FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Retail Product Matching

Francisco Teixeira Ferreira

Mestrado Integrado em Engenharia Informática e Computação

Supervisor: Vera Miguéis

Company Supervisor: Paulo Maurício

July 26, 2021

Retail Product Matching

Francisco Teixeira Ferreira

Mestrado Integrado em Engenharia Informática e Computação

July 26, 2021

Abstract

To properly define products' prices and their market positioning, retailers have always attempted to identify the products provided by the competition. With the growing evolution of technology, some price comparison platforms began to appear. However, one big challenge these platforms face is to correctly match similar products sold by different sources. Most retail product matching state-of-art attempts struggle with limitations regarding the image component of the products. Images corresponding to the same product can differ in terms of version, illumination, angles, or even colours. Additionally, most state-of-art studies' learning models face problems due to the large amount of product categories retailers sell.

In this dissertation, a proposal of a product matching solution based on retail product images is provided. In other words, the thesis aims at developing a solution capable of determining the corresponding market product given an input image. To correctly develop this solution, a dataset with images from Portuguese retail e-commerce platforms was created.

Due to the peculiarities of retail product matching, two approaches were proposed and their results were compared: the first one combines traditional feature extractor methods, like Scale-Invariant Feature Transform, with FLANN and brute-force feature matchers; and the second approach consists of using two convolutional neural networks, with the help of a string filter. Both approaches were developed using Python. The approaches had their parameters tuned utilizing a grid search methodology and several metrics were utilized to measure their performance.

The experiments conducted revealed that the approach that utilized combinations between classic feature extractors and matchers obtained the best results, both in terms of accuracy and execution times. Additionally, this study showed that some specific retail categories compose a bigger challenge regarding image matching due to its peculiar characteristics, namely *Bebidas* and *Frescos*.

Keywords: Retail, Image Matching, Image Features, Convolutional Neural Networks

Resumo

Desde sempre, o setor do retalho tentou definir o preço dos seus produtos e estabelecer a sua posição no mercado, analizando os preços praticados pela competição. Com a crescente evolução da tecnologia, surgiu o conceito de plataformas de comparação de preço. No entanto, um dos maiores desafios enfrentados por estas plataformas é a correta identificação de que artigos vendidos por diferentes retalhistas correspondem de facto ao mesmo produto. Na tentativa de estabelecer correspondências entre produtos, estudos semelhantes encontraram dificuldades relativamente à componente visual dos artigos. Isto porque imagens correspondentes ao mesmo produto podem divergir em termos de versão, iluminação, ângulos, ou mesmo cores. Em adição, a maioria dos modelos de aprendizagem propostos para estabelecer correspondências entre produtos encontraram limitações devidas à enorme quantidade de classes de produtos.

No âmbito desta dissertação, é fornecida uma proposta de correspondência de produtos, recorrendo às suas imagens. Por outras palavras, a solução proposta é capaz de determinar o produto de retalho correspondente, recebendo apenas uma imagem como entrada. Foi utilizada uma fonte de dados criada a partir de imagens de produtos provenientes de plataformas digitais de retalho português. Devido às peculiaridades da correspondência de produtos do retalho, duas abordagens foram propostas: a primeira combinando métodos tradicionais de extração de características únicas de cada imagem , como o algoritmo SIFT, com métodos de correspondência entre as características extraidas, nomeadamente as abordagens FLANN e brute-force; a segunda abordagem proposta consiste no uso de duas redes neuronais convolucionais, com o auxilío de um filtro de texto. Ambas as abordagens foram desenvolvidas em Python. Os parâmetros de cada abordagem foram afinados recorrendo a uma metodologia de grid search e múltiplas métricas foram utilizadas para avaliar os seus desempenhos.

Concluindo, os testes feitos revelaram que a abordagem que recorreu a combinações entre métodos clássicos de extração de características únicas com métodos de correspondência obteve resultados superiores, tanto em termos de exatidão como de tempos de execução. Adicionalmente, este estudo revelou que algumas categorias de retalho apresentam maiores desafios à correspondência de imagens, devido à peculiaridade das suas características, nomeadamente as categorias *Bebidas* e *Frescos*.

Keywords: Retalho, Correspondência de Imagens, Redes Neuronais Convolucionais

Acknowledgements

To my advisors, Professor Vera Miguéis, for the continuous guidance and dedication, and Paulo Maurício, for all the knowledge shared upon the development of the project.

To my colleagues at Deloitte, in special to Francisca Cerquinho, for the tremendous amount of support provided.

To my mother, for the kindness, to my father, for the strength, and to my brother, for the inspiration.

To the rest of my family, in particular Tala and Té, for the constant assistance.

To all my friends, specially to the ones that make me laugh on a daily basis, for the authenticity.

Finally, to the four beautiful stars I have in the sky, for the love.

Francisco Ferreira

"He who says he can,
and he who says he can't,
are both usually right."

Henry Ford

Contents

List of Figures

List of Tables

Abbreviations

API	*Application Programming Interface*
BF	*Brute-Force Feature Matcher*
CNN	*Convolutional Neural Network*
DoG	*Difference-of-Gaussian*
EAN	*European Article Number*
FLANN	*Fast Library for Approximate Nearest Neighbors*
MSE	*Mean Square Error*
ORB	*Oriented FAST and Rotated BRIEF*
ReLu	*Rectified Linear Unit*
SAAS	*Software as a Service*
SIFT	*Scale-Invariant Feature Transform*
SSIM	*Structural Similarity Index*
SURF	*Speeded-Up Robust Features*
WWW	*World Wide Web*

Chapter 1

Introduction

This chapter contains an introduction to this dissertation, regarding its context, motivation, objectives and scope. Section 1.1 refers to the problem and the environment in which the solution is going to be developed. Section 1.2 mentions the stimulations behind this thesis. Section 1.3 regards the main goals this study is meant to achieve. Finally Section 1.4 defines the structure of the rest of this report.

1.1 Context

Retailing is a socio-economic system consisting of all the activities that include goods sold to people in exchange for some consideration in return, with the ultimate purpose of consumption. Hence, retail is an enormous industry with a very diverse range of products and stores, ranging from small groceries to gigantic supermarket chains and shopping malls. One particular activity that possesses a substantial percentage in the retail world is food retailing. It is a continuously evolving subsection due to the economy, consumer trends, competition amongst the players in the same market, and technology development [25].

By the year 1980, Portugal's food market consisted only of small groceries working as independent atoms. Still, five years later, it all changed due to the first supermarket's appearance, changing the entire food distribution. Successively, new supermarkets began to appear. Nowadays, Portugal has over 1800 structures of this type, consisting of a ratio of supermarket per person only comparable to the most advanced countries in the world. The introduction of supermarkets impacted the demand of the consumers as they could find a wide variety of products in the same place. In addition to the expansion of retail companies, larger quantities of products were sold, eventually leading to lower prices due to the supply chains' propitious conditions.

According to data provided by Sonae MC in 2018, this player leads the food retail market with market shares of 21.9% being closely followed by Jerónimo Martins SGPS.

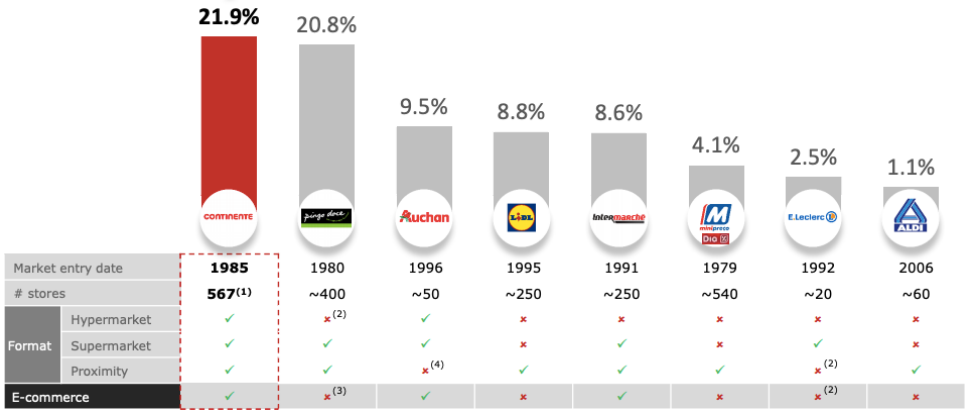

Figure 1.1: 2018 Portuguese Market Share

Source: Sonae Planet Retail RNG as of June 2018

Various dimensions influence the market players' success, one of the biggest being the price. Although it is easy to understand the impact product pricing has, it is challenging to determine the ideal cost of a product. It varies with many factors such as location, period, consumption trends, economy, promotional offers, and many more. Additionally, prices also vary between stores from the same player as they can have different characteristics. Hypermarkets such as Continente from Sonae MC characterize themselves by being enormous physical structures with an immense diversity of products. On the other hand, supermarkets like Continente Bom Dia or Pingo Doce from Jerónimo Martins SGPS are sightly smaller structures with less variety. Still, they are located in strategic places that provide convenience to the consumer. Moreover, the consumption frequency is higher than hypermarkets, which incentivizes promotional offers. An example of a promotional offer is the usage of loyalty cards that benefit consumers by visiting the stores multiple times in the same week or month.

In the last years, another format has been rising, namely, e-commerce, also known as online markets. In this distribution channel, the goods are presented in virtual stores, and the customers have a shopping cart that they can checkout after choosing all the desired products and then proceeding to the payment. Online payment is a subsection of e-commerce that also had a significant evolution in the last years, and now the user has a wide range of methods to conclude the transaction. To summarize, this process has three major phases that affect customer experience, the first being related to the ease of navigation and platform design, the second regarding the transaction, where additional factors to the price need to be considered, such as safety. The last phase is the delivery, where product conditions and processing times are critical. Combining these three steps is a goal of the online market players as it leads to greater client satisfaction [15].

Besides improving the whole e-commerce process, companies invest their resources in technology because, in a competitive business such as retail with constant data volume increments, it became necessary to include advanced data analytics in their procedures to save time and gain essential knowledge about the market. Advanced analytics is the autonomous inspection of data

or content using sophisticated methods and tools to discover deeper insights, make predictions, or generate recommendations to optimize current structures. In the retail context, the usage of advanced analytics can help the identification of growth opportunities, the development of new negotiation procedures and the improvement in the relationships with clients, which then translates in higher sales and profits [11].

In a new era of retail, both clients and sellers have a deeper awareness of the whole market, due to the rise of technology. With computers tablets, smart phones, the Internet, and social media, it is very easy for anyone to gather all types of information towards a specific product. This makes it harder for retail companies to differentiate themselves from the competition with their traditional approaches, such price variations. On the other hand, with the growing usage of these devices, it also became easier to understand how customers think and act, based on the digital breadcrumbs they leave behind. The ability to get more accurate insights about both customers and the market, comparatively to the competition, allows retailers to elaborate faster and better strategies to attract more and more clients. With this being said, the usage of analytics in retail is mandatory for companies that want to thrive and distanciate themselves from the competition [11].

1.2 Motivation

The increasing development in technologies made the retail market extremely competitive. The players need to continually upgrade their procedures in terms of operations, revenues, market shares, and better user experience. This intense contest obligates the players to find new and better strategies to increase their results, even with limited costs.

One particular concept that started to gather interest from both retailers and customers are product comparison platforms. In sum, these platforms extract information about retail products from multiple sources and display comparisons between similar products available in different stores, as well as their respective prices. There are currently options in the product comparison market that will be explained in more detail later in this report. Still, despite providing some results for comparing similar products between different stores and different brands, the platform users often face problems such as the inability to adapt. The maintenance of these platforms is also exceptionally high due to the ever-changing retail market. Another problem with the usage of these engines by the retail world players is that it is not possible to come up with a pricing strategy because of the lack of rigor of the data, as the slightest comparison mistake or data insufficiency may compromise the entire goal of the operation.

A platform capable of successfully establishing correspondeces between similar products sold by different retailers, and provide price comparisons between them, would be an extremely innovative addition to the Portuguese retail market. For its proper development, two main components have huge influence in the product: data extraction and product matching. As mentioned before, the first step is crucial because if the data has some kind of incongruity, it translates in the platform providing wrong insights and prejudicing its users. The second step, product matching, is the core

of the whole platform, as the ability of the platform to organize the products accordingly to their similarities constitutes its main purpose.

However, product matching is a very challenging task that can be divided into two main components: string matching and image matching. Due to their complexity, the aim of this study is to focus on the image matching component of the problem and to develop multiple approaches to match retail products based on their images. This part of the matching process will then be integrated in a major product comparison platform, that can provide insights about the Portuguese retail market. It will lead to more efficient decision-making in sales, marketing, user experience, and operations.

The project's development will be undertaken at Deloitte Consultores S.A..

1.3 Objectives

Matching producs by their images is an extremely complex task, due to some characteristics of the retail area. Firstly, there is a huge amount of data, as a single category of an e-commerce platform can have thousands of products. Additionally, some different products can have minor image differences. For instance, two yogurts from the same brand with a different flavour have very similar images, with only few colour variations in certain parts. These difficulties, allied with others that will be mentioned in the next sections of this paper, make it very hard for the development of a robust approach to match retail products based on their images.

With the development of this project, it is intended to provide a solution to this problem by creating a method that can accurately match identical retail products by analyzing their images. Additionally, it is expected to integrate that solution in a platform capable of extracting data from e-commerce platforms and match those extracted products, followed by a visualization interface to analyze the Portuguese retail market.

To summarize, this thesis is expected to develop a solution capable of:

- Classifying Portuguese retail product images, attributing each image a product class;
- Integrating the product matching component of a Portuguese retail product comparison platform.

The developed solution is expected to be capable of adapting to product additions or removals in a market context. Moreover , the model is also expected to be robust independently of the quantity of products, and every performance test will be evaluated with real Portuguese retail images. The ultimate goal of the solution is to achieve high accuracy upon the match of images into products, knowing that this is a difficult task, even for the human eye.

1.4 Document Structure

This thesis is divided into five chapters. Chapter 1 provides a brief introduction to the project, whereas Chapter 2 describes the State of the Art regarding approaches that were proposed to tackle related problems. In Chapter 3, the problem is explained in more detail and a solution to circumvent it is proposed. Chapter 4 explains the strategies that were used to validate the work developed and the results obtained. Finally, Chapter 5 reflects the achievements of the project, its goals and future improvements.

