UNIVERSITY
INSTITUTE
OF LISBON

OMECO - Generating Personalized Business Card Designs from Images

Nuno Francisco Castro da Gama Antunes

Master in **Integrated Business Intelligence Systems**

Supervisor
Doctor João Carlos Amaro Ferreira, Assistant Professor
ISCTE- University Institute of Lisbon

Co-Supervisor
Doctor Elsa Alexandra Cabral da Rocha Cardoso, Assistant Professor
ISCTE- University Institute of Lisbon

October, 2020

TECNOLOGIAS
E ARQUITETURA

OMECO - Generating Personalized Business Card Designs from Images

Nuno Francisco Castro da Gama Antunes

Master in **Integrated Business Intelligence Systems**

Supervisor
Doctor João Carlos Amaro Ferreira, Assistant Professor
ISCTE- University Institute of Lisbon
Co-Supervisor
Doctor Elsa Alexandra Cabral da Rocha Cardoso, Assistant Professor
ISCTE- University Institute of Lisbon

October, 2020

Resumo

O aumento da concorrência nos setores retalhista e hoteleiro, especialmente em destinos e densamente povoados e turísticos, é uma preocupação crescente para muitos empresários, que desejam apresentar a sua estratégia de comunicação de marca ao público-alvo. Muitos destes negócios dependem do marketing boca-a-boca, entregando cartões de visita aos clientes. Além disso, a falta de uma equipa de marketing dedicada e de orçamento para a consolidação da imagem e criação de design limita muitas vezes a capacidade de expansão da marca. O objectivo deste estudo é propor um novo protótipo de sistema que possa sugerir desenhos personalizados para cartões de visita, com base numa imagem de cartão de visita já existente. Utilizando técnicas de transformação de perspectiva, extracção de texto e redução de cor, conseguímos obter características da imagem original do cartão de visita e gerar um desenho alternativo, personalizado para o utilizador final. Conseguímos gerar cartões de visita personalizados para diferentes tipos de negócio, com informação textual e uma palete de cores personalizada que corresponde à imagem original apresentada. Todos os módulos do sistema demonstraram ter resultados positivos para os casos de teste e a proposta respondeu à principal questão de pesquisa. É necessária mais investigação e desenvolvimento para adaptar o sistema atual a outros imprimíveis de marketing, tais como folhetos ou cartazes.

Palavras-chave: Visão Computacional, Inteligência Artificial, Marketing, Geração de Design, Processamento de Língua Natural.

Abstract

Rising competition in the retail and hospitality sectors, especially in densely populated and touristic destinations is a growing concern for many business owners, who wish to deliver their brand communication strategy to the target audience. Many of these businesses rely on word-of-mouth marketing, delivering business cards to customers. Furthermore, the lack of a dedicated marketing team and budget for brand image consolidation and design creation often limits the brand expansion capability. The purpose of this study is to propose a novel system prototype that can suggest personalized designs for business cards, based on an existing business card picture. Using perspective transformation, text extraction and colour reduction techniques, we were able to obtain features from the original business card image and generate an alternative design, personalized for the end user. We have successfully been able to generate customized business cards for different business types, with textual information and a custom colour palette matching the original submitted image. All of the system modules were demonstrated to have positive results for the test cases and the proposal answered the main research question. Further research and development is required to adapt the current system to other marketing printouts, such as flyers or posters.

Keywords: Computer Vision, Artificial Intelligence, Marketing, Design Generation, Natural Language Processing.

Acknowledgements

I would like to express my gratitude to my supervisors João Ferreira and Elsa Cardoso for their immense support and encouragement, not only during the development of this dissertation, but throughout my academic progress. This research would not have been possible without their guidance.

I would like to thank INOV INESC Inovação and Pedro Santos for the opportunity to develop this dissertation in collaboration with the institution and to my project manager Ricardo Ribeiro for always being patient and understanding in times of dissertation deliverables.

I thank my work colleagues and close friends Luís Elvas, João Boné and João Henriques, with whom it has been a great pleasure to share a similar education and research path with. Your friendship and support has been amazing throughout the years.

Finally, a special thanks to my parents and brother, who have always been a great source of inspiration and support, especially in this very convoluted year of 2020.

Contents

List of Figures

Abbreviations

BL	**B**ottom **L**eft (Corner) (see page 42)
BR	**B**ottom **R**ight (Corner) (see page 42)
CIE L*a*b	**C**ommission **I**nternationale de l'**É**clairage **L**ightness **a*** and **b*** (see page 46)
DoF	**D**egree **o**f **F**reedom (see page 46)
GAN	**G**enerative **A**dversarial **N**etwork (see page 10)
GFTT	**G**ood **F**eatures **T**o **T**rack (see page 40)
GPT	**G**eometric **P**erspective **T**ransform (see page 64)
HSL	**H**ue, **S**aturation and **L**ightness (see page 48)
HSV	**H**ue, **S**aturation and **V**alue (see page 46)
ICDAR	**I**nternational **C**onference on **D**ocument **A**nalysis and **R**ecognition (see page 49)
MSME	**M**icro, **S**mall and **M**edium-sized **E**nterprise (see page 5)
OCR	**O**ptical **C**haracter **R**ecognition
OMECO	**O**ne-Stop Shop **M**arketing **ECO**System (see page 6)
RGB	**R**ed, **G**reen and **B**lue (see page 46)
SOA	**S**tate **O**f the **A**rt (see page 10)
TL	**T**op **L**eft (Corner) (see page 42)
TR	**T**op **R**ight (Corner) (see page 42)

Chapter 1

Introduction

The effects of marketing and the constructs underlying the notion of brand in consumer behaviour have received much attention for decades, having been proven that a strong and identifiable brand image is associated with an enhancement of brand performance [1]. For this reason, companies yearly invest a significant portion of their revenues in marketing, fortifying the brand image the company portrays to customers and expecting a growth in sales that justifies the investment. The underlying phenomenon of revenue, market share or market value alteration derived from the investment in marketing to make the brand desirable [2], is explained by the concept of brand equity.

1.1 Overview

1.1.1 Brand Equity

Yoo, Donthu and Lee [3] refer to the contributions of Farquhar and Ijiri (1991), Kamakura and Russell (1993), Park and Srinivasan (1994) and Rangaswamy and Oliva (1993) to define brand equity as the incremental utility or value added to a product by its brand name. In its 2014 paper, Lee, James and Kim [4] go beyond it, defining brand equity to be:

> "The customers' perceived added value associated with a particular product that is accrued by a brand beyond the functional or utilitarian value of the product."

The latter definition of brand equity explores how people are willing to pay extra for a product that includes aspects that are often associated with a specific brand. For example, the main colours used by a brand or the design style used in products or product packages can factor in the perceived added value associated with the product. The connection between the brand name and brand equity has been demonstrated and presented by Rangaswamy et al. in 1991 [5]. The authors present the utility for a brand that a consumer might perceive as a combination of three components: (i) utility for the physical attributes of the brand in the parent category (i.e. the products overall attributes such as colour or shape), (ii) the utility for the brand name and (iii) the utility due to the interaction of the brand's physical attributes with the brand name (i.e. the products overall attributes associated with the brand that might make customer opt for it).

When faced with a just-launched product in a new category from a brand name with a favourable opinion in the eyes of the customer (ii), the customers tended to have higher intentions of purchase of that product to the detriment of others.

Brand equity consists of four different dimensions [6]:

- Brand associations;
- Brand awareness;
- Perceived quality of brand;
- Brand loyalty.

Alterations registered in any of these structural components that define a brand can have a positive or negative impact on brand equity, which is why they must be taken into consideration and monitored. Despite the complexity inherent with attempting to change these dimensions, marketing, whose aim, according to Drucker and Maciarello is branding [2], can have a strong influence by defining a brand identity strategy and shape a brand image perception.

1.1.2 Brand Image and brand identity

Brand image is of undoubted importance for the its overall value and revenue, being a central driver of brand equity [7]. These facts justify the importance of understanding what the definition of a brand image entails and the contours of it. Brand image has been used in different contexts by researchers, taking diverging definitions. Previous definitions have been categorized as being (i) General blanket definitions, (ii) with emphasis on symbolism, meanings or messages, (iii) personification of the brand as an entity or (iv) emphasizing cognitive or psychological attitudes of consumers towards the brand [4, 8]

Lee, James and Kim [4]'s attempt to contextualize and group brand image definitions resulted in the proposition of a revised definition, which will be used throughout this dissertation. According to the authors, brand image is defined as:

> "The sum of a customer's perceptions about a brand generated by the interaction of the cognitive, affective, and evaluative processes in a customer's mind."

This definition accounts for both extrinsic (brand imagery) and intrinsic (utilitarian) attributes of products, without necessarily considering an order of relevance of attributes that might influence the customer or order of mental images associated with the brand, as the customer very rarely has a defined ordered list of reasons for choosing a specific product. These limitations have been pointed out in earlier definitions of the topic, justifying the need of the previously stated definition and the choice of this definition for the dissertation. It is important to denote the difference between brand image and brand identity. As stated before, brand image concerns an external perspective of the company, reflected by the perceptions of customers about it. Brand identity denotes the desired perceptions of customers towards a brand, being the core character of the brand [9]. Strictly speaking, Black and Veloutsou [10] explore both Aaker(1996) [11] and Balmer and Greyser (2006)'s [12] contributions to defining brand identity as an internal perspective of how the brand aspires to portray itself in the market, the associations

desired to be made by the customer and the identifying symbols used (such as logo, colour palette or others). A perfect alignment between brand identity and image, meaning the internal view of how the brand is supposed to be portrayed in the market and the way customers really see the brand is the desirable scenario. This alignment would mean that the targeted market segment is, in fact, aware of what the brand represents and how it differentiates from its competitors. Focusing on a competent marketing and advertising plan that can display the business' identity to the segment can be a hugely effective strategy to align these two concepts and contribute to higher brand equity.

1.1.3 Consumer Behaviour

The importance of a focused advertising plan that can influence brand image is, in practice, seen in positive changes on consumer behaviour towards purchase intentions [13], having this factor a direct implication in brand equity. The reasoning behind this premise is manifold. Firstly, branding impacts quality perception, as customer prefer to buy products from well-known brand names and with distinguishable packages, as stated by Malik et al. [13], referring to the contributions of Khasawneh and Hasouneh (2010) [14] and Prince (2010) [15]. Brand association, meaning the constructs present in the consumers' minds about a brand, also have a positive relationship with buyer purchasing [16], being advertisements a mean for companies to disseminate their message and stay in the consumer's mind. The idea of increased quality perception and the presence of the brand through the form of advertisements or other marketing elements can lead to an increase in brand loyalty and customer retention. As stated by Lee, James and Kim [4], "Customers with brand loyalty demonstrate patterns of repeat purchasing of the preferred brand, and they often do not leverage alternatives. They are less vulnerable to price fluctuation, and they willingly pay premium prices.". Therefore, a strong brand image will reflect itself on brand equity, which is the desired scenario for a company competing in any market.

1.2 Motivation and Scope

The highly-competitive red ocean scenario [17] observed in many market segments, where many companies compete for a sizeable market share means not everyone survives and only a few are expected to grow. Whereas companies with a higher market share typically have a solid business plan with defined budgets for every major area of the company and a dedicated marketing department, that is not always the case for Micro, Small and Medium Enterprises (MSMEs) and companies operating in retail and hospitality sectors. For many MSMEs, there is no defined marketing department or budget, which can impede the execution of the strategy and achievement of the desired goals. A common situation observed in hostels is having only one person handle the entire marketing and publicity or having people accumulating functions within the company. In businesses such as restaurants or small retail, it is common practice not to have a marketing plan at all. As a result, due to its higher prices, little attention is paid to forms of advertisement such as television or the internet. The necessity of MSME companies inserted in highly competitive markets to stand out and communicate their brand with the desired target proves to be a problem when a formal marketing structure and plan is lacking. Most of these businesses rely on word-of-mouth marketing, expecting their previous clients to carry the brand's message and disseminate it to potential new customers. A common strategy is to empower the customer with tools that can simplify the process of sharing this brand image with a potential new customer, specifically print advertising. These tools, such as business cards, flyers, leaflets, brochures or handouts, can carry information about the brand's identity and personality and useful information for the customer such as directions or open times. Elements such as the overall style of design of the printed element, explainable text, slogans or even the font and colours used can give the individual a sense of what the brand represents. The desire to change the design of one of these elements can pose as a problem for a company with no specified marketing budget as it implies hiring a designer and going through the iterative design process cycle, with design revisions on every iteration. This approach presents two problems. Firstly, the time it takes in an iterative process, where every iteration may imply

significant changes to the previous design, especially in early stages. Specifically, the ideas of the designer and client for the print element must converge into a final design that the client is satisfied with. The cost is also a problem, as hiring a professional designer to create a unique design implies a sometimes unaccounted for expense from the company. The time-consuming process and, consequentially, the higher price-point of these printable marketing elements can be a result of the difficulty of communicating requisites between the client and the designer. Ideally, the designer would prefer concise technical language that exactly described all final design requisites. However, these are often not well established, meaning the customer does not exactly know what he desires and does not have a vision of how the finished product might look like.

The present work, jointly developed at ISCTE-IUL and INOV-INESC Inovação, presents an effort to address a problem in the field of computer vision. This dissertation is part of the One-Stop Shop Marketing Ecosystem (OMECO) project, financed by the Portugal2020 research and development support programme.

1.3 Objectives

The focus of this dissertation is to provide a prototype that can automate the creation and personalization of business card designs based on an existing business card, adapted to the user. Considering the limitations prior mentioned with the traditional iterative design, in this work, we propose a solution that, based on the natural image of an existing print advertising element that the user is partial to, can extract features, automatically creating a template-based customized design for the user. The newly created design must take into consideration the predominant colours and the text in the element. The presented study shows the effectiveness of each of the components and its contribution to the proposed system, taking as a case of study the design of business cards. We provide validation scenarios for each individual component of the system, in order to detail failure cases. Finally, we validate the system, by demonstrating its results on a real use case.

1.4 Contributions

Taking into consideration the defined objectives, in order to answer the research question, it was required to develop the components that work together in order to achieve the defined objective. The areas of Computer Vision and Natural Language Processing are crucial for the development of this dissertation's contributions. In the current work, we present a system that can generate and personalize a business card design, according to a user's needs, based on a submitted business card natural image. The contribution consists of a process flow system, in which the business card image is preprocessed, features are extracted, a design template is selected and personalized according to the extracted information. The pipeline's overall architecture is further explained in Chapter 3, where each system module is detailed.

1.5 Dissertation Structure

In this section, we provide an overview of the dissertation structure and main sections of each Chapter. The design science research methodology process [18] is used as the structural baseline of the present work, as presented in Figure 1.1.

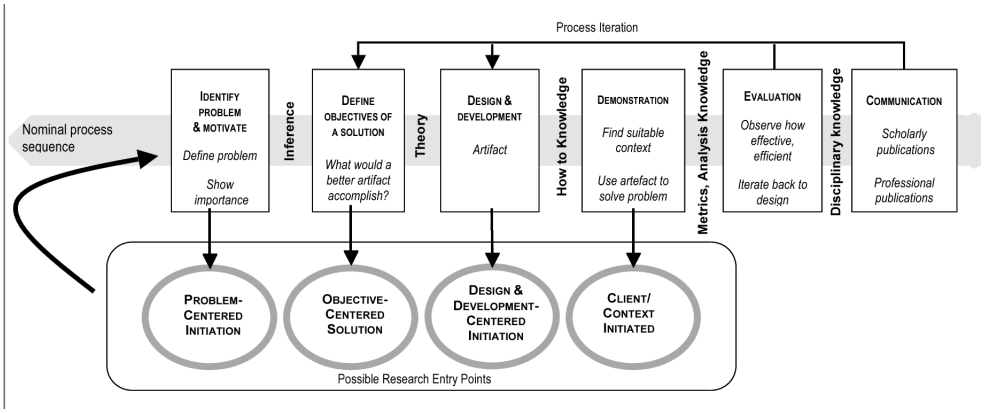


FIGURE 1.1: Design Science Research workflow [18]

In Chapter 2, we review the current relevant state of the art for the creation of the proposed system. The Chapter is organized into three sections. Firstly, a review of the current state of conditional image generation techniques is provided.

Following, we present existing works on design generation systems for marketing in literature. Finally, the state of the art for text detection and extraction is presented. In chapter 3, we present the system architecture, by explaining the functionality and importance of each module, as well as the architecture of each system component. In Chapter 4, the proposed method for business card design generation is presented and explained. The system modules mentioned in Chapter 3 are further developed and described. In Chapter 5, the system is evaluated and validated. First, the evaluation results for each of the developed components are provided, validating the choices for the system architecture detailed in Chapter 3. A system demonstration is provided based on the business card generation case study. The set context for this demonstration is based on the requirements presented by the corporation responsible for the research project. In Chapter 6, we conclude the dissertation, presenting an overview of the developed work and research questions answered, discussing the contributions and limitations to the field and expounding future directions of research.

Chapter 2

State of The Art

Creating personalized designs for marketing campaigns has been an important task for centuries, leveraging brand awareness and reputation, and delivering brand image to the customer [19]. Digital transformation has had a major impact in the generalization and diffusion of advertising, as it allowed inexpensive and easier access to design creation tools. Furthermore, this transformation, empowered medium and small companies with simpler and affordable ways of advertisement and portraying brand image, through digital mediums and marketing printables. As previously discussed in Chapter 1, customizing a design for a client in an automatic way, without human intervention, further reduces the cost and time to generate a personalized design.

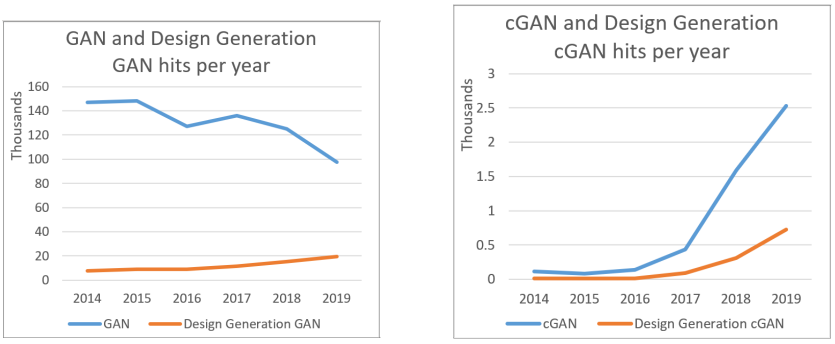
Automatic design generation is a challenging problem in the field of computer vision. The subjective aspect inherent to graphic design associated with individual likings, different perceptions of beauty and aesthetics, where different people have distinct preferences and tastes makes the problem increasingly convoluted. The rules of "what goes well together" are often complicated to derive, which is also associated with the subjectivity of the problem. To further aggravate the complexity of the research question, the diversity of needs for each client can prove to hamper research on the field. Creating a system with consistently acceptable results, fit for clients' needs and aesthetically pleasing is, therefore, an exceedingly complex challenge.

There are several design aspects to take into consideration [20] when designing a business card, namely the (1) legibility, the contrast between background and text, complexity of background and size of text are important factors; (2) complementing colours, maintaining coherence and adequability of colour palette; (3) adequate scaling, where icons and pictures must keep the aspect ratio and quality after upscaling or downscaling; (4) graphical element placement, in respect to the alignment of graphical elements in the final design; (5) fonts used must be adequate in type and amount; (6) emphasis and visual hierarchy, where key points in the design must be emphasized by colour, size, placement, amongst others; (7) spacing, where elements must be evenly spaced and there must be an adequate amount of white space to increase readability; and (8) design consistency, meaning all pieces in the final design should make sense together, being the design unequivocally representative of the brand behind the marketing campaign.

In this chapter we provide an overview of the state of the art (SOA) on the task of conditional image generation, including recent advances with the introduction of Generative Adversarial Networks (GANs) . Following, we provide an overview of existing applications of conditional design generation for the field marketing. Finally, we present recent innovations and important papers in the area of text detection, extraction and recognition, providing a theoretical background and listing commonly used features for text detection. The assessment and comprehension of these concepts is crucial for the development of the current work, as they justify the choice of the pretrained text detecting module and text recognition module.

2.1 Conditional Image Generation

Goodfellow's introduction of generative adversarial networks [21] in 2014 allowed for the generation of low-resolution image samples, with little to no control of the trained network output. Conditional models are a type of generative models, meaning models who take a training set with a predefined distribution and which learn to represent an estimate of this distribution.



(A) GAN and Design Generation GAN hits per year

(B) cGAN and Design Generation cGAN hits per year

FIGURE 2.1: Google Scholar result count on GAN, cGAN and Design Generation based on the search queries shown in Section 2.1.1

With the insurgence of conditional GANs, proposed by Mirza et al. [22], it is possible to condition the data generation process based on, for example, data classes. The recent improvements in both high-resolution image generation and conditional generative algorithms allowed for a substantial rise of interest in GANs and its applications to other fields like aviation [23], medicine [24], communication [25], image manipulation [26, 27], amongst others.

The possibility of generating high-quality graphics according to given conditions has led to an increase of interest in design generation since 2017. The conditions applied to the generative adversarial network provide us with a way to communicate the type and placement of features according to the used training dataset.

Depicted in Figure 2.1, we can observe the prominence of each subtopic over the years, measured by the number of keyword hits in Google Scholar. Figure A shows the google scholar results for queries 1 and 2 and Figure B shows the results of queries 3 and 4 of Section 2.1.1. We can conclude that the conditional GAN subtopic is in upsurge, along with its application to design generation.

2.1.1 Materials and Methods

This Survey was conducted on Scopus search and Google Scholar databases, where published English-written articles in the last 5 years (2015 to 2019) were analysed

and categorised as relevant or not relevant for our problem and case study. Papers with less than eight per-year citations were disregarded for this SoTA review. These results were based on four queries:

1. ("GAN" OR "Generative Adversarial Net" OR "Generative Adversarial Network")
2. ("GAN" OR "Generative Adversarial Net" OR " Generative Adversarial Network") AND "design" AND ("generate" or "generation")
3. ("CGAN" OR "Conditional Generative Adversarial Net" OR "Conditional Generative Adversarial Network")
4. ("CGAN" OR "Conditional Generative Adversarial Net" OR "Conditional Generative Adversarial Network") AND "design" AND ("generate" or "generation")

A more in-depth analysis was performed to the queries 3, with 580 results on Scopus and query 4, whose search string resulted in 48 results. All results were analysed and categorised according to its relevance to the investigation. The research summary process consisted of categorising the found articles according to the methodology, techniques used and results found. The most relevant articles were scrutinised and compared in this state-of-the-art review.

2.1.2 Generative Adversarial Networks

Generative Adversarial Networks, introduced by Goodfellow et al. [21] in 2014, are an example of generative models. Generative models, such as the GAN, take as input a set of images taken from the same distribution and outputs a probability distribution which represents an estimate of the given distribution. This allows generation models to excel at either or both tasks of image data density estimation and image sample generation. Even though GANs can perform both tasks, they were primarily designed to tackle the task of sample generation, where they have been proven to outperform all previously proposed generative models. According to Goodfellow [28], there are two main types of generative models:

- Explicit density generative models: These models define an explicit density function for the data, using the computed density function to calculate the gradient and potentially generate an image sample. However, creating a model that can capture the complexity of the data in a computationally feasible manner is a challenge;
- Implicit density generative models: These models don't require defining a density function. The training of these models usually corresponds to indirectly sampling from the found model.

GANs are implicit density models, created as an alternative to the Markov chain implicit density models, which required multiple steps with a high computation cost in order to generate an image. GANs are still the only implementation of generative models, which does not need an implicit density function and that can generate an image in a single pass through the network. GANs are constituted of two main components: a generator and a discriminator that work as adversaries in the network architecture. The discriminator's function in the network is to classify an input image as being drawn from the wanted distribution (real) or not (fake). The generator's job is to learn how to create images that are, during training, increasingly more convincing, and that can fool the discriminator into thinking the generated image is indeed drawn from the original distribution. The discriminator takes both real data and the generator's data as input, learning to classify the images as real or fake by minimizing its loss function. When training a GAN architecture, both generator and discriminator are trained independently in the following way [29]:

1. The discriminator trains for one or more epochs.
2. The generator trains for one or more epochs.
3. Repeat steps 1 and 2 to continue to train the generator and discriminator networks.

The separate training allows for the GAN to converge, as the different learning rates between generator and discriminator would make for a very difficult if not

impossible conversion. As the generator gets better, the discriminator's performance drops until it cannot tell the difference between real and fake images. As the discriminator's performance improves, the task of generating convincing images that can fool the discriminator gets more problematic, which leads to the need for the generator to improve its performance. The game between an increasingly better generator and an increasingly better discriminator is what makes the network learn how to create more convincing images.

2.1.3 GAN Variations for Image Content Generation Tasks

Since the proposal of the basis GAN model by Goodfellow et al. [21], many GAN architecture variations have been suggested, specialized to tackle other problems. In this section, we provide an overview of significant GAN variations and its usage in literature.

DCGAN: Radford et al. [30] propose a deep implementation of a GAN, where the generator is composed of several upsampling convolutional layers that do not use either max pooling or fully-connected layers. The discriminator uses regular convolutional layers to classify the images as real or fake. Batch normalization is applied to both the generator and the discriminator.

Conditional GAN (cGAN): Mirza and Osindero's conditional GAN [22] introduce a way to control the generation process by conditioning the model with additional information. The generator learns to generate images with the desired characteristics. The authors suggest feeding conditional information to both the generator and discriminator. The generator gets random noise and conditional information as input. The discriminator gets as inputs real and fake data, along with the conditional information.

Progressive Growing GAN (PGGAN: Karras et al.'s progressive implementation of GANs [31] has shown improvements in the system training stability, training time and image variation for high-resolution images. Progressive growing GANs start training with small 4x4 images and really shallow networks until they converge. The number of layers and image size are progressively increased during

training for both networks until we achieve the desired resolution. This approach results in a more stable network and quicker network conversion as early layers converge very fast. Since only a few layers are trained at a time, the training time is significantly reduced.

Image-to-Image Translation: Isola et al. propose a general framework for image-to-image translation problems [32]. The general architecture is based on the architecture proposed of DCGANs [30], using both the generator and discriminator convolutional modules with batch normalization. The generator follows the general U-net shape, used for image segmentation, along with skip connections between layers, in order to prevent the bottleneck resulting from upsampling an image through the network from a downsampled representation. The network learns how to generate an image based on the lower dimensional representation of the image associated with the bottleneck, along with information from previous layers, which included a less downsampled input image representation. The discriminator architecture works by classifying NxN image patches as real or fake, instead of the entirety of the picture. This allows for the discriminator to be faster and able to better penalize pictures whose details are not very sharp.

2.1.4 Challenges with GANs

Despite the positive results documented in the past 5 years, there are many challenges inherent to GANs. According to literature, the most significant problems that may arise when implementing a generative solution are:

1. Amount of data required for training: Most successful implementations of generative image models existing in literature had as basis existing datasets with a substantial dimension, enough to create a natural image manifold that we can use to generate new images. Getting a fit dataset for the problem is the basis for a GAN, as the image quality and diversity will determine how good the final outcome will turn out to be [33]. As examples, (1) the CelebA dataset used by the works in [31] and [34] contains 200,000 images, labelled with 40 binary attributes; (2) the CIFAR-10, with 60,000 images

labelled in 10 classes, used by [31]; (3) the CityScapes Dataset, with 25.000 annotated images, used by [27]; (4) MS COCO, with 330,000 object images, over 2/3 of them labelled, used by [35]; amongst others. As we can observe, the datasets used for the proof of concept of GAN architectures are very large and adapted to the specific task.

2. Failure to Converge: The training progress of a GAN can create situations where the network fails to converge. As training progresses, the generator and discriminator become increasingly better, becoming the feedback less and less meaningful over time. If the network is trained over the point that the discriminator is giving random feedback, we will be feeding the generator with feedback with no real information, originating the degradation of quality of the generator [29].

3. Vanishing Gradients or Saturation Problem: This problem occurs when the gradient used for the generator to train is diminished to the point where the generator cannot learn from it anymore [33]. Goodfellow et al. [21] state that it is caused by overtraining the discriminator to the point that it can reject generator images with high confidence. This leads to the vanishing of the gradient passed from the discriminator to the generator, making it impossible for the generator to learn from feedback.

4. Mode Collapse: During training it can happen that the generator, instead of generalizing the problem and creating a spatial natural image manifold, from which to generate new and distinct samples, it collapses into a much lower dimension space, failing to generalize the problem and generating the same set of sample images every time.

5. Evaluation: Evaluation for model benchmark and selection can be convoluted, as model evaluation though a calculated metric may not well represent the validity of the generated images. Manual evaluation implies an increased cost and time needed to gather the results.

Despite the research question requiring the generation of a customized business card for the end user, similar to the work by Isola et al [32] Image-to-Image

Translation, we believe the application of a generative model is not the appropriate approach for the problem for two reasons. Firstly, business card images contain a lot of noise that we do not want a generative model to learn from. Namely, text in different positions, which varies according to the designer choice (some business cards have text on the bottom, others don't, some have double column text, others have centered text), company names in large fonts, occupying a big portion of the business card and logotypes, which should differ according to the user. The presence of so much noise from which we do not want to learn from will negatively impact a model and lead to unwanted results. This problem leads us to a second related problem, which is bound to the question "which features would be useful to learn from the business cards?". Considering the text and logos are always different depending on the user, there are very little features worth learning from business cards in order to generate unseen samples.

2.2 Design Generation Systems

2.2.1 Materials and Methods

We have performed a systematic search in the Scopus database on English-written published papers related to Design Generation systems. Only articles and conference papers have been considered for this review. All the results were analysed and categorised according to its relevance and similarity to the problem and research question of the present work. These results were based on two queries performed on titles, abstracts and keywords:

1. ("Automatic" AND "Design" AND (generation OR personalization) AND "marketing")
2. ("Automatic Design" AND (generation OR personalization))

These queries were the starting point for the paper selection. The results were exported to excel and further analysed and refined.

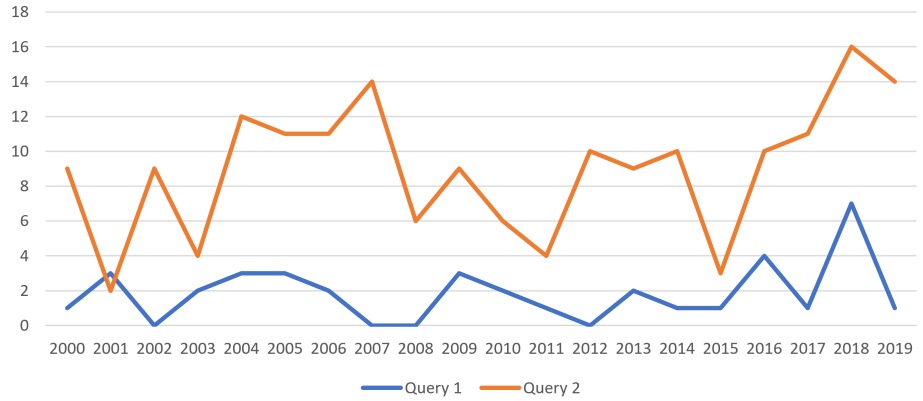


FIGURE 2.2: Number of topic hits per year on Scopus Search. Information from 14 December 2019.

The first query attempts to find any published works that relate design generation to marketing and retrieved 42 documents. The second query attempts to find any work on design automation, without considering the scope of marketing. This query retrieved 239 documents. The number of papers per year since the year 2000 is depicted in Figure 2.2. We performed a manual analysis on all gathered documents, in order to find similar works. All works retrieved from the first query had the abstract scrutinised in order to eliminate all works that are not relevant for the present problem. Due to the fact that the second query is more permissive, the results from this query had two stages of elimination. Firstly all titles were analysed and the works with no relevant connection the the current problem were eliminated. Following, we analysed all paper abstracts, removing works with no connection to the current problem. After manual evaluating the results, two papers, one of which result to both queries, have been selected.

2.2.2 Similar Works

Liang et al. [36] propose the automatic generation of textual advertisement for video advertising. The idea is to automatically generate textual advertisement and insert it into video without occluding important video information and maintaining contrast between text and the video background, so that the text is legible. The position of text takes into consideration the multiple frames of video, in order to

pick a candidate region where it least occludes important video information. Even though this work combines image and text for advertising.

Jahanian et al. [37] propose a recommendation system for automatic design of magazine covers for non-designer users. The proposed system takes as input from the user a design style from a predefined list. The design style is then used to generate a colour palette for the magazine cover, by relating certain colours to a certain design style. The authors cite the work of Kobayashi [38], which further justifies the style-colour relationship. The found colour palette is then used to pick a background image from the system user image gallery. The authors then compute candidate regions for text placement, meaning image regions that are "less busy" relative to the rest of the image. The colour of the text is picked according to the background image colour in the text candidate region, in order to contrast with the background. The design font was set to Helvetica in every design.

2.3 Text Detection, Extraction and Recognition

Over the past decade, extracting text from images has been a widely studied subject, with major breakthroughs in what was considered state-of-the-art systems. Sahare et al. [39] define text extraction or text segmentation as the process of separating text from images. Two key subtopics are explored with regard to text extraction systems: text extraction from documents and text extraction from natural images, the latter being considered a more prevalent topic, particularly with the proliferation of smartphones that enable people to quickly capture digital images.

We can analyse the prominence of each subtopic over the years in Figure 2.3, measured by the number of keyword hits in Google scholar. Despite the extra complexity of detecting text in natural images, we can see a steady increase in interest in this subtopic, which will be the focus of this state-of-the-art review. Due to the variation in the complexity of images, each having a very different set of features and amount of noise, the problem of recognizing text in natural images

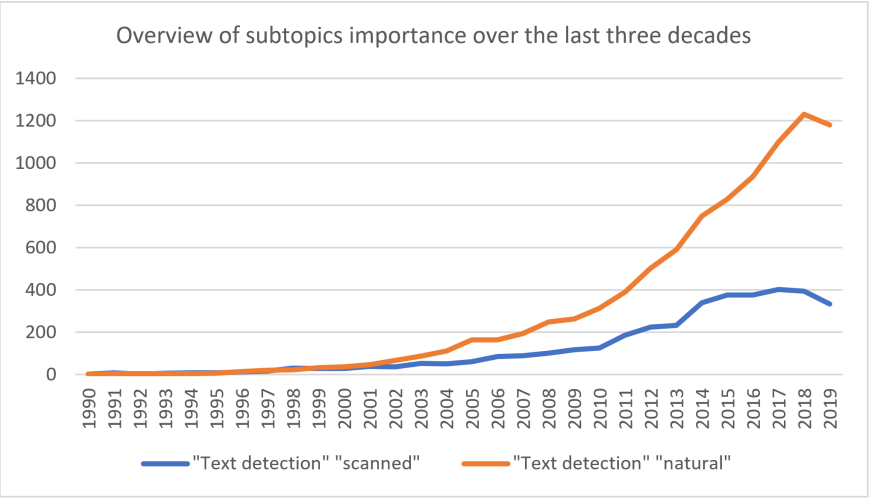


FIGURE 2.3: Number of topic hits per year text detection on google scholar. Information from 14 December 2019.

has been found to be much more difficult to solve. Such features transform it into a convoluted problem, adding complex backgrounds and textures, distant, slanted or perspective text, various light conditions and multiple fonts.

There are two important tasks on a text detection system: text extraction and text recognition. It is necessary to find the text in the picture before interpreting it [40], as it can happen in natural images that the larger part of the image is non-text. This is considered to be a much more complex task than text recognition itself [41] as text can blend with backgrounds and many non-text slices of images can very easily be mistaken and classified as text. Text detection and text recognition are crucial for processing natural images in a fully functional, end-to-end system [41]. For this reason, text detection systems often comprise several different submodules working in serial, where each submodule is responsible for a single task in the workflow. Tian, S. et al. [42] identify four main steps typically comprised in said systems: character candidate detection, false character candidate removal, text line extraction and text line verification. These steps are, in one way or another, always present in end-to-end text detection systems, albeit modern approaches comprise more complex approaches using different types of neural networks, where the boundaries of said steps are often fuzzy.

The image is first preprocessed in order to prepare it for text detection. This step takes the original image and transforms it using a set of defined rules that

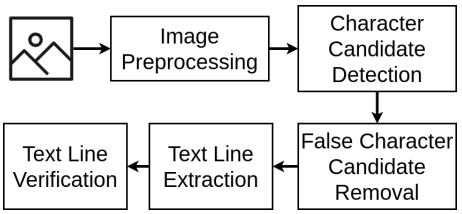


FIGURE 2.4: Text detection pipeline, according to [42]

ideally will increase the text extraction precision, making it easier to differentiate between text and non-text. After preprocessing, the system attempts to detect potential characters by going through the image and classifying portions of it as text or non-text. A common approach to this step is a sliding window method, as described by Wang et al. [43]. The result of the previous step is run though a false character candidate removal, which suppresses misclassified windows that either are repeated characters (where the same character is discovered twice in a row) or are non-character. The candidate can be classified as a non-character when, for example, as Epshtein, B. et all state [44]: (1) The component size is too small or too large; (2) There are components surrounding text (any shape surrounding text should not be considered); (3) The component has an unusual aspect ratio that is usually not associated with text (for example, the component is very wide or very tall). Following the character extraction, the letters are grouped together, which is considered to be a significant step in further reducing noise and character false detections, as single letters usually do not appear in images. The gap between characters, character styles (character width and height and stroke width) often give us an idea of when the characters should be banded together [44]. The method culminates in a validation of the text lines, where an OCR algorithm is run on the identified candidate words, extracting and analysing the results. One of the main problems these systems are said to have is that since the steps are sequential, the system pipeline leads to a gradual error accumulation in each step [42], which can hinder the overall system performance. Especially considering that, if we analyse this process with a traditional ceiling analysis method, we will most likely discover that most of the error comes from the character candidate detection unit, which is located at an earlier stage of the chain and whose error will propagate to the subsequent steps, where it will be amplified.

2.3.1 Features Used for Natural Image Text Extraction

Various features have been explored by different investigators to identify text in natural images, not existing a consensus to the best possible approach to the problem. In this section, we will be exploring these features, referencing to the original papers.

Edge: Edge information has been one of the most common features used in works related to text extraction. An edge detector is used to compute the edge locations in the image, as explored by Cho et al. [45], Epshtein et al. [44] and Huang et al. [46]. Some limitations have been pointed out to methods using text extraction with edge features alone, mainly because the edges are sensible to more complicated scenarios, such as multiple connected characters, segmented stroke characters and non-uniform illumination [47], which do often occur in natural images.

Texture-based features: Textures can provide us with valuable information about the various elements in an image. Since the text in real images tends to be contained in well-defined regions, characterized by texture and having the text itself a different texture, researchers have used this knowledge about contrasting textures to set a classifier between text and non-text. According to Grover, S. [48], there are two approaches to texture-based text segmentation: pixel-based and block-based. The pixel-based approach computes the probability of each pixel being part of text based on the neighbouring pixel textures. The author states that the problem with this approach is setting an appropriate threshold for the classifier. The block-based approach splits the image into blocks with, for example, a sliding window, where each block is labelled as text or non-text based on the textural information contained in it.

Colour: Methods based on the colour feature assume that the pixels of each character have a similar colour and that pixels belonging to characters can be segmented from the background by colour clustering, as explored in the work by Yan and Gao [49]

Connected components: As stated by Liu, Shen and Wang [41], connected component mixes several low-level properties, like the gradient, stroke width, colour, amongst others, in order to discover existing components in the picture. For two adjacent letter candidates, if they share similar properties, it is very possible that those should be connected components. Koo and Kim [50] define adjacent as components classified as text that are no more than two characters apart from each other.

Stroke: Epshtein et al.'s approach to text classification commends the stroke as a critical feature for the task [44]. The authors state that the almost constant stroke width is a feature which distinguishes text from other components, allowing regions that might contain text to be discovered based on the presence of such strokes. According to the authors, stroke information can also be useful for text-line extraction, grouping neighbouring components together that have a similar stroke width, in order to form words.

2.3.2 State of the Art for Text Detection

Due to the similarities of our problem with the well-researched topic of focused scene text, we have decided to evaluate methods designed to work for this particular problem. According to Karatzas et al. [51], focused scene text refers to images especially focused around the text content of interest, where the user explicitly directs focus of the camera to the element which contains the text of interest.

The ICDAR Robust Reading Competition, has since 2013 been updated with the results of new state of the art detectors on the robust reading dataset [52]. We have compared the results from the text location task, deciding on the method with the highest recall score. We only consider methods that have been featured in published papers. For this reason, we have opted for the Craft++ text detector [53], with a recall score of 94.36%, higher than the 116 other participants.

2.3.3 State of the Art for Text Recognition

We have followed the same method for selecting the state-of-the-art algorithms for text recognition that we have described in Section 2.3.2. Comparing the results for the Focused Scene Text competition for the task of word recognition, we have opted for the CLOVA-AI v2 [54], as it was the best performer out of the 28 compared methods, being the only method with a published paper.

Chapter 3

Framework Design and Development

In previous chapters, the concepts required for the architecture that is response to the studied problem were developed. In this chapter, we present and provide an overview of the business card design generation framework along with an overall description of the system architecture. The process of generating personalized designs for marketing begins with gathering features from the submitted natural image to be used as inputs to determine the design template and customisation made according to the client's needs.

3.1 System Functionalities and Constraints

Based on the studied problem, the following functionalities have been identified:

I System Input:The system must take as input a JPEG, PNG or BMP image.

II information Retrieval: The system must extract the necessary information from the uploaded business card in order to generate an alternative design for the client. Specifically, the necessary information for the system is:

(a) Textual elements present in the design

(b) Colour palette inferred from the business card

(c) Business type

III Text Categorization: The extracted textual elements must be categorised according to the portrayed information into one of the following categories: email, phone number, website, address, facebook, twitter.

IV Design Generation: The system must generate a personalised business card design based on information taken from the natural image. The personalized business card must contain a similar colour palette to the one in the submitted business card, along with a style fitted to the business type.

V Design Personalisation: The business card must be personalised with the extracted text from the submitted element.

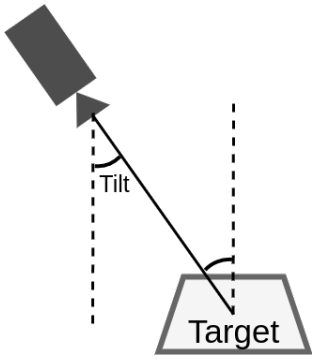


FIGURE 3.1: Camera tilt

Additionally, in order for the system to work, a few constraints must be met, regarding the image used as input:

I The business card must have a rectangular shape and horizontally oriented. Square business cards are also admitted;

II The business card must be the main object present in the image, occupying at least 40% of the image pixels;

III The business card must not be partially occluded. All four corners of the business card must be visible and in frame, in the submitted picture;

IV The image must not be submitted upside down or in uncommon rotations. Even though the system accounts for this scenario, rotations over 85º of its original orientation mean it becomes very difficult or even impossible to predict to which side the system should correct the rotation;

V The image must not have an exaggeratedly steep perspective. The tilt of the camera, depicted in Figure 3.1, must be kept under 40°, as increasing it any further will result in hard to read text and loss of details;

VI There must be a reasonable contrast between the background and the business card colours. The contour of the business card in pictures where the card has the same colour as the background might not be picked up.

3.2 System Architecture

Figure 3.2 depicts the proposed architecture for business card design, where the system is decomposed into work packages. The system is characterized by 6 individual modules, on which the input data (business card) is treated in order to extract information for the final module, responsible for design personalisation.

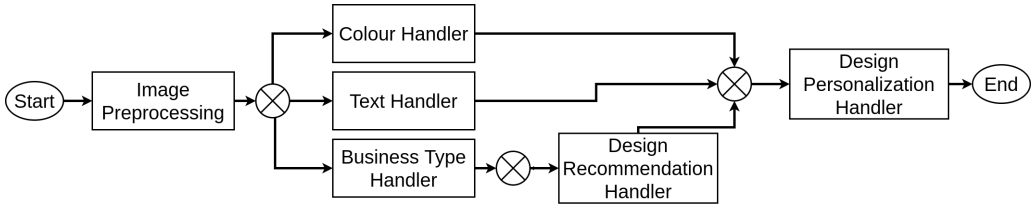


FIGURE 3.2: High-level System Architecture

Each system module corresponds to a unique detachable pipeline of atomic tasks that are executed in sequence to achieve the module's final objective/output. The inherent modularity of the system architecture poses some advantages, especially for future development and improvements to the system, for three main reasons:

1. It allows for easy ceiling analysis and independent testing each module, benchmarking them as a single software piece. This allows for a better understanding of where and how the system is failing and what the steps might be, in order to improve the overall system performance;

2. The system modules are hot-swappable. The system is decomposed into abstract function modules, each expecting specific inputs in order to generate

an output, but not depending on the remainder of the process in order to create these inputs. In other words, each of these works as a black box and, providing input and output formats are respected, these can be updated;

3. The system is expandable. At the moment, we propose a system that extracts text, colour and the business type from the input. Adding new components to the system can be done without changing the already implemented data extraction modules, updating only the subsequent system modules in order to handle additional input information.

As we are aware that the proposed system is an early development prototype, with much room for improvement, it is crucial to allow for easy improvements and system updates, which this architecture does. Examples of said improvements are described in Section 6.3, Future Work.

3.2.1 Components:

As shown in Figure 3.2, this dissertation's artifact will require the development of the following components:

1. Development of a model that segments the marketing element from the natural image, determining the polygonal contour of the business card, extracting and straightening it;

2. Main colours extraction model development;

3. Sentence merging, which transforms the individual words that have been detected into text boxes, grouping them into sentences;

4. Text categorization, which determines the meaning of each textual element in the printable, categorizing the elements according to what they represent;

5. Design selection and personalization, which selects an appropriate design and customizes it with colour, text and logos;

3.3 Module Specification

The modules presented in the previous chapter are themselves constituted of either a unique component or a pipeline of task components, that allow for them to transform the input data.

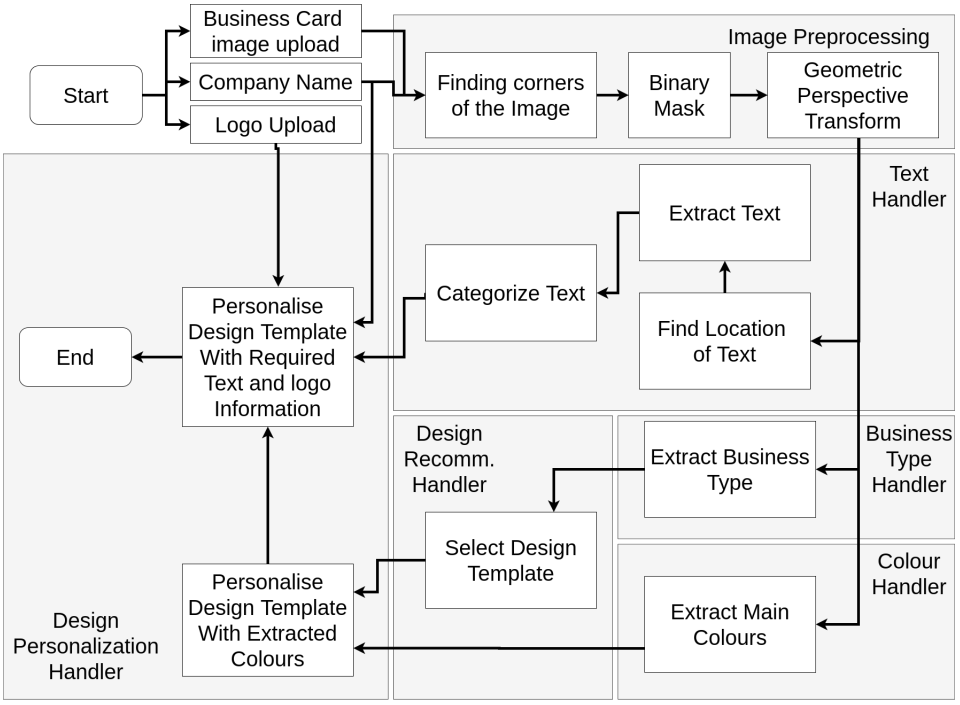


FIGURE 3.3: High-level Module Specification

In Figure 3.3, a more detailed system diagram is presented. The gray boxes correspond to the modules specified in Figure 3.2. This overview allows for a better understanding of the atomic tasks being performed in each module. The system starts with the upload of a business card image from the user. The image is preprocessed, removing the perspective of the card (angle between the camera and the surface where the business card is laying) and the background that is not relevant for the task, Section 4.1. The preprocessed image and company name then goes through the three main information extraction components: the text handler, the colour handler and the business type handler. The dominant colours are extracted using a K-means further explained in Section 4.2. In order to extract textual information, firstly textual regions of interest are established, the text is extracted and categorized according to what each expression represents, Section 4.3. Finally, the business type is identified and extracted.

Taking into consideration the business type, a recommendation for a design template is selected from the available on the system. This template is then personalized with the colours previously extracted and the text coming from the text handler module. A customized image of an alternative design for the business card, based on the submitted design, is the output of the system.

3.3.1 Image Preprocessing: Unskewing and Background Removal

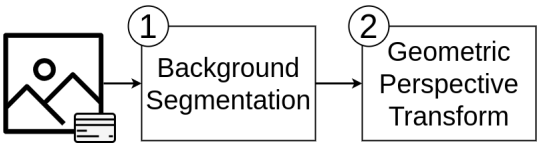


FIGURE 3.4: Overview of the Image Preprocessing Module

In this subsection, we present the component 1 of the system. In order to prepare the image, we perform two consecutive steps, as seen in Figure 3.4: First, (1) we attempt to segment the background from the card present in the image, discarding all pixel information that is not part of the business card. This is performed by first finding the four corners of the business card and applying a binary mask to the image. The pipeline used for background segmentation is further described in Subsection 4.1.1. Following this step, (2) we apply a geometric transformation in order to modify the perspective in which the business card is seen, straightening it into a rectangular image, as it would be seen if the camera tilt, described in better detail in Section 3.1, was zero. Details are further explained in Subsection 3.3.1.1 and 3.3.1.2 and the implementation in Subsection 4.1.

3.3.1.1 Background Segmentation

In order to segment the background from the object, we must find the location of the object in the image. Since the object we are dealing with has the shape of a trapezoid (due to the possible perspective slant), we can think of the shape being defined by four non collinear points. Once we find the four points that correspond

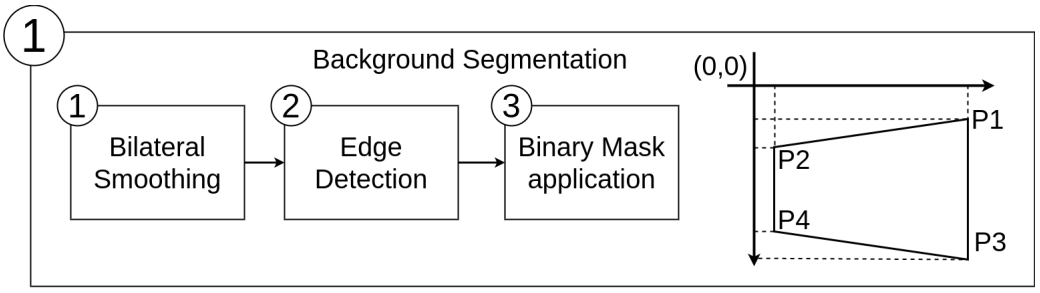


FIGURE 3.5: Overview of the Background Segmentation Submodule

to the vertices of the object, we apply a binary mask in order to discard non relevant pixels. This pipeline can be seen in Figure 3.5.

For background segmentation, we first apply bilateral smoothing, an an edge-preserving noise-smoothing function, to remove excessive noise from the image. Following, we apply the Canny edge detection [55] to find the quadrilateral contour of the business card, further detailed in Chapter 4.1.1. Finally, we apply a binary mask, removing all background information from the business card.

3.3.1.2 Geometric Perspective Transform

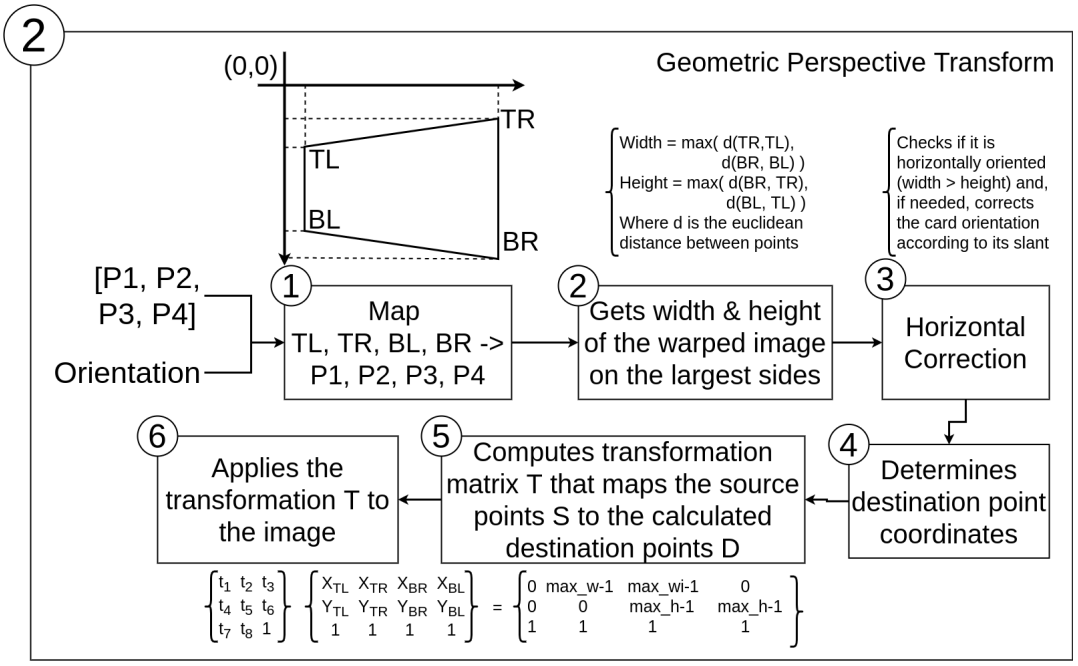


FIGURE 3.6: Geometric Perspective Transform Submodule

Considering we know the exact coordinates of the card vertexes and the angles between sides, also knowing the internal angles of a generic business card, 90º,

we can apply a geometric transformation to the original image in order to change the perspective of the business card from its original orientation, corresponding to a polygonal shape. This transformation would map this shape to a top-down view, as if there was a camera tilt of 0^{o}. This makes it so the resulting image is a rectangle. As input, this submodule takes the original image, along with the coordinates of the corner points detected by the submodule described in 3.3.1.1 and the default orientation of the business card, allowing the specification of a horizontal or vertical card orientation.

In order to transform the image, after knowing the four corners of the business card, we perform six consecutive steps, as seen in Figure 3.6: First, (1) we attempt to label the four found corners as top left, top right, bottom left or bottom right of the card, according to its location in the image, as described in 4.1.2.1. Then, we (2) calculate the width and height of the card, used as input information for the following two modules. This is further described in 4.1.2.2. Following, we apply an (3) orientation correction module, which uses information from the width and height values of the business card and its slant in the image to determine if the card is in the right orientation, also described in 4.1.2.2. Using the width and height information, we then (4) determine the destination coordinates of each corner point according to the aspect ratio of the business card, detailed in 4.1.2.3. In order to apply the geometric transformation we take the calculated destination point and source point coordinates and (5) attempt to find a transformation matrix that maps these two geometric spaces, as shown in 4.1.2.4. Finally, (6) we use the calculated matrix to apply a geometric transformation in order to modify the perspective in which the business card is seen. Details on implementation are provided in 4.1.2.4.

3.3.2 Colour Handler

Following the image preprocessing, and considering we no longer have background pixels in the image, we can now extract colour information. This section presents the implementation details of the component 2.

This module attempts to find a solution for the discovery of the main colours in the business card, when faced with an image encoded in the RGB colour space. As we are dealing with natural images, with non controlled lighting conditions, we face images where the business card is not evenly illuminated, resulting in patches of the same colour present in the business card seem lighter in some places (direct exposure or closer to the light source) and darker in others. Empirically, pixels representing the same colour may have slightly different values for r, g and b, meaning we must perform colour reduction in order to find and appropriate palette. The usefulness of the module explained in Chapter 3.3.1 is more evident in this module, as we only want to perform colour palette extraction on the pixels corresponding to the business card, disregarding any background information.

In order to find the most predominant colours we performed colour reduction operations, so that we can transform the multi-million colour space to a small subset of colours that most represent the input image. The colour reduction was performed by progressively grouping pixels that are close to each other in the colour space, until we have a colour palette of 5 colours. The colour reduction was performed by applying a k-means algorithm to the HSV colour space, with 5 initial centroids. The extracted colour palette corresponds to the HSV colour value of each found centroid, further explained in Section 4.2. The colours are then converted to the HSL colour space, where any colour with lightness values under 15 or over 85 is disregarded.

3.3.3 Text Handler

This section is related to the implementation details of the component 4, whose ultimate objective is to extract meaning from the textual information present in the business card image. Inherent to information extraction from text present in an image are two tasks that must be performed consecutively. First, described in Section 4.3.1, (1) we attempt to extract from pixel information, text present in the image. Then, as shown in Section 4.3.3, (2) we tag the extracted text appropriately, according to the information present in excerpts of text.

As seen in Chapter 2, the problem of extracting text from natural images is split up into two distinct thesis: text detection and text recognition. The first problem (1) corresponds to finding the exact location of text characters in the images, in order to remove all non-character pixels contained in the image which, for this module, are noise. The result of text detection is a heat map or bounding polygons, where the locations of characters are emphasized. The pixel information contained inside the found bounding polygons is then (2) used as input to a text recognition system, which attempts to determine the word contained in each bounding box by applying a character recognition model.

In Figure 3.7 we can observe, in high-level, the text extraction diagram, composed of four consecutive steps that must be performed prior to text tagging.

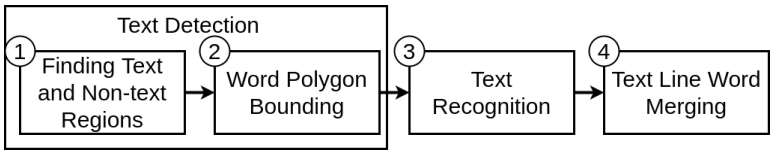


FIGURE 3.7: General Text extraction Flow Diagram.

Firstly, considering the submitted image is a matrix of pixels with no extra information regarding its content, we must determine in which of these pixels it is more likely for text to be present in. Ultimately, this corresponds to classifying portions of the base image as text or non-text via, for example, a sliding window approach. Image portions classified as having text must then be merged together and encompassed in the same polygon if they are close to each other, being the concept of closeness defined by a learned or predefined hyperparameter, varying according to the implementation. The found words, defined by a series of pixels encompassing a word inside the bounding polygon will then be used as input to the text recognition module, which will output the predicted word for each of the found regions. Finally, we must merge the found words into ordered text, able to be interpreted. This can be done by evaluating the position of the bounding polygon in the original image, where polygons that are close together and have the same orientation and a similar height, are recursively merged.

3.3.3.1 Text Detection and Text Recognition

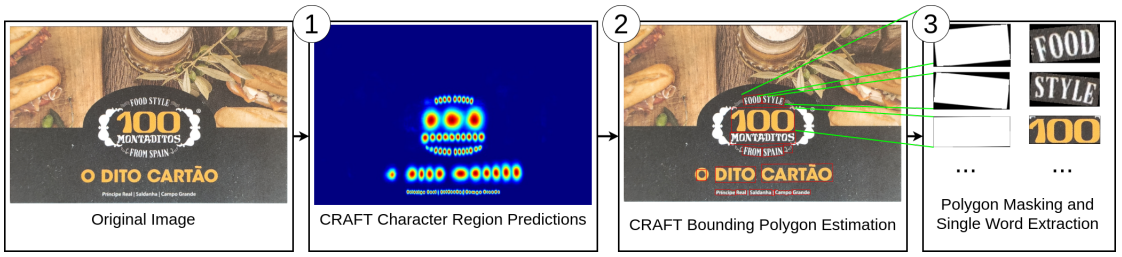


FIGURE 3.8: Text detection Flow Diagram.

In Figure 3.8, we can observe the data flow for the text handler that was implemented for the current project, using the text extraction diagram steps defined in Figure 3.7. The system applied the CRAFT text detector to the original business card image, extracting the heatmap of the character regions of interest (1). CRAFT then estimates the bounding polygons for each word (2). In this step, we also collect additional information bound to each of the found polygons position and size in the picture, to be used by the text merging submodule, explained in Subsection 3.3.3.2. Finally, we apply a binary mask to isolate individual words (3).

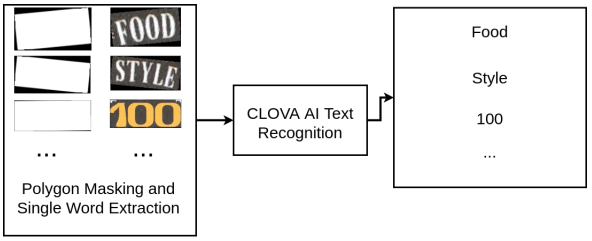


FIGURE 3.9: Text recognition Flow Diagram.

The text recognition module takes as input the pixel information inside each bounding polygon found by the CRAFT text detector, using the work published by CLOVA AI [54] to get the word present in each bounding polygon and the probability associated to it. Depicted in figure 3.9 are the inputs and outputs of this module.

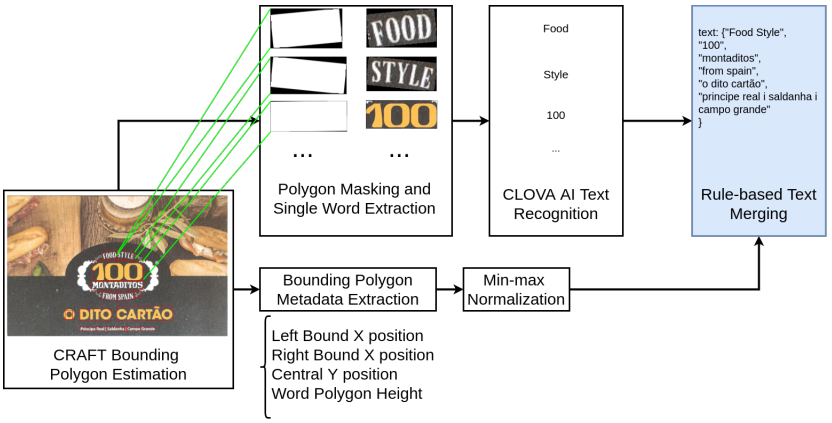


FIGURE 3.10: Text Merging Flow Diagram.

3.3.3.2 Text Merging

In this section we propose the creation of the component 3, a rule-based word merging module, capable of merging words into text boxes according to its position in the business card and style. The importance of merging words into text boxes comes from the fact that interpretability is required in order to extract information from the business card text. Text must be interpreted as full sentences to tag excerpts according to the information they portray.

In order to create this rule-based system, we have extracted features off of each bounding polygon, saving in json format location and size attributes related to each word, as seen in Figure 3.10. The extracted features from each bounding box are the left bound X coordinate, the right bound X coordinate, the Y coordinate of the bounding box centroid and the word polygon height. We also use the top bound y coordinate and bottom bound y coordinates, which are calculated using the centroid y coordinate and half of the word polygon height.

Since we accept all kinds of image sizes and aspect ratios, we cannot work with absolute values, as the same value of bounding box distances can represent text far away or close, depending on the size and aspect ratio of the original card. The coordinate values are all min-max normalized between 0 and 1, so that distances are comparable between different sized cards and we can manually tune hyperparameters to work on most cases.

The algorithm recursively iterates over all words in the dataset, sequentially merging words into sentences if they are close enough to each other, according to a predefined threshold, for the x and y directions. The words are only merged if they have similar-sized fonts, defined by the bounding box y size.

In Subsection 4.3.2, we provide additional details on how the word merging rules are applied.

3.3.3.3 Text Categorization

We propose a text categorization system that can evaluate the text expressions found by the submodule 3.3.3.2 and categorize it into the type of information portrayed. This module attempts to find social network information, namely facebook and twitter, as well as websites, addresses, emails and phone numbers. In Figure 3.11, we can visualize the text tagging flow diagram performed for each text box found and the output json containing the information found on every text box.

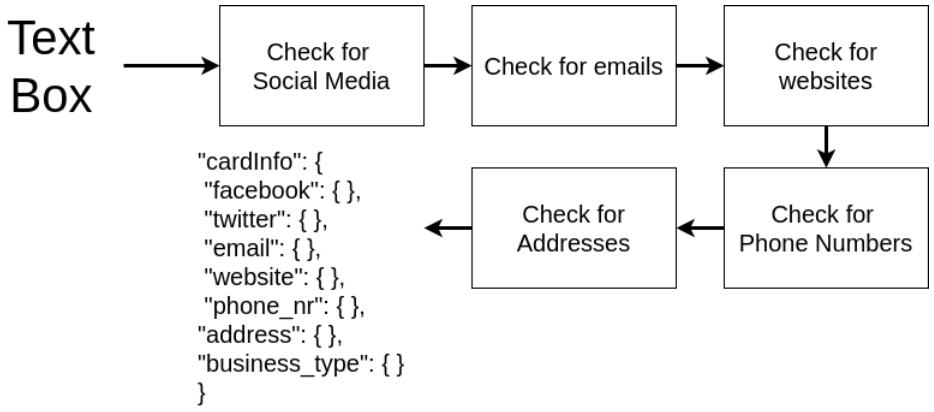


FIGURE 3.11: Text Tagging Flow Diagram.

We applied regular expressions for each text tag, that can evaluate if the text excerpt contains a specific tag. The address verification does not use regular expressions. Instead, we search text for indicator words that indicate that the evaluated text string is an address. These indicators include state names, cities and municipalities and other cues, such as the words "street", "rua", "st.", "road", "rd.", amongst others. Additional implementation information regarding the regular expressions and word indicators are provided in Subsection 4.3.3.

3.3.4 Design Personalisation Handler

The final module of the system, component 5, takes in all gathered information and generates a business card, personalized according to the user's business card information. This module takes as inputs the name of the business and logo (submitted by the user, as this information would be very hard to extract from a business card), the colour palette found, the categorized text and the business type.

Business Type Identification: We employ the Google Places service, available through API call, in order to identify the type of business in the business card. Since there can be many businesses with a similar name worldwide, we utilise geolocation information to search for nearby places, in a radius of under 40Km, with the name requested. This service outputs a json which includes the business types related to a particular business, in order of relevance.

The system first selects one of the available design styles according to the type of business of the user. The design style is adapted, meaning the templates are distinct between restaurants, retail stores, hotels, etc. The design is then customized with the colour palette extracted from the submitted business card image, being the original colours reflected in the new design. Finally, the system customizes the template with the business name and logotype, as well as all the information found in the original business card (social networks, email, address, website and phone number).

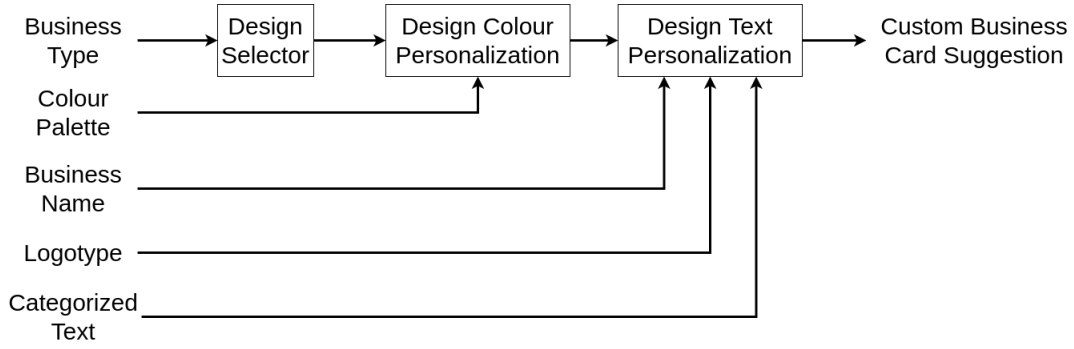


FIGURE 3.12: Design Personalization Flow Diagram.

The output of the module is a suggestion of a custom business card, generated from the original submitted image, as can be observed in Figure 3.12.

Chapter 4

Module Development Specification and Concepts

In this chapter we provide the implementation details for the dissertation artifacts and developed working packages listed in Section 1.4 and referenced in the Chapter 3, in order to address the problem of personalized business card generation. This chapter is structured following the system data flow. In Section 4.1, we provide the implementation details of the image preprocessing module. Following, the colour handler and text handler are each described in detail in Sections 4.2 and 4.3 respectively. Finally, in Section 4.4, the template personalization process is detailed.

4.1 Image Preprocessing: Unskewing and Background Removal

In this section, we will detail the implementation of the system artifact 1. In the architecture described in Chapter 3, we propose performing an image preprocessing step, which takes the original submitted natural image, turning it into a visual representation of the business card present in the image, without the presence of the background. It is advantageous to perform this step as all information contained in the background is considered unnecessary for the proposed task. All

information required to extract in order to generate a new design of a business card, as explained in Section 3.1, is contained in the card itself.

4.1.1 Background Segmentation

Two approaches were explored in order to find the corners of the business card: based on corner detection algorithms and based on edge detection algorithms. The results of each of the tested algorithms, which further justify the choice made are detailed in Chapter 5.

The first tested methods corresponded to the employment of corner detectors to the natural images, as they are rotation invariant and can detect changes of intensity when they occur in the x and y directions. We have tested the Harris corner detector, as proposed by Chris Harris [56] and the Shi & Tomasi's Good features to track (GFTT) detector [57]. In the tests conducted, further described in Chapter 5, we can observe the Harris corner detector had overall better results in the task at hand, managing to detect the corners in more instances than GFTT However, this has proven not to bee a good approach in the tests conducted for two major reasons.

Firstly, due to the nature of these printables, the contrast between background and its elements are often very high in order to increase readability. This means black text on white background is commonly seen and so are sharp symbols, logos and images. Most of the times, these have a more significant change of intensity when the Harris window shifts in x and y directions, than the pixels that make up the corners of the picture. In the Figure 4.1, we can observe that the corners and edges are much sharper on text than they are on the business card corners.

Since business cards are mostly constituted of text, it was observed that these algorithms easily picked up on the corners corresponding to each letter, but often missed the corners of the business card itself. Secondly, since text in itself corresponds to multiple individual letters that should have a very high intensity change when compared to the background, there are many corner points in each of the

FIGURE 4.1: Corner point pixel information

letters, which adds a lot of noise when trying to identify which of these are the true corners of the card.

One major feature that distinguishes the corner points that constitute the contents of the business card from the actual corner points of the business card are the edges associated to these specific corner points. Whilst the content of the business card is associated to short strong edges, the corner points should be associated to four much longer edges in the form of a quadrilateral. Formally, each intercept point C, $C(x, y) = (E_h \cap E_v)$, where E_h is a horizontal edge and E_v is a vertical edge will correspond to one of the vertexes. Thus, we employ edge information in order to find the corners of the image.

Since algorithmic speed is not a major concern for our problem, we will not utilise satisfying metrics, having only one optimising metric, accuracy, defined by the similarity between the resulting binary mask and the ground truth. We apply an edge preserving smoothing function, in our case we opted for bilateral smoothing, as it smoothes image noise, preserving edge information better than other blur functions, such as gaussian blur, as explained by Tomasi [58]. Following, we apply the Canny edge detection, as described in the original paper by John

F. Canny [55] to detect the most prominent edges in the image. We then take the edge map and attempt to find contours defined by these edges. A contour is defined by a closed polygon with values superior to 0 in every pixel of the edges that define the polygon in the edge map. We store every contour polygon in the nodes of a hierarchical tree structure, where the root tree node C_0^0 corresponds to the entire image. Each child node corresponds to a contour that is encompassed inside the parent polygon. A node can have several child subtrees with variable depth.

As previously stated, the polygon that defines the business card should be the contour with the largest area present in the picture after the root node, as we assume that the business card is the largest object present in the image. This means the contour corresponding to the card will be in depth 1 of the tree. If there are more than one contour in this level (which can happen in busy backgrounds or when other objects are also in frame, we assume the largest contour defined by four points (quadrilateral) to be the one that defines the card, as stated in Section 3.1.

Following, we apply a binary mask to the image, corresponding to the found quadrilateral contour. Every pixel information on the inside of the quadrilateral is kept. The remaining pixel values are discarded.

4.1.2 Geometric Perspective Transform

4.1.2.1 Coordinate Mapping

The system starts by mapping the coordinates of the four points to the corner locations top left (TL) , top right (TR) , bottom left (BL) , bottom right (BR) . If TL = (X_{TL}, Y_{TL}), TR = (X_{TR}, Y_{TR}), BL = (X_{BL}, Y_{BL}), BR = (X_{BR}, Y_{BR}), and considering that the image is represented on a Cartesian coordinate system as depicted in figure 3.6, the top right corner will be the one whose sum of X and Y coordinates is minimum. Formally, $X_{TL} + Y_{TL} < \min((X_{TR} + Y_{TR}), (X_{BL} + Y_{BL}), (X_{BR} + Y_{BR}))$. Using the same logic, the bottom right corner is the one whose

sum of arguments is maximum, $X_{BR} + Y_{BR} < \min((X_{TR} + Y_{TR}), (X_{BL} + Y_{BL}), (X_{TL} + Y_{TL}))$.

In order to determine the TR and BL corners, we apply a similar method. We compute the difference between the X and Y values of every point and the point whose difference is larger will correspond to the TR corner, since $X_{TR} > Y_{TR}$ and $X_{BL} < Y_{BL}$.

4.1.2.2 Calculating the Width and Height and Horizontal Correction

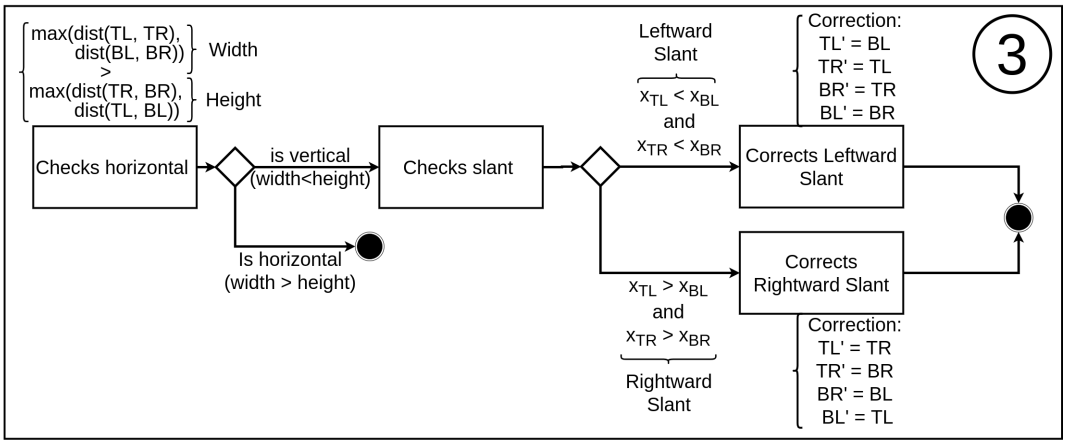


FIGURE 4.2: Horizontal Orientation Correction: Corrects the figure to be horizontally oriented if the previous module picked it up as being vertically oriented.

We then evaluate if the points given label is according to what a horizontal business card would look like. In other words, a horizontal business card would have a larger width value than height value. In order to calculate this from our image, we take the coordinates of the points and measure the distance between corners of the business card. A horizontal card would hold the value of True for the following expression: max(dist(TL, TR), dist(BL, BR)) > max(dist(TR, BR), dist(TL, BL)). If this is not verified, it means the business card present in the image is slanted further than 45°, as depicted in Figure 4.2, or is indeed a vertical business card, which is currently not accepted by our system and included in the Section 3.1.

In case we detect the card is rotated, two distinct situations can happen: a left or right slant. In order to rotate it accordingly, we assess which of these is

the case, by comparing the x coordinates of the corner points. If the BL corner is to the right of the TL corner and the BR corner is to the right of the TR corner (the values of the x coordinate of BL and BR must be higher than the values of x in TL and TR respectively), then we know for sure there is a leftward slant. If none of those is true, then we are facing a rightward slant. Formally, if $X_{TL} - X_{BL} < 0 \wedge X_{TR} - X_{BR} < 0$, we are facing a leftward slant. If on the other hand $X_{TL} - X_{BL} > 0 \wedge X_{TR} - X_{BR} > 0$ we are facing a rightward slant.

In very rare occasions it can happen that one of the conditions is true and the other one is false, when we are facing card rotations over 80° and a very high camera tilt. In this case, our system will fail to adjust the business card orientation, which is a scenario that has been accounted for in the system requirements, described in Section 3.1. The points are then remapped in order to rotate the card to the horizontal position. In case of a leftward slant, the corner tags rotate counterclockwise and in case of a rightward slant, the tags rotate clockwise, as can be seen in figure 4.3.

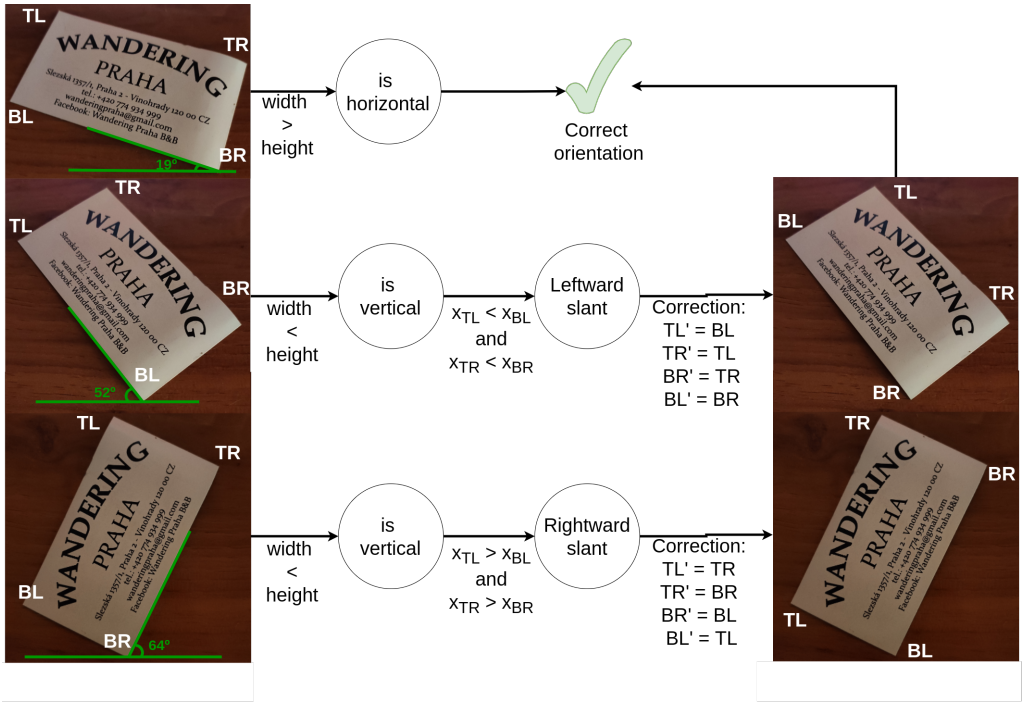


FIGURE 4.3: Applied example of the Horizontal Orientation Correction submodule.

4.1.2.3 Computing Destination Point Coordinates

In order to apply a perspective warp transform to map a trapezoid to a rectangle, we must first determine the exact coordinates where the corners will be located in the resulting image. Considering we want to remove all the pixels outside the bound delimited by the four corner points, also rotating the card so the top left corner coincides with the top left corner of the resulting image, it makes sense that the destination coordinate of TL will be the point 0,0. Having calculated the max width and max height of the business card in 4.1.2.2, we can infer the exact destination coordinates, represented by matrix D (destination), into which we want to warp our image corners, represented by matrix S (source). Our coordinate mapping will correspond to the following:

$$S \longrightarrow D = \begin{bmatrix} X_{TL} & Y_{TL} \\ X_{TR} & Y_{TR} \\ X_{BL} & Y_{BL} \\ X_{BR} & Y_{BR} \end{bmatrix} \longrightarrow \begin{bmatrix} 0 & 0 \\ max\_width - 1 & 0 \\ max\_width - 1 & max\_height - 1 \\ 0 & max_height - 1 \end{bmatrix}$$

4.1.2.4 Computing Transformation Matrix and Applying the Perspective Warp

We can think of the business card picture as a perspective projection on a 2d plane of an object in 3D space. The four 2D coordinates correspond to a projection of the points in 3D space to a plane, being homogenous coordinates of the corresponding point in the world, per definition of homography. For this reason, we can apply a transformation in order to change the perspective of the object to the desired representation. Additionally, we know that a non linear transformation will be required for this task, as the purpose of the task is to transform lines that are not parallel in the current image, into parallel lines (business card sides), due to the empirical knowledge we have on the depicted object.

We must now find the transformation matrix T, which can achieve the transformation described in 4.1.2.3, mapping every point p from matrix S, to a point

q, present in matrix D. In other words, we must find the matrix T that satisfies the condition q = T*p, let p and q be the coordinate vectors of two points in the source and destination image.

Considering we know the exact coordinate vectors of the source and destination images, we employ the concept of projective transformations in 2D space, in order to determine the matrix T. This non-linear transformation utilises a 3x3 transformation matrix with 8 DoF , where per definition T_3^3 is 1. Formally,

$$T \begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} = \begin{bmatrix} x_q \\ y_q \\ 1 \end{bmatrix} \Leftrightarrow \begin{bmatrix} t_1 & t_2 & t_3 \\ t_4 & t_5 & t_6 \\ t_7 & t_8 & 1 \end{bmatrix} \begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} = \begin{bmatrix} x_q \\ y_q \\ 1 \end{bmatrix}$$

Applying the knowledge we know about matrix S and D onto the equation, we can find the exact projectivity matrix T that satisfies the equality for every single corner point transformation, as shown in the equality below.

$$\begin{bmatrix} t_1 & t_2 & t_3 \\ t_4 & t_5 & t_6 \\ t_7 & t_8 & 1 \end{bmatrix} \begin{bmatrix} X_{TL} & X_{TR} & X_{BR} & X_{BL} \\ Y_{TL} & Y_{TR} & Y_{BR} & Y_{BL} \\ 1 & 1 & 1 & 1 \end{bmatrix} =$$

$$= \begin{bmatrix} 0 & max\_width - 1 & max\_width - 1 & 0 \\ 0 & 0 & max\_height - 1 & max\_height - 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

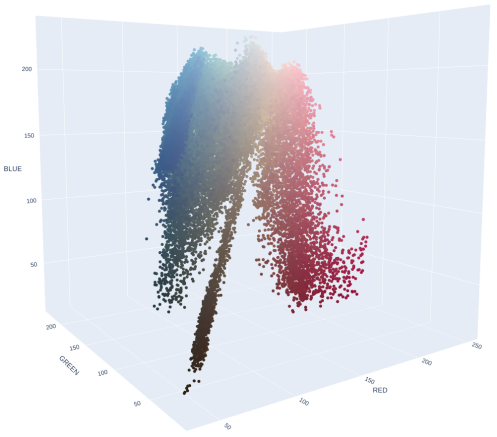
Finally, we apply the calculated matrix T to the entire image, morphing space so that the image perspective is changed, changing the location of every point.

4.2 Colour Handler

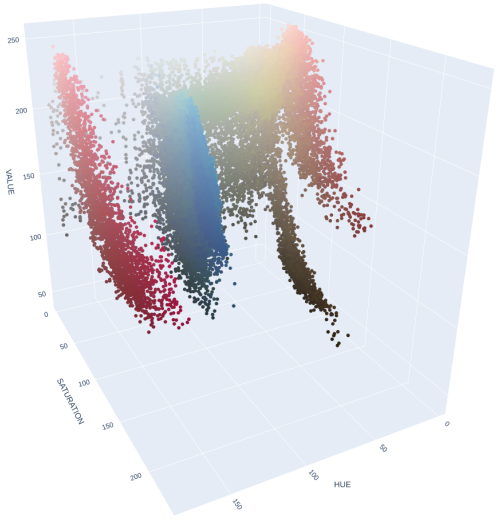
The choice of the colour space in which to perform the colour grouping and reduction has a significant impact in the resulting found colours. Three colour spaces were considered for the problem: the (1) RGB colour space, the (2) HSV colour space and the (3) CIE L*a*b* colour space.

We began by comparing the spacial distribution of pixels from different business card examples across the colour spaces. Depicted in Figure 4.4, we can observe the differences between each colour space when plotting the image pixels on one of the analysed examples.

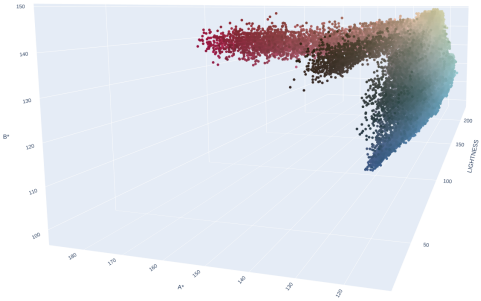
(A) Original image



(B) RGB colourspace



(C) HSV colourspace



(D) LAB colourspace

FIGURE 4.4: (a) Original image and (b), (c), (d) visualizations of the original image pixels displayed in the corresponding colour space

From the visualization standpoint solely it was clear that, in most business card examples we've tested, the HSV and CIE L*a*b* better segmented distinct colours, as the brightness/lightness information is separated from the actual colour hue information [59, 60]. An example of the distribution of colours in distinct colour spaces for one business card example is shown in Figure 4.4. From the two

possible alternatives, we have decided to choose the HSV colour space for the task as, according to Bora et al. [61], the HSV colour space performed better than CIE L* a* b* in colour segmentation.

We have opted for a k-means implementation for colour reduction. The k-means algorithm was applied to the HSV colour space with a K defined at 5 and a pseudo-random centroid initialization with k-means++. The algorithm was run 10 times for each case, only keeping the attempt with the lower cost function, in order to minimize the problem of local optima.

The process of extracting all colours from the business card will result in the module picking up undesired colours, very close to white or very close to black, common business card background colours. In order to discard these neutral colours, we convert every detected colour to the HSL colour space and evaluate each colour lightness. Any colour with lightness values under 15 or over 85 (considering the lightness scale ranges from 0 to 100) will be discarded.

4.3 Text Handler

4.3.1 Text Extraction

4.3.1.1 Approaches Tested

In the present work, we compare three approaches to tackle this problem. The results of the employed methods are described in detail in Chapter 5. We have created a unique pipeline for the task, comprised of a state-of-the-art text detector, a state-of-the-art text recognition system and a rule-based system for text ordering and word grouping. This approach was then compared to the python package EasyOCR.

Text Extraction Pipeline

As mentioned in Chapter 2, the selected models have been proven to be the best performers in the Robust Reading Competition on the COCO-Text 2017 dataset

with publicly available published paper, source code and network weights. Considering the image has already been preprocessed for the text to be easily be picked up, there was no need to account for heavily slanted text or perspective text. For this reason, we have selected, without performing extensive comparisons between available algorithms, the CRAFT Text Detector [53] as it was the best performing algorithm overall with published source code, according to the evaluated metrics in the Text Localization task of the Robust Reading Competition Focused Scene Text challenge, promoted by the International Conference on Document Analysis and Recognition (ICDAR) [62].

For text recognition we have opted for the CLOVA-AI deep text recognition [54] as it was the best performer in the similar task of recognizing words in the ICDAR Robust Reading Competition Focused Scene Text challenge [62].

We have also evaluated the EasyOCR Python package for text detection and recognition, created by the company Jaided AI. It is a sequential framework which utilizes the CRAFT text detector and a modified version of the CLOVA-AI text extraction module, trained to account for a larger dictionary of characters and symbols and, thus, supporting more languages. Considering the limitations of the original CLOVA-AI algorithm in respect to the symbols and graphical accents supported, we decided to utilise this modified version of the published work, trained with a more extensive set of symbols.

4.3.2 Text Merging

The next step in the text handling pipeline corresponds to merging the found text words into sentences. As explained in Chapter 3.3.3.2, this module takes as inputs the words extracted by OCR and its corresponding metadata associated to each word's bounding box. The following fields are extracted from each bounding box:

1. LeftX - left bound x coordinate, corresponding to the distance, in pixels, from which the bounding box left edge is to the left edge of the image;

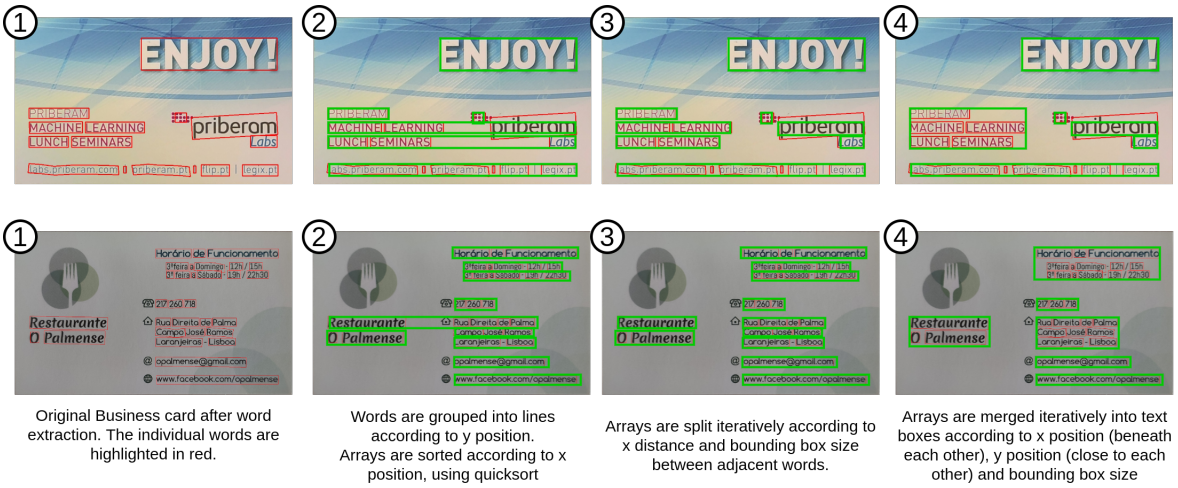


FIGURE 4.5: Word ordering and merging flow diagram

2. RightX - the right bound x coordinate, corresponding to the distance, in pixels, from which the bounding box right edge is to the left edge of the image;

3. y - the y position of the word, corresponding to the distance, in pixels, from which the centroid of the bounding box is to the top edge of the image;

4. h - the word height, corresponding to the euclidean distance between the bottom edge and the top edge of the bounding box (y coordinate subtraction).

An additional two variables are used, corresponding to the bottom edge and top edge y coordinate position (BottomY and TopY). However, these are calculated based on 3 and 4. As mentioned in Chapter 3.3.3.2, all of these values are min-max normalized to account for variances in image size and image aspect ratio.

Y Position Ordering: The module flow diagram is represented in Figure 4.5. The module first takes in all bounding box information as a list, using quicksort to order the list according to the y position of each bounding box.

Text Line Vectors: Using the sorted array, words are iteratively merged together into text line arrays by their y position, and the average y position of each merged array is calculated. In the end, all text line arrays are ordered using quicksort, according to the x position of the word in the image. Defining the vector

S of size m as the vector that originally contains all individual words, and being $S^{(i)}$ the i^{th} element in the vector S and $S^{(i)}_{LeftX}$, the value of the attribute LeftX from the data element in the i^{th} position. A standard iteration of this module computes the distance between $S^{(i+1)}_{y} - S^{(i)}_{y}$ and if the distance is inferior to the threshold defined, the words are considered in line and grouped together in a single vector. this step is repeated for i in the range of 0 to m-1. This can be observed in Figure 4.5 (2). We have tweaked the threshold, setting it to a distance of 0.01.

When two vector words are merged, the new resulting vector word will then have LeftX to be $S^{(i)}_{LeftX}$ and RightX to be $S^{(i)}_{RightX}$. The y and h values are computed every iteration and stored as parameters in two auxiliary vectors, computer and updated on every iteration.

The resulting text line vectors are sorted using quicksort, by LeftX values, so that words are ordered from left to right, as they appear in the business card.

Text Line Splitting: Given that business card text is often organised in text boxes or columns of text, not every line of text corresponds to the same sentence or information. Consequently, the resulting text line vectors must be split according to the X distance between words. The X distance between two adjacent words can be calculated by subtracting the value of X from the right bound of the leftmost word to the left bound X value of the rightmost word $S^{(i+1)}_{LeftX} - S^{(i)}_{RightX}$. The threshold was tweaked and the value of 0.12 was used, meaning if the word distance is above 0.12, the text line is split into two.

Additionally. word height information is used for text splitting. If two bounding boxes with very distinct heights are close to each other, they are considered to be not from the same text, as it's standard to utilize the same font size throughout a meaningful sentence. Formally, if $(S^{(i+1)}_{h} - S^{(i)}_{h})^2 > t$,where t is the threshold, the text line is split into two. The threshold was adjusted and we have found the value 4e-4 to work the best for our dataset. This means if the size difference between boxes is over 0.02, the text line is split into two.

Text line splitting is a recursive process, which stops when no more splits are required. The result of this operation can be seen in Figure 4.5 (3).

Text box Merging: The last step of the submdule corresponds to merging each individual text excerpts present in different text lines into text boxes, where three cues are used:

1. BottomY - bottom edge y coordinate, corresponding to the distance, in pixels, from which the bounding box bottom edge is to the top edge of the image. It is calculated as $S^{(i)}_{BottomY} = S^{(i)}_{y} + \frac{1}{2}S^{(i)}_{h}$;

2. TopY - top edge y coordinate, corresponding to the distance, in pixels, from which the bounding box top edge is to the top edge of the image. It is calculated as $S^{(i)}_{BottomY} = S^{(i)}_{y} - \frac{1}{2}S^{(i)}_{h}$;

3. AvgH - average text box height on a text excerpt;

4. LeftX - left bound x coordinate, corresponding to the distance, in pixels, from which the leftmost word bounding box left edge is to the left edge of the image;

5. RightX - the right bound x coordinate, corresponding to the distance, in pixels, from which the rightmost word bounding box right edge is to the left edge of the image.

The Y position is used to determine if the text lines are close enough to each other to be merged. The text X position is used to determine if the text line with the biggest Y position is directly underneath the other text line. Finally, the text line height is used to determine if the two lines have the same font size, meaning it is likely that they are part of the same text excerpt or sentence. This recursive method merges text lines until no more merges are possible.

For each text excerpt, the submodule calculates its average font size, average y position and the left and right positions of the text. It compares all possible pairs of text excerpts, evaluating font size similarity, x and y positions and deciding if a merge is required or not. Taking as an example the text excerpts pair i and j, we evaluate the following expressions:

1. $(S^{(i)}_{AvgH} - S^{(j)}_{AvgH})^2 < t_h$, the squared average font size between text excerpts is under a predefined threshold t_h;

2. $(S^{(i)}_{TopY} - S^{(j)}_{BottomY})^2 < t_y$, the distance between the two text excerpt bounding boxes is under the threshold t_y;

3. $S^{(j)}_{LeftX} < \frac{1}{2}(S^{(i)}_{LeftX} + S^{(i)}_{RightX}) < S^{(j)}_{RightX}$, the x coordinate of the centroid point of the bounding box of the word i is between the left bound X coordinate and the right bound X coordinate of the word j, meaning the text excerpt i is directly below j.

The thresholds have been tweaked to work on most business cards on our dataset. The threshold t_h was set to 4e-4, corresponding to a normalized distance of 0.02 between font sizes. The threshold t_y should be relative to the font size that is to be merged. This means that larger fonts are to be expected to have a bigger spacing between text lines. For this reason, we defined t_y in function of the font sizes of the text excerpts i and j. t_y was defined to be $\frac{1}{3}(S^{(i)}_{AvgH} + S^{(j)}_{AvgH})$, where the constant $\frac{1}{3}$ was tweaked to work on most test cases.

4.3.3 Text Categorization

4.3.3.1 Regular Expressions

For the implementation of the text tagging module, we utilise regular expressions and basic string comparison. The text boxes' stings output by the text handler module are individually used as input for text tagging. We have defined the following regular expressions:

1. Website:

```
    (((http|https):\/{2})?(([0-9a-z_-]+\.)+(aero|asia|biz|cat|com|coop|
edu|gov|info|int|jobs|mil|mobi|museum|name|net|org|pro|tel|travel|ac|ad|
ae|af|ag|ai|al|am|an|ao|aq|ar|as|at|au|aw|ax|az|ba|bb|bd|be|bf|bg|bh|bi|
bj|bm|bn|bo|br|bs|bt|bv|bw|by|bz|ca|cc|cd|cf|cg|ch|ci|ck|cl|cm|cn|co|cr|
cu|cv|cx|cy|cz|cz|de|dj|dk|dm|do|dz|ec|ee|eg|er|es|et|eu|fi|fj|fk|fm|fo|
fr|ga|gb|gd|ge|gf|gg|gh|gi|gl|gm|gn|gp|gq|gr|gs|gt|gu|gw|gy|hk|hm|hn|hr|
ht|hu|id|ie|il|im|in|io|iq|ir|is|it|je|jm|jo|jp|ke|kg|kh|ki|km|kn|kp|kr|
kw|ky|kz|la|lb|lc|li|lk|lr|ls|lt|lu|lv|ly|ma|mc|md|me|mg|mh|mk|ml|mn|mn|
mo|mp|mr|ms|mt|mu|mv|mw|mx|my|mz|na|nc|ne|nf|ng|ni|nl|no|np|nr|nu|nz|nom
|pa|pe|pf|pg|ph|pk|pl|pm|pn|pr|ps|pt|pw|py|qa|re|ra|rs|ru|rw|sa|sb|sc|sd
|se|sg|sh|si|sj|sj|sk|sl|sm|sn|so|sr|st|su|sv|sy|sz|tc|td|tf|tg|th|tj|tk
|tl|tm|tn|to|tp|tr|tt|tv|tw|tz|ua|ug|uk|us|uy|uz|va|vc|ve|vg|vi|vn|vu|wf
|ws|ye|yt|yu|za|zm|zw|arpa)(:[0-9]+)?((\/([~0-9a-zA-Z\#\+\%@\.\/_-]+))
?(\?[0-9a-zA-Z\+\%@\/&\[\];=_-]+)?)?))\b
```

2. email:

```
(^[a-zA-Z0-9_.+-]+@[a-zA-Z0-9-]+\.(aero|asia|biz|cat|com|coop|edu|gov|
info|int|jobs|mil|mobi|museum|name|net|org|pro|tel|travel|ac|ad|ae|af|ag
|ai|al|am|an|ao|aq|ar|as|at|au|aw|ax|az|ba|bb|bd|be|bf|bg|bh|bi|bj|bm|bn
|bo|br|bs|bt|bv|bw|by|bz|ca|cc|cd|cf|cg|ch|ci|ck|cl|cm|cn|co|cr|cu|cv|cx
|cy|cz|cz|de|dj|dk|dm|do|dz|ec|ee|eg|er|es|et|eu|fi|fj|fk|fm|fo|fr|ga|gb
|gd|ge|gf|gg|gh|gi|gl|gm|gn|gp|gq|gr|gs|gt|gu|gw|gy|hk|hm|hn|hr|ht|hu|id
|ie|il|im|in|io|iq|ir|is|it|je|jm|jo|jp|ke|kg|kh|ki|km|kn|kp|kr|kw|ky|kz
|la|lb|lc|li|lk|lr|ls|lt|lu|lv|ly|ma|mc|md|me|mg|mh|mk|ml|mn|mn|mo|mp|mr
|ms|mt|mu|mv|mw|mx|my|mz|na|nc|ne|nf|ng|ni|nl|no|np|nr|nu|nz|nom|pa|pe|
pf|pg|ph|pk|pl|pm|pn|pr|ps|pt|pw|py|qa|re|ra|rs|ru|rw|sa|sb|sc|sd|se|sg|
sh|si|sj|sj|sk|sl|sm|sn|so|sr|st|su|sv|sy|sz|tc|td|tf|tg|th|tj|tk|tl|tm|
tn|to|tp|tr|tt|tv|tw|tz|ua|ug|uk|us|uy|uz|va|vc|ve|vg|vi|vn|vu|wf|ws|ye|
yt|yu|za|zm|zw|arpa)$)
```

3. Phone number: `r"[(]?[+]?[(]?[0-9]{1,4}[)]?[\s\.0-9]{8,15}"`

The website expression has two main capturing groups. The first capturing group, `((http|https):\/{2})?`, is optional, meaning its presence is not required for a website to be tagged. This portion will trigger on strings beginning with http:// or https://. The second portion of the expression includes four sequential verifiers. Firstly, the module attempts to match any sequence of numbers and letters followed by a dot. Then, we verify the existence of a valid top-level domain name, listed in the regular expression. We then verify the existence of a port number in the domain, with the expression `(:[0-9]+)?`, which will match any

string that starts with a colon, followed by a sequence of numbers. Finally, we attempt to capture any subdomains, sub-subdomains and so on, with the expression `((\/([~0-9a-zA-Z\#\+\%@\.\/_-]+))?(\?[0-9a-zA-Z\+\%@\/&\[\];=_-]+)?)?`. This expression matches the character literally, followed by a single or sequence of numbers, letters or special characters.

The email expression has three components that must be present. First, there must be a sequential set of letters, numbers and/or symbols followed by the symbol @. We then check, once again, for the presence of a sequential set of letters, numbers or dashes, representing the email provider. Finally, we check for the email ending sequence, with the character "." (dot) and a valid top-level domain name.

The phone number verifier can be divided into two parts. Firstly, we verify the existence of an international indicator, which is optional but, if present, will be captured by the expression `[(]?[+]?[(]?[0-9]{1,4}[)]?`. This will allow for indicators surrounded by parenthesis or not, including the plus sign or with the numerical indicator only. Indicators such as (+1), +(351), (+44) and +2 will all be captured by this expression. Following, we capture any sequence of numbers, dots and white spaces, with a size of 8 to 15 characters, as most phone numbers in the world are no shorter than 8 characters. We have opted for capturing white spaces and dots as it is common to use these symbols to group digits and improve readability.

For each text box, the system first attempts to match any character succession with the regular expressions of emails, websites or phone numbers. If any match is found, the found strings are tagged and saved. It important to denote that the email search is performed prior to the website search and, if an email is found, the text is removed from the array of expressions to tag. This is due to the nature of formatting of an email, where the portion after the symbol @ is often a website (for example, the provider gmail.com).

4.3.3.2 Rule-based search

We verify the presence of social network references by searching text boxes for the text "facebook", " fb " or "twitter". This information is saved according to the type of social network portrayed. Since some business cards may contain a direct link for the platform, we evaluate the presence of facebook and twitter links before searching for websites with the regular expression.

Since home addresses do not follow a specific pattern that can be recognised by an expression, we have opted for a rule-based system, which attempts to find cue keywords that can indicate the text box is an address. The submodule was designed to work with addresses from Portugal and the United States of America. The cue keywords are of three types:

1. Address indicators: We have compiled a list of the most used address indicators in English and Portuguese. The presence of one of the words "rua", "avenida", "av.", "praça", "estrada", "estr" often indicate we are evaluating a Portuguese address. The words "street", "st.", "road", "rd.", "avenue" and "av." are indicators for English-speaking country addresses.

2. USA States: We check for the presence of USA state names on each text box, as a reference to the state is often present in American addresses. We have decided not to check for state abbreviations, commonly used as an alternative way of including state information in addresses, as some abbreviations are also valid English and Portuguese words or abbreviations for other words (examples of "Co" for Colorado, "De" for Delaware, "Hi" for Hawaii, "In" for Indiana, "Ia" for Iowa, amongst others).

3. Portuguese Cities and Municipalities: We have compiled a CSV with all Portuguese cities and municipalities information, taken from the website dados.gov.pt.

We have decided not to use country names as address indicators, as they are more prominently present in company slogans than state and city names, which could result in wrong tags. However, the possibility that some slogans or others can also

include state, city or municipality names is not an option to discard, which could lead to wrong tags. This situation is further explored in Section 5.3.

The presence of any of these keywords will trigger the module to tag the text box with "Address".

4.4 Design Personalisation Handler

In order to implement the design personalization handler, we first query the Google Places API for the business name. We utilise the geocoder library in order to acquire the current approximate location of the user based on its IP and search for businesses with a similar name as input by the user, in a 40 kilometer radius. The resulting json response is parsed, extracting the array of "types" associated to the business. We are specifically looking for the main type, which better describes the business. As such, we save the type present in the position 0 of the array that was response to our query. This module supports the generation of double sided business cards, having these two files for the design generation, each corresponding to a face of the card.

The designs are organised in folders, according to its type. In the first level of the filesystem, we encounter folders for each business type currently supported. The current types are restaurant, accommodation, retail and miscellaneous. Each of these folders contains one folder per associated template. The templates include 4 components: a fonts folder, which includes any .ttf or .otf files necessary for the generation of the template, an icons folder, which includes all icons necessary for the template such as email and social network icons, maps icon, phone icon, amongst others, an images folder, which includes any images necessary for the design generation and the design.py file.

The design.py contains all code necessary for the generation of the business card. The fonts and icons are loaded and placed in the appropriate location. The placeholder for the text elements and company logo are also defined in this file. The module takes as input the text extracted and the submitted logo and applies it to the placeholders defined on each business card. The gathered colour information

is also utilised as input of the module, passed in as a list in order of importance. This list is used to colourize specific parts of the business card. The design can have defined shapes, backgrounds or fonts to include colours that are related to the brand. Each of these elements can be tagged with colours MAINCOLOUR, representing the colour in position 0 of the list, SECONDCOLOUR, respresenting the first position of the list and TERTIARYCOLOUR, representing the second position of the listr

Chapter 5

Evaluation and Demonstration

In order to evaluate the developed modules, we have selected a set of twelve local business cards, supposed to represent the most common cases of submitted images, from different types of businesses, with different backgrounds, font styles, design styles and information portrayed. These business card images have been taken in different background scenarios, totalling 24 images.

Businesses: Six restaurant business cards, one hotel business card, three retail shop cards, one conference card and one driving school business card.

Backgrounds: Four images on each of three common table colours, in order to represent surfaces on which the client might take a picture on. We have picked a wooden texture table, a light couloured table and a dark coloured background.

Content: We have included business cards with custom fonts, where some letters appear warped, arched text, different colour text, text over complex backgrounds and with low background contrast.

5.1 Image Preprocessing: Unskewing and Background Removal

This section is organised into three sets of tests. Firstly, we will provide the results for the corner detectors, which explain why this approach has not been further

pursued. Following, we present the results achieved by the method in use for the current work, which takes edge information into consideration. Finally, we present the results for the image unskewing module.

5.1.1 Corner Detection

We first compared the Harris and Shi-Tomasi corner detectors in order to determine the best fit for the task. It was found that, even though neither had particularly good results, not being able to detect all corners in most pictures, the Harris corner detector performed better for the test set, having found 11 out of the 48 corner points, whilst the Shi-Tomasi only managed to pick up 6. Additionally, a lot of unwanted corner points have been detected, mostly corresponding to the corners of each single character present in the business card text. The use of blurring filters in order to remove any unwanted points has proven to be unsuccessful as we have found out that the corner points of the business card are often softer than the hard edge contrast between black letters on white background in the business card. The application of noise reduction algorithms resulted in an adverse effect than desired, with even less business card corner points being detected. In image 5.1, we can observe an example of the results of both of these methods. In Appendix A Figure A.1, we include a comparison between other examples present in the dataset.

(A) Shi-Tomasi (B) Harris

FIGURE 5.1: Example of corner detection results with Shi-Tomasi and Harris algorithms. We can observe that many corner points that do not correspond to the business card have been picked up in both scenarios and not all business card corner points have been picked up.

5.1.2 Edge Detection

The edge detection algorithm applied, as explained in Section 4.1.1 has proven to be a much more successful approach to the problem. However, there are still some limitations and failure cases, which will be explored in this section.

FIGURE 5.2: Edge Detection Module results on different background types and business cards.

In the tests performed, the algorithm performed consistently well on all business cards that were predominantly white, regardless of the background colour and complexity behind the business card (as long as the constraint VI, referenced in Section 3.1 regarding colour contrast is respected). We can observe some success cases for different business cards in distinct lighting conditions and background

FIGURE 5.3: Edge Detection Module failure results.

types in Figure 5.2. Depicted is the original input image, the edge map after the bilateral smoothing and canny edge detector have been applied and the resulting bounding quadrilateral.

Despite the positive results shown, this module is prone to failures. Using the dataset of images described in the present Chapter, we have evaluated the most common failure scenarios, attempting to reason why the failure has occurred. In Figure 5.3, we can observe some of the failure scenarios:

1. In a), the complexity of the image present on the left edge of the business card could be responsible for this fail case.

2. In images b) and c), two similar scenarios are presented, where the drop shadow present under the business card has softened the edge on the business card black background.We can observe that the edges in which the business cards are dark and a drop shadow is cast, the edge is not visible in the edge map.

3. Despite the edge map looking seemingly unaffected by the similarities in colour between the business card and the background in d), the module still failed to output a valid bounding polygon.

We believe this module is crucial for the system and its failure may be a bottleneck for the remaining components, especially for the colour extraction module. For this reason, we provide recommendations for further improving its results in Section 6.2.

5.1.3 Geometric Perspective Transform

In the tests performed, the Geometric Perspective Transform module achieved near perfect results, successfully removing the background and straightening the business card to the correct orientation. These tests have been performed independently to the result of any previous modules, meaning we assume the coordinates of the business card corners have been correctly inferred. In Figure 5.4, we can observe examples of the module results.

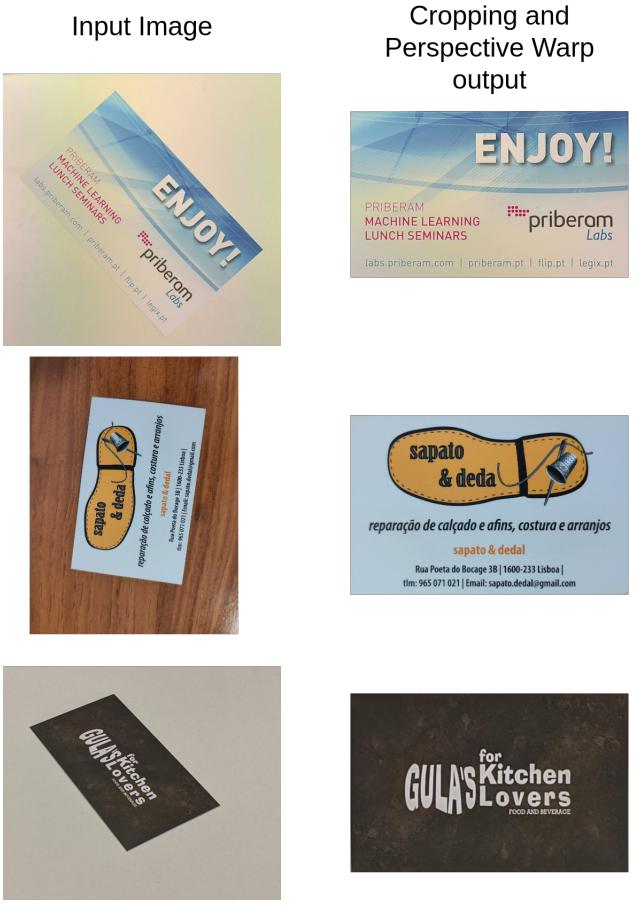


FIGURE 5.4: Geometric Perspective Transform results.

In attempt to explore the potential failure cases of the module, we performed the perspective transform operations on unusual perspectives, which violate the system Constraint V, specified in Section 3.1. We can observe that the result of such transformation has an incorrect aspect ratio, due to the fact that the module utilises the size of the width and height in the original image to approximate the original aspect ratio of the card. This situation can be observed in Figure 5.5

FIGURE 5.5: Geometric Perspective Transform failure case.

5.2 Colour Extraction

We have tested the colour extraction module with the images resulting from the GPT module, which correspond to the transformed business card without background information.

FIGURE 5.6: Colour Extraction test cases.

In Figure 5.6, we can observe the colours extracted by the module from a variety of images. Below each picture, we can observe the list of colours extracted, ordered by its representativity. The representativity is measured by the number of pixel points associated to each colour cluster, meaning an important colour should

theoretically be more well represented in the business card. Below each colour, we can observe the value of L in the HSL colour system, which makes sure the colour is not too close to white or too close to black to be relevant.

Overall, the module achieved good results, being able to extract the most represented colours in each business card and eliminating very light and very dark tones. However, it can be noted that (i) the exclusion of unwanted colours, taken into consideration the lightness of each colour, has limitations. This situation can be seen in Figure 5.6 d), where the main colour is in the second position of the vector due to the slight blue tinge of the background associated to the lighting conditions when the picture was taken. (ii) The 5 colour palette may not be adequate to some business cards with a different number of colours. This can be seen in images b) and e) from Figure 5.6, where the same colour on different lighting conditions appears twice on the palette. This situation was to be expected in a solution with a fixed number of colours K. In Section 6.2, we provide future recommendations to improve this module.

5.3 Text Extraction

5.3.1 Text Detection

As explained in chapter 4, we have applied the CRAFT text detector, not providing a comparison with similar state-of-the-art models. In order to evaluate the results we set up a dataset of manually annotated images corresponding to the ground truth expected for the algorithm to produce and we manually compared it to the output bounding polygons produced by the model.

Four types of distinct error types were identified: Type-1 error, corresponding to a false-positive word found, type-2 error, corresponding to a false negative error, cropped word error, where the output bounding box does not encapsulate the full word and agglomeration error, where two or more words were wrongly agglomerated. In our testing dataset, the detector achieved an accuracy[1] of 97.785%, being

[1]For the calculation of accuracy, we have considered the cropped word and agglomeration to be special cases of type-1 errors, despite being in different columns on Table 5.1

Business Type	Img	Correct	Type-1	Type-2	Cropped-Word	Agglomeration
Restaurant	1	43	0	0	0	0
	2	4	0	0	0	3
	3	15	1	0	0	0
	4	19	0	0	0	0
	5	31	0	0	0	0
	6	25	0	1	0	0
Driving School	1	28	0	0	0	0
Hostel	1	19	0	0	1	0
Conference	1	14	1	0	0	0
Retail	1	26	0	0	0	0
	2	30	0	0	0	0
	3	55	0	0	0	0
Total		309	2	1	1	3

Table 5.1: CRAFT text detector results

the detailed results reported in Table 5.1. In Appendix A, Figure A.2, we can observe errors on the CRAFT model applied to the business card dataset.

5.3.2 Text Recognition

We performed tests on the text recognition module independently, meaning every error on the text extraction module has been ruled out, in order to only account for mistakes made by the text recognition model. This means any bounding polygons that either do not correspond to a word, that crop words in any way or that agglomerate words in a single image are not used to test the module.

In order to find the module accuracy we first labeled every word image with a true label of the word represented in each image. Any type of word capitalization was disregarded, being all labels lowercase and the output of the models tested converted to lowercase before comparing. Some of the words were assigned more than one true label if they contained graphic accents or punctuation. For example for the true label "correção,", the variants "correcão", "correçao", "correcao", "correcão,", "correçao," and "correcao," would also be accepted.

Cause	# of Errors	Percentage
". , : ;" mistaken by a letter	11	0.2037
Zip Code – or / mistaken by number	8	0.1481
International phone indicator "+"	6	0.1111
Currency indicator	6	0.1111
website not recognized	5	0.0926
Email @ not recognized	4	0.0741
Superscripts	3	0.0556
Ampersand	2	0.037
Graphical accents	2	0.037
Dash	1	0.0185
Other mistakes	**6**	**0.1111**
Total	54	1

Table 5.2: CLOVA-AI Deep Text Recognition error analysis

5.3.2.1 CLOVA-AI Deep Text Recognition

The Deep Text Recognition [54], developed by CLOVA-AI scored an accuracy value of 0.82, lower than what was initially anticipated from the published results on more complex text shapes.

When performing error analysis, we came to the conclusion that the errors mainly occurred in words with special symbols. So, for the 54 misclassified examples out of the 300 total cases, 89%, corresponding to 48 were mistakes due to special characters not used in training and, therefore, not classified. In the Table 5.2 is discerned the misclassified examples between categories. Considering the high amount of errors due to an inappropriate training set, we have determined this method is not the best fit for the problem. In Figure 5.7, we can observe examples of the most common error types.

5.3.2.2 Tesseract

We then performed OCR using the popular library Tesseract on the bounding words extracted with CRAFT text detector. Considering it was trained including a broader dictionary of symbols and special characters, with multi-language support, it is relevant to be tested for our problem.

Correct Label	Blank Prediction	Word Error	Total
105	183	11	299

TABLE 5.3: Tesseract results

In Table 5.3, we can observe the results of the model. Despite the very high accuracy rate of 90.5172% when comparing the errors with the sum between correct predictions and errors, the vast majority (61.2%) of the predictions of the algorithm were blank, meaning it did not manage to find text in the image. If we add in the blanks to the calculation of accuracy, we only achieved 35.12% of correct predictions, which is substantially lower than the method described in 5.3.2.1.

One of the reasons that might explain the poor performance of Tesseract is the use of the bounding box words instead of the full text picture. To test this theory, we have performed tests using Tesseract in the full business card image. In Appendix A, Table A.1, we can analyse the types of errors that occurred using Tesseract for both the text detection and text recognition.

We can observe the results of detecting and recognizing text with Tesseract were substantially better than the previous CRAFT+Tesseract approach. However, one of the biggest problems with this method is the weak text detection component, which is reflected in a very high number of Type 2 errors. This has

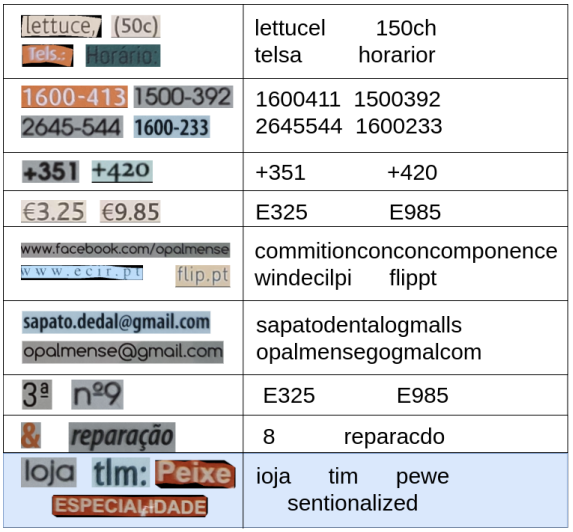


FIGURE 5.7: Deep Text Recognition errors. All, excluding the row in blue, have been caused by characters not supported by the model.

resulted in a heavy impact on the results of the system, with a final accuracy[2] of 63.924%, lower than the previously tested combination of CRAFT and CLOVA-AI text detector.

By performing error analysis we can conclude that text with a low contrast to the background, slanted text and arched text were the main causes of type-2 errors, which can be seen in Appendix A, Figure A.3

5.3.2.3 EasyOCR

This package uses a modified version of the CLOVA-AI text recognizer, which underperformed on the tests in 5.3.2.1 due to the training dictionary not including symbols. In Table 5.4, the error analysis is summarized. We can observe that, compared to our pipeline, the error rate decreased drastically, mainly due to the fact that the errors related to special characters are now reduced to only 50% of the total of errors. Using EasyOCR, we managed to achieve an overall system accuracy of 86.913%. In Table 5.5, we provide a more detailed breakdown of the error types detected, whcih can be compared to the original CLOVA-AI algorithm, in Table 5.2.

Business Type	Img	Correct	Type1	Type2	Wrong Label
Restaurant	1	24	0	0	13
	2	4	0	0	1
	3	10	0	1	3
	4	17	1	0	2
	5	27	0	0	3
	6	22	0	2	3
Driving School	1	28	0	0	0
Hostel	1	18	0	0	2
Conference	1	12	0	0	0
Retail	1	26	0	0	1
	2	27	0	0	1
	3	45	0	0	5
Total		259	1	3	35

Table 5.4: EasyOCR text detection and recognition results

[2]For the calculation of accuracy, we have considered the wrong classification to also be a type-1 error, despite being in a different column on Appendix A, Table A.1

Cause	# of Errors	Percentage
. , : ; mistaken by a letter	0	0
Zip Code – or / mistaken by number	1	0.02941
International phone indicator "+" mistaken by a number	1	0.02941
Currency indicator mistaken by a letter (usually E)	10	0.29412
website not recognized	0	0
Email error	1	0.02941
Superscripts	3	0.08823
Ampersand	0	0
Graphical accents mistaken as different letter	0	0
Dash	1	0.02941
Other errors	17	0.5
Total	34	1

TABLE 5.5: EasyOCR text recognition error analysis

5.3.3 Text Categorization

In this chapter, we will discuss the success and failure scenarios of the text categorization Regular Expressions, specified in Section 4.3.3. Regarding the website regular expression, it successfully captured a variety of website formatting standards, defining the communication protocol or not (http:// or https://), starting with "www." or not. It also successfully captures any levels of subdomains on an address. The email address expression is prepared to handle any email provider, only verifying the top-level domain. The expression allows for the use of letters, numbers and certain symbols on both the email account name and email provider. We have tested the phone number regular expression with Portuguese, United States and British phone numbers, including or not the international indicator. For all test cases, the expression managed to pick up the phone numbers correctly. Even though no extensive tests were conducted with phone numbers from other countries, the module should correctly categorize any phone number under 15 characters long. Result examples of the two regular expressions are provided (see Appendix A, Figure A.4)

5.4 Design Personalization and Final Demonstration

In this section we will provide a demonstration of the full system the design personalization module. We will then discuss the obtained results. The pipeline demonstrated in the present chapter correspond to the architecture defined in chapter 3

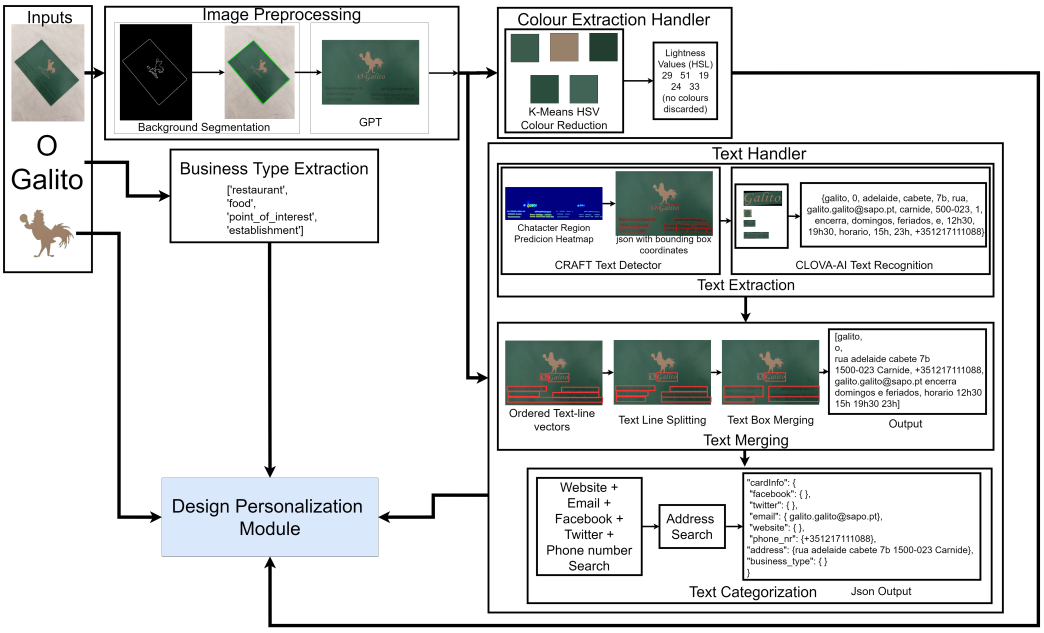


FIGURE 5.8: System Demonstration.

We begin by submitting the image of a business card, along with the company name and logo. We can observe all outputs of each stage of the system pipeline in image 5.8.

The business type is extracted from the inputted business name, by querying the Google Places API. In this case, the system recognised that the client is a restaurant.

The image is first preprocessed, where we perform background segmentation and geometric perspective transform. The background segmentation is executed with the creation of an edge map. We can observe that the module successfully generated a flat representation of the business card design. Following, we extract colour and textual information from the Image Preprocessing output.

The colour extraction module first converts the image to HSV and applies a K-means algorithm for colour reduction, with a K set as 5. This algorithm is ran 10 times, where the result with the lowest value for the cost function is used. The found colours are then converted to HSL colour system, where the lightness value is evaluated. In this test case, all HSL lightness values are between 15 and 85, meaning no colour is discarded.

The text handler is composed of three consecutive steps:

1. Firstly, we extract the text, using the CRAFT text detector followed by the CLOVA-AI Text Recognizer, being the main output a list of words found, along with the metadata associated to each bounding box, as explained in 4.3.2. We can observe the output words in Figure 5.8. The bounding box metadata information is not displayed due to its size.

2. Following, we merge text into text boxes, by consecutively merging words into lines, splitting lines into segments and merging each text segment into boxes. The output of the module is a list textboxes with ordered text. We can observe that the results of this module were flawed, as for example, the words "O" and "Galito" were not merged. This was due to the bounding box size difference, where the algorithm recognized they were written in different fonts.

3. Finally, the text is categorized. In this example, the system has found an email, a phone number and an address.

The results of the text handler, the colour handler and the business type extraction, along with the submitted logo, are used as input for the design personalization module, detailed in Figure 5.9. The module selects a design from the available designs list for the category "Restaurant". Following, the design was personalized with colour, text and logo information extracted, according to the set of rules in the selected design. We can see a visual representation of a non-personalized design, with placeholders for text and colour, and the final design in Figure 5.9.

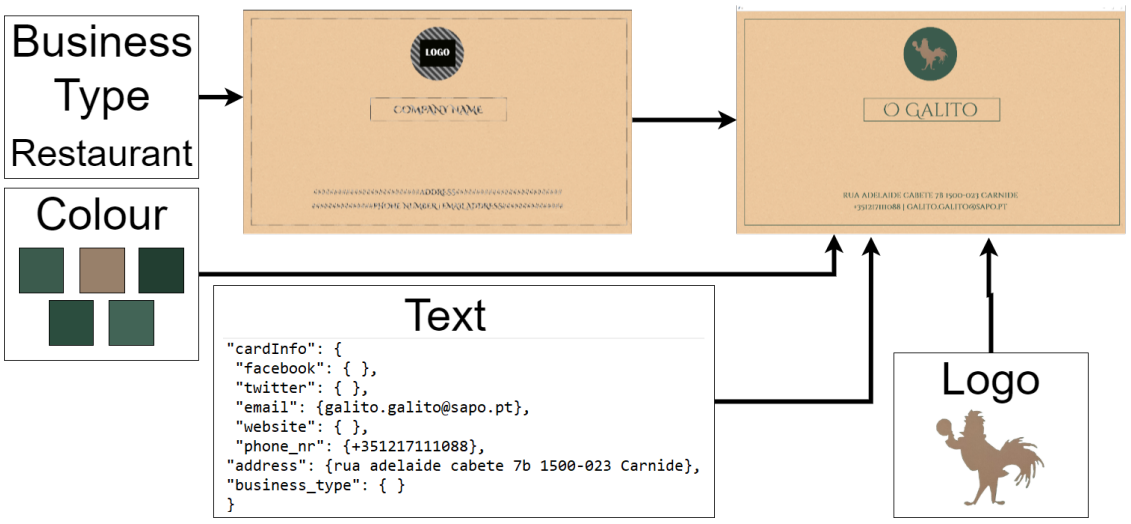


FIGURE 5.9: Design Personalization Module Demonstration.

(A) Business Card front

(B) Business Card back

FIGURE 5.10: Generated business card front and back

Rerunning the module with the same business card input can result in different solutions. In Figure 5.10, we can observe an alternative result of the design personalization module for the same example. In this case, a double-sided business card was generated, with the MAINCOLOUR being used in the word "Restaurante" and the colour bar along the bottom of the front side, the SECONDCOLOUR used to colourize the circle around the logo and the name of the restaurant in the front side, and the background in the back side. The icons, the slogan "Life's too short for boring food", the knife and fork and the casserole on the back design are all icons part of the design template.

We can also observe that the placeholders for the Facebook and Twitter include filler text, as the information was not present in the original design. The idea is that the client may want to add this information to a newer design, even though it was not present in the submitted image. We provide recommendation for future

work in order to better control the designs generated according to the information the user wants to have portrayed in the new design. These recommendations are listed in Sections 6.2 and 6.3.

Chapter 6

Conclusion

In the present work, we have successfully developed a prototype that fully answers the research question mentioned in Chapter 1.3, which was to provide a prototype that can automate the creation and personalization of business card designs based on an existing business card, adapted to the user. The motivation for this research comes from the necessity of providing a cheaper, less time consuming and more interactive alternative for MSMEs to create marketing printouts. The importance of such elements is related to the necessity of small enterprises, which often do not have a dedicated marketing team or budget, to communicate their brand image with the desired target, further explained in Chapter 1.2.

In order to select and personalize the template, we have taken into consideration the business type, the predominant colours in the element and the text elements, as described in Chapter 3. In order to achieve this objective, we have developed the following modules:

1. An image background removal and image geometric perspective transform module, described in Section 3.3.1 and detailed in 4.1, so that all pixel information that is not related to the business card is discarded;

2. A colour extraction module, described in Section 3.3.2 and detailed in Section 4.2, which extracts colour information from the business card image

3. A text extraction pipeline, which successfully identifies and extracts text from the business card, categorizing the information according to what is portrayed, described in Section 3.3.3 and detailed in 4.3.

4. A design personalization handler, which selects an appropriate template according to the user's business type, and personalizes it with the colour and textual information extracted, described in Sections 3.3.4 and detailed in 4.4.

The overall system tests have shown that the prototype can successfully personalize business cards according to the user's input image, as can be seen in Section 5.4.

