Multimedia Tools and Applications (2022) 81:25051–25073
https://doi.org/10.1007/s11042-022-12416-y

Generating personalized business card designs from images

Nuno Antunes[1,2] · João Carlos Ferreira[1,2] · Elsa Cardoso[1]

Received: 2 December 2020 / Revised: 4 April 2021 / Accepted: 25 January 2022 /
Published online: 22 March 2022

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

✉ Nuno Antunes
nuno_francisco@iscte.ptnuno

João Carlos Ferreira
joao.carlos.ferreira@iscte.pt

Elsa Cardoso
elsa.cardoso@iscte.pt

[1] ISTAR, Instituto Universitário de Lisboa (ISCTE-IUL), 1649-026 Lisboa, Portugal

[2] INOV Instituto de Engenharia de Sistemas e Computadores Inovação, Rua Alves Redol, 9, 1000-029 Lisbon, Portugal

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 [13]. 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. Brand image is defined by Kim et al [19] 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.". 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. Focusing on a competent marketing and advertising plan that can display the business identity to the segment can be a highly effective strategy to align these two concepts and contribute to higher brand equity.

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), such as 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. 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. Restaurants, hostels and small shops are often family businesses with many years in the active, means that occasionally business cards are required to be redesigned or reprinted, when the stock runs out. 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 scope of the project is to provide a prototype that can automate the creation and personalization of business card designs based on an existing business card. The automatic generation of customized printed advertisement is especially important for two main actors: design and printing companies, which are provided with a system that allows the clients to more easily prototype and decide on a design that meets their requirements and expectations, and the clients themselves, which benefit from a more affordable solution that is guaranteed to match the selected generated design.

The possibility of generating new business card designs with the upload of a couple images and instantly sending the preferred generated style to print makes it very convenient for the client, which does not have to go through an iterative design process, saving the client time and money with the designer. Additionally, it is a major asset for design and printing firms, as they can offer a service which is a waypoint for more client orders. This way, the service benefits both the consumer and the provider.

This study presents a problem-solving approach to the research question. For this exploratory study, we utilised an experimental research method, presenting a theoretical development of the implemented solution and evaluation of its applicability.

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. 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. The prototype 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 present work provides a contribution to the fields of computer vision and Natural Language Processing, by combining existing techniques on both areas in order to create an innovative system that can generate designs based on a natural image of a business card design by extracting the colour and text with computer vision and processing the extracted text.

2 Related work

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.

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))

The results were exported to excel and further analysed and refined. The queries retrieved 42 and 239 documents respectively. 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 to 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.

Liang et al. [20] propose the automatic generation of textual advertisement for video advertising. The scope of the work was to automatically generate and insert textual advertisement 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. This was achieved by the use of visual significance estimators, using human face localization and saliency detection, in order to estimate the busy and important parts of video, on which text should not be inserted.

Jahanian et al. [12] 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 [16], 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.

Despite the found works relating design generation with marketing, we could not find any published papers that attempt to solve the research question at aim by the present work.

2.1 Image analysis

Wang et al. [30] propose a system for image-to-image translation, which transforms sketch images into natural images of its representation. For this task two CNNs are used, which attempt to retrieve category information from the input. This information is then utilised for re-ranking, using a category similarity measurement in order to re-rank the categories according to the similarity between the input sketch and the retrieval result. Nogueira et al. [6] propose an automatic system for image captioning based on an encoder-decoder structure that can extract image features and multimodal gated recurrent units for image caption generation. These approaches pose some problems for our system. Firstly, not always the most visually similar design to the one submitted is the best for the client, as he might want to change styles. In this regard, attempting to find a similarity between the input card and alternative designs would not always yield the best result. Second, the nature of these printables, being rich in textual information and often with no distinct images of what the business card entails, which means such systems would mostly fail to categorize the business card with the present information. Finally, as further explained in Section 5.1, training an algorithm of this type would require a sizeable dataset.

2.2 Image generation

Recent advancements in conditional image generation, with the insurgence of new types of GANs, such as Conditional GANs [24], Deep Convolutional GANs [25], Progressive Growing GANs [15], amongst others, allowed for a substantial rise of interest in the technology and its applicability to other fields such as aviation [36], medicine [33], communication [23], image manipulation [18, 31] and image-to-image translation [11]. However the specificity of the research question poses some problems for the usage of a GAN. Considering the business card designs are tailored to the specific client, with the exact logotype desired by the user, and since a big part of a business card design is text, most of the features necessary to generate a new business card would not be features that a GAN could learn.

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. [26] 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.

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 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 [34], 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 [21] 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 [21]. 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. [28] 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 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. [29]. 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 [7]:

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 [7]. 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 [28], 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. [5], Epshtein et al. [7] and Huang et al. [10]. 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 [35], 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. [8], 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 [32]

Connected components As stated by Liu, Shen and Wang [21], 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 [17] 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 [7]. 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. [14], 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 [22]. We have compared the results from the text location task, deciding for the Craft++ text detector [2] with the highest recall score of 94.36%. We only considered methods that have been featured in published papers.

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 [1], as it was the best performer out of the 28 compared methods, being the only method with a published paper.

3 Framework design and development

In order to respond to the present research question, it was required to conduct the following tasks, represented by the modules depicted in Fig. 1.

1. Preprocessing the input image, removing its background and adjusting the perspective of the card (angle between the camera and the surface where the business card is laying), obtaining a projection of the object into a rectangular plane. Using these techniques, we discard the maximum amount of pixels that do not belong to the business card design;

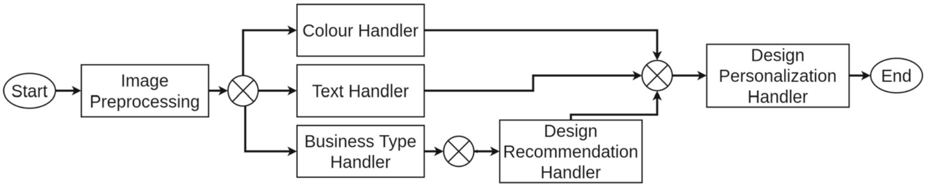


Fig. 1 High-level Architecture

2. Extracting the most prominent colours from the image and building a colour palette;
3. Finding, extracting and interpreting text in the image, categorizing each excerpt of text according to the portrayed information into one of the following categories: email, phone number, website, address, Facebook, Twitter.
4. Finding the company's business type;
5. Recommending a design template based on the business type;
6. Personalizing the design template with colour, textual information and a submitted logotype image automatically.

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 personalization.

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, the text coming from the text handler module and a logotype image file, uploded by the user. A customized image of an alternative design for the business card, based on the submitted design, is the output of the system.

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. 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;
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 a prototype, it is crucial to allow for easy improvements, which this architecture does. Examples of said improvements are described in Section 5.3, Future Work.

This architecture also poses a major disadvantage. The accumulating error between modules means that each module is impacted by the error of all previous module operations. This can lead to very poor results from all other modules if, for example, the Image Preprocessing module fails.

3.1 Image preprocessing

In order to prepare the image, we perform two consecutive steps, as seen in Fig. 2. First,

1. we 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. 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 was zero.

3.1.1 Background segmentation

In order to segment the background from the business card, we must find its location in the image. Since the object we are dealing with has the shape of a quadrilateral due to the possible perspective slant of the photo, it is defined by four non collinear points.. On account of the angle of the camera, the projection of the business card onto the image plane is not necessarily a rectangle. Once we find the four points that correspond to the vertices of the object, we apply a binary mask in order to discard non relevant pixels. This pipeline can be seen in Fig. 3. We first apply bilateral smoothing, an edge-preserving noise-smoothing function, to remove excessive noise from the image. Following, we apply the Canny edge detection [4] to find the quadrilateral contour of the business card, defined by a closed area defined by four points, with brightness values superior to 0 in every pixel of the perimeter edges. We assume the largest polygon defined by four non-collinear points to be representative of the business card. In Fig. 4, we can observe the module in action, first calculating the edge map and then inferring the appropriate polygon. Finally, we apply a binary mask, removing all background information from the business card. On the tests conducted, an edge-based approach outperformed other techniques based on the Harris [9] and Shi & Tomasi's GFTT [27] corner detectors.

3.1.2 Geometric perspective transform

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 morph 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 0º. 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 background segmentation

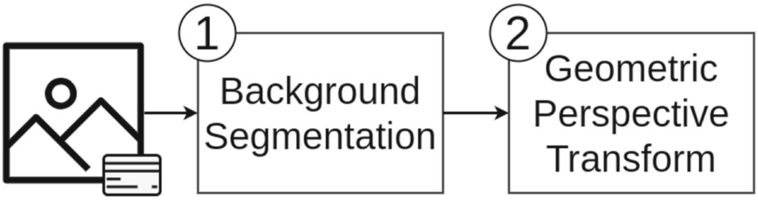


Fig. 2 Overview of the Image Preprocessing Module

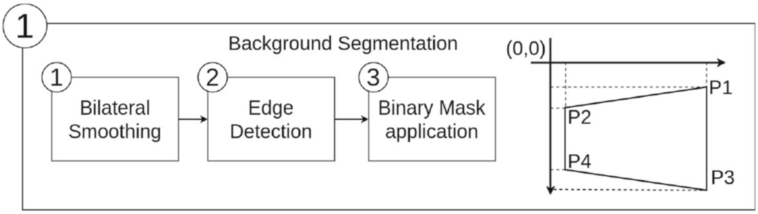


Fig. 3 Overview of the Background Segmentation Process

process, described in Section 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 Fig. 5.

1) Four Corner Mapping We first map 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}),

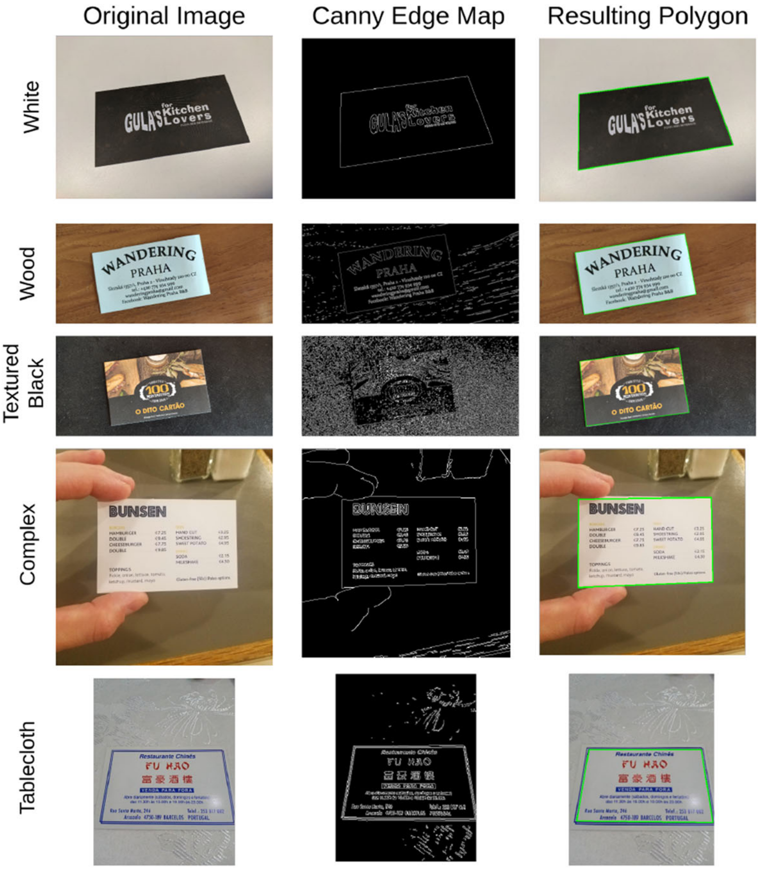


Fig. 4 Edge Detection Results on test cases

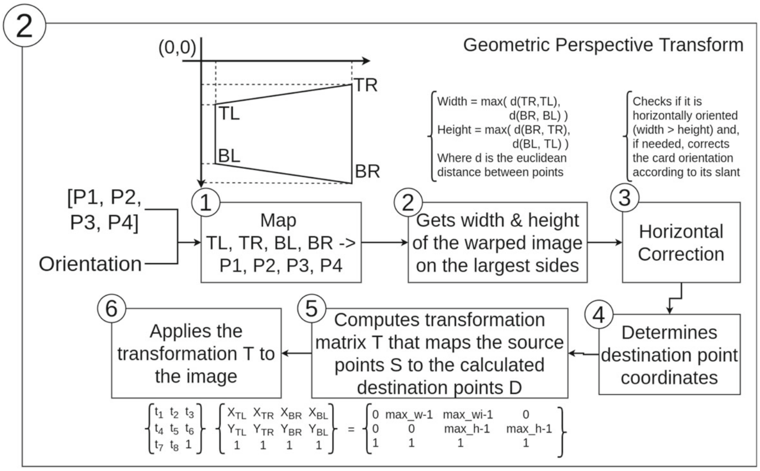

2
Geometric Perspective Transform
(0,0)
TL
TR
BL
BR
Width = max(d(TR,TL), d(BR, BL))
Height = max(d(BR, TR), d(BL, TL))
Where d is the euclidean distance between points
Checks if it is horizontally oriented (width > height) and, if needed, corrects the card orientation according to its slant
[P1, P2, P3, P4]
Orientation
1
Map TL, TR, BL, BR -> P1, P2, P3, P4
2
Gets width & height of the warped image on the largest sides
3
Horizontal Correction
4
Determines destination point coordinates
5
Computes transformation matrix T that maps the source points S to the calculated destination points D
6
Applies the transformation T to the image
t1 t2 t3 / t4 t5 t6 / t7 t8 1
XTL XTR XBR XBL / YTL YTR YBR YBL / 1 1 1 1
=
0 max_w-1 max_wi-1 0 / 0 0 max_h-1 max_h-1 / 1 1 1 1

Fig. 5 Geometric Perspective Transform Process

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 Fig. 5, 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, computing 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 $X_{BL} < Y_{BL}$.

2) Width and Height Calculation The width of the destination image is calculated as being the max(dist(TL, TR), dist(BL, BR)). The height of the card is set to be the max(dist(TR, BR), dist(TL, BL)).

3) Orientation Correction We first check if the card is already horizontally oriented. A horizontal card would mean the calculated value for width should be higher than the calculated value for height. If that is not the case, it means the card is slanted further than 45 degrees. In order to rotate it accordingly, we assess if we are facing a left or right slant, by comparing the x coordinates of the corner points. 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. 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 Fig. 6.

4) Determining Destination Point Coordinates Regardless of the input, since we intend to disregard all background information, the destination image will have the same width and height as before calculated. Our coordinate mapping will, therefore, correspond to the following:

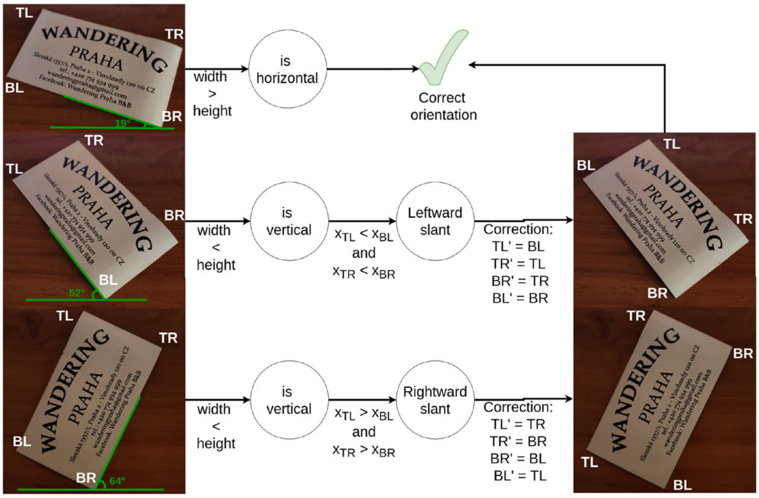


Fig. 6 Applied example of the Horizontal Orientation Correction submodule

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

5) Computing transformation matrix 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 object perspective. The premise is to find the transformation matrix T which maps every point p from matrix S, to a point q in matrix D, 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. 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.

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

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

6) Applying the perspective warp The final step of the module corresponds to applying the calculated transformation matrix T to the entire image.

3.2 Colour handler

Considering we no longer have background pixels in the image, we now extract the main colours in the business card from the RGB image. As we are dealing with natural images with non controlled lighting conditions, the business card in the image is often 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.

In order to find the most predominant colours we performed colour reduction operations, so that we can transform the colour space to a small subset of colours that most represent the input image. The colour reduction was performed by applying a k-means algorithm to the HSV colour space, with 5 initial centroids, progressively grouping pixels that are close to each other in the colour space. The algorithm was run 10 times in order to mitigate the problem of local optima. The extracted colour palette corresponds to the HSV colour value of each found centroid. We have decided to work with HSV values as, according to Bora et al. [3], the HSV colour space performed better than CIE L* a* b* in colour segmentation tasks.

The colours are then converted to the HSL colour space, where any colour with lightness values under 15 or over 85 is disregarded due to it being too bright or too dark to be used in the final design.

3.3 Text handler

This module's 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 three tasks that must be performed consecutively. First, described in Section 3.3.1,

1. we attempt to detect text present in the image and label it with the depicted word. Then, as shown in Section 3.3.2,
2. we merge the extracted text into text boxes. Finally, in Section 3.3.3
3. we label the extracted text according to the information it portrays.

In Fig. 7 we can observe, in high-level, the text extraction diagram, performed prior to text tagging.

3.3.1 Text detection and text recognition

The problem of extracting text from natural images is split up into two distinct thesis: text detection and text recognition. Firstly,

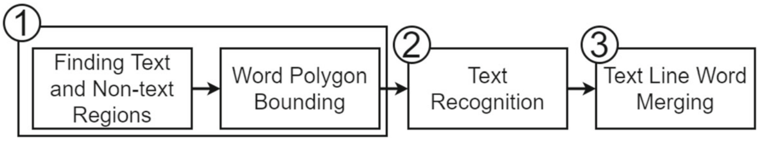


Fig. 7 General Text extraction Flow Diagram

1. 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. Image portions classified as having text are then
2. merged together and encompassed in the same polygon if they are close to each other. The found words, defined by a series of pixels encompassing a word inside the bounding polygon will
3. be used as input to the text recognition module, which will output the predicted word for each of the found regions.

Text detection corresponds to finding the exact location of text characters in the images, in order to remove non-character pixels, as seen if Fig. 8. We have applied the CRAFT text detector [2] 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, used for text merging rules, explained in Section 3.3.2. Finally, we apply a binary mask to isolate individual words (3).

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 [1] to predict the word present in each bounding polygon and the probability associated to it.

3.3.2 Text merging

This module merges words into text boxes according to its position in the business card. The importance of merging words into text boxes comes from the fact text must be interpreted as full sentences to tag excerpts according to the information they portray.

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;
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

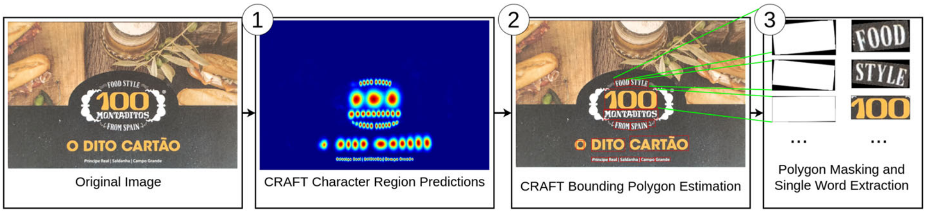

1
2
3
FOOD STYLE
100
MONTADITOS
FROM SPAIN
O DITO CARTÃO
Original Image
CRAFT Character Region Predictions
CRAFT Bounding Polygon Estimation
FOOD
STYLE
100
...
...
Polygon Masking and Single Word Extraction

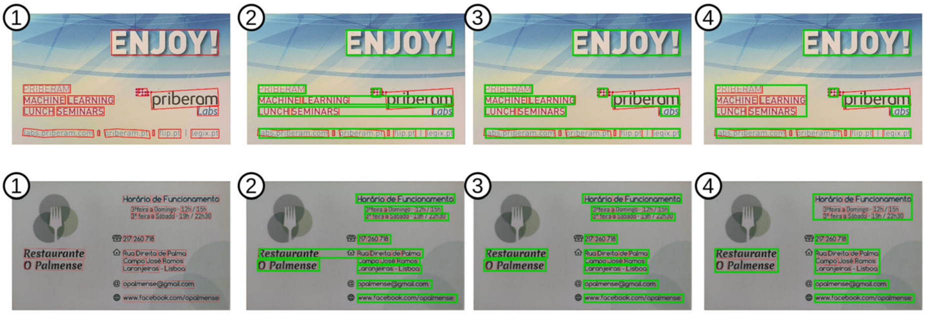
Fig. 8 Text detection Flow Diagram

respectively. All of these values are min-max normalized to account for variances in image size and image aspect ratio. Four consecutive steps are performed, as seen in Fig. 9. First

1. Y Position Ordering: We first apply quicksort to order the list according to the y position of each bounding box;
2. 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. 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. The threshold was set to a distance of 0.01. The resulting vector will have LeftX $S^{(i)}_{LeftX}$ and RightX to be $S^{(i)}_{RightX}$. 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.
3. 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 is defined as $S^{(i+1)}_{LeftX} - S^{(i)}_{RightX}$. The threshold for text-line splitting was set as a word distance of 0.12. 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. 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.
4. Text box Merging: Text present in different lines is finally recursively merged into text boxes. We utilise five measurements for text merging:

   (a) BottomY, bottom edge y coordinate, where $S^{(i)}_{BottomY} = S^{(i)}_y + \frac{1}{2}S^{(i)}_h$;

   (b) TopY, top edge y coordinate, where $S^{(i)}_{TopY} = S^{(i)}_y - \frac{1}{2}S^{(i)}_h$;



Original Business card after word extraction. The individual words are highlighted in red.

Words are grouped into lines according to y position. Arrays are sorted according to x position, using quicksort

Arrays are split iteratively according to x distance and bounding box size between adjacent words.

Arrays are merged iteratively into text boxes according to x position (beneath each other), y position (close to each other) and bounding box size

Fig. 9 Word ordering and merging flow diagram

(c) AvgH, average text box height on a text excerpt;
(d) LeftX, left bound x coordinate;
(e) RightX, the right bound x coordinate.

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:

(a) $(S^{(i)}_{AvgH} - S^{(j)}_{AvgH})^2 < t_h$, the squared average font size between text excerpts is under a predefined threshold t_h, set to a distance of 0.02 between font sizes;
(b) $(S^{(i)}_{TopY} - S^{(j)}_{BottomY})^2 < t_y$, the distance between the two text excerpt bounding boxes is under the threshold t_y; 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.
(c) $S^{(j)}_{LeftX} < \frac{1}{2}(S^{(i)}_{LeftX} + S^{(i)}_{RightX}) < S^{(j)}_{RightX}$, 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 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.

3.3.3 Text categorization

The text categorization module attempts to find social network information, namely facebook and twitter, as well as websites, addresses, emails and phone numbers on thee extracted text. In Fig. 10, 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. We have utilised a rule-based approach resorting to regular expressions for socialm network, websites, emails and phone numbers. The address verification searches text for indicator words such as state names, cities, municipalities or words that are often associated to addresses “street”, “st.”, “road”, “rd.”, “avenue”, “av.”, amongst others.

3.4 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 businesses with similar names worldwide, we utilise geolocation information to search for nearby places, in a radius of 40Km, with the name requested. This service outputs a json which includes the business types related to a particular business, in order of relevance.

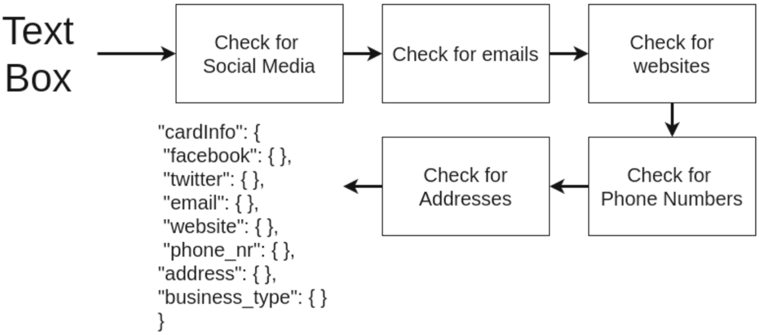


Fig. 10 Text Tagging Flow Diagram

3.5 Design recommendation and personalization

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).

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, an icons folder, an images folder and the design file. Each component folder encompasses the necessary elements to generate the design. The design file contains all code necessary for the generation of the business card. The fonts and icons are loaded and placed in the appropriate location, as specified by the design file. 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.

The output of the module is a suggestion of a custom business card, generated from the original submitted image, as can be observed in Fig. 11.

4 Experimental evaluation

In this section we will provide a demonstration of the full system the design personalization module. The pipeline demonstrated in the present Section correspond to the architecture defined in Section 3

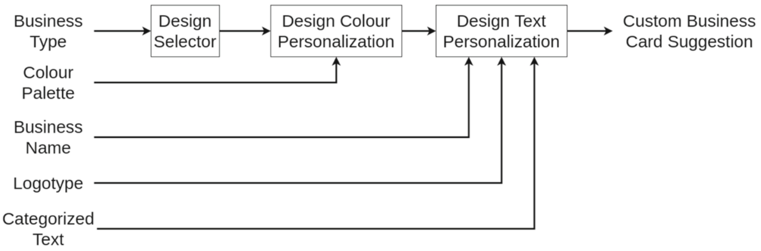

Business Type
Design Selector
Design Colour Personalization
Design Text Personalization
Custom Business Card Suggestion
Colour Palette
Business Name
Logotype
Categorized Text

Fig. 11 Design Personalization Flow Diagram

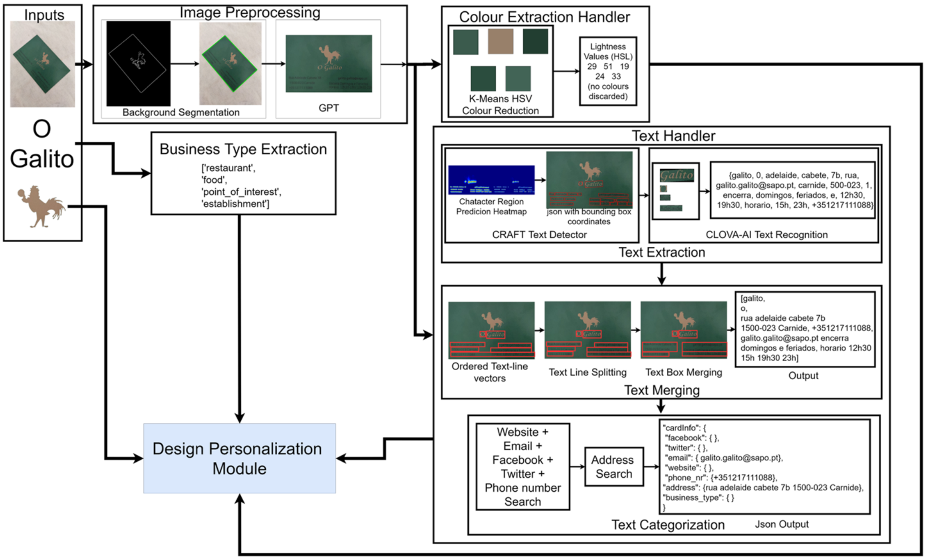
The system inputs are an image of a business card, along with the company name and logo. We can observe all outputs of each stage of the system pipeline in Fig. 12.

The business type is extracted from the input 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, performing 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 was discarded.

The text handler is composed of three consecutive steps:



```
{galito, 0, adelaide, cabete, 7b, rua,
galito.galito@sapo.pt, carnide, 500-023, 1,
encerra, domingos, feriados, e, 12h30,
19h30, horario, 15h, 23h, +351217111088}
```

```
[galito,
o,
rua adelaide cabete 7b
1500-023 Carnide, +351217111088,
galito.galito@sapo.pt encerra
domingos e feriados, horario 12h30
15h 19h30 23h]
```

```
"cardInfo": {
 "facebook": { },
 "twitter": { },
 "email": { galito.galito@sapo.pt},
 "website": { },
 "phone_nr": {+351217111088},
"address": {rua adelaide cabete 7b 1500-023 Carnide},
"business_type": { }
}
```

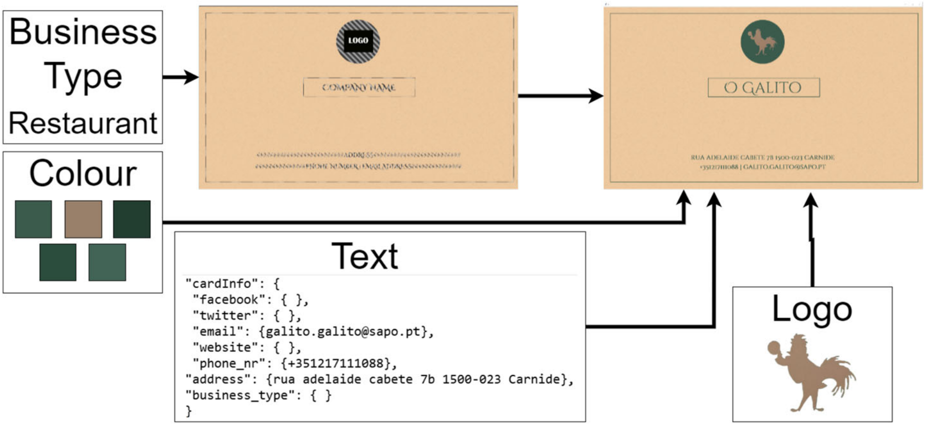
Fig. 12 System Demonstration

1. Firstly, the system extracts 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. We can observe the output words in Fig. 12.
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 Fig. 13. 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 Fig. 13.

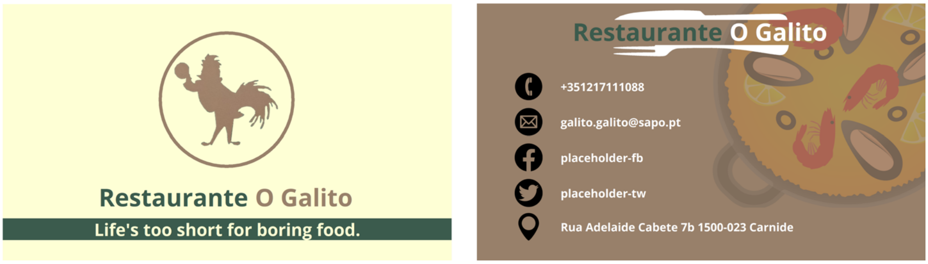
Rerunning the module with the same business card input can result in different solutions. In Fig. 14, we can observe an alternative result of the design personalization module for the same example. In this case, a double-sided business card was generated, with the most represented colour in the input design being used in the word “Restaurante” and the colour bar along the bottom of the front side, the second most represented colour 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


Business
Type
Restaurant
Colour
LOGO
COMPANY NAME
ADDRESS
PHONE NUMBER | EMAIL ADDRESS
O GALITO
RUA ADELAIDE CABETE 7B 1500-023 CARNIDE
+351217111088 | GALITO.GALITO@SAPO.PT
Text
Logo

```
"cardInfo": {
 "facebook": { },
 "twitter": { },
 "email": {galito.galito@sapo.pt},
 "website": { },
 "phone_nr": {+351217111088},
"address": {rua adelaide cabete 7b 1500-023 Carnide},
"business_type": { }
}
```

Fig. 13 Design Personalization Module Demonstration



(a) Business Card front (b) Business Card back

Fig. 14 Generated business card front and back

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 5.2 and 5.3.

5 Conclusion

In the present work, we have successfully developed a prototype that fully answers the research question, 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.

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 through following modules:

1. an image background removal and image geometric perspective transform module in order to remove pixels that are not part of the business card design,
2. a colour extraction module,
3. a text extraction pipeline and
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.

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 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. Future improvement recommendations based on testing are summarized in Section 5.2.

In Section 5.1 we provide a list of the found limitations of the developed project. Finally, in Section 5.3, we present an overview of the possible future work related to this study, including possible directions of research.

5.1 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 bifold. 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. In Section 5.2, we provide recommendations in order to further improve these modules. The lack of real test cases and validation with the requesting company was a limitation for the results shown in Section 4. A thorough evaluation and approval of the requester would have been the ideal scenario.

5.2 Recommendations

In this Section we provide recommendations for the work development based on the tests performed. We have concluded it would be advisable to further improve the background segmentation algorithm in the image preprocessing module, described in 3.1. Currently, due to the lack of image data necessary, as explained in Section 5.1, 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.

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 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.

5.3 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. 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 labeled 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.

Acknowledgements This work was supported by EEA Grants Blue Growth Programme (Call #5). Project PT-INNOVATION-0045 - Fish2Fork.

References

1. Baek J, Kim G, Lee J, Park S, Han D, Yun S, Oh SJ, Lee H (2019) What is wrong with scene text recognition model comparisons? dataset and model analysis. Proceedings of the IEEE International Conference on Computer Vision 2019-October:4714–4722. https://doi.org/10.1109/ICCV.2019.00481
2. Baek Y, Lee B, Han D, Yun S, Lee H (2019) Character region awareness for text detection. Proc IEEE Comput Soc Conf Comput Vision Pattern Recog 2019:9357–9366. https://doi.org/10.1109/CVPR.2019.00959
3. Bora DJ, Gupta AK, Khan FA (2015) Comparing the Performance of L*A*B* and HSV Color Spaces with Respect to Color Image Segmentation. arXiv:1506.01472 5(2):192–203
4. Canny J (1986) A computational approach to edge detection. IEEE Trans Pattern Anal Mach Intell 6:679–698. https://doi.org/10.1109/ASICON.2011.6157287
5. Cho H, Sung M, Jun B (2016) Canny text detector: fast and robust scene text localization algorithm. Poc IEEE Comput Soc Conf Comput Vision Pattern Recog 2016:3566–3573. https://doi.org/10.1109/CVPR.2016.388
6. do Carmo Nogueira T, Vinhal CDN, da Cruz Júnior G, Ullmann MRD (2020) Reference-based model using multimodal gated recurrent units for image captioning. Multimed Tools Appl 79(41-42):30615–30635. https://doi.org/10.1007/s11042-020-09539-5
7. Epshtein B, Ofek E, Wexler Y (2010) Detecting text in natural scenes with stroke width transform. In: 2010 IEEE computer society conference on computer vision and pattern recognition, IEEE, pp 2963–2970
8. Grover S, Arora K, Mitra SK (2009) Text extraction from document images using edge information. Proceedings of INDICON 2009 - An IEEE India Council Conference, https://doi.org/10.1109/INDCON.2009.5409409
9. Harris C, Stephens M (1988) A Combined Corner and Edge Detector, pp 23.1–23.6. https://doi.org/10.5244/c.2.23
10. Huang W, Lin Z, Yang J, Wang J (2013) 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. https://doi.org/10.1109/ICCV.2013.157
11. Isola P, Zhu J-Y, Zhou T, Efros AA (2017) Image-to-image translation with conditional adversarial networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1125–1134
12. Jahanian A, Liu J, Lin Q, Tretter D, O'Brien-Strain E, Lee SC, Lyons N, Allebach J (2013) Recommendation system for automatic design of magazine covers. In: Proceedings of the 2013 international conference on Intelligent user interfaces, pp 95–106

13. Jamal A, Goode MM (2001) Consumers and brands: A study of the impact of self-image congruence on brand preference and satisfaction. Mark Intell Plan 19(7):482–492. https://doi.org/10.1108/02634500110408286
14. Karatzas D, Shafait F, Uchida S, Iwamura M, i Bigorda LG, Mestre SR, Mas J, Mota DF, Almazan JA, De Las Heras LP (2013) Icdar 2013 robust reading competition. In: 2013 12th International conference on document analysis and recognition, IEEE, pp 1484–1493
15. Karras T, Aila T, Laine S, Lehtinen J (2017) Progressive growing of gans for improved quality, stability, and variation. arXiv:1710.10196
16. Kobayashi S (1981) The aim and method of the color image scale. Color Res Appl 6(2):93–107
17. Koo HI, Kim DH (2013) Scene text detection via connected component clustering and nontext filtering. IEEE Trans Image Process 22(6):2296–2305. https://doi.org/10.1109/TIP.2013.2249082
18. Kupyn O, Budzan V, Mykhailych M, Mishkin D, Matas J (2018) Deblurgan: Blind motion deblurring using conditional adversarial networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 8183–8192
19. Lee JL, James JD, Kim YK (2014) A reconceptualization of brand image. Int J Bus Admin 5(4):1–11. https://doi.org/10.5430/ijba.v5n4p1
20. Liang Y, Liu W, Liu K, Ma H (2018) Automatic generation of textual advertisement for video advertising. In: 2018 IEEE Fourth international conference on multimedia big data (BigMM), IEEE, pp 1–5
21. Liu Z, Shen Q, Wang C (2018) Text Detection in Natural Scene Image with Text Line Construction. IEEE Int Conf Inf Commun Sign Process 44(12):2113–2141. https://doi.org/10.16383/j.aas.2018.c170572
22. Lucas SM, Panaretos A, Sosa L, Tang A, Wong S, Young R, Ashida K, Nagai H, Okamoto M, Yamamoto H et al (2005) Icdar 2003 robust reading competitions: entries, results, and future directions. IJDAR 7(2-3):105–122
23. Michelsanti D, Tan Z-H (2017) Conditional generative adversarial networks for speech enhancement and noise-robust speaker verification. arXiv:1709.01703
24. Mirza M, Osindero S (2014) Conditional generative adversarial nets. arXiv:1411.1784
25. Radford A, Metz L, Chintala S (2015) Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv:1511.06434
26. Sahare P, Dhok SB (2017) Review of Text Extraction Algorithms for Scene-text and Document Images. IETE Tech Rev (Inst Electron Telecommun Eng, India) 34(2):144–164. https://doi.org/10.1080/02564602.2016.1160805
27. Shi J, Tomasi C (1994) Good features to track. In: Proceedings of the IEEE computer society conference on computer vision and pattern recognition, pp 593–600
28. Tian S, Pan Y, Huang C, Lu S, Yu K, Tan CL (2015) Text flow: A unified text detection system in natural scene images. Proc IEEE Int Conf Comput Vis 2015 Inter:4651–4659. https://doi.org/10.1109/ICCV.2015.528
29. Wang K, Babenko B, Belongie S (2011) End-to-end scene text recognition. Proceedings of the IEEE International Conference on Computer Vision, pp 1457–1464. https://doi.org/10.1109/ICCV.2011.6126402
30. Wang L, Qian X, Zhang Y, Shen J, Cao X (2020) Enhancing Sketch-Based Image Retrieval by CNN Semantic Re-ranking. IEEE Trans Cybern 50(7):3330–3342. https://doi.org/10.1109/TCYB.2019.2894498
31. Wang T-C, Liu M-Y, Zhu J-Y, Tao A, Kautz J, Catanzaro B (2018) High-resolution image synthesis and semantic manipulation with conditional gans. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 8798–8807
32. Yan J, Gao X (2014) Detection and recognition of text superimposed in images base on layered method. Neurocomputing 134:3–14. https://doi.org/10.1016/j.neucom.2012.12.070
33. Yang G, Yu S, Dong H, Slabaugh G, Dragotti PL, Ye X, Liu F, Arridge S, Keegan J, Guo Y et al (2017) Dagan: Deep de-aliasing generative adversarial networks for fast compressed sensing mri reconstruction. IEEE Trans Med Imaging 37(6):1310–1321
34. Zhang H, Zhao K, Song YZ, Guo J (2013) Text extraction from natural scene image: a survey. Neurocomputing 122:310–323. https://doi.org/10.1016/j.neucom.2013.05.037
35. Zhang S, Lin M, Chen T, Jin L, Lin L (2016) Character proposal network for robust text extraction. ICASSP, IEEE Int Conf Acoust Speech Sign Process- Proc 2016:2633–2637. https://doi.org/10.1109/ICASSP.2016.7472154
36. Zhang Y, Sun H, Zuo J, Wang H, Xu G, Sun X (2018) Aircraft type recognition in remote sensing images based on feature learning with conditional generative adversarial networks. Remote Sens 10(7):1123