Chapter 2

State of Art

This chapter is divided into two main parts. Section 2.1 provides an explanation of the concept of image matching. Section 2.1.1. explains image matching traditional approaches as well as analyzing its implementations in state of art projects. Section 2.1.2. mentions the other type of image matching approaches, which are convolutional neural networks. Section 2.1.3. refers to some evaluation metrics available to evaluate the performance of the algorithms. Finally, Section 2.2. describes Retail Product Matching Platforms and its current state of art.

2.1 Image Matching

Computer vision is a scientific field with the primary goal of developing computational models of the human visual system to provide these machines perception capabilities to sense environments or learn from experiences and improve future performance. In other words, from an engineering point of view, computer vision ultimate goal is to study how computers can obtain a more profound understanding from images and videos to achieve tasks that the human visual system can or even to surpass them in some cases [19].

This area was born in 1960 when Larry Roberts, a thesis student at MIT, discussed the extraction of 3D geometrical information from 2D perspective views of blocks [37]. Later, researchers realized the potential of tackling real-world images, and the first computer vision tasks and algorithms began to appear, more precisely related to edge detection and segmentation. The evolution of this field of informatics is also due to merging with other related areas, such as image processing, camera calibration, and computer graphics.

Recently, there has been a significant growth regarding the demand for computer vision systems to perform tasks. However, most of the existing models have their failures, and in most cases, there is no space for mistakes. Therefore, the current state of art computer vision is far from being able to perfectly deal with real-world problems such as navigation, manufacturing, or even face recognition when compared to the power of the human visual system. On the other hand, this

immense growth in demand also translates into a significant evolution in algorithms and solutions [41].

A particular relevant field of computer vision relevant for this thesis is image matching, which in this specific case is related to product matching. Image matching is already present in multiple real-world applications such as object recognition. It has the goal of establishing correspondences between two images of the same scene. One of the simplest categories of image matching methods is gray-based matching, which is based on the grayscale feature of pixels. Together with template matching, this approach can provide an easy and fast to execute solution, comparatively to more advanced matching approaches [44]. Template Matching returns a grayscale image, where each pixel represents how much the pixel's region matches with the template. Its usage to match images is ideal if the template of an image is exactly within the other. However, this method presents flaws matching an image with other where the contents of one of them are rotated as the pattern is not the same anymore. Template matching techniques are achieve better results with intermediate complexity, although with an increasing number of templates, the efficiency of the algorithm is negatively affected [47].

On the other hand, a more complex approach consists of detecting a set of interest points, each associated with image descriptors from image data. Once the features and their descriptors are extracted from multiple images, the next task is to establish preliminary feature matches between these images. This approach and a few others are explained in the following sections. Feature matching approaches are a better fit to more complex problems as they are able to perform a more extensive range of matches, due to its robustness to rotations or translations. These approaches can be divided in two main categories: classical feature matching methods, and convolutional neural networks.

Classical feature matching methods establish correspondences between extracted features of two images and attribute a matching score to the pair of images. This approach can be adapted to solve classification problems by iterating through multiple images and converting the multiple scores into a final prediction. Oppositely, convolutional neural networks are pure classifiers, whose function is to predict a class, given an image. In the next sections, both approaches will be explained in greater detail and an analysis to their implementations towards retail product matching will also be provided.

2.1.1 Traditional Feature Matching

A feature is a chunk of information pertinent to solving the computational task related to a particular application. Features can be specific structures in the image, such as points, edges, or objects. Features may also be the result of a general neighborhood operation or feature detection applied to the image. Features can be clustered in two types: the ones that are in specific locations of an image, such as the corners of a building, called key-point features (or corners), and the features that can be matched based on their orientation and local appearance, called edges, which indicate object limits and occlusion events in the image sequence [44].

The traditional approach of feature matching can be divided in two major components: feature extraction and feature matching. Section 2.1.1.1 provides a deeper look towards the core concepts of both feature extraction and matching, and an explanation of some of the algorithms that were developed to perform those tasks. Section 2.1.1.2 explains some approaches that utilized the previously explained algorithms in a retail product matching environment.

2.1.1.1 Core Concepts

The growth of computer vision allowed the appearance of algorithms capable of extracting relevant parts of images. But to extract these distinct features of an image, some steps may be necessary previously to maximize the accuracy of the extraction, such as the image preprocessing where noise and redundant data remotion helps improve the quality of the image for the subsequent tasks. Image preprocessing mainly includes segmentation, transformation, and enhancement [49].

In order to achieve success in the image match, the most critical step is to maximize the accuracy of the feature extraction, as the match is dependent on the features extracted. A *Feature Extraction* algorithm takes an image and returns feature vectors, also called descriptors. Feature descriptors encode crucial information into a series of numbers and act as an unique identifier that is then utilized to differentiate one feature from another. As explained in the previous chapter, features are unique characteristics of an image. Feature analysis is crucial to tell whether two images are the same, not only in computer vision, but humans do it subconsciously as well.

One of the first feature extraction algorithms was developed by Chris Harris, and Mike Stephens with the name of Harris Corner Detection in 1998 [16]. The algorithm bases itself on a mathematical formula. It finds the difference in intensity for a displacement of vectors in all directions. Despite being revolutionary, this method's performance depended on the scale features of the images to remain constant, which led to the appearance of better algorithms, such as Scale-Invariant Feature Transform, SIFT [46].

In 1990, David Lowed proposed the Scale Invariant Feature Transform algorithm [31]. SIFT features have several benefits as they provide rotation and translation invariance as well as infinite scaling [49]. This algorithm has four significant steps: Scale-space extrema detection, key-point localization, orientation assignment, and key-point description. Scale-space extrema detection consists of the scan of all the image locations, using a Difference-of-Gaussian function. DoG aim is to identify interest point prospects [4]. DoG is accomplished by the convolution of the image

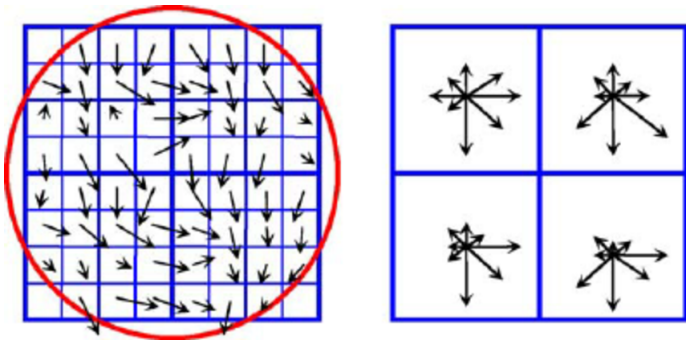

Figure 2.1: SIFT Feature Descriptor

Source: 3D CAD model matching from 2D local invariant features [51]

applying Gaussian filters at different scales. Then, the difference between the several blurred images is performed, extracting the key-points as the minimum of the difference between the values in these Gaussian scales. The next step in the SIFT algorithm utilization is the key-point location. The key-point location and scale information is computed by applying a quadratic Taylor expansion of the Difference-of-Gaussian function employed in the antecedent step. Following this task and before the key-point description is performed, orientations are allocated to every key-point location based on the operations that were applied in the image data [4]. The last phase of the SIFT algorithm consists of the computation of the key-point descriptors by measuring local image gradients at the selected scale in the proximal regions around each key-point. These gradients are then converted into a depiction that allows shape deformations and illumination changes.

In 2006, scientists developed a new method called Speeded Up Robust Features , SURF, [7], which was based on the SIFT characteristics but with the goal of reducing the features calculation times comparatively to the previously mentioned algorithm [49]. SURF was based on the multi-scale space theory [43], which makes the algorithm robust to different scales and illuminations. As in SIFT, SURF can also be divided into four stages that perform the key-point detection and description: the integral image generation, the key-point detection, the descriptor orientation assignment, and the descriptor generation. The integral image generation is the representation of the initial image as the sum of the gray pixel values. This step is vital to guarantee a fast computation speed. Then, in order to detect the key-points, Hessian matrices are employed to determine the location of the interest points. Similarly to the SIFT algorithm, key-point detection is also performed recurring to image convolutions with the difference that the Laplacian of Gaussian is applied instead of the DoG. After having the location of the interest points, to guarantee that the algorithm is robust to rotations, an orientation assignment step is crucial [4]. Despite being a faster version of the SIFT algorithm and with good results in most of the cases, the SURF method calculation times were still not enough to satisfy some feature detection problem requirements. In 2006, Edward Rosten and Tom Drummond proposed a new algorithm, called "Features from Accelerated Segment Test", abbreviated to FAST, [38]. This algorithm starts by selecting pixels and by evaluating

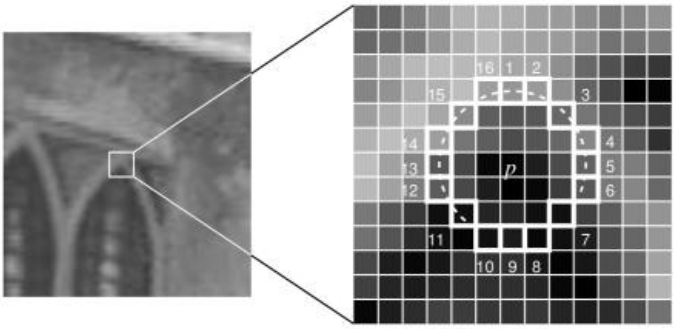

Figure 2.2: FAST - Evaluating if pixel p is a corner

Source: OpenCV - FAST Algorithm for Corner Detection [45]

if those pixels can be identified as interest points. To test whether it can be an interest point or not it starts by selecting a circle with the 16 pixels around the one that is being tested, as shown in figure 2.2. It then determines if that pixel is a corner by checking if there is a set of continuous pixels in the circle that are either all brighter or all darker than the pixel under evaluation. Despite being fast, this approach contains a few problems due to multiple features being detected adjacent to one another. This characteristic translates in a lack of robustness against image noise and the algorithm is dependent on a threshold value in the number of pixels to examine near the one under test [45].

One last traditional feature extraction algorithm worth mentioning is Oriented FAST and Rotated BRIEF, ORB. It was developed by Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary R. Bradski in 2011 [39]. This method mixes some of the algorithms mentioned before. It starts by utilizing FAST to find key-points, followed by the application of Harris corner measure to find the best points among them. However, FAST does not compute orientation. To increase ORB's robustness against rotation variance, the authors came up with the computation of the weight of the centroid of the patch with the located corner. The direction of its vector determines the orientation. ORB utilizes BRIEF [8], a method that provides a fast feature calculation and matching.

Image matching does not end with feature extraction. In fact, despite being such an important part of the process, feature extraction would be worthless if it was not followed by a feature matching algorithm. *Feature matching* is the task of establishing correspondences between similar features extracted from two different images.

Suppose that p is a point previously detected in an image associated with a descriptor k.

$$\phi_k(p) = \left(f_1^p, f_2^p, ..., f_n^p\right) \tag{2.1}$$

The objective is to find the best match q in another image from the set of N interest points Q=$q1,q2,...,qN$ by comparing the feature vector $\phi_k(p)$ with those of the points in the set Q. Accordingly, a distance measure between the two interest points descriptors $\phi_k(p)$ and $\phi_k(q)$ can be

defined as

$$d_k(p,q) = |\phi_k(p) - \phi_k(q)| \tag{2.2}$$

Depending on the distance d_k, the points of Q are sorted in ascending order individually for each descriptor creating the sets. A match between the pair of interest points (p,q) is validated only if p is the best match for q in relation to all the other points in the first image and q is the best match for p in relation to all the other points in the other image. However, in order to perform all these validations, it is primordial to devise an efficient method to be as fast as possible [17].

A straightforward algorithm to match features is the Brute-Force matcher. It takes the descriptor of one feature in the first set and is matched with all other features in the second set using distance calculation and then returning the closest feature. The algorithm takes only two parameters, the first one being the normType, regarding the distance measurement that can vary according to the feature extraction algorithm used previously. For binary string-based descriptors such as ORB, it is recommended to use the Hamming distance. The other parameter is a boolean whose function is to return only the matches that fulfill the condition mentioned in the previous paragraph where a feature from one set was the best match from a feature from another set and vice-versa. Turning this value to false can lead to worse results. Brute-Force matcher is also flexible as it can return the best match or the k nearest neighbor matches where k is a value specified by the user.

Fast Library for Approximate Nearest Neighbors, also known as FLANN, contains a collection of algorithms optimized for quick nearest neighbor search in populous datasets and high dimensional features. It works faster than Brute-Force matcher for large datasets [14].

FLANN-based matcher receives a dictionary of search parameters as input, which specify the number of recursions the trees in the index should be transversed. Despite leading to higher computation costs, the higher the values, the better the results.

Although it is imperative to choose the best feature matching algorithm and properly optimize its parameters, the performance of the matching methods is related to the properties of the interest points extracted and the previously selected image descriptor. What this means is that a proper selection of the Feature Descriptor Algorithm should be made.

2.1.1.2 Traditional Feature Matching Implementations

Some of the simplest approaches utilized to calculate matching similarity were through the implementation of the Mean Square Error or the Structural Similarity Index, also known as MSE and SSIM, respectively. MSE computes the mean square error between the pixels for two images, whereas SSIM will do the opposite and look for similarities within the pixels, such as identical alignments or density values [32]. Despite being more straightforward approaches to the problem, they can be very effective when there are no object rotations. As Salunke *et al.* [48] mentioned, using a pixel approach provides the extra measure regarding the image quality, which can be beneficial in some cases by giving additional detail but can also lower the matching accuracy due to circumstances where two similar images are considered as different if one of them is blurred. Fonseca *et al.* [10] implemented a platform that extracted data from Portuguese retail e-commerce platforms and then matched those products with the SSIM method, with two different approaches: one with the original extracted images and another one with a cut to the image limits. The results obtained showed that applying the cut was the approach with the better results as 68.55% of the data had higher similarity rates when compared to the 31.45% where the original image was kept.

Matching retail images using SSIM or MSE approaches is fast, however it does not provide robustness against image rotations or scalings. Other studies utilized more complex approaches and were able to achieve better results. Hebbarb *et al.* [4] attempted to develop a model of image recognition utilizing traditional approaches such as the SIFT and SURF methods for the feature extraction and Fast Library for Approximate Nearest Neighbors for feature match. This study can be compared to retail product matching due to its high complexity caused by parameters such as illumination, scaling, rotations and translations that can undermine the algorithm. The proposed system consisted of a flow that started with the selection of an input image from an open-source data set, followed by a key-point detection and the feature description. These two previous steps were variated by performing the key-point detection with SIFT and then followed it with a SURF feature extraction and vice versa. Then, the FLANN descriptor match was utilized to communicate whether there was a match or not [4]. The first approach was conducted using the SURF key-point detection by applying the stages of Integral Image Generation, the Interest Point Detection using Hessian Detector, and the Descriptor Orientation Assignment. Then, Feature Description was performed using the SIFT algorithm. On the other hand, the second approach had the techniques applied in the reverse order as SIFT was implemented in the stages of Scale-space Extrema Detection, Key-Point Location and Orientation Assignment and feature description was performed using the SURF algorithm. This study was conducted with two significant open-source datasets with approximately 15000 normalized images each [18]..

Karami *et al.* [23] conducted a study where different types of transformations were applied in a set of images and then proceeded to the calculation of matching parameters, namely image key-points, matching percentage, and the computation times for the different algorithms applied. The ultimate goal of the study was to find out which algorithm was better for each image distortion operation. Regarding the approaches, state of art algorithms were implemented to compare their

performances, specifically SIFT, SURF, and ORB. In terms of image distortions, the ones applied to the images were rotation, scaling, horizontal and vertical deformities, and fish-eye distortions, whose goal was to create hemispherical panoramic images. An example of a fish-eye distortion is in meteorology to capture cloud formations. Results demonstrated that despite ORB being the algorithm with the most negligible computation costs, SIFT performed better in most scenarios. Similar to the previously mentioned study, Shehata *et al.* [24] also evaluated the performance of a feature description method against various image deformations, with a specific focus in the evaluation of the accurate positive matches comparatively to the false positives counterpart. For each deformation, the percentage of true positive matches was calculated as well as data from the orientation differences between the matched interest points. These orientation differences allowed establishing a threshold in which its value determined whether a positive match could be considered true or false.

Joglekar *et al.* [22] also presented a SIFT image matching approach with a standard feature extraction approach but with a few innovations regarding those features match by the utilization of probabilities. The key-point extraction consisted of gathering interest points based on a histogram of magnitude and direction gradients. Then, probabilities were assigned using the Baye's theorem according to categories, considering the key-point assignment as a classification task. The probabilistic estimates were improved with iterations that applied a relaxation labeling technique. Relaxation techniques' goal is to constantly reduce the ambiguity and disagreement of the initial labels by employing a consistency measure for the set of correspondences [33].

To sum up, each study had its own variations and most were conducted with normalized open-source image datasets. However, it was clear that utilizing an approach based on traditional feature extractors like SIFT, SURF, and ORB provided quality results in most cases, when combined with the respective feature matchers. These approaches also showed great robustness against rotations, scalings, and other deformities, making it ideal for retail image matching.

2.1.2 Convolutional Neural Networks

This section mentions the other image feature matching based approach, which is the usage of convolutional neural networks . Section 2.1.2.1 provides an explanation of the core concepts of CNNs, and section 2.1.2.2 mentions a few of its usages in a retail matching context.

2.1.2.1 Core Concepts

With the evolution of the computer vision area, new algorithms and approaches began to appear, making the contemporary techniques much quicker in run times than before, and most importantly, they also provide better results. As seen in the previous subsection, there are several approaches regarding computer vision and, more precisely, image matching. However, according to the state of art research, convolutional neural networks are considered the best performers in such tasks.

The idea of utilizing machine learning algorithms in the area of computer vision is not recent but only in the last decade was able to be implemented due to the rise of computer power. Within machine learning, the most successful algorithms are the deep learning techniques, capable of identifying several levels of image representation with considerable detail. More specifically, convolutional neural networks revolutionized the entire field of computer vision by starting to learn the basic shapes in the first layers and then evolving to learn the image features in the deeper layers, which leads to an increment in image comparison and classification accuracy. Convolutional neural networks were inspired by the visual cortex of a cat, which translated into a hierarchical representation of neurons by Hubel and Wiesel in 1962 [20].

Convolutional Neural Networks, also known as CNN, were first proposed by LeCun and later improved [28]. Still, the first network proposed did not have much success due to the lack of computational power and the short amount of training data. Nevertheless, in 2012, CNNs had a giant leap when AlexNet was invented [27] in the field of image recognition. This achievement was because, with deep neural networks, CNN now had a flexible architecture, with layers of neurons being added, improving its learning capacity. To summarize, this lead to bigger training datasets as well as utilizing the upcoming rise of the computational power of GPUs to fetch the training data.

What makes a CNN convenient in computer vision is its capability of detecting patterns. A convolutional neural network consists of five layers where each layer performs a distinct task with the input data. The first layer of the CNN is the *input layer*, which reads the input images and is then followed by the *convolutional layer*. It performs a convolution operation and calculates the dot product between the filters and the image patches, which translates into the computation of an output volume, passed then to the next layer. It is within the convolutional layer that the image features are extracted. The following layer is the *pooling layer*, used to reduce computation costs by decreasing the spatial volume of the received input after the convolution. It is then followed by the *fully-connected layer* which receives the previous inputs and outputs a single dimension array with the class scores. Finally, the last step of a CNN is the *output layer*, which analyzes the scores and predicts a class [42].

Convolutional Neural Networks have the capability of extracting image features and converting them into lower dimensions maintaining its characteristics. Its main advantages are the simplicity of each layer as it only performs one specific task and the ability to achieve the best results in most computer vision problems. Another benefit of utilizing this approach is its flexibility, as further layers can be added to the previous paragraph's ones to enhance training performance. For instance, the dropout layer is a layer that can be integrated with a CNN to drop random neurons during the input function and remember the neurons that are left to bring regularization. The model is then able to learn robust features independent of the neurons, and this is a way of avoiding the overfitting [21]. However, a few disadvantages are the lack of encoding the position and the orientation of the object in the predictions and the higher computational costs compared to other algorithms. In addition, not every problem can be solved with a CNN due to its high requirements towards the training datasets [21].

The next section provides a deeper look on how some convolutional neural networks were implemented to classify retail products.

2.1.2.2 CNN Implementations

This section describes a few implementations of convolutional neural networks in matching environments. These approaches can be divided in implementations where the authors solved their problem by directly applying CNNs to predict the end result, or in implementations where only a set of CNN's functionalities was used to attribute matching scores between two images, and other techniques are responsible for converting the scores into predictions. Additionally, instead of analyzing one image and predicting its respective product, few studies trained CNNs to receive an input of two images, where the prediction would be a binary classification corresponding to whether or not those images were a match.

Starting with more conventional CNN use cases, Masood *et al.* [34] developed an experiment where he trained a model of a 16 layer architecture previously trained on several images. The model allowed the extraction of features from the train and test data. The image was passed through a stack of convolutions with 3x3 filters with a ReLu activation function as in the other convolutional neural networks. ReLu stands for The Rectified Linear Unit. Its implementation helps prevent exponential growth in the resources required to operate the neural network by making the gradient calculation of a neuron computationally inexpensive. Pooling was then performed through a max layer of a 2x2 window and was then followed by three fully connected layers. In addition, the network was compiled with categorical cross entropy for loss function [6].

Jogin *et al.* [21] also opted for a more classical CNN implementation, inspired by AlexNet [27] with a typical CNN with a ReLu activation function. The network layers were: input, convolutional, ReLu , pooling, fully connected, and output. CIFAR-10 [26], a famous training dataset, was utilized, with normalized image size and three color channels. The convolutional layer computed the local dot between the filters and the image patches. The resulting dimensions of this operation had several different values as the number and size of filters used in the convolutional layer were varied. The max-pooling layer had the common objective of downsampling values

and reducing their dimension. Then, the fully-connected with 10 neurons layer computed the class scores, which were mapped using a SoftMax activation layer. SoftMax is a generalization of logistic regression and can be utilized for multi-class classification [50].

In contrast, Schroff *et al.* [40] showed the effectiveness of CNNs in image matching, but the role of CNNs in his solution was only to extract the features and not to predict the final result. The utilized approach was based on mapping images to a compact Euclidean space where distances directly correspond to a measure of face similarity. Once this embedding was produced, computer vision tasks became straightforward, as image recognition became a k-NN classification problem. Some limitations of the classical CNN approach are that it does not guarantee the bottleneck layer capability of appropriately generalizing new images, and the image requirements are also very high in terms of quality. On the other hand, Schroff's method did not train the CNN to generalize the match matches but as a way of optimizing the face embeddings. The dataset utilized was LFW [18], and the conducted study reached an accuracy of 99.63%.

Dusmanu *et al.* [13] proposed a method where a convolutional neural network acted firstly as a feature descriptor and only after as a feature detector. Its advantages were that by postponing detection, the acquired key-points were more robust than they would be otherwise. The goal of this study was to match extracted features to identify large-scale reconstructions. The proposed approach started by calculating feature mappings with a deep convolutional neural network. Following this step, the feature maps were used to calculate the descriptors and find the image key-points. To summarize this architecture, the author stated that by performing the description previously, the detection was able to use feature maps from deeper layers, which translated into a higher link between the descriptor and the detector. In other words, by splitting the description and the detection, the author could perform feature detection with a higher level of knowledge. Another unorthodox approach to what is the default CNN implementation was brought by Malisiewicz *et al.* [12]. He came up with a fully convolutional neural network architecture called SuperPoint, which operates with original-sized images and produces interest point detections accompanied by fixed-length descriptors in a single forward pass. The model starts by encoding the images to reduce their input dimensionality. After this phase, the model splits its learning procedure into two parts: one for key-point and another for key-point description. Similar to the previously mentioned approach, the author was able to achieve with this approach the ability to share calculations and representations across the feature descriptors and detectors, something that the most classic strategies are unable to do.

Conversely, some researches used CNNs to classify retail products in a binary approach. In other words, the process of classifying an image was to iterate through all the dataset images and for each pair the neural network would predict whether those images were a match or not. More *et al.* [35] utilized this strategy to identify retail products based on the images, as well as utilizing string attributes, such as product title and description. The approach identified the products' key attributes such as orientation or color from the respective images and measuring the disparities found regarding attribute extraction and detections. Performing the image comparison was problematic due to the small amount of labeled data and cases where the same product could

have different images with variations in perspective, proportions, or color. With the purpose of avoiding the task of collecting data for the vast amount of individual products, an indirect approach was to train an image similarity model based on an auxiliary taxonomy with nodes representing the web pages' navigation to facilitate the process, for example, Food -> Cookies -> Oreos. This allowed the image comparison across nodes and came up with conclusions regarding the products being the same or whether zoom or color variations were found. The author also stated that several approaches were beneficial as different models were responsive to different image characteristics.

Finally, Schmid *et al.* [36] proposed a matching algorithm with the name of DeepMatching, which was also based in a binary classification. DeepMatching is a multi-layer convolutional network and was developed with the purpose of classifying image matches as a 0 or 1. The algorithm's architecture had multiple layers, breaking down the patches into a hierarchy of sub-patches. The advantage of this approach is that it allowed working at several scales and mainly to handle repetitive textures. In addition, local matches are propagated up within each of the layers in the hierarchy. In each step, there is the possibility of identifying an incorrect match and progressively discarding it, reinforcing the robustness of the approach. To calculate similarity, images were converted into a 4x4 cell grid where the image descriptor, R, could be divided into 4 quadrants: *R = [R1, R2, R3, R4]*, being p_s the position of the respective quadrant. Each quadrant was then subdivided into 4 sub-quadrants and so on until a minimum patch size was reached, also called atomic size. As the following formula shows, given two image descriptors, R and R', the correlation mapping was implemented in a bottom-up fashion until the atomic size was reached, and then, the similarity was computed by doing the reverse operation.

$$sim\left(R,R'\right) = \frac{1}{4}\sum^{4}\max_{p} sim\left(R,R'\left(p\right)\right) \tag{2.3}$$

To conclude, convolutional neural networks state of art analysis demonstrated that implementations of this model can lead to very good results. However, in most cases adaptations needed to be performed to fit the model's requirements. Some methodologies opted to utilize only parts of these models, where others opted to convert multiclass problems into binary classification approaches to facilitate the learning phase. None of this CNNs retail matching implementations treated the problem as a simple classification one, where an image would be provided as input and its respective product class would be returned. A solution capable of overcoming such barriers and to treat the problem in a more direct approach would be very beneficial, but it could not be found.

2.1.3 Evaluation Metrics

A good predictive model is one that, given a new input of data, can predict the correct output. For instance, in the case of retail image matching problem, a good predictive approach would have to be capable of predicting the output corresponding to the product that image would indeed match. Still, the algorithm should also be capable of predicting the ones that could not. With this being said, no evaluation metric is perfect as some are better suited for a specific type of problem and vice versa [9].

As mentioned in the previous section, few studies on retail image classification were based on a binary classification approach. Instead of receiving one image and predicting its corresponding product, two images were received instead and the prediction consisted in whether or not those images were representations of the same product. More *et al.* [35] conducted his experiments following a binary classification, utilizing a confusion matrix. A Confusion Matrix shows the number of true positives, which are the values with positive ground truth and a positive prediction outcome; the false negatives, which consist in the cases with a positive ground truth but a false outcome; the false positives, concerning positive predictions with a false ground truth; the true negatives, which are cases where both the prediction outcome and the ground truth were negative. From this matrix, metrics like sensitivity, which consists in the true positive cases ratio, or specificity, regarding the true negatives ratio, can be estimated.

However, most experiments in the state of art of retail image matching do not treat the problem as a binary classification. Instead, they only require one image as input, and the prediction output is the corresponding product identifier. For instance, Schroff *et al.* [40] utilized a very commonly used metric, called accuracy. Classification Accuracy is the ratio of number of correct predictions to the total number of input samples. The usage of accuracy as an evaluation metric is important to determine the quality of the proposed model, as it directly evaluates the exactness of the prediction. Additionally, to tackle eventual classification imbalance towards retail categories, category accuracy was also utilized.

$$Accuracy = \frac{Numberof CorrectPredictions}{Numberof Predictions} \tag{2.4}$$

Other approach utilized to measure the performance of the models was Accuracy@T. Schmid *et al.* [36] created this metric which consists in the calculation of the percentage of correct pixels between the input image and the predicted product's image. This percentage was determined for a specified error threshold.

One final metric utilized in the state of art was the number of features extracted. This evaluation approach is directly related to feature extractor component of the classification process, which is considered the most important phase. Within the same algorithm, extracting more features from images is expected to achieve better matching results. However, this may not be the case when comparing the number of extracted features from two different approaches. For instance, Karami *et al.* [23] was able to extract more features with ORB comparatively to SIFT, but the second model achieved a better results overall.

2.2 Product Matching Platforms

Despite being such a complex process, image matching is only a small component in the bigger environment that are retail product comparison platforms. A price comparison application allows the user to compare product prices that multiple retailers sell. It then shows the differences between different deals from the same product, specially regarding prices. Some tools require a barcode scan while others allow the customer to type the product name, both of these options being followed by the list of retailers selling that respective product.

By combining image with string matching methods it is possible to achieve a quality approach to match similar products. However, similarly to the image process, string matching also has a few challenges. The same product may be described in different manners and have slightly different names. These are factors that change according to sellers' demands. One other difficulty are synonyms, which appear not only in titles but also in the products' descriptions. For instance, in a hair conditioner, the words Function and Effect may have the same purpose but result in a slightly different title. To conclude, product matching is no easy task, specially considering title, attributes, and image comparison simultaneously [29]. Figure 2.3 provides an example of how different retailers can present variations in the way they present the same product.

Figure 2.3: Same Product in Different E-Commerce Platforms

Source: Deep cross-platform product matching in e-commerce [30]

Combining string and image matching algorithms is not the only way to match products extracted from different e-commerce platforms. Another approach is to compare the EANs of each product. EAN stands for European Article Number, also known as International Article Number, and is a standard descriptor of a barcode symbology and numbering system used in global trade to identify a specific retail product from a particular manufacturer. In other words, EAN product matching is straightforward because two articles with the same EAN are the same product, which makes it a computationally cheap process with 100% accuracy. However, despite being the ideal solution, not all stores are interested in providing their data to comparison tools and do not deliver the products' EANs, which explains the usage of approaches like the combination of string and image matching algorithms. Additionally, product matching isn't the only arduous task when maintaining a product comparison platform, as gathering product data from different retail

websites is also a difficult process, and when incorrectly performed may compromise the whole platform. To gather this data, there are two main approaches: either the companies have interest in having the extra visibility provided by the product comparison platform and pay for their products to integrate those web pages; or the platform extracts the companies online products from their home pages, via API requests or by the usage of web crawlers.

Despite the challenges presented, few product comparison platforms were already developed with success. KuantoKusta is a Portuguese price comparison platform, leading by far the Portuguese market of price comparators. It is a simple platform that comprises the Portuguese market with detail, being trusted by retail consumers [2]. Google Shopping, previously known as Google Product Listing Ads, is a global resource for business owners looking to find competitors selling the same articles. Google is top of the list for quality engineering options, and it is not different when performing a difficult task such as product matching. In addition to KuantoKusta and most pricing comparator platforms, Google Shopping also allows sellers to add their products, selling them directly [1]. After analyzing these and other tools, the only quality price comparison tool in the Portuguese market is KuantoKusta. However, it is not retail oriented, and is also based in EAN matching. Therefore, its approach is not feasible in cases where stores do not provide these identifiers, similarly to what happens with retail stores.

To sum up, there is a lack of solutions to compare Portuguese retail products, and the development of one, independent of EAN matching, would be a great addition to such competitive market.

2.3 Summary

Although there are several international price comparison tools, the Portuguese food retail market lacks one. To implement such platform, the usage of a product matching approach with high performance would be required. After analyzing the state of art matching approaches, it is notorious that two main methods can help achieve the desired performance: classic feature matching algorithms and convolutional neural networks. However, despite studies showing that CNNs perform better than the other mentioned approach in most cases, its implementation in the retail food market may not be viable. This is because, to train the neural network properly, a high amount of data per a small number of classes should be provided to maximize its classification accuracy. When it comes to the retail sector, there are easily 50000 different products, each one with 2 or 3 identifying images, which is the exact opposite of what would be ideal for neural network learning. One particular approach that tackled this issue was the usage of data augmentation techniques to increase the number of images per product. In opposition, classic feature matching methods don't have such high dataset requirements and are ideal for tackling some of the most significant image matching challenges, namely rotations and scale variations.

In the next chapter, a deeper look at the challenges of implementing a product matching platform will be provided, and two proposed approaches for this problem, one with a variation of classic image matching approaches and the other concerning a CNN.

Chapter 3

Problem & Solution

This section addresses retail image matching implementation problems, as well as the methodology adopted to tackle them. Section 3.1. describes the main challenges each stage of the methodology encompasses, and Section 3.2. details the proposal capable of circumventing the previously mentioned challenges. The proposal is divided into the data selection, the developed approaches, the methodology to validate them, as well as a brief mention to the evaluation metrics.

3.1 Problem Description

As mentioned in previous chapters, retail companies are incredibly competitive. With the continuous price oscillations in such a volatile environment, the competitors need to adapt their offers as soon as possible. Therefore, the development of a product matching platform, rich in reliable data that is constantly being updated based on the Portuguese retail digital platforms, can provide accurate insights capable of revolutionizing this market. More precisely, having access to such application provides a big leverage, specially price wise.

However, to implement such platform some steps need to be taken into consideration before the actual product matching task, namely the data extraction and its storage. As those procedures are requirements and do not constitute the core of this thesis, only a brief explanation of their problems will be provided. Firstly, data extraction has a few complications. The first one concerns the different formats of each web page. To maximize the quality and quantity of the data acquired, both web scraping and web crawling should be used. Besides, some online platforms do not want their data to be extracted and defend their pages against web scraping. After gathering the data, it is impossible to immediately proceed to its analysis , as it needs to be cleaned and standardized first. Data cleaning stands for the approach of guaranteeing that data is not corrupted and has no missing or incorrect values, as that can lead to misleading results. Data standardization is used because the information acquired comes in different quality and formats. Without correct and homogeneous data, it is not even possible to proceed into the product matching.

Figure 3.1: Two very similar images, yet different products

Only after storing this standardized data, it is possible to apply the matching algorithms to identify which similar products are being sold by different retailers. This stage of the development of the proposed platform is the hardest task and can be split into two major parts: string matching and image matching. The main problem of string attributes matching is that the same product sold in two different online stores may have a slightly different name, may have synonyms in its title and may even have product characteristics appended to its title, which in some cases may hijack the result of a match.

Regarding the main focus of this thesis, the image matching component of the problem, there are several challenges to deal with, specially because of the peculiarity of retail product matching when compared to common object detection approaches. These difficulties can be grouped into four major aspects, i.e. the large-scale classification and the data limitation, intraclass variation and flexibility.

Regarding the large-scale classification, the number of distinct products to be identified in a supermarket can be humongous, approximately several thousands, for a medium-sized store, which exceed the ordinary capability of object detectors. What this means is current recognition approaches are not suited for the retail product matching problem, due to their constraints towards large-scaled categories.

Concerning data limitation, deep learning-based approaches, such as Convolutional Neural Networks, require large amounts of labeled data for training, raising notorious issues in cases where only small number of examples are provided. In addition, the best datasets provided by scholars are created towards the distinction of very different classes of images, which does not suit the retail problem, as two different retail products can have nearly identical images. In contrast, retail datasets have a bigger number of classes but fewer examples per each one, which is far from ideal for most state-of-art algorithms. With this being said, it is very important to gather the most examples possible per class in order to maximize the product recognition results. Some techniques were developed to deal with this challenge, namely data augmentation, which is the creation of synthetic data from the existing images to increase the number of information per class.

Intraclass variation is also a major problem that affects the quality of retail matching approaches. One big specificity of matching a set of retail product images contrasting to other object images is that within each class, there is a huge variety of products. For instance, if a model was trained to recognize different types of vehicles, it would be relatively easy to learn the

differences between the category car and the category helicopter as most examples inside each category would be identical. On the other hand, in retail that does not apply as a bib and a pacifier have completely different images but can be both find within the category "Baby". Furthermore, objects from similar minor categories often present few very small differences in a certain area of their appearance, as shown in figure 4.12, making the image classification even challenging for the human eye. Other important factor is the environment in which the product picture was taken, as different lighting angles or different backgrounds have a big impact in the classification of intraclass products.

Finally, the last major challenge of retail image matching is flexibility. Historically, there has been an increment in the number of products, as stores need to import new products frequently to attract new clients. Besides, existing products images are constantly updated over time. For these two reasons, an accurate matching system should be flexible to these changes with the minimum amount of retrain possible whenever there is a new product in the system or an old one is updated. However, this is not the case in most state-of-art algorithms. For instance, Convolutional Neural Networks are unable to identify some formerly trained images when converted in a new task. This leads to the need of the algorithm to be totally retrained whenever a change happens, which is extremely inefficient, both in terms of time and computer resources. Therefore, the development of a product classifier with long-term memory is a problem worthy of study.

3.2 Proposal

In this section, the solution's proposal will be discussed. Subsection 3.2.1. provides a brief explanation of the solution, as well as a description of the technologies utilized. Subsection 3.2.2. refers the selection of the dataset. Subsections 3.2.3. and 3.2.4. present two proposed approaches to tackle the retail image matching problem. Subsection 3.2.5. provides a look on the parameters' tuning and validation methodology. Finally, Subsection 3.2.6. mentions the metrics utilized to evaluate the solution's performance.

3.2.1 Solution Overview

The goal of this thesis is the development of a powerful image classification solution, able to properly match the Portuguese retail products. This solution will be then integrated in a platform responsible for extracting products' data from different retailers, and storing it in a database, to provide a future analysis of the Portuguese retail market. To achieve this goal two main approaches were developed, both based on image individual features. The first resorts in traditional feature detection algorithms to extract image features, being followed by a feature matching algorithm. The second approach utilizes convolutional neural networks to assign pictures to products. By implementing feature matching algorithms, the platform benefits from an increment in robustness against image scalings and rotation variations, which is an extremely frequent phenomenon in the retail e-commerce platforms.

Aiming to benefit from the state-of-art methods, the first approach will consist of a combination of Scale-invariant Feature Transform, SIFT, Speeded-up Robust Features, SURF, and ORB with FLANN and brute-force matching algorithms. By utilizing traditional methods, it is possible to deal with most challenges presented in section 3.1, as these algorithms are trained to detect distinct features of an image and then to match those features, returning a score. What this means is that this approach does not require retraining upon any update and that it is capable to deal with large amounts of products with few images.

On the other hand, this logic does not apply to convolutional neural networks. So in order to utilize the best performing method according to the state-of-art literature, a few adaptations need to be implemented. Therefore, the second strategy relies in the utilization of a properly trained CNN to identify which product category the images belong to. By doing so, the dataset is split into only ten categories such as Hygiene or Leisure and each one having a decent quantity of images. This way, the neural network requirements are fulfilled and the predicted category is returned. After receiving the category, the platform then applies a filter to the products within that category and a second neural network is trained to predict which of the filtered products matches with the input image.

To ensure that the methods employed are suitable for their intended use, both are submitted to a validation methodology as mentioned in the next section. Additionally, to make the study results as realistic as possible, the image dataset utilized was extracted from Portuguese retail e-commerce platforms, as explained in the next section.

Regarding the technology utilized, the product data is directly stored in an Amazon RDS Postgresql database. Amazon RDS allows tasks such as migration, backup, recovery and patching. The delivery of these resources was provisioned by Serverless Framework, a software that builds, compiles, and packages code for serverless deployment. Algorithm wise, the code was developed using Python with the help of a few libraries depending on the approach. One approach utilized was OpenCV and the other was Tensorflow Keras. OpenCV is an open-source library that includes several hundreds of computer vision algorithms. TensorFlow is an end-to-end, open-source machine learning platform. Keras is the high-level API of TensorFlow: an approachable, highly-productive interface for solving machine learning problems, with a focus on modern deep learning. It provides essential abstractions and building blocks for developing and shipping machine learning solutions with high iteration velocity [5].

Finally, experiment visualization was provided by Tableau [3]. Tableau is a powerful and fastest growing data visualization tool used in the Business Intelligence Industry. It helps in simplifying raw data in a very easily understandable format. Data analysis is very fast with Tableau tool and the visualizations created are in the form of dashboards and worksheets.

3.2.2 Dataset Selection

The usage of a proper dataset is extremely important to maximize any model's performance. It needs to be labelled with the right information, to provide context to the algorithms so that they correctly understand the world around them. In such a specific environment such as retail, the datasets to train any model, have extremely rare characteristics and are not available as open sources. In addition, as the goal of the project is not only to study the efficiency of several image matching models in the Portuguese e-commerce market, but also to integrate the solution in a platform capable of performing these comparisons on a daily basis, the datasets utilized in this thesis were directly extracted from the websites of four local retailers with the usage of web scraping techniques.

After being extracted, attributes like product name, brand, package size, and obviously product image were stored in an Amazon RDS database. In addition, one other very relevant attribute worth mentioning is the European Article Number, also known as EAN. As mentioned in section 2.2, it is a standard descriptor used in global trade to identify specific products. In other words, if a source company has one product with the same EAN of a product in other source, it means that those products are the same. On the other hand, if the EANs are different, then those products do not match. The extraction of this attribute was crucial to label the products. Two database tables were created: a dimensional table, *product_ref*, and a factual table, *product_match*. Each row in the *product_ref* has a different EAN, implying that each row is an unique universal product. As shown in the UML Diagram of the database in figure 3.2, additionally to the core attributes of a product, the dimensional table attribute *product_code_source1* represents the product code of that respective source. This is the retailer's identifier of a product, and different retailers have different product codes for the same product. With this being said, in order to find if a source company has a product, all it takes is to check whether there is a product code in the *product_code_sourceX*. The various company images are also stored in the *productImages array*, so that each product reference has every extracted image of the product and not only the ones belonging to a specific store.

The implementation of this database easily allowed for the creation of two datasets: one where images were labelled by its *product_ref* identifier and a second one where the label was the combination of the product category with the *product_ref* identifier. The usage of two different datasets is explained by the needs of the two different approaches. The first approach, mentioned in section 3.2.3 only requires to have a set of images, each one labelled with the *id* of the respective product in the database table *product_ref*. Oppositely, the approach described in section 3.2.4 requires the data to be labelled both with its *id* but also with the respective category.

An exploratory analysis of the data utilized will be provided in section 4.1. The next section explains the first proposed approach to classify image products.

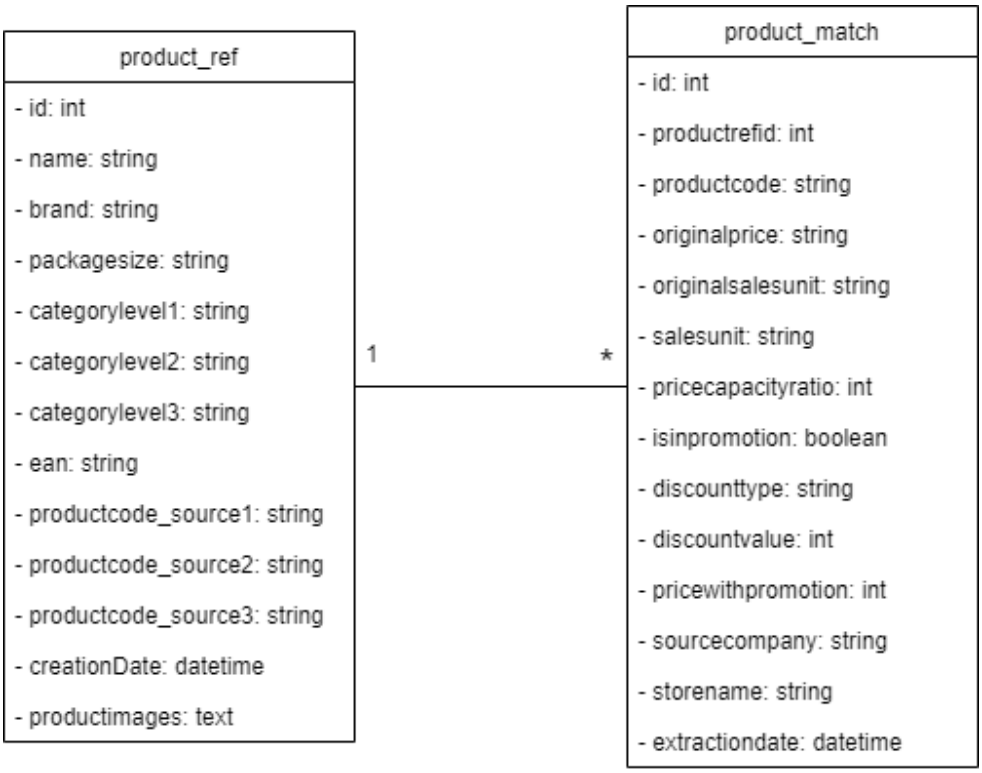

Figure 3.2: UML Diagram

3.2.3 Traditional Feature Extractors with FLANN and brute-force matchers

The first proposed approach for matching retail products images is a combination between traditional feature detection methods and feature matching methods. The feature detectors utilized were Scale-Invariant Feature Transform, *SIFT*, Speeded-Up Robust Features, *SURF*, and Oriented FAST and Rotated BRIEF, *ORB*. Regarding the match of the extracted descriptors, Brute-Force matcher, and Fast Library for Approximate Nearest Neighbors, *FLANN* were utilized. Each algorithm has its specific parameters and a validation step was performed to ensure the obtainment of the best possible results.

This approach utilized the dataset with images labelled by ids, implying that two images with the same id are images of the same product. In sum, after gathering a set of test images, the process for predicting the product id requires iterating through all the products and attributing a match score between the test image and that specific product images. After iterating through all the products, the product with the highest matching score is the most probable match. To assign a matching score between two images, the first step is to extract each image descriptors and to store them in a list. Both descriptor lists are then submitted to the image matching algorithm that returns a set of matches. Each match within the set represents the matching algorithm prediction that one descriptor in one image was the same descriptor in the other, attributing a distance to this relation. The smaller the distance, the higher the probability that those two descriptors are indeed representative of the same feature. With this being said, before attributing a score to the

Algorithm 1 Image Match with Classic Feature Extraction and Match Methods

Input: image, i,
product, p,
minimum matching distance, d

$scoreList \leftarrow []$
$P \leftarrow getImages(p)$
$N1 \leftarrow length\ of\ P$
for $k \leftarrow 1$ to $N1$ **do**
    $desc_i \leftarrow extractFeatures(i)$
    $desc_p \leftarrow extractFeatures(P_k)$
    $matches \leftarrow matchFeatures(desc_i, desc_p)$
    $goodMatches \leftarrow []$
    $N2 \leftarrow length\ of\ matches$
    for $j \leftarrow 1$ to $N2$ **do**
        if $matches_j.distance >= d$ **then**
            $goodMatches.append(matches_j)$
        $score \leftarrow calculateScore(goodMatches)$
        $scoreList.append(score)$
$max \leftarrow \max(scoreList)$
return max, maximum score obtained between the test image and the product images.

set of matches, it is also performed the removal of the matches that do not verify the predefined minimum distance to be considered a good match. This minimum distance threshold is a value that is defined in the validation stage for each approach. After having the minimal set of good matches, the matching score is calculated according to the amount of equivalent features. The pseudo-code for this process is provided in algorithm 1.

The choice of the algorithms to perform both feature extraction and feature matching is expected to have a big impact in the results of the matches, as according to the state-of-art research SIFT and ORB seem to have higher accuracy while SURF tends to be faster. While accuracy is the highest requirement when matching images, from the retailers point of view, it is also very important to minimize the operation costs. To be more precise, as retail prices vary in a daily basis and new products are constantly being added to the market, there is a urge to keep up with these changes from a product match platform point of view. In other words, thousands of matches between products are done every day, which increases the importance of building a model capable of achieving accurate results but also in the minimum time possible.

When it comes to the feature matching algorithms, each one has several parameters that influence the quality of the solution. Brute-Force is simpler than FLANN, as it takes the descriptor of one feature from one image and is matched with all the features from the other image, using some distance calculation, returning the closest one. The approach to calculate this distance varies according to the characteristics of the features extracted. For instance, for binary based descriptors such as ORB, the distance between the matches is measured utilizing the Hamming distance. Basically, it measures the minimum number of substitutions required to change one descriptor into

the other. In contrast, when features are extracted utilizing SIFT or SURF the distance is measured by the following Euclidean distance formula, where $list_1$ and $list_2$ are the lists with the extracted features from two images.

$$euclideanDist(list_1, list_2) = \sqrt{\sum(list_1(I) - list_2(I))^2} \tag{3.1}$$

On the other hand, Fast Library for Approximate Nearest Neighbors has a few more parameters to configure then simply features distance related, which makes it more complex but also provides better results when correctly calibrated. These parameters include the number of trees or key size and are mentioned in section 3.2.5. Similarly to the Brute-Force matcher, depending on the type of descriptors, the algorithm suffers a few variations. If the descriptors were extracted using SIFT or SURF, the FLANN matching approach will be based on a k-dimensional tree recursion, whereas if ORB was used as the extraction method, locality-sensitive hashing is utilized instead.

In the next section, an explanation on the other proposed product classification approach will be provided.

3.2.4 Convolutional Neural Networks

The second proposed approach to classify retail products images utilizes the state-of-art best performing method, Convolutional Neural Networks. This approach is slightly different from the previously presented one as it directly classifies the image as a product, contrarily to the traditional approach where matching scores are attributed first. In section 3.1, major problems concerning the implementation of this type of neural network in the context of the retail area were presented and to circumvent this issues, few modifications to the typical state-of-art CNN approaches were performed.

Algorithm 2 Image Classification with double CNN and string filter

Input: image, i,
category list, C,
string filter, f

1: $cnnCategories \leftarrow trainModel(C)$
2: $predictedCategory \leftarrow cnnCategories.predict(image)$
3: $products \leftarrow getCategoryProducts(predictedCategory)$
4: **if** f **then**
5: $\quad filteredProducts \leftarrow stringFilter(products)$
6: $\quad cnnProducts \leftarrow trainModel(filteredProducts)$
7: **else**
8: $\quad cnnProducts \leftarrow trainModel(products)$
9: $predictedProduct \leftarrow cnnProducts.predict(image)$
10: **return** *predictedProduct*, product id of the algorithm prediction

Instead of trying to predict the product each image belongs, a two step prediction is adopted. This means that a Convolutional Neural Network is firstly trained to predict the category of the test image, and then a second model is trained with a customized dataset of products within the category that was predicted.

The biggest problems for the usage of a neural network in the retail market context are the dataset requirements, as ideally a model should be trained with a small amount of classes, each one with a big amount of information. If a model was to be trained with an image dataset grouped by products, the results would be very poor as there would be thousands of classes, each one with one to three images. By splitting the approach into two steps, none of them has to deal with these dataset limitations.

Thus, the first model consists of splitting in categories, so that there are ten classes, each one with thousands of images, fulfilling the requirements of a good dataset. The ten categories are: *Animais*, *Bebé*, *Bebidas*, *Congelados*, *Frescos*, *Higiene*, *Lacticínios*, *Lazer*, *Limpeza*, and *Mercearia*.

The second model is trained with products within the predicted category. Despite being an improvement when compared to all the products in the dataset, each category has nearly 1000 classes so if the second algorithm was to be trained with that amount of data it would still not be a decent option regarding prediction accuracy. To tackle this issue, a filter was applied to

reduce the amount of products within that selected category. The filter used product string data, more precisely name, brand and package size, which were the attributes that showed the higher level of singularity. In sum, after the first model predicts a certain category, a dataset is created with all the products within that category. A string comparison algorithm is then applied to each product in the new dataset and each one has a similarity score attributed, which is the result of applying the Levenshtein Distance between the core attributes of that specific product and the test image product. After iterating the whole dataset, the algorithm returns the most similar products regarding string attributes. The number of most similar products returned is a parameter of the algorithm that can be oscillated to test the impact of the filter usage in this process. After the application of the filter, that translates in the final dataset, the second model is trained and the product prediction is performed.

The Levenshtein distance between two strings is the minimum number of single-character alterations required to change one string into the other. The alterations can be character insertions, removals or replacements. The equation provided below shows how the distance calculation between two strings a and b is done, where i represents the length of string a, and j the length of string b.

The next subsection mentions the validation methodology implemented to optimize the results of both approaches.

$$levenshteinDist_{a,b}(i,j) = \begin{cases} max(i,j) & \text{if } min(\text{i, j}) = 0, \\ min \begin{cases} distance_{a,b}(i-1,j)+1 \\ distance_{a,b}(i,j-1)+1 \\ distance_{a,b}(i-1,j-1)+1 \end{cases} & \text{otherwise} \end{cases} \tag{3.2}$$

3.2.5 Parameters' Tuning and Validation Methodology

Regarding the retail product images dataset, it is split in the following way: 10% constitute validation data, 20% represent the test data, and the other 70% are the training data. The usage of the validation set has the goal of evaluating each trained model's performance. The model with the best performance on the validation set is then selected to perform on the test set. To ensure that each model performs the best, it is necessary make a few optimizations rather than just executing image matching algorithms like FLANN with the predefined parameters' values. One particular algorithm responsible for parameters' tuning is *Grid Search*. It performs an exhaustive search over a pre-determined parameter grid, evaluating all combinations of chosen values. Each approach has its set of parameters that utilize the Grid Search tuning approach, as will be described in the following paragraphs.

Brute-Force matchers, only have two parameters to tune: the minimum distance between matches and the cross-check flag. As mentioned before, the minimum distance between matches represents the threshold of matches that can be considered good matches indeed. On the other hand, when true, the cross-check flag returns only those correspondences with value (i,j) such that i-th descriptor in set A has j-th descriptor in set B as the best match and vice-versa. That is, the two features in both sets should match each other. The FLANN matcher is slightly more complex as its parameters vary according the descriptors utilized. If the descriptors are descendant from SIFT or SURF, it is necessary to tune the number of trees. When the descriptors are extracted with ORB, three new parameters need configuration: table-number, key-size and multi-probe-level. In addition, the minimum minimum distance is also a FLANN parameter that needs to be tuned independently of the descriptor utilized.

Similarly, convolutional neural networks also need optimization, specially the first CNN, that is used to predict the category after being trained with a constant dataset, in contrast to the second CNN, where the dataset is generated dinamically according to the predicted category. The main variables that are tuned in this approach are the number of epochs, the number of layers the amount nodes per layer, and the number of dense layers utilized. Section 4.2 shows how the results of the Grid Search approach.

3.2.6 Evaluation Metrics

To evaluate the quality of the proposed solutions, it comes with no surprise that accuracy is the most relevant metric. To be more precise, accuracy is the ratio of the number of correct predictions to the total number of input samples, which in this case, are retail product images. As discussed in the state-of-art section, there are other ways to measure image matching approaches, more concretely, based on confusion matrices. However, that type of evaluation in a retail product matching scenario requires the problem to be treated as a binary classification problem, due to the huge amount of product classes. Image matching can, indeed, be considered a binary classification problem, where true matches are true positives and false matches false positives, and the opposite for the false cases. As the goal of the proposed solution is to assign an image to a product, instead of determining if two images correspond to the same product, confusion matrices are not the best option. As explained in the previous section, even though one of the proposed approaches is based on attributing match scores between two images, those scores are then converted into a classification. To be more concrete, the accuracy measurement is performed by comparing the product id label of the image under evaluation with the id that the classifier predicts. Additionally, it is also worth mentioning that it is unrealistic to expect incredibly high accuracy results, as retail product matching is not feasible utilizing only images. In fact, there are cases that are only distinguishable recurring to string comparison methods.

Moreover, other important way implemented to evaluate the models other than overall accuracy is category accuracy. The model is capable of analyzing how accurate its results are for each individual category, as not all categories are expected to have the same results. For instance, packaged products have less image variations when compared to a piece of fruit. By identifying any category matching flaw, it is possible to plan future improvements towards the proposed approaches.

While accuracy is extremely important, from the platform's perspective it is also very important to reduce the time costs, as the requirements of retail matching platforms obligate the model to perform thousands of daily comparisons, which make the models obsolete if those comparisons are not carried out under a specific duration. To sum up, the proposed solutions quality is also measured accordingly to their executions times as shown in the experiments conducted in the next chapter.

3.3 Summary

This chapter described the major challenges of retail product classification utilizing images and then, two different approaches were proposed to circumvent these issues and provide quality solutions. The proposal firstly mentions the dataset selection and is then followed by a description of the solutions implemented, one providing a combination between traditional feature extractors and feature matchers, and the other one utilizing convolutional neural networks. Additionally, the whole process of the dataset creation is explained, from the requests to e-commerce stores websites to the creation of the retail product image datasets. Besides, a look at the validation methodologies to optimize the algorithms performances was provided, as well as an explanation of which metrics are used in the evaluation of the proposed solutions.

The peculiarities of retail product image matching translate in adaptations to state-of-art approaches. As mentioned before, the goal of this thesis is to achieve the maximum accuracy product classification approach, having in consideration that it will hardly be perfect, as some products can only be classified through their string attributes. Therefore, the proposed solution consists in the development of a solution that can properly classify images of the Portuguese retail market, so that it integrates a more complex platform whose objective is to extract data and to perform product matches on a daily basis.

Next chapter will provide insights on the experiments that were conducted as well as an analysis of the results obtained by each technique.

Chapter 4

Experimental Setup & Results

This chapter aims to show the results obtained in the project, referring to the proposed approaches. Firstly, Section 4.1 describes the collected dataset. Secondly, Section 4.2 mentions the parameters' tuning step and its results. Finally, Section 4.3 contains several experiments conducted and an analysis to their results.

4.1 Exploratory Data Analysis

In section 3.2.2, an overview of the dataset selection was given. As mentioned, two datasets were created, because of the necessities of the different models, but the only difference between one dataset and the other is the way data is labelled, as the dataset utilized to train CNNs had its data splitted in retail categories. With this being said, both dataset have exactly the same products, only organized in a different way, which means that the data analysis presented bellow is valid for the two datasets.

A total of 9667 singular products were extracted from 4 Portuguese retailers. The figure bellow shows the number of products per retailer.

Retailers	Number of Products
Source A	2072
Source B	8336
Source C	5193
Source D	8392

Table 4.1: Number of Products per Source Company

Source Availability	Number of Products
2	9667
3	4325
4	1125

Table 4.2: Number of Products in Multiple Sources

Note that the sum of the products per retailer is bigger than the amount of products, as the same product is available in several retailers. On the other hand, not every product matches with the four retailers. In fact, only approximately 11% of the products are available in the 4 retailers. From a dataset point of view, this translates in the majority of product classes having 2 to 3 images each to perform the matches. This data is represented in table 4.2.

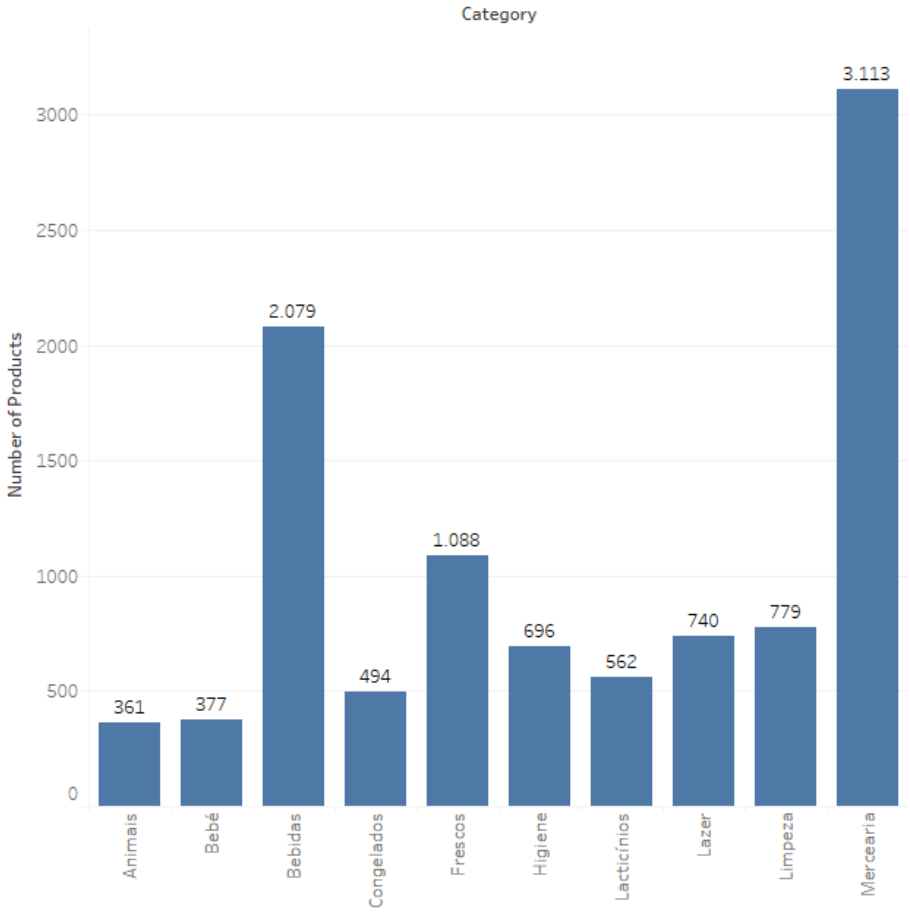

Figure 4.1: Amount of Products per Category

Another important conclusion obtained by analyzing the dataset is that there is a clear imbalance towards the amount of products per category, as shown in Figure 4.1. For instance, *Mercearia* and *Bebidas* have more products than the other categories, which can increase the classification

difficulty, as there are more similar products in those categories. It is also important to mention that certain products may belong to more than one category.

4.2 Parameters' Tuning

In section 3.2.5, the process of tuning the parameters for each approach was explained, as well as the parameters themselves. A validation dataset was defined and a quick search approach was conducted to measure the differences in performance using different values as parameters. Traditional algorithms utilizing Brute-Force matchers only had their distance and cross-check parameters tuned, where the ones using FLANN had a few more variations. Approaches with SIFT and SURF, had their distance and number of trees configured in contrast to ORB, where variations in table-number, key-size, and multi-probe-level were performed. The following figures are not representative of the total amount of parameters and their variations, as it would lead to hard to interpret graphics.

Starting with Scale Invariant Feature Transform, a better performance was obtained by using a minimum matching distance of 0,5 in both Brute-Force matcher and FLANN. The algorithm also performed better with Brute-Force by having cross-check verification and in FLANN by having 5 trees. Figure 4.2 represents the distance validation of both matchers. Each value corresponds to the percentage of the total accuracy for the sum of the methods' distances.

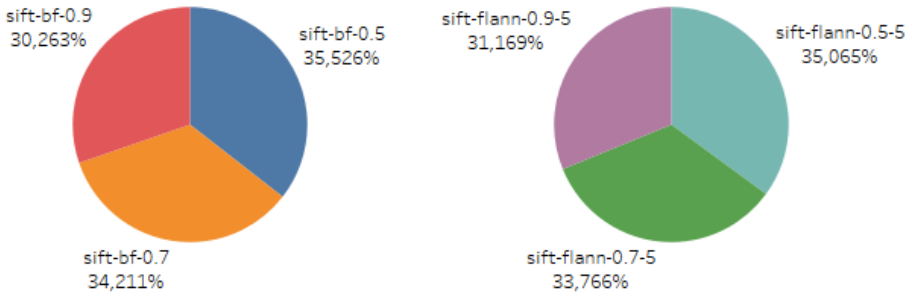

Figure 4.2: SIFT Distance Validation with BF and FLANN

Regarding Speeded Up Robust Features, the usage of the Brute-Force matcher achieved better results with a minimum distance of 0.9 where FLANN had its peak with 0.5. Similarly to SIFT, Brute-Force also obtained superior results with cross-check enabled and FLANN with 5 trees. Figure 4.3 shows the distance validation between a few values using the two matchers.

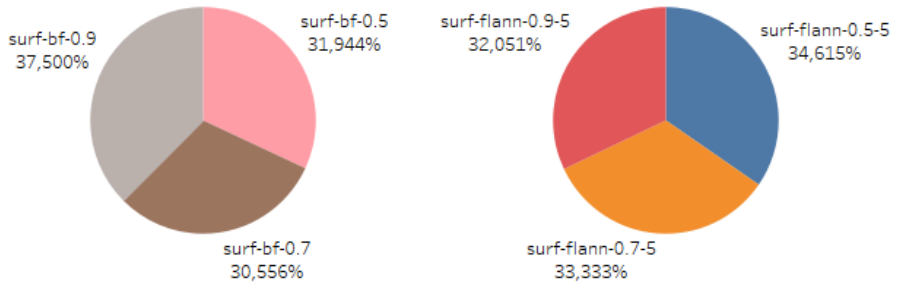

Figure 4.3: SURF Distance Validation with BF and FLANN

The last traditional method utilized was ORB, which performed better using a distance of 0.9 when matching the descriptors using brute-force and 0.5 when using FLANN. In addition, the usage of FLANN with a table-number of 12, key-size of 20, and a multi-probe-level of 2 as the best parameters.

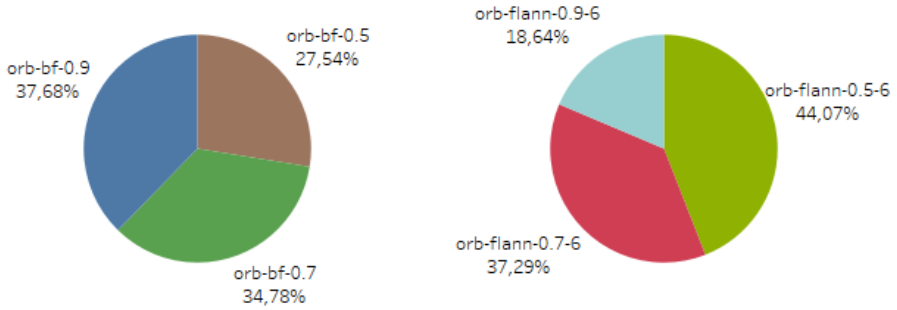

Figure 4.4: ORB Distance Validation with BF and FLANN

Similarly to the traditional approach, the CNN also had its values tuned, more specifically the number of epochs, the size of the layer, the number of convolutional layers, and the number of dense layers at the end.

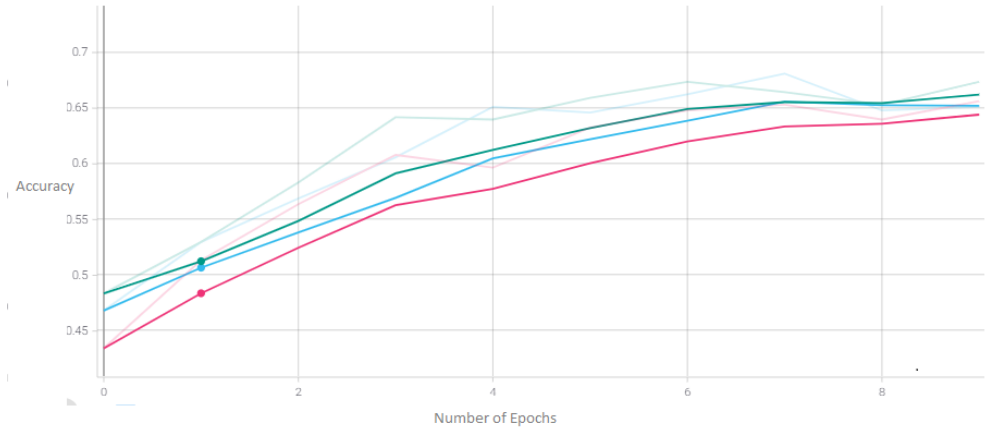

Figure 4.5: CNN Parameters' Tuning - Epoch Accuracy

Pink: 128 Layer Size, Blue: 64 Layer Size, Green: 32 Layer Size

To analyze the impact of each parameter in the model per each epoch, Keras Tensorboard was utilized. Each model had not only its accuracy evaluated but also its loss. In figures 4.5 and 4.6, it is possible to see three models in the colors of pink, green and blue, which are a sample of the CNN variations performed. In this case, all of them have 2 convolutional layers and 1 dense layer at the end, and are trained for the same number of epochs. However, pink has a layer size of 128, blue of 64 and green of 32. For these 3 samples, green would be considered the best model as it has both the higher accuracy and the smaller loss.

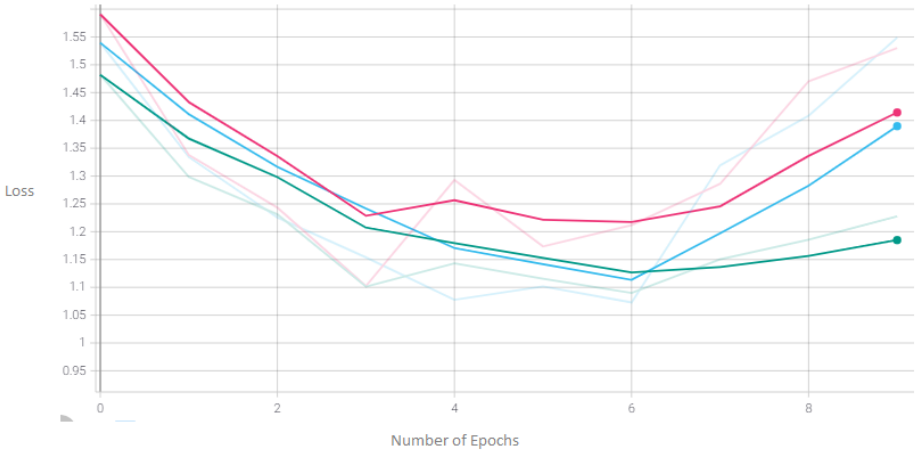

Figure 4.6: CNN Parameters' Tuning - Epoch Loss

Pink: 128 Layer Size, Blue: 64 Layer Size, Green: 32 Layer Size

Following 54 variations in the parameters of the CNN, the values that achieved the best results were obtained by training the model for 10 epochs, utilizing a layer size of 32, with 3 convolutional layers and 0 dense layers at the end.

After determining what the best configurations for each model were, various experiments were conducted. These experiments are explained in the next section.

4.3 Experiments

The most important aspect of a retail product classification solution is to study how well the proposed classifier performs, utilizing the accuracy metric. A few experiments to the accuracy were conducted, using the standard state-of-art gray-based pixel images but also their original version. Each model's accuracy was evaluated in these color variations, to find out if one approach was better than the other in general or just in some punctual situations. In addition, an average overall accuracy and accuracy per category were studied, to verify the solidity of the proposed solutions and to discover points of future improvement. Moreover, to measure the impact of the amount of features extracted in the accuracy of the solution, experiments regarding its influence towards feature extractors and categories were also performed. Finally, execution times were also measured, as algorithms with good predictions but with high time costs may not be appropriate for the problem.

4.3.1 Accuracy per Method - Black and White Images

The first experiment involved the utilization of different gray-based images. The results are shown in graph 4.7. Through the graph it is possible to conclude that the Convolutional Neural Network failed to accomplish relevant results when compared to the classical image matching approaches. As explained in the previous chapter was divided in two layers, one for predicting the category, and the second one, that with the predicted category and a filter restricting the range of products. It is important to point out that the first CNN had an accuracy of 86% to predict the category. The problem was the second CNN that clearly lacked data per each product to properly learn to classify them.

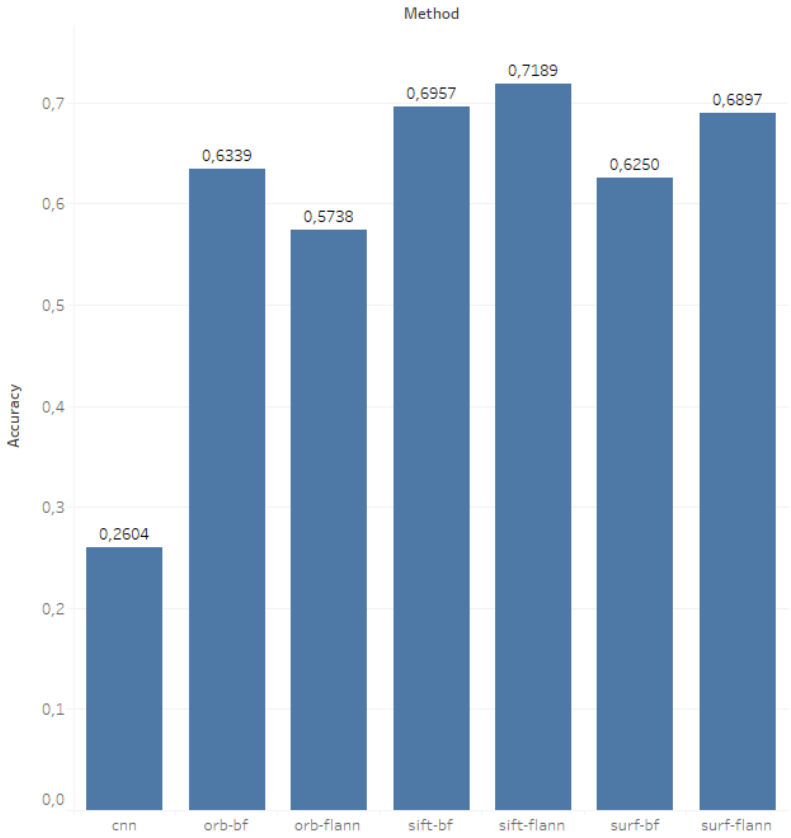

Figure 4.7: Accuracy per Method - Black and White Images

Regarding the traditional methods, good results were achieved, specially considering the lack of the string matching component of every product matching problem. The usage of the SIFT algorithm in combination with a FLANN matcher achieved the best results, having an accuracy of approximately 71%. SIFT with Brute-Force matcher and SURF with FLANN were the second and third most accurate options with accuracies near 70% and 69%, respectively.

4.3.2 Accuracy per Method - Coloured Images

The second experiment was also accuracy related but using coloured images instead. According to the state-of-art, most algorithms seem to perform better with black and white images. However, in some cases, colors are the differentiating factor between two products, even for the human eye. In order to verify what would be the best option for the algorithms, an identical experiment to the one in section 4.3.1 was performed, with the only difference that the utilized images were coloured. The obtained results demonstrated that almost every method performed better utilizing the full range of colours of each image, with the exceptions of ORB with FLANN and SURF with FLANN, that demonstrated to be independent of the colour approach used.

Similarly to the first experiment, SIFT with FLANN achieved the best results, obtaining an accuracy of almost 76%, and the biggest growth from one experiment to the other was ORB with BF that saw its accuracy score increase almost 8%. The accuracy of approach with Convolutional Neural Networks also incremented, however, the results were still too low to be considered an acceptable solution to the retail image matching problem.

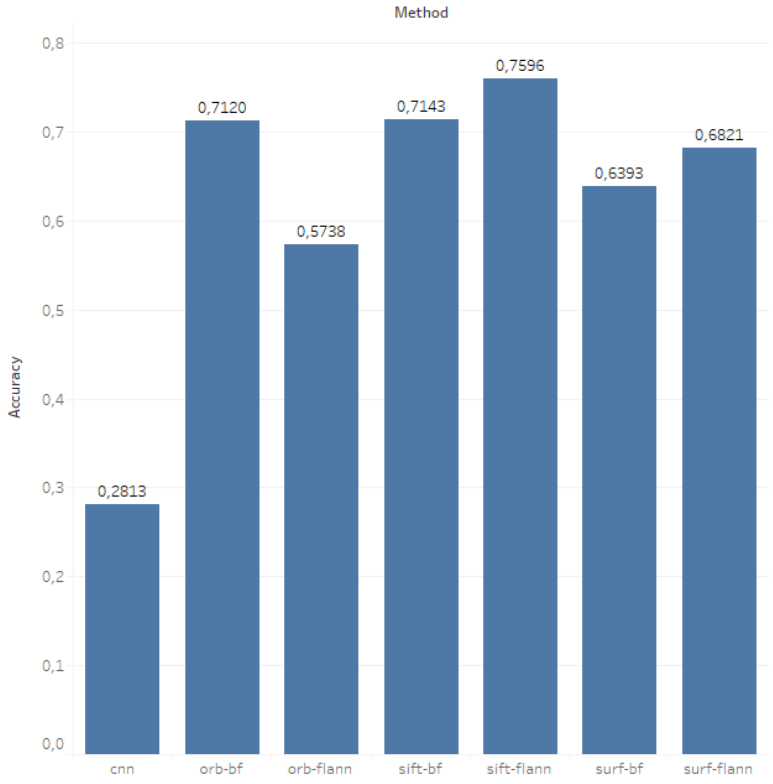

Figure 4.8: Accuracy per Method - Coloured Images

4.3.3 Accuracy per Category

In order to understand possible flaws of the solution proposed, the correct matches rate was also evaluated by variating each category, as shown in figure 4.9. Two particular categories obtained results below the average, *Bebidas* and *Frescos*. By analyzing these two categories, possible hypothesis explaining the low accuracy rates were formulated. Regarding *Bebidas*, it is a category with a huge amount of products, being the second highest with a count of 2079, as shown in figure 4.1, which combined with the similarities between each product within the category can be a huge problem for any classifier. For instance, two different bottles of the same whine can only be distinguished by the respective labels, as the rest of the body of the product is identical. On the other hand, *Frescos* classification suffers from the opposite problem, which is lack of similarity, not only between different products, but between the exact same product in different stores. An example is a variety of an apple, where each store can take its own picture of the product and the images can look completely different. This problem does not happen in packages categories, which results in higher accuracies of categories such as *Mercearia*.

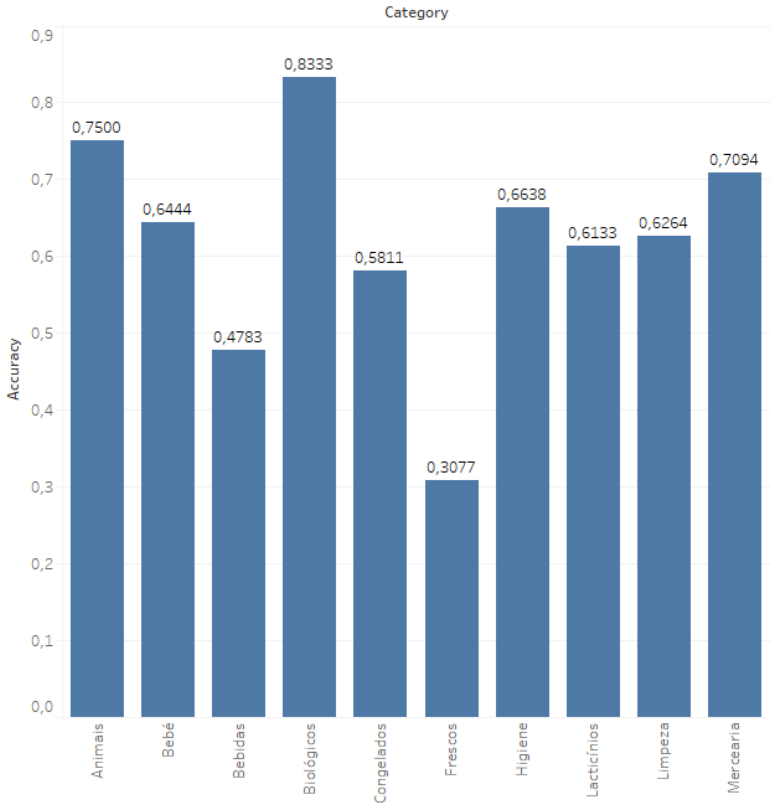

Figure 4.9: Accuracy per Category

4.3.4 Extracted Features per Method

Another conducted experiment was to measure the impact of the features extracted, comparatively to the accuracy of the method. As the number of features only varies according to the extractor utilized, and is completely independent from the feature matchers, there was no need to evaluate their combination, oppositely to the accuracy experiments. Instead, only the three feature extractors, SIFT, SURF, and ORB, were evaluated. Figure 4.10 represents the average number features extracted per method.

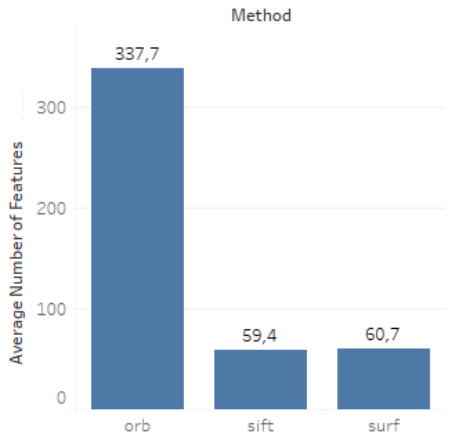

Figure 4.10: Average Number of Features Extracted per Method

By observing the results of the experiment, it is possible to conclude that the amount of features extracted doesn't have an impact in the accuracy of the classification. SIFT overall had the most accurate results and had the lowest average of features extracted with only 59.4 per image. On the other hand, SURF and ORB had similar results regarding accuracy, and their average number of features extracted per image is completely different. SURF only averages 60,7 features per image, contrarily to the impressive 337,7 features extracted for each image analyzed by ORB.

To sum up, this experiment showed that having a higher quantity of features extracted does not translate in having more accurate results. Instead, it is the quality of the features that dictates how accurate the classification is.

4.3.5 Extracted Features per Category

In parallel to the last experiment, another one was conducted regarding the average number of extracted features, but this time category related. The objective was to find out if any correlation regarding categories accuracy could be discovered. The results of the experiment can be visualized in figure 4.11.

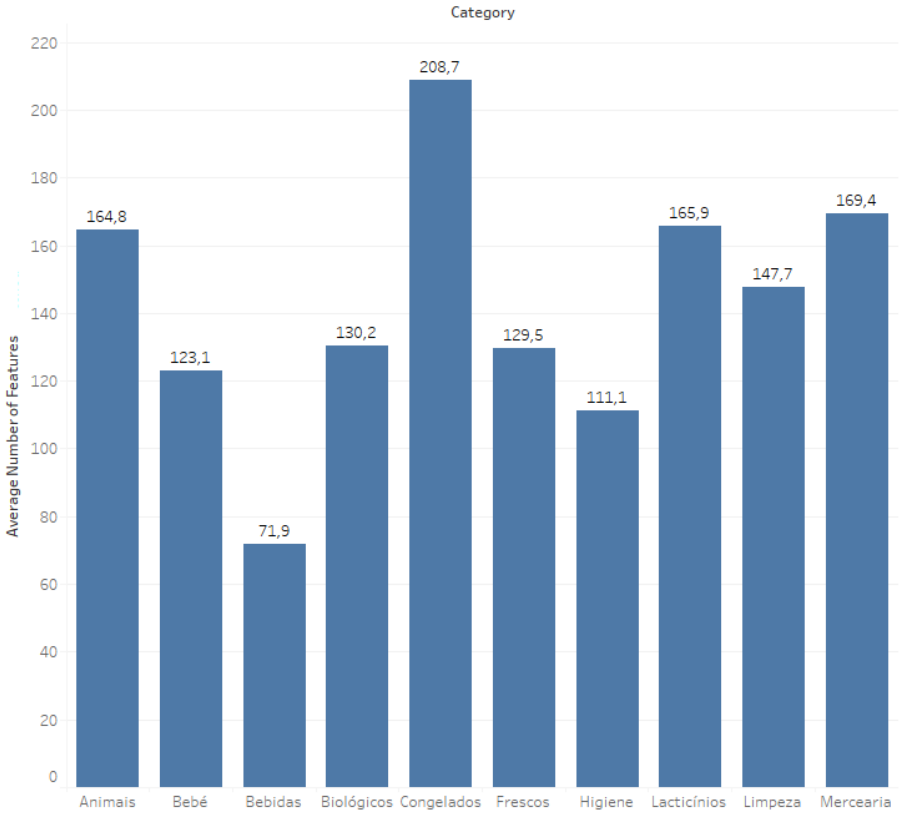

Figure 4.11: Average Number of Features Extracted per Category

After analyzing the results, it is hard to establish a direct connection between the quantity of features extracted and the category's accuracy. For instance, the category *Congelados* is the category with the highest average amount of features extracted, but is the third least accurate category with only 58% correct predictions. However, to have such a variation in the average number of extractions, using the exact same procedures between categories like *Congelados* and *Bebidas*, it is possible to conclude that some categories contain images that provide an easier feature extraction than others. In particular, traditional feature extractors seem to have difficulties extracting enough features in the category of *Bebidas*. A possible explanation could be that most images in this category simply contain a bottle with a small label in a minor portion of the image. As the body of the bottle has very few corners with little colour variations, it is harder for the extractor to find features, comparatively to packaged products.

4.3.6 Execution Times

After conducting experiments towards accuracy, the average execution times for the prediction of a match per method were also calculated. Some unexpected results were obtained, as SIFT was able to calculate each match in lower times than ORB or SURF, by obtaining the minimum average time per match, which was 20,65 seconds. On the other hand, as expected, the CNN approach took the longest average time to gather its conclusions taking almost twice the time traditional approaches needed to work. Those results are shown in figure 4.12.

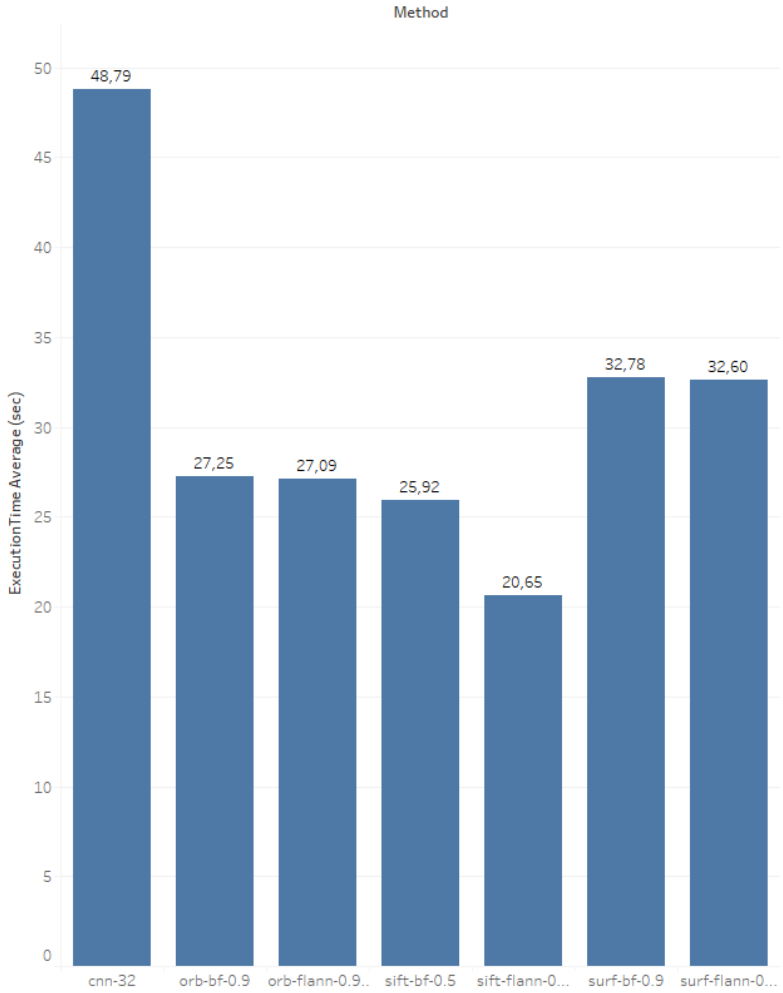

Figure 4.12: Average Classification Time per Method

4.4 Summary

In this chapter, an analysis to the dataset utilized was conducted as well as the explanation of how each individual approach was optimized to perform the best results possible within their capacities. Finally, several experiments were performed to evaluate which approach could provide the best results between the usage of traditional methods or convolutional neural networks. The first option clearly accomplished the best results, both in terms of image matching accuracy but also of execution times. The usage of coloured images also proved to be beneficial as most methods performed better with this option when compared to the usage of black and white images.

To sum up, the approach that accomplished the best performance was the use of SIFT as a feature extraction mechanism, followed by a FLANN to match the extracted descriptors, with an accuracy of almost 76% and an average execution time of 20, 65 seconds, utilizing coloured images. Contrastingly, the failure of the CNN classification approach was due to the second CNN. Possible reasons are that even though the amount of product classes were diminished by the application of a filter, each class only had a maximum sample of 4 images, which was clearly not enough to train the model to properly classify products.

Chapter 5

Conclusion

Retail markets present a huge growth and, consequently, extremely competitive environments. Therefore, every company attempts to innovate through the usage of technology to ease its customers shopping experience. The appearance of internet and its impact in this area translated in the globalization of the market, where local stores gave place to an omni-channel sector, combining physical with e-commerce shops. Thus, in a scenario where the clients can easily compare multiple products and brands, via their web browsers, source companies desire do distanciate themselves from the competition lead to the search for new solutions to revolutionize the market.

This project aimed at proposing a solution to help retailers position themselves in the retail market. The solution consists in a software that is able to match similar products from multiple e-commerce platforms from different source companies. Several matching approaches were performed in an attempt to understand which one fits better in this peculiar area that is retail. Despite being a matching focused platform, the whole process starts in an extraction of data from multiple sources and its storage in a database and ends in the product match. The work developed in this thesis culminates in the image matching. Matching results obtained by the platform could be improved by combining string similarity algorithms with the image matching approaches presented, but the ultimate focus of this project was the optimization of the comparison between products' image methods, in a retail context.

Various experiments were conducted in order to understand which matching approach would be ideal for a project with such specific requirements. Two main approaches were proposed: the first one having a variation of classic feature extractors, being them SIFT, SURF and ORB, followed by either Brute-Force or FLANN matcher; and the second proposed approach was based on convolutional neural networks. After the literature review on the specific conditions in which a CNN can perform, a two step CNN model was proposed. In the first part of the approach, the goal was to predict the product category instead of the product itself. Then, a string filter was applied in order to reduce the amount of products in the predicted category, so that the second CNN could be trained with fewer classes.

By comparing the two approaches, and even though the efforts to maximize CNN's potential, the approach conducted with classic extractors and matchers obtained much better results. Within this approach, the best combination of algorithms was the one utilizing SIFT to extract the image features and FLANN to match them. This approach was then added to the AWS platform, to store the products that are daily scraped from e-commerce platforms on database responsible for storing the matches. In addition, product categories also proved to be impactful in the results obtained, namely *Bebidas* and *Frescos*. These categories achieved the lowest results and compose a challenge to feature matchers due to the overall characteristics of their images. Contrarily, categories contained packaged products like *Mercearia* allowed to algorithms to achieve their best results.

5.1 Main Difficulties

Various adversities were faced along the development of the solution. In the first place, there is a huge amount of retail products, in a magnitude of several of thousands. From a classification point of view this is a big challenge as image classification algorithms have their best performances the lower the number of classes. On the opposite side, the amount of information per class is short as retail products have at most four images each. To summarize, retail image matching is a hard process as there is a huge number of products with small images each, which are the exact opposite conditions for the prosperity of an image matching classifier. Additionally, in terms of image similarity, different products may have almost identical images. For instance, most bottles of wine in the category *Bebidas* have a very identical look, despite having minor differences in their labels which compose a small component of the image. Convolutional neural network image classification also involved several additional difficulties. As mentioned in previous chapters, a first CNN was implemented to detect the category of the product image. However, due to the intraclass variation in terms of category, the learning phase of the neural network was a very challenging process.

5.2 Main Contributions

This thesis contributed to the study of several image matching and classification approaches for the retail product matching problem, namely traditional feature extractors and convolutional neural networks. The first approached achieved quality results and proved to be capable of being integrated in a product comparison platform. Oppositely, the second approach did not achieve high performances, but was able to circumvent the limitations restraining the usage of convolutional neural networks in such problem, and with future improvements could be able to revolutionize the retail image matching solutions. This project also provided a deeper understanding of the retail market and which procedures could be implemented to maximize matching results. In addition, the datasets were directly extracted from retailers websites, which made the study much more realistic and conclusive as the approaches were tested in an environment that simulated the challenges presented in a real retail product matching scenario.

5.3 Future Work

To conclude, future improvements could be added both in algorithm and platform level. Algorithm wise, there are two possible ways of improving the convolutional neural network image matching approach. One way would be to tackle the problem of having very few images per product class, recurring to data augmentation techniques, which increase the amount of data by adding slightly modified copies of already existing data or newly created synthetic images. The other approach that could be utilized in the future is to convert the CNN approach from a pure classification problem into a matching problem that would be then adapted into a classification one. This means that instead of receiving an image and trying to predict which product corresponds to it, the algorithm would take two images and evaluate whether those images are matches or not by attributing a score, similarly to the approach presented with traditional methods. After iterating through all product images, a classification based on the matching scores would be attributed. This approach tackles dataset restrictions as to train it would not require to utilize the restrictive retail image dataset.

To improve the product matching procedure, string matching must be combined with its image equivalent. As explained previously, this thesis was focused on the product image component, but images alone are not the best identifier of a product, as there are cases where it is even difficult to identify a match for the human eye. The analysis of attributes such as name, brand or package size are essential to distinguish one product from another. Therefore, as a future improvement, a similar study should be conducted with string matching approaches and then the most accurate one should be combined with the developed image approaches.

References

[1] Google shopping. Website. URL: https://www.google.com/shopping. Accessed on 2021-06-25.

[2] Kuantokusta: Comparador de preços online. Website. URL: https://www.google.com/shopping. Accessed on 2021-06-25.

[3] Tableau, business intelligence and analytics. Website. URL: https://www.tableau.com. Accessed on 2021-06-25.

[4] Vinay .A, Dixit Hebbar, Vinay S., Kannamedi Balasubramanya Murthy, and Natarajan Subramanyam. Two novel detector-descriptor based approaches for face recognition using sift and surf. *Procedia Computer Science*, 70:185–197, 12 2015.

[5] Martín Abadi, Ashish Agarwal, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[6] Baeldung. How relu and dropout layers work in cnns. August 2020.

[7] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. volume 3951, pages 404–417, 07 2006.

[8] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. Brief: Binary robust independent elementary features. volume 6314, pages 778–792, 09 2010.

[9] André Miguel Ferreira da Cruz. Fairness-aware hyperparameter optimization. July 2020.

[10] Francisca Leão Cerquinho Ribeiro da Fonseca. Developing a tool to help retailers better position themselves in the food retail market. July 2020.

[11] Deloitte. Analytics in retail going to market with a smarter approach. 2013.

[12] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. 12 2017.

[13] Mihai Dusmanu, Ignacio Rocco, Tomas Pajdla, Marc Pollefeys, Josef Sivic, Akihiko Torii, and Torsten Sattler. D2-net: A trainable cnn for joint description and detection of local features. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8084–8093, 2019.

[14] Cong Fu and Deng Cai. Efanna : An extremely fast approximate nearest neighbor search algorithm based on knn graph. 09 2016.

[15] Kujtim Hameli. A literature review of retailing sector and business retailing types. *ILIRIA International Review*, 8, 07 2018.

[16] C. G. Harris and M. Stephens. A combined corner and edge detector. In *Alvey Vision Conference*, 1988.

[17] Mahmoud Hassaballah, Abdelmgeid Ali, and Hammam Alshazly. *Image Features Detection, Description and Matching*, volume 630, pages 11–45. 02 2016.

[18] Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007.

[19] T. Huang. Computer vision: Evolution and promise. 1996.

[20] D. Hubel and T. Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of Physiology*, 160, 1962.

[21] Manjunath Jogin, Mohana Mohana, M Madhulika, G Divya, R Meghana, and S Apoorva. Feature extraction using convolution neural networks (cnn) and deep learning. pages 2319–2323, 05 2018.

[22] Jyoti Joglekar and Shirish Gedam. Image matching with sift features - a probabilistic approach. 38:7–12, 01 2010.

[23] Ebrahim Karami, Siva Prasad, and Mohamed Shehata. Image matching using sift, surf, brief and orb: Performance comparison for distorted images. 11 2015.

[24] Ebrahim Karami, Mohamed Shehata, and Andrew Smith. Image identification using sift algorithm: Performance analysis against different image deformations. 11 2015.

[25] Rajashekhar Karjagi. The anatomy of retail success. 12 2020.

[26] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

[27] Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton. Imagenet classification with deep convolutional neural networks. *Neural Information Processing Systems*, 25, 01 2012.

[28] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[29] Juan Li, Zhicheng Dou, Yutao Zhu, Xiaochen Zuo, and Ji-Rong Wen. Deep cross-platform product matching in e-commerce. *Information Retrieval Journal*, 23, 04 2020.

[30] Juan Li, Zhicheng Dou, Yutao Zhu, Xiaochen Zuo, and Ji-Rong Wen. Deep cross-platform product matching in e-commerce. *Information Retrieval Journal*, 23, 04 2020.

[31] Tony Lindeberg. *Scale Invariant Feature Transform*, volume 7. 05 2012.

[32] Iftekher Mamun. Image classification using ssim. *Towards Data Science*, January 2019.

[33] David Marshall. Relaxation labelling methods. *Cardiff School of Computer Science and Informatics*, 1997.

[34] Sarfaraz Masood, Shubham Gupta, Abdul Wajid, Suhani Gupta, and Musheer Ahmad. *Prediction of Human Ethnicity from Facial Images Using Neural Networks*, pages 217–226. 06 2018.

[35] Ajinkya More. Product matching in ecommerce using deep learning. September 2017.

[36] Jerome Revaud, Philippe Weinzaepfel, Zaid Harchaoui, and Cordelia Schmid. Deep convolutional matching. 06 2015.

[37] Lawrence Roberts. *Machine Perception of Three-Dimensional Solids*. 01 1963.

[38] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. volume 3951, 07 2006.

[39] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *2011 International Conference on Computer Vision*, pages 2564–2571, 2011.

[40] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 815–823, 2015.

[41] Nicu Sebe, Ira Cohen, and Ashutosh Garg. *Machine Learning in Computer Vision*, volume 29. 01 2005.

[42] Md Shahid. Convolutional neural network. *Towards Data Science*, February 2019.

[43] Springer Verlag GmbH, European Mathematical Society. Scale-Space Theory, Encyclopedia of Mathematics. Website. URL: https://www.encyclopediaofmath.org/. Accessed on 2021-05-20.

[44] Deepanshu Tyagi. Introduction to feature detection and matching. January 2019.

[45] Open Source Computer Vision. Fast algorithm for corner detection. URL:https://docs.opencv.org/master/df/d0c/tutorial_py_fast.html . Accessed on 2021-05-20.

[46] Open Source Computer Vision. Harris corner detection. URL:https://docs.opencv.org/master/dc/d0d/tutorial_py_features_harris.html . Accessed on 2021-05-20.

[47] Open Source Computer Vision. Template matching. URL:https://docs.opencv.org/master/d4/dc6/tutorial_py_template_matching.html. Accessed on 2021-05-20.

[48] Anil Wadhokar, Krupanshu Sakharikar, Sunil Wadhokar, and Geeta Salunke. Ssim technique for comparison of images. *International Journal of Innovative Research in Science, Engineering and Technology*, 03:16281–16286, 09 2014.

[49] Yuchen Wei, Son T. Tran, S. Xu, B. Kang, and Matthew Springer. Deep learning for retail product recognition: Challenges and techniques. *Computational Intelligence and Neuroscience*, 2020, 2020.

[50] Thomas Wood. Softmax function. *DeepAI*.

[51] Kunpeng Zhu, Y.S. Wong, W.F. Lu, and H.T. Loh. 3d cad model matching from 2d local invariant features. *Computers in Industry*, 61:432–439, 06 2010.