FACULTY OF ENGINEERING AND ARCHITECTURE

Do you recognize your choice? An analysis of the influence of image properties and similarities on choice blindness

Ward Vereecken
Student number: 01713073

Supervisors: Prof. dr. ir. Pieter Simoens, Dr. Yara Khaluf
Counsellor: Ir. Stef Van Havermaet

Master's dissertation submitted in order to obtain the academic degree of
Master of Science in Information Engineering Technology

Academic year 2020-2021

GHENT UNIVERSITY

Word of thanks

During an extensive period of six months, I have performed research on Choice Blindness and written my thesis on this subject. It was a period in which I not only learned to build a valid and reliable research, but also to better evaluate, improve and appreciate my work. I have not done this alone. Therefore, I would like to thank those individuals who have provided, helped, and supported me with wise counsel during my research.

Firstly, I would like to thank my mentor, Dr. Yara Khaluf, for her weekly, indispensable advice. Her mentoring and support helped me power through many challenges. Many important choices and decisions were a result of her counseling, and without her encouragements, I would never have gotten as far as I did. Secondly, I want to thank Ir. Stef Van Havermaet, for providing me with occasional counseling and a tremendous help writing my thesis. I am also grateful to Prof. Dr. Ir. Pieter Simoens for providing advice and feedback for writing my thesis and allowing me the opportunity to provide participants for my study by arranging a budget through the University of Ghent.

Additionally, I would like to express gratitude to Professor Peter Veelaert for providing me with valuable information and explanations through the course "Computervisie". This course supplied me with the tools and inspiration that I needed to tackle the problems and successfully complete my dissertation.

Next to them, I want to thank my family who was always there to listen to me and believe in my work. Not to forget, I want to thank all the people who took their time to test out my study to provide helpful insights and criticism before hiring participants. And lastly, but certainly not least importantly, I want to thank my amazing friends and fellow students for their unwavering support, both technically and emotionally.

Many thanks,
Ward Vereecken

Permission for use of content

Do you recognize your choice? An analysis of the influence of image properties and similarities on choice blindness

Ward Vereecken

Supervisor(s): Dr. Yara Khaluf, Prof. dr. ir. Pieter Simoens

Counsellor: Ir. Stef Van Havermaet

Abstract—**Each day, a person makes hundreds to thousands of choices. But what if these choices turn out to be easily manipulated? Choice blindness is the paradigm in which a person is blind for their own choice, mere seconds after making it. This paper aims to study the influence of face and landscape image properties and similarities on choice blindness, through the means of an online study. The relationship between choice blindness and personal preference, confidence, and option types is also investigated. Multiple scene properties are investigated for correlations with choice blindness chances. To this end, colour histograms, colourfulness, busyness, texture vectors, facial landmark shapes, contrast and sharpness values are extracted. Next to this, three image similarity measures are defined and inspected for correlations with choice blindness chances: global similarity, defined by low-level image features, GIST similarity, defined by the spatial layout of the image, and neural net similarity, a black-box algorithm. Results show GIST similarity correlates the strongest with detection rates in landscape type images, while neural net similarity correlates with detection rates in both face and gray scaled face type images. Of all manipulated questions, 32.05% of undetected and 6.16% of detected manipulations resulted in a change of preference. Furthermore, participants were found to use more time to argue their choice as image similarities increase, indicating possible lower confidence levels. Out of the three option types, landscape images resulted in the lowest detection rate of 33%, as opposed to 41% for grayscaled faces and 50% for coloured faces. Lastly, out of all extracted properties, the only statistically significant correlations were found between texture, contrast, and nose bridge shape distances with detection rates.**

Keywords—**choice blindness, image similarity, image features, introspection**

I. Introduction

In their day to day life, a person makes a substantial amount of choices. To a large extent, people identify themselves, and other people identify them, with their choices. These choices include the choice of their occupation, their life partner, their living area, which political party to support, and even which shoes to buy. The role of choice plays a bigger role in life than most people realize. This role is threatened to be undermined if it were shown that people attribute choices to themselves that they never made and even defend them. Concepts surrounding this theme are widely researched in the psychology field. One of these concepts is **choice blindness**, of which the thought process behind it has not yet been fully understood. Better understanding the psychological effects that underlie human reasoning, decision making and communication, which are often seemingly irrational, is also one of the most daunting challenges that lie in improving human-AI[1] interaction. In this day and age, humans increasingly interact with AI-driven systems, making this field progressively important. For this reason, the choice blindness paradigm is the focus of this thesis.

[1]AI or Artificial Intelligence: the simulation of human intelligence in machines that are programmed to think like humans and mimic their actions

A. Choice blindness

Put concisely, choice blindness is the failure to detect mismatches between intention and outcome in a simple decision task. In the choice blindness paradigm, participants are presented with two choices. After choosing a certain answer, they are presented with manipulated answers (the answer they did not choose). When choice blindness occurs, the participants **fail to notice this manipulation** and even **offer thoughtful reasoning** as to why they chose said (manipulated) answers. In recent years, this effect has received increasing attention as more studies are being performed to unravel the mysteries and scientific reasons behind it. A promising direction might be the relationship between image similarity and choice blindness.

B. Choice blindness and image similarity

The main focus of this thesis revolves around the influence of the similarity between images on the chances that a manipulated question will be noticed, or in other words, that choice blindness occurs. This topic has been previously touched by Taya et al. [1], where it was suggested that similarity is no significant predictor for choice blindness (coeff = 20.02, p = 0.83, t = 20.22). It was, however, shown by Hall et al. [2] that while pairs with exceedingly low similarity get detected considerably more than pairs with fairly high similarity, no correlation was found for pairs in the "gray zone" of similarity. However, an important detail for both these studies, is that similarity rates were not determined by an unbiased algorithm, but rather by the opinions of participants. Thus, to further investigate this topic, this thesis determines the **similarity rates in multiple, unbiased ways**. To be more precise, computer vision algorithms are used and/or combined to calculate the similarity rates between certain image pairs and ultimately look for correlations with choice blindness chances.

To determine similarity in images, image pixels need to be interpreted in some way to retrieve higher-level information, also called **semantic information**. Such high-level information can range from objects to ideas or even the general shape of a scene. For example, they could correlate an extracted colour such as blue with the sea or sky, white with a building, and so on. Low-level information on the other hand, looks at patterns and values

in the pixels, but does not try to extract semantic information from them. Instead, these values are extracted almost raw and then compared at face value. For example, the texture or color histogram of two images could be compared for similarity, or two images could be scanned for recurring corners or edges in the pixel values. Following these notions, this thesis also investigates the difference in influence of semantic vs low-level feature similarities between images on choice blindness chances.

While investigating influencing factors of choice blindness is an interesting research topic, examining what humans really **think during** manipulations might help gaining more insights into the process behind choice blindness. This leads to the following research topic of introspection.

C. Choice blindness and introspection

In this thesis, introspection[2] is researched in two particular ways: **preference change interpretations** and **confidence levels**. The first idea resides in the notion that preference change due to choice blindness could be attributed to two main interpretations: (i) *choice-error*, where the participant gives reasons about the wrong choice and is in fact wrong about what her choice is, or (ii) *choice change*, when the participant gives reasons and she is right about what her choice is, but she does not realize that her choice has been changed [3]. To investigate this notion, manipulated questions are repeated to the same participant to check whether the participants' preferences changed as a consequence of said manipulation. Secondly, introspection in participants is determined by measuring their confidence levels in terms of time used per argumentation. If participants take longer to answer a question, it could be argued that their level of confidence is lower than if they answer relatively quickly, indicating some mechanism of doubt or introspection in their thought process.

D. Choice blindness and difference in option types

Many of the found studies use different methods to analyze the choice blindness effect. However,what seems to be rarely studied, is the difference between different types of options and/or criteria. To give some examples, in a study by Hall et al. [2], the taste and smell of two different consumer goods were used. In another study by Hall et al. [4], political statements were used, and in a study by Steenfeldt-Kristensen and Thornton [5], touch, or haptic sensations were used, etc. However, although these studies sometimes briefly discuss the difference with other studies and methods, actually comparing multiple options and/or criteria types in one study is rarely done. After research on the topic, many option type comparisons were proposed (to give some examples: images with texts or long texts with short texts). In the end, the decision was made to compare different **image scene types** for their influence on choice blindness. More specifically, landscapes, faces, and 'filtered' (grayscaled, blurred) faces are compared among each other.

[2]The Oxford dictionary defines introspection as the careful examination of your own thoughts, feelings and reasons for behaving in a particular way. In other words, it could refer to how much a person is aware of why they chose a certain option when presented with two options. In that sense, it is highly related to the choice blindness paradigm.

E. Conclusion

Choice blindness is a paradigm that could lead to important insights into the human psyche and the mechanisms behind humans' often seemingly illogical thought processes. The goal of this thesis is ultimately to test certain hypotheses about choice blindness in order to gain more insights into the process behind it. The following research questions are defined and attempts to answer them are made through the means of an online study.

- *RQ 1: Which computational features of an image correlate with choice blindness chances?*
  - *RQ 1.1: How do different types of image properties or criteria influence the chances of choice blindness occurring?*
- *RQ 2: How much could choice blindness result in changes in personal preference?*
- *RQ 3: To what degree does image (dis)similarity affect confidence levels of participants?*
- *RQ 4: How do different types of options compare when it comes to the chance of choice blindness occurring?*

II. Methodology

In order to answer these questions as sufficiently as possible, first, multiple image analysis and similarity measures are investigated and defined. Secondly, an online study is designed, implemented and launched. Based on said measures and the results acquired from testing them on image data sets, this study's question structure and order is set up.

A. Image similarity measures

The most commonly used techniques to determine image similarity or perform classification are based on image features. Concisely put, image features define certain properties of groups of pixels in an image. For example, one type of image feature is edges. Edges are points where there is a boundary (or an edge) between two image regions. As previously mentioned, image features can be divided into two main categories: (i) low-level features and (ii) high-level or semantic features. Low-level image features are image characteristics that are captured by computers for the purpose of recognition and classification (such as pixel intensity, pixel gradient orientation, colour distribution, etc.), while semantic image features are the features that are commonly used by humans to describe images (objects, actions, etc.), and might have the most substantial impact when it comes to image preferences for humans (Ibarra et al. [6]).

A.1 Low-level image features

Colour

The colour feature is extracted by calculating the **colour histogram** of the image. A colour histogram is a representation of the distribution of colours in an image. In a digital image, this represents the amount of pixels containing certain colours from a list of colour ranges that span the image's colour space (e.g. RGB). To compare images based on this color histogram, four main methods exist: correlation, chi-squared, intersection and Bhattacharyya distance. Testing results show how correlation, intersection and Bhattacharyya distance are relatively good

measures for colour similarity. After consideration, correlation was chosen as similarity measure, using equation 1 to compare two histograms H_1 and H_2 in an image I.

$$d(H_1, H_2) = \frac{\sum_I (H_1(I) - \bar{H_1})(H_2(I) - \bar{H_2})}{\sqrt{\sum_I (H_1(I) - \bar{H_1})^2 \sum_I (H_2(I) - \bar{H_2})^2}} \quad (1)$$

Texture

To overcome difficulties such as variations in colour distribution, scale, illumination, rotation or affine transform, texture features can also be extracted. Texture analysis is a way of describing the spatial distribution of intensities, which makes it useful in the classification of similar regions in different images. To extract these features, the **local binary pattern histogram** is extracted, which is a simple yet very efficient texture operator which labels the pixels of an image by thresholding the neighbourhood of each pixel and considering the result as a binary number. The histograms are then compared using the same correlation method as previously mentioned.

Image moments

An image's moment represents a particular weighted average (moment) of the image pixels' intensities. Simple properties of the image which are found via image moments include area (or total intensity), its centroid, and information about its orientation. In other words, they capture information about the shape of an object in a binary image because they contain information about the intensity I(x,y), as well as position x and y of the pixels. To calculate moments that are invariant to translation, scale, and rotation, the **Hu moment** needs to be calculated. Hu Moments (Hu moment invariants) are a set of seven numbers calculated using central moments that are invariant to image transformations. Lastly, the log scale of these Hu moments is calculated to achieve a comparable scale.

Local feature descriptors

Local feature descriptors quantify local regions of an image by looking at certain patters, corners, gradients, etc. While these types of features are widely used in image matching and classification, it was found to not be applicable to the problem at hand. This is likely a consequence of this method's tendency to focus on relatively small-scale details in images, which is not necessarily a good indicator of human-based similarity.

Combination

Except for local feature descriptors, all of the above-defined features were combined into one flat vector and compared between images by calculating the euclidean distance between them. This similarity measure is further referred to as the **global** similarity measure.

A.2 High-level or semantic image features

Machine learning

Determining semantic image similarity is somewhat more of a challenge. Machine learning algorithms exist that can transform a multitude of low-level image features into semantic information. Two machine learning models are used, namely the Apple TuriCreate image similarity model for landscapes and the face-recognition Python library facial recognition model for faces. These models use convolutional neural networks, processing the raw pixels information through multiple layers, to finally yield some semantic category for the given input image. The distance based on this output then yields a similarity score of sorts.

GIST features

Another approach of quantifying semantic similarity between images, as proposed by Oliva and Torralba [7], looks at a very low-dimensional representation of the scene, called the **Spatial Envelope**. They propose that a holistic representation of the scene informs about its probable semantic category. In other words, that images could be categorized based on a vague shape of the scene rather than intricate details. The feature descriptor that describes this spatial layout is called the GIST feature descriptor, and is extracted using the Python library Lmgist. Importantly, this GIST descriptor is meaningful to how humans observe images: "the spatial envelope model organizes scene pictures as human subjects do, and is able to retrieve images that share the same semantic category", Oliva and Torralba [7]. Thus, computing a distance between the GIST descriptors of two images could yield an accurate result in terms of semantic similarity, seeing as it gives a notion of the distance between their respective semantic categories.

Other semantic features

To further investigate the influence of semantic similarity on choice blindness, some more specific semantic features are extracted. Firstly, facial landmark coordinates are extracted using the machine learning library described above. These coordinates are used to quantify the similarity of **landmark shapes** between different faces by calculating the moment of these shapes and determining the cosine similarity between them (e.g. the difference in nose shape between two faces). Next to this, a **colourfulness** metric, as defined by Hasler and Suesstrunk [8], is extracted as a semantic feature. This extraction is based on the opponent colour space representation of the image along with the mean and standard deviations of these values. Lastly, a **busyness** metric was defined, attempting to quantify how "busy" or "crowded" an image is. This is acquired by performing bilateral filtering, Otsu's thresholding, erosion, and edge detection on the image, dividing objects from the background and then counting the number of separate "objects".

B. Design of the Online Study

To attempt to answer the proposed research questions, an Online Study is designed and implemented using the PsyToolkit framework, a software package for running psychological experiments, best classified as behaviour experiment software [9]. Two image data sets were retrieved from the internet. The first set contains multiple landscape scenes, ranging from mountainous areas to beaches and deserts. The second set contained a large number of faces, out of which only images were chosen with a uniform, white background, and where the faces faced forward in the same orientation. Question order and structure

were based entirely on the calculated similarity scores as described above. The three main similarity measures, namely global, neural net and GIST similarity, are used in a round-robin fashion to choose which pair of images to manipulate. Next to this, each of the three image types is chosen in a round-robin fashion. In other words, the first question's image pairs could be chosen based on their similarity as calculated by the global similarity measure, while the next pair would be chosen based on their similarity as calculated by the neural net. This leads to the question structure as portrayed in Appendix A.

Due to the psychological nature of this study, several test runs and configurations were necessary before a final configuration was chosen upon for the launch of the study. Small groups of participants were gathered through social media to test the study. Each testing phase consists of two separate sets to test the best configurations. To give an example, in a first testing phase ($N = 16$), manipulation pairs were kept in the 80-100% similarity range, while all none-manipulated questions were uniformly chosen in the 0-100% range. Two sets were created, one where a text box is used to ask the participants for argumentation, and one where radio buttons are used. Table IV in Appendix B shows how text boxes resulted in the lower manipulation detection rate. This could possibly be a result of higher suspicion levels because the radio buttons hint at a sort of manipulation. After all testing phases, text boxes are chosen as detection recognition methods, an artificial delay is added between questions to simulate real-life studies, and the mean of manipulated question pair's similarity scores was chosen to be higher. Please refer to Appendix B for the results of each testing phase. These changes result in a manipulation detection rate comparable to previous studies on choice blindness, making the study ready to be launched.

III. Results

In test phase 3, a detection rate of 34% was observed for 13 participants. The final study ($N = 173$) saw an increase of about 6% to this detection rate. Out of 881 manipulated questions, 356 were detected, resulting in a 40.41% detection rate. In 293 false-positive cases, the participants' answers were categorized as detections while the question was not actually manipulated. Manipulation detections were recognized by filtering argumentations on certain key-words indicating a detection, for example, 'didn't', 'not', 'remember', 'chose', etc..

A. Influence of similarity rates

Most pairs that lied in the lower detection range (5 to 40 %) seemed to also be the questions that, on average, appeared the most in the **first half** of the study. The relationship between question order and detection rate turns out to be direct, suggesting that as pairs turn up later in the study, the chances of their manipulation being detected rise.

Importantly, the correlations between the three main similarity measures and choice blindness chances were investigated. Figure 1 shows each of the three measures and their correlation with detection rates for each of the three image types. These results show that while for landscapes and polished faces, all three correlations remain inverse, face images result in a direct correlation for the global similarity measure.

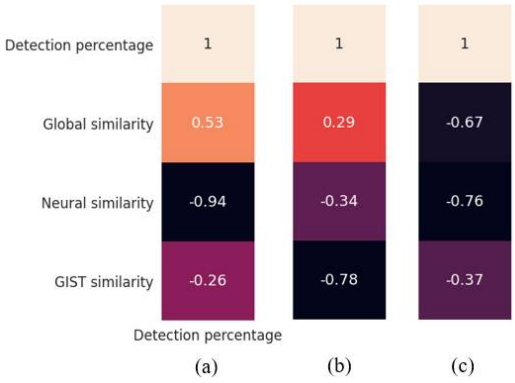


Fig. 1: Extracted columns of correlation matrices of collected data, for (a) face images, (b) landscape image and (c) polished face images.

More precisely, GIST similarity best correlates with detection rates in landscape images, while neural net similarity best correlates with detection rates in face images. Overall, neural net similarity seems to be the best predictor. To quantify these predictors' strength, the R-squared scores of the relationship between these variables are calculated. In statistics, the R-squared value is the proportion of the variance in the dependent variable that is predictable from the independent variable(s) (Carpenter [10]). This value ranges from 0 to 1, indicating either a poor or a very strong prediction factor.

	R-squared value		
Image type	Global	Neural net	GIST
Landscapes	0.0822	0.1148	0.61567
Faces	0.2834	0.8848	0.0675
Polished faces	0.4494	0.5829	0.1361
All	0.0757	0.5482	0.1630

TABLE I: R-squared values for each algorithm-image type combination.

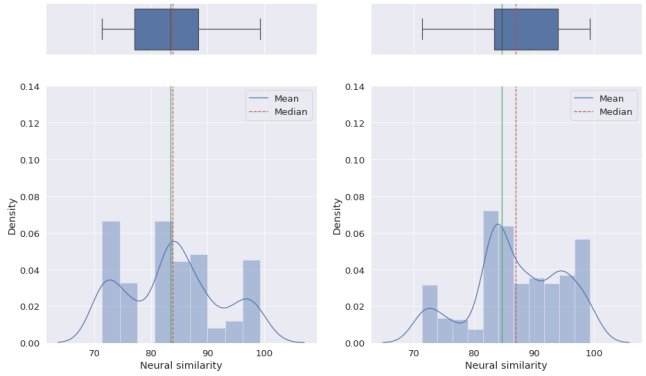
Table I shows the R-squared scores for each of the algorithm-image type combinations. These results suggest the global similarity measure might be a good predictor for detection rates in polished faces. This could be attributed to the fact that the global measure focuses for a big part on textures in the image. As the polished face lack colour differences, participants might be more eager to concentrate on the texture of facial images, because, as discussed by Taya et al. [1], texture is one of the other more prominent features people tend to observe when recognizing faces. The high R-squared score for GIST similarity in landscape images is most likely a result of the fact that the GIST algorithm concentrates on scene shapes and layout, which is of high importance when it comes to similarity between landscapes. The neural net similarity yields a significantly high R-squared score for face images (0.8848), indicating a very high variance explanation by this variable.

Two functions are defined from the linear regression models created from these findings. Function 2 defines the relationship

between detection chances and landscape image similarity as defined by the GIST algorithm. Function 3 then defines the relationship between detection chances and face similarity as determined by the used neural net.

$$y = -44.78321x + 104.72 \tag{2}$$

$$y = -47.1372x + 104.4028 \tag{3}$$



(a) Detected manipulations (b) Undetected manipulations

Fig. 2: Neural net similarity histogram- and box plot for (un)detected manipulations.

Figure 2 illustrates histograms and box plots of neural net similarities for manipulated questions. Concluding from the inter-quartile range of these box plots, exactly half of the undetected manipulations have a similarity rate of 83 to 94, while for detected manipulations, this range is 77 to 88. This indicates that [83-88] might serve as a vague cut-off **range** for manipulation detection in terms of neural network similarity. When inspecting a joint plot of the same questions, detected questions are observed to be in the majority for similarity rates of 79 and under, indicating a more clear cut-off similarity **rate** of 79.

B. Introspection

Averaged over all image types, about one in five manipulations resulted in a change of opinion on the question at hand, regardless of detection state. However, as shown in Table II, when dividing these questions on detection state, about one third of all undetected questions, and even 6.16% of all detected questions, resulted in a preference change. These results might imply the presence of choice change in a lot of cases, where participants effectively changed opinions, and thus are correct about what their choice actually is. When comparing the three image types, landscape type image resulted in the highest opinion change rate of a staggering 37.73%, followed by faces with 31.4% and polished faces with 24.34%. These findings might indicate that as option types are lesser bound to a person's identity (as is likely the case with landscapes as opposed to faces), their likelihood of being choice blind rises.

In terms of time taken to argue their choices, participants seemed to take slightly longer as manipulated option similarity rose. In other words, for all three measures, manipulated image pair similarity seemed to correlate directly with the mean time

	Rate of changed opinions [%]	
Image type	Undetected	Detected
Landscapes	37.73	7.87
Faces	31.4	7.02
Polished faces	24.34	3.44
All	32.05	6.16

TABLE II: Rate of changed opinions for each image type, depending on detection

used to argue for these pairs, possibly indicating a sort of unconscious detection of self-deception, as discussed by Rieznik et al. [11].

C. Option types

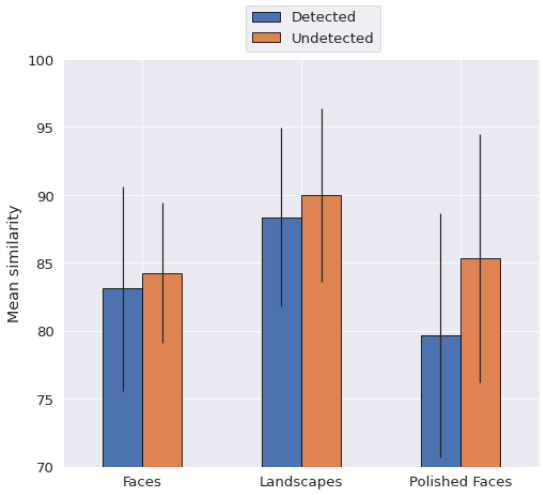


Fig. 3: Mean across all similarity measures bar plotted for (un)detected manipulations for each type.

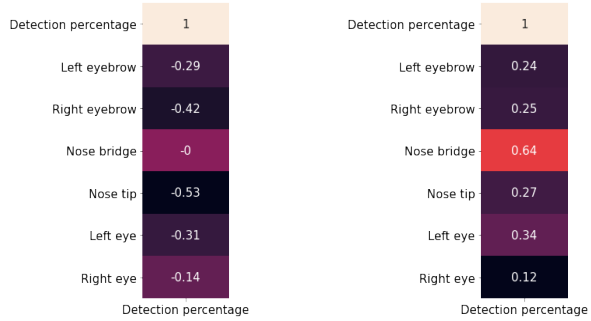
Out of the three option or image types, landscape pairs resulted in the lowest detection rate, followed by polished faces and faces (see Figure 3). Also, the deviation of the detection percentages seems to be relatively low, indicating a more steady and predictable detection rate for landscapes than for faces and polished faces. This result supports the conclusions drawn from subsection III-B, where images of type landscape showed a significantly higher rate of opinion alteration, indicating that opinions on landscapes might have a smaller impact on a person's self-identity than opinions on faces, resulting in a more loose attitude towards them. Next to this, from this graph can be concluded that, for this study, undetected manipulations indeed had a higher mean similarity than detected manipulations. Also, for polished faces, this difference seems to be the highest.

D. Semantic properties

Lastly, the aforementioned properties are analyzed for correlations or interesting patterns in terms of influence on detection rates. Next to this, sharpness and contrast are extracted and analyzed. Results imply that texture and contrast most strongly correlate with detection chances. Texture distance correlates highly

with manipulation detection in polished face images as opposed to coloured face images, suggesting humans tend to focus more on texture in polished faces due to the lack of colour features. Results also revealed a formed **cluster** in colour histogram plots, resulting in a defined range of [0.8-1.2] for colour histogram difference in which all questions result in a detection rate ranging from 0.3 to 0.5.

For facial landmarks, it was found that all facial feature shape distances in pairs inversely correlate with detection rates in coloured faces, while they directly correlate with detection rates for grayscaled faces. Out of all landmarks, the nose bridge and left eye shape distances correlate the strongest with detection rates, supporting the notion that the center of the face might be of most importance when it comes to face recognition and thus choice blindness. Lastly, a multivariate linear regression analysis was performed to define a function that might serve as a detection rate predictor. To accommodate for adding variables of too low statistical significance, the adjusted R-squared score is calculated for multiple possible variable combinations until the adjusted R-squared score no longer drops. A function using image type, question number, neural net similarity, contrast difference and texture difference results in the highest adjusted R-squared score of 0.5482 for detection rates as the target variable. A function was also defined for polished face images including the nose bridge shape distance variable, resulting in a staggering adjusted R-squared score of 0.727.



(a) Face landmarks. (b) Polished face landmarks.

Fig. 4: Extracted columns from (a) normal and (b) polished face landmark correlation matrices.

IV. Conclusion

This thesis sets out to answer several questions surrounding the choice blindness paradigm. To answer these questions as sufficiently as possible, an online study was built up using the PsyToolkit framework. This study was performed by 173 hired participants, after which the collected data was analyzed using the Python scripting language.

Results show that all three of the defined image similarity measures correlate inversely with chances of choice blindness occurring, with neural net similarity showing the strongest correlation (RQ1). More specifically, data suggests that similarity as defined by the GIST algorithm is the strongest predictor for choice blindness in landscape images (R^2= 0.6157), while the neural network best predicts choice blindness for faces (R^2 = 0.8848), polished faces (R^2 = 0.5829) and all types combined (R^2 = 0.5482). Also, a neural net similarity cut-off range of [83-88] and cut-off rate of 79 is defined for manipulation detection. Of all manipulated questions, 32.05% of undetected and 6.16% of detected manipulations resulted in a change of preference, possibly indicating that *choice change* is occurring (RQ2). Furthermore, participants were found to use more time to argue their choice as image similarities increase, indicating a possible sort of unconscious detection of self-deception (RQ3). Out of the three option types, landscape images resulted in the lowest detection rate of 33%, as opposed to 41% for grayscaled faces and 50% for coloured faces, hinting at the notion that as options are lesser connected to a person's sense of identity (as is likely the case for landscapes as opposed to faces), this person is more likely be choice blind (RQ3). Lastly, out of all extracted properties, the only statistically significant correlations were found between texture, contrast and nose bridge shape distances with detection rates (RQ1.1). For colour histogram distances, it was found that a [0.8-1.2] colour histogram distance most likely results in a detection percentage ranging from 30 to 50%.

These results show that choice blindness in images might be more predictable and preferences might be more easily manipulated than initially thought, when enough variables are available. When taking into account the importance of choice and preference in the daily lives of people, this notion is certainly not to be ignored, and further research is definitely of interest.

References

[1] F. Taya, S. Gupta, Ilya Farber, and O. Mullette-Gillman, "Manipulation detection and preference alterations in a choice blindness paradigm," *PLoS ONE*, vol. 9, 2014.

[2] Lars Hall, Petter Johansson, Betty Tärning, Sverker Sikström, and Thérèse Deutgen, "Magic at the marketplace: Choice blindness for the taste of jam and the smell of tea," *Cognition*, vol. 117, no. 1, pp. 54–61, 2010.

[3] L. Bortolotti and Ema Sullivan-Bissett, "Is choice blindness a case of self-ignorance?," *Synthese*, pp. 1 – 18, 2019.

[4] Lars Hall, Thomas Strandberg, Philip Pärnamets, Andreas Lind, Betty Tärning, and Petter Johansson, "How the polls can be both spot on and dead wrong: using choice blindness to shift political attitudes and voter intentions," *PloS one*, vol. 8, no. 4, pp. e60554–e60554, Apr 2013, 23593244[pmid].

[5] Catherine Steenfeldt-Kristensen and Ian M. Thornton, "Haptic choice blindness," *i-Perception*, vol. 4, no. 3, pp. 207–210, 2013, PMID: 23799197.

[6] Frank F. Ibarra, Omid Kardan, MaryCarol R. Hunter, Hiroki P. Kotabe, Francisco A. C. Meyer, and Marc G. Berman, "Image feature types and their predictions of aesthetic preference and naturalness," *Frontiers in Psychology*, vol. 8, pp. 632, 2017.

[7] Aude Oliva and Antonio Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *International journal of computer vision*, vol. 42, no. 3, pp. 145–175, 2001.

[8] David Hasler and Sabine E. Suesstrunk, "Measuring colorfulness in natural images," in *Human Vision and Electronic Imaging VIII*, Bernice E. Rogowitz and Thrasyvoulos N. Pappas, Eds. International Society for Optics and Photonics, 2003, vol. 5007, pp. 87 – 95, SPIE.

[9] Gijsbert Stoet, "Psytoolkit: a software package for programming psychological experiments using linux," *Behavior research methods*, vol. 42, no. 4, pp. 1096—1104, November 2010.

[10] R. G. Carpenter, "Principles and procedures of statistics, with special reference to the biological sciences," *The Eugenics Review*, vol. 52, no. 3, pp. 172–173, Oct 1960, PMC2972823[pmcid].

[11] Andrés Rieznik, Lorena Moscovich, Alan Frieiro, Julieta Figini, Rodrigo Catalano, Juan Manuel Garrido, Facundo Álvarez Heduan, Mariano Sigman, and Pablo A Gonzalez, "A massive experiment on choice blindness in political decisions: Confidence, confabulation, and unconscious detection of self-deception," *PloS one*, vol. 12, no. 2, pp. e0171108–e0171108, 2017.

APPENDICES

APPENDIX A - STRUCTURE OF QUESTIONS IN THE ONLINE STUDY

Set	Global	Neural net	GIST
Landscapes	1	10	19
	2	11	20
	3	12	21
Faces	22	4	13
	23	5	14
	24	6	15
Polished faces	16	25	7
	17	26	8
	18	27	9

TABLE III: The structure of a set in the used study.

APPENDIX B - RESULTS OF THE TESTING PHASES

Recognition method	Detection rates (%)
Text box	71
Radio button	86

TABLE IV: Detection rates using different manipulation detection recognition methods

Delay	Detection rates (%)
Yes	51
No	73

TABLE V: Detection rates depending on delay usage.

Similarity rate mean (%)	Detection rates (%)
78	54
87	34

TABLE VI: Detection rates depending on image similarity mean.

Analyse van de invloed van afbeeldingseigenschappen en -gelijkaardigheden op keuzeblindheid

Ward Vereecken

Promotors: Dr. Yara Khaluf, Prof. dr. ir. Pieter Simoens

Begeleider: Ir. Stef Van Havermaet

Abstract— **Elke dag maakt een mens honderden tot duizenden keuzes. Maar wat als deze keuzes gemakkelijk gemanipuleerd blijken te kunnen worden? Keuzeblindheid is het paradigma waarbij een persoon blind is voor zijn eigen keuze, enkele seconden nadat hij die gemaakt heeft. Deze thesis wil de invloed van eigenschappen en gelijkenissen van gezichts- en landschapsafbeeldingen op keuzeblindheid bestuderen aan de hand van een online studie. De relatie tussen keuzeblindheid en persoonlijke voorkeur, vertrouwen, en optie types wordt ook onderzocht. Meerdere scène-eigenschappen worden onderzocht op correlaties met de kans op keuzeblindheid. Hiertoe worden kleurhistogrammen, kleurrijkheid, drukte, textuurvectoren, gezichtsmerkvormen, contrast- en scherptewaarden geëxtraheerd. Daarnaast worden drie maatstaven voor afbeeldingssimilariteit geïnspecteerd en onderzocht op correlaties met de kans op keuzeblindheid: globale similariteit, gedefinieerd door low-level beeldkenmerken, GIST-similariteit, gedefinieerd door de ruimtelijke indeling van het beeld, en neurale net similariteit, een black box algoritme. De resultaten tonen aan dat GIST similariteit het sterkst correleert met detectiepercentages in landschapstype afbeeldingen, terwijl neurale net similariteit correleert met detectiepercentages in zowel gezicht als grijs geschaalde gezichtstype afbeeldingen. Van alle gemanipuleerde vragen resulteerde 32,05% van de onopgemerkte en 6,16% van de opgespoorde manipulaties in een verandering van voorkeur. Verder bleek dat deelnemers meer tijd nodig hadden om hun keuze te beargumenteren naarmate de gelijkenissen tussen de afbeeldingen toenamen, wat wijst op een mogelijk lager vertrouwensniveau. Van de drie soorten opties resulteerden landschapsfoto's in het laagste detectiepercentage van 33%, tegenover 41% voor gezichten met grijstinten en 50% voor gekleurde gezichten. Tenslotte werden van alle geëxtraheerde eigenschappen, de enige statistisch significante correlaties gevonden tussen textuur, contrast en neusbrug vorm afstanden met detectie percentages.**

Keywords— **keuzeblindheid, afbeeldingssimilariteit, afbeeldingskenmerken, introspectie**

I. Introductie

In zijn dagelijks leven maakt een mens een groot aantal keuzes. Mensen identificeren zichzelf, en andere mensen identificeren hen, in grote mate met hun keuzes. Die keuzes omvatten de keuze van hun beroep, hun levenspartner, hun woonomgeving, welke politieke partij ze steunen en zelfs welke schoenen ze kopen. De rol van de keuze speelt een grotere rol in het leven dan de meeste mensen beseffen. Deze rol dreigt te worden ondermijnd als zou blijken dat mensen zichzelf keuzes toeschrijven die ze nooit hebben gemaakt en deze zelfs verdedigen. Concepten rond dit thema worden in de psychologie veelvuldig onderzocht. Een van deze concepten is **keuzeblindheid**, waarvan het denkproces erachter nog niet volledig is begrepen. Een beter begrip van de psychologische effecten die ten grondslag liggen aan het menselijk redeneren, het nemen van beslissingen en de communicatie, die vaak ogenschijnlijk irrationeel zijn, is ook een van de grootste uitdagingen voor de verbetering van mens-AI[1] interactie. In deze tijd interageren mensen steeds meer met AI-gestuurde systemen, waardoor dit gebied steeds belangrijker wordt. Daarom wordt in deze thesis het keuzeblindheidsparadigma centraal gesteld.

A. Keuzeblindheid

Keuzeblindheid is kort gezegd het niet detecteren van discrepanties tussen intentie en uitkomst in een eenvoudige beslissingstaak. In het keuzeblindheidsparadigma krijgen deelnemers vooraf twee keuzes voorgeschoteld. Nadat ze een bepaald antwoord hebben gekozen, krijgen ze gemanipuleerde antwoorden te zien (het antwoord dat ze niet hebben gekozen). Wanneer er sprake is van keuzeblindheid, **merken de deelnemers deze manipulatie niet op** en geven zij zelfs een **doordachte redenering** waarom zij de genoemde (gemanipuleerde) antwoorden hebben gekozen. De laatste jaren is dit effect steeds meer in de belangstelling komen te staan omdat meer studies worden verricht om de mysteries en wetenschappelijke redenen erachter te ontrafelen. Een veelbelovende richting zou de relatie tussen afbeeldingssimilariteit en keuzeblindheid kunnen zijn.

B. Keuzeblindheid en afbeeldