useful information for what we are trying to extract, it's actually a slogan created to catch the consumers attention, but very hard to distinguish when extraction the information. This example would generate a false positive for each field (brand, description, type of promotion, promotion, price and last price) that product/promotion pair is composed of.

5.2 Future work

More data and with better quality is import for better assessment of the proposed methods as well as for further developments.

There are multiple strategies that could perform better on the problem presented in this thesis.

o improve the methods proposed, it is important to gain a better understanding of the behaviour of the sub steps of each method. Additionally, for the method developed in Section 3.3.5, results could be improved by using negative images of leaflets instead of random negative images.

Exploring more information extraction techniques should be done in order to improve precision results using deep learning.

A topic that would be interesting for this project is semantic segmentation. Having information about the type of product found is a plus for company strategy as inspected in the data set provided. Semantic segmentation could really help on this aspect, by recognizing the images of the products are labelling them.

Finally, to make the methods developed useful in practice, it is essential to develop a tool that provides an easy human interface to them.

References

[Bar17] Telmo Barbosa. Reversing shopview analysis for planogram creation, 2017. Available in https://repositorio-aberto.up.pt/handle/10216/106141/.

[Bha12] Mehta-K.A Bhammar, M.B. Survey of various image compression techniques. *International Journal on Darshan Institute of Engineering Research & Emerging Technologies*, pages 85–90, 2012.

[BKTT15] Sebastian Bittel, Vitali Kaiser, Marvin Teichmann, and Martin Thoma. Pixel-wise segmentation of street with neural networks. *CoRR*, abs/1511.00513, 2015.

[BM04] Siddiqui N. Q. Barlow, A. K. and M. Mannion. Developments in information and communication technologies for retail marketing channels. *International Journal of Retail & Distribution Management 32*, pages 157–163, March 2004.

[CMGS11] D. C. Ciresan, U. Meier, L. M. Gambardella, and J. Schmidhuber. Convolutional neural network committees for handwritten character classification. In *2011 International Conference on Document Analysis and Recognition*, pages 1135–1139, Sep. 2011.

[CY04] X. Chen and A. L. Yuille. Detecting and reading text in natural scenes. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 366–373, Los Alamitos, CA, USA, jul 2004. IEEE Computer Society.

[EAHAA+14] Moftah Elzobi, Ayoub Al-Hamadi, Zaher Al Aghbari, Laslo Dings, and Anwar Saeed. Gabor wavelet recognition approach for off-line handwritten arabic using explicit segmentation. *Image Processing and Communications Challenges 5*, 233, 01 2014.

[EOW10] B. Epshtein, E. Ofek, and Y. Wexler. Detecting text in natural scenes with stroke width transform. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2963–2970, June 2010.

[FAZAQ+16] A. Farhat, A. Al-Zawqari, A. Al-Qahtani, O. Hommos, F. Bensaali, A. Amira, and X. Zhai. Ocr based feature extraction and template matching algorithms for qatari number plate. In *2016 International Conference on Industrial Informatics and Computer Systems (CIICS)*, pages 1–5, March 2016.

[Fly19] Flyertown. Flyertown homepage, 2019. Available in https://www.flyertown.ca/.

[GBT14] A. K. Gaur, D. S. Bharangar, and M. C. Trivedi. A survey on ocr for overlapping and broken characters in document image: Problem with overlapping and broken characters in document image. In *2014 International Conference on Computational Intelligence and Communication Networks*, pages 138–141, Nov 2014.

[GNP17] Hashem Ghaleb, P Nagabhushan, and Umapada Pal. Segmentation of offline handwritten arabic text. pages 41–45, 04 2017.

[Goo] Google. Tesseract repository. Available in https://github.com/tesseract-ocr/tesseract/.

[Goo01] Joshua T. Goodman. A bit of progress in language modeling. *Computer Speech Language*, 15(4):403 – 434, 2001.

[HK15] Tasnuva Hassan and Haider Adnan Khan. Handwritten bangla numeral recognition using local binary pattern. 05 2015.

[HZRS16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[IIN16] Noman Islam, Zeeshan Islam, and Nazia Noor. A survey on optical character recognition system. *ITB Journal of Information and Communication Technology*, 12 2016.

[KJB16] Andreas Krämer, Martin Jung, and Thomas Burgartz. A small step from price competition to price war: Understanding causes, effects and possible countermeasures. *International Business Research*, 9:1–13, 01 2016.

[LBDG13] Markus Löchtefeld, Matthias Böhmer, Florian Daiber, and Sven Gehring. Augmented reality-based advertising strategies for paper leaflets. In *Proceedings of the 2013 ACM Conference on Pervasive and Ubiquitous Computing Adjunct Publication*, UbiComp '13 Adjunct, pages 1015–1022, New York, NY, USA, 2013. ACM.

[LLL+11] J. Lee, P. Lee, S. Lee, A. Yuille, and C. Koch. Adaboost for text detection in natural scene. In *2011 International Conference on Document Analysis and Recognition*, pages 429–434, Sep. 2011.

[LNW15] C. Liyanage, T. Nadungodage, and R. Weerasinghe. Developing a commercial grade tamil ocr for recognizing font and size independent text. In *2015 Fifteenth International Conference on Advances in ICT for Emerging Regions (ICTer)*, pages 130–134, Aug 2015.

[LW02] R. Lienhart and A. Wernicke. Localizing and segmenting text in images and videos. *IEEE Transactions on Circuits and Systems for Video Technology*, 12(4):256–268, April 2002.

[LZ16] Lianzhi Huo Lijun Zhao, Ping Tang. Feature significance-based multibag-of-visual-words model for remote sensing image scene classification. *Journal of Applied Remote Sensing*, 10:10 – 10 – 21, 2016.

[LZH+06] Chong Long, Xiaoyan Zhu, Kaizhu Huang, Jun Sun, Yoshinobu Hotta, and Satoshi Naoi. An efficient post-processing approach for off-line handwritten chinese address recognition. 02 2006.

[MB09] Millward-Brown. Using neuroscience to understand the role of direct mail. 2009.

[MC15] Priyanka Mukhopadhyay and Bidyut B. Chaudhuri. A survey of hough transform. *Pattern Recognition*, 48(3):993 – 1010, 2015.

[NM11a] L. Neumann and J. Matas. Text localization in real-world images using efficiently pruned exhaustive search. In *2011 International Conference on Document Analysis and Recognition*, pages 687–691, Sep. 2011.

[NM11b] Lukas Neumann and Jiri Matas. A method for text localization and recognition in real-world images. In Ron Kimmel, Reinhard Klette, and Akihiro Sugimoto, editors, *Computer Vision – ACCV 2010*, pages 2067–20178, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[NM12] L. Neumann and J. Matas. Real-time scene text localization and recognition. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3538–3545, June 2012.

[Opea] OpenCV. Canny edge detection. Available in https://docs.opencv.org/3.1.0/da/d22/tutorial_py_canny.html.

[Opeb] OpenCV. Face detection using haar cascades. Available in https://docs.opencv.org/3.4.1/d7/d8b/tutorial_py_face_detection.html.

[Opec] OpenCV. Hough line transform. Available in https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_houghlines/py_houghlines.html.

[Oped] OpenCV. Scene text detection. Available in https://docs.opencv.org/3.0-beta/modules/text/doc/erfilter.html.

[PHL09] Y. Pan, X. Hou, and C. Liu. Text localization in natural scene images based on conditional random field. In *2009 10th International Conference on Document Analysis and Recognition*, pages 6–10, July 2009.

[QA09] M. T. Qadri and M. Asif. Automatic number plate recognition system for vehicle identification using optical character recognition. In *2009 International Conference on Education Technology and Computer*, pages 335–338, April 2009.

[RSa] Hewlett-Packard Ray Smith. Tesseract (software). Available in https://en.wikipedia.org/wiki/Tesseract_(software).

[RSb] Hewlett-Packard Ray Smith. Tesseract (software). Available in https://hub.docker.com/r/bmakowe/opencv-haar-training/.

[RSc] Hewlett-Packard Ray Smith. Tesseract (software). Available in https://memememememememe.me/post/training-haar-cascades/.

[San12] Ana Sanlez. Pingo doce sob investigação da asae um dia depois do caos das promoções, 2012. Available in PingoDocesobinvestigaç~aodaASAEumdiadepoisdocaosdaspromoç~oes.

[Sat12] D.A Satti. Offline urdu nastaliq ocr for printed text using analytical approach. MS thesis report Quaid-i-Azam University: Islamabad, Pakistan, 2012.

[Ser15] Ana Serafim. Guerra de preços nos supermercados veio para ficar, 2015. Available in https://sol.sapo.pt/artigo/126664/guerra-de-precos-nos-supermercados-veio-para-fica.

[Sim18] Osvaldo Simeone. A very brief introduction to machine learning with applications to communication systems. *CoRR*, abs/1808.02342, 2018.

[SKC+17] J. S. Sevak, A. D. Kapadia, J. B. Chavda, A. Shah, and M. Rahevar. Survey on semantic image segmentation techniques. In *2017 International Conference on Intelligent Sustainable Systems (ICISS)*, pages 306–313, Dec 2017.

[Smi07] R. Smith. An overview of the tesseract ocr engine. In *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, volume 2, pages 629–633, Sep. 2007.

[VJ01] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, volume 1, pages I–I, Dec 2001.

[YBCK10] Z. Yin, R. Bise, M. Chen, and T. Kanade. Cell segmentation in microscopy imagery using a bag of local bayesian classifiers. In *2010 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pages 125–128, April 2010.

[ZK11] Jing Zhang and Rangachar Kasturi. Character energy and link energy-based text extraction in scene images. In Ron Kimmel, Reinhard Klette, and Akihiro Sugimoto, editors, *Computer Vision – ACCV 2010*, pages 832—-844, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[ZZ05] Li Zhuang and Xiaoyan Zhu. An ocr post-processing approach based on multi-knowledge. In Rajiv Khosla, Robert J. Howlett, and Lakhmi C. Jain, editors, *Knowledge-Based Intelligent Information and Engineering Systems*, pages 346–352, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

Appendix A

Methodology

A.1 Scene text location

Table A.1: OpenCV - Approach methods, parameters and their default values

Methods	Parameter	Description	Default
createERFilterNM1	thresholdDelta	Threshold step in subsequent thresholds when extracting the component tree	1
	minArea	The minimum area (% of image size) allowed for retrieved ER's	0.00025
	maxArea	The maximum area (% of image size) allowed for retrieved ER's	0.13
	minProbability	The minimum probability P(er|character) allowed for retrieved ER's	0.4
	nonMaxSuppression	Whenever non-maximum suppression is done over the branch probabilities	true
	minProbabilityDiff	The minimum probability difference between local maxima and local minima ERs	0.1
createERFilterNM2	minProbability	The minimum probability P(er|character) allowed for retrieved ER's	0.3
erGrouping	orientation	ERGROUPING_ORIENTATION_HORIZ or ERGROUPING_ORIENTATION_ANY	ERGROUPING_ ORIENTATION _HORIZ
	minProbablity	The minimum probability for accepting a group. Only to use when grouping method is ERGROUPING_ORIENTATION_ANY.	0.5