For each module we have conducted a battery of validation tests, which have proven its efficiency in the specific task and denoted the current limitations and failure cases. The test reports can be observed in Chapter 5 and future improvement recommendations based on testing are summarized in this chapter's Section 6.2.

An adaptation of the present work has been submitted to the Applied Sciences special issue journal "Applications of Emerging Digital Technologies: Beyond AI & IoT" section Computing and Artificial Intelligence. This adaptation includes references to Sections 1.2, 1.3, 1.4, 2.2, 3.2, 3.3, 4, 5.1, 5.2 and 5.4.

In Section 6.1 we provide a list of the found limitations of the developed project. Finally, in Section 6.3, we present an overview of the possible future work related to this study, including possible directions of research.

6.1 Limitations

The findings of this study have to be seen in light of some limitations. The ongoing COVID-19 pandemic has resulted in the project kickoff and deliverables to be postponed to a future time. This situation's implications to the present work were trifold. Firstly, the lack of a promised image dataset resulted in the

necessity to adapt the implementation of certain modules, which has hindered the performance of the image preprocessing and Design Personalization modules, observed in Sections 5.1.1 and 5.4. In Section 6.2, we provide recommendations in order to further improve these modules.

The time constraints related to the development of the dissertation, not corresponding to any project deliverable was a limitation for the development. Due to the postponing of the project kickoff, the development of the dissertation started later than what was initially predicted.

Finally, the lack of real test cases and validation with the requesting company was a limitation for the results shown in Chapter 5. A thorough evaluation and approval of the requester would have been the ideal scenario, not possible due to the prior mentioned circumstances.

6.2 Recommendations

In this Section we provide recommendations for the work development based on the tests performed in Chapter 5. We have concluded it would be advisable to further improve the background segmentation algorithm in the image preprocessing module, described in 3.3.1.1 and 4.1.1. Currently, due to the lack of image data necessary, as explained in Section 6.1, the task is being performed by a rule-based system, which applies an edge detector to find the business card contour. This module would likely be outperformed by a deep-learning segmentation model, pretrained with ground truth annotated business card image data. It is expected for the implementation of the present work to collect sufficient image data in order to train the proposed model in the future.

The colour extraction module, whose results are detailed in Section 5.2 could be further improved. We have shown that the necessity of defining a fixed factor K for the colour reduction can influence the found results, as the number of colours on each business card is not known. An implementation on which the value of K would adjust to the scenario would be an improvement and would lead to more consistent results.

The system modularity explained in the System Architecture, Section 3.2 allows for a constant update of the text extraction module. Despite the satisfactory results achieved with both the text extraction and text recognition modules, as better-performing state-of-the-art systems are published, the models should be updated.

The design personalization module could be improved in two ways: firstly, the selected designs do not take into consideration the textual information gathered. This means that if one or more textual categories are not found but the selected template is expecting this information, the template will include placeholder text in the final design. We recommend the future implementation of a filtering system, where the designs are categorized by information required. Secondly, in the tests conducted some generated designs had text or images placed on top of backgrounds with very similar colours, which is not ideal. We recommend the implementation of a system which can check if a text or image with a specific colour can be placed in a certain background, by measuring the similarity between both colours.

6.3 Future Work

There are many ways in which we can improve the present work. The development of interfaces for both the user and a designer submitting a template are interesting features for future work. The existence of a user interface would allow users to generate images without submitting a picture, by manually selecting colours and submitting textual information. The user interface would also permit the user to modify the text or tweak colours that have been incorrectly picked up by the text and colour extraction modules respectively. Another way of further extending the present work is to simplify the creation of business card templates by designers. The existing business card templates have been created as described in Section 4.4, using a Python canvas and interface creation library. A frontend solution that allows for simple element drag-and-drop operations on images, text boxes and placeholders for logos, permitting the customization of font sizes and styles and saving the code for the design template would be ideal.

Currently, the selection of the template is solely based on the company type. The choice between the several adequate templates to the company type is still not controlled. For future work, it would be beneficial for template recommendation to develop a deep-learning model that can detect the style of the marketing printable and further filter the templates that are more adequate to the client. This would require a large labaled dataset of business card styles, which is expected to be collected by system proposed in the present work.

Finally, we believe this solution can be adapted to other marketing printouts, such as flyers, leaflets, brochures, handouts or posters. However, this would require broadening the scope of research and adapting the currently developed modules, which would result in the necessity of a further research and new validation scenarios.

Appendices

Appendix A

Evaluation Appendix

A.1 Image Preprocessing

FIGURE A.1: Shi-Tomasi and harris Corner Detector Application.

A.2 Text Handler

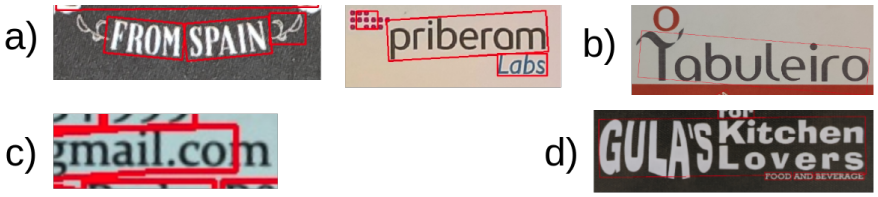


FIGURE A.2: All errors made by CRAFT text detector. a) Type-1 errors; b) Type-2 error; c) Cropped Word error; d) Word agglomeration.

Business Type	Img	Correct	Type1	Type2	Wrong classification
Restaurant	1	21	6	5	1
	2	2	3	2	0
	3	5	0	9	2
	4	2	0	17	0
	5	31	0	0	2
	6	1	0	24	0
Driving School	1	31	1	5	6
Hostel	1	19	0	1	0
Conference	1	10	1	0	2
Retail	1	6	0	20	0
	2	30	0	0	0
	3	44	2	5	0
Total		202	13	88	13

TABLE A.1: Tesseract text detection and recognition results

FIGURE A.3: Tesseract Text Detection errors

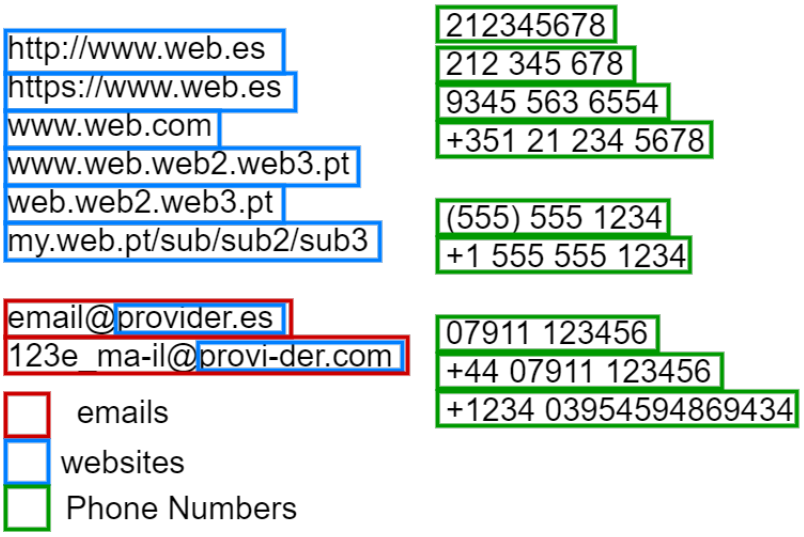


FIGURE A.4: Text Categorization regular expression tests.

References

[1] A. Jamal and M. M. Goode, "Consumers and brands: A study of the impact of self-image congruence on brand preference and satisfaction," *Marketing Intelligence & Planning*, vol. 19, no. 7, pp. 482–492, 2001.

[2] P. F. Drucker and J. A. Maciariello, *Management, revised edition.* New York: Collins, 2008.

[3] B. Yoo, N. Donthu, and S. Lee, "An examination of selected marketing mix elements and brand equity," *Journal of the Academy of Marketing Science*, vol. 28, no. 2, pp. 195–211, 2000.

[4] J. L. Lee, J. D. James, and Y. K. Kim, "A Reconceptualization of Brand Image," *International Journal of Business Administration*, vol. 5, no. 4, pp. 1–11, 2014.

[5] A. Rangaswamy, R. R. Burke, and T. A. Oliva, "Brand equity and the extendibility of brand names," *International Journal of Research in Marketing*, vol. 10, no. 1, pp. 61–75, 1993.

[6] B. Yoo and N. Donthu, "Developing and validating a multidimensional consumer-based brand equity scale," *Journal of Business Research*, vol. 52, no. 1, pp. 1–14, 2001.

[7] A. L. Biel, "How brand image drives brand equity," *Journal of Advertising Research*, vol. 32, no. 6, pp. 6–12, 1992.

[8] Y. Zhang, "The Impact of Brand Image on Consumer Behavior: A Literature Review," *Open Journal of Business and Management*, vol. 03, no. 01, pp. 58–62, 2015.

[9] M. L. Barnett, J. M. Jermier, and B. A. Lafferty, "Corporate Reputation: The Definitional Landscape," *Corporate Reputation Review*, vol. 9, no. 1, pp. 26–38, 2006.

[10] I. Black and C. Veloutsou, "Working consumers: Co-creation of brand identity, consumer identity and brand community identity," *Journal of Business Research*, vol. 70, pp. 416–429, 2017.

[11] D. A. Aaker, "Measuring Brand Equity Across Products and Markets," *California Management Review*, vol. 38, no. 3, pp. 102–120, 1996.

[12] J. M. Balmer and S. A. Greyser, "Corporate marketing: Integrating corporate identity, corporate branding, corporate communications, corporate image and corporate reputation," *European Journal of Marketing*, vol. 40, no. 7-8, pp. 730–741, 2006.

[13] M. E. Malik, M. M. Ghafoor, and H. K. Iqbal, "Impact of Brand Image and Advertisement on Consumer Buying Behavior," *World Applied Sciences Journal*, vol. 23, no. 1, pp. 117–122, 2013.

[14] K. Khasawneh and A. B. I. Hasouneh, "The effect of familiar brand names on consumer behaviour: A Jordanian perspective," *International Research Journal of Finance and Economics*, vol. 43, pp. 33–56, 2010.

[15] A. E. Price, "How Brand Name and Packaging Quality Affect the Consumer Choice Process," pp. 1–51, 2010.

[16] A. B. Del Río, R. Vázquez, and V. Iglesias, "The effects of brand associations on consumer response," *Journal of Consumer Marketing*, vol. 18, no. 5, pp. 410–425, 2001.

[17] W. C. Kim and R. Mauborgne, *Blue ocean strategy, expanded edition: How to create uncontested market space and make the competition irrelevant*. Harvard business review Press, 2014.

[18] K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee, "A design science research methodology for information systems research," *Journal of management information systems*, vol. 24, no. 3, pp. 45–77, 2007.

[19] R. Church, "New perspectives on the history of products, firms, marketing, and consumers in Britain and the United States since the mid-nineteenth century," *Economic History Review*, vol. 52, no. 3, pp. 405–435, 1999.

[20] A. W. White, *The elements of graphic design: space, unity, page architecture, and type*. Skyhorse Publishing, Inc., 2011.

[21] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[22] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.

[23] Y. Zhang, H. Sun, J. Zuo, H. Wang, G. Xu, and X. Sun, "Aircraft type recognition in remote sensing images based on feature learning with conditional generative adversarial networks," *Remote Sensing*, vol. 10, no. 7, p. 1123, 2018.

[24] G. Yang, S. Yu, H. Dong, G. Slabaugh, P. L. Dragotti, X. Ye, F. Liu, S. Arridge, J. Keegan, Y. Guo *et al.*, "Dagan: Deep de-aliasing generative adversarial networks for fast compressed sensing mri reconstruction," *IEEE transactions on medical imaging*, vol. 37, no. 6, pp. 1310–1321, 2017.

[25] D. Michelsanti and Z.-H. Tan, "Conditional generative adversarial networks for speech enhancement and noise-robust speaker verification," *arXiv preprint arXiv:1709.01703*, 2017.

[26] O. Kupyn, V. Budzan, M. Mykhailych, D. Mishkin, and J. Matas, "Deblurgan: Blind motion deblurring using conditional adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8183–8192.

[27] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional gans," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8798–8807.

[28] I. Goodfellow, "Nips 2016 tutorial: Generative adversarial networks," *arXiv preprint arXiv:1701.00160*, 2016.

[29] Google, "GAN Training | Generative Adversarial Networks | Google Developers." [Online]. Available: https://developers.google.com/machine-learning/gan/training

[30] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[31] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of gans for improved quality, stability, and variation," *arXiv preprint arXiv:1710.10196*, 2017.

[32] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1125–1134.

[33] N. Takano and G. Alaghband, "Srgan: Training dataset matters," *arXiv preprint arXiv:1903.09922*, 2019.

[34] R. A. Yeh, C. Chen, T. Yian Lim, A. G. Schwing, M. Hasegawa-Johnson, and M. N. Do, "Semantic image inpainting with deep generative models," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5485–5493.

[35] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. N. Metaxas, "Stackgan++: Realistic image synthesis with stacked generative adversarial networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 8, pp. 1947–1962, 2018.

[36] Y. Liang, W. Liu, K. Liu, and H. Ma, "Automatic generation of textual advertisement for video advertising," in *2018 IEEE Fourth International Conference on Multimedia Big Data (BigMM)*. IEEE, 2018, pp. 1–5.

[37] A. Jahanian, J. Liu, Q. Lin, D. Tretter, E. O'Brien-Strain, S. C. Lee, N. Lyons, and J. Allebach, "Recommendation system for automatic design of magazine covers," in *Proceedings of the 2013 international conference on Intelligent user interfaces*, 2013, pp. 95–106.

[38] S. Kobayashi, "The aim and method of the color image scale," *Color research & application*, vol. 6, no. 2, pp. 93–107, 1981.

[39] P. Sahare and S. B. Dhok, "Review of Text Extraction Algorithms for Scene-text and Document Images," *IETE Technical Review (Institution of Electronics and Telecommunication Engineers, India)*, vol. 34, no. 2, pp. 144–164, 2017. [Online]. Available: http://dx.doi.org/10.1080/02564602.2016.1160805

[40] H. Zhang, K. Zhao, Y. Z. Song, and J. Guo, "Text extraction from natural scene image: A survey," *Neurocomputing*, vol. 122, pp. 310–323, 2013. [Online]. Available: http://dx.doi.org/10.1016/j.neucom.2013.05.037

[41] Z. Liu, Q. Shen, and C. Wang, "Text Detection in Natural Scene Image with Text Line Construction," *IEEE International Conference on Information Communication and Signal Processing*, vol. 44, no. 12, pp. 2113–2141, 2018.

[42] S. Tian, Y. Pan, C. Huang, S. Lu, K. Yu, and C. L. Tan, "Text flow: A unified text detection system in natural scene images," *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2015 Inter, pp. 4651–4659, 2015.

[43] K. Wang, B. Babenko, and S. Belongie, "End-to-end scene text recognition," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1457–1464, 2011.

[44] B. Epshtein, E. Ofek, and Y. Wexler, "Detecting Text in Natural Scenes with Stroke Width Transform," *Microsoft Corporation*, no. October, pp. 2963–2970, 2010. [Online]. Available: https://www.microsoft.com/en-us/research/publication/stroke-width-transform/

[45] H. Cho, M. Sung, and B. Jun, "Canny Text Detector: Fast and Robust Scene Text Localization Algorithm," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-Decem, pp. 3566–3573, 2016.

[46] W. Huang, Z. Lin, J. Yang, and J. Wang, "Text localization in natural images using stroke feature transform and text covariance descriptors," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1241–1248, 2013.

[47] S. Zhang, M. Lin, T. Chen, L. Jin, and L. Lin, "Character proposal network for robust text extraction," *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, vol. 2016-May, pp. 2633–2637, 2016.

[48] S. Grover, K. Arora, and S. K. Mitra, "Text extraction from document images using edge information," *Proceedings of INDICON 2009 - An IEEE India Council Conference*, 2009.

[49] J. Yan and X. Gao, "Detection and recognition of text superimposed in images base on layered method," *Neurocomputing*, vol. 134, pp. 3–14, 2014. [Online]. Available: http://dx.doi.org/10.1016/j.neucom.2012.12.070

[50] H. I. Koo and D. H. Kim, "Scene text detection via connected component clustering and nontext filtering," *IEEE Transactions on Image Processing*, vol. 22, no. 6, pp. 2296–2305, 2013.

[51] D. Karatzas, F. Shafait, S. Uchida, M. Iwamura, L. G. i Bigorda, S. R. Mestre, J. Mas, D. F. Mota, J. A. Almazan, and L. P. De Las Heras, "Icdar 2013 robust reading competition," in *2013 12th International Conference on Document Analysis and Recognition*. IEEE, 2013, pp. 1484–1493.

[52] S. M. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong, R. Young, K. Ashida, H. Nagai, M. Okamoto, H. Yamamoto *et al.*, "Icdar 2003 robust reading competitions: entries, results, and future directions," *International Journal of Document Analysis and Recognition (IJDAR)*, vol. 7, no. 2-3, pp. 105–122, 2005.

[53] Y. Baek, B. Lee, D. Han, S. Yun, and H. Lee, "Character region awareness for text detection," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2019-June, pp. 9357–9366, 2019.

[54] J. Baek, G. Kim, J. Lee, S. Park, D. Han, S. Yun, S. J. Oh, and H. Lee, "What is wrong with scene text recognition model comparisons? dataset and model analysis," *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2019-October, pp. 4714–4722, 2019.

[55] J. Canny, "A computational approach to edge detection," *IEEE Transactions on pattern analysis and machine intelligence*, no. 6, pp. 679–698, 1986.

[56] C. Harris and M. Stephens, "A Combined Corner and Edge Detector," pp. 23.1–23.6, 1988.

[57] J. Shi and C. Tomasi, "Good features to track," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1994, pp. 593–600.

[58] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 839–846, 1998.

[59] K. McLAREN, "XIII—The Development of the CIE 1976 (L* a* b*) Uniform Colour Space and Colour-difference Formula," *Journal of the Society of Dyers and Colourists*, vol. 92, no. 9, pp. 338–341, 1976.

[60] S. Sural, G. Qian, and S. Pramanik, "Segmentation and histogram generation using the HSV color space for image retrieval," *IEEE International Conference on Image Processing*, vol. 2, pp. 589–592, 2002.

[61] D. J. Bora, A. K. Gupta, and F. A. Khan, "Comparing the Performance of L*A*B* and HSV Color Spaces with Respect to Color Image Segmentation," vol. 5, no. 2, pp. 192–203, 2015. [Online]. Available: http://arxiv.org/abs/1506.01472

[62] ICDAR, "Results - Focused Scene Text - Robust Reading Competition." [Online]. Available: https://rrc.cvc.uab.es/?ch=2&com=evaluation&task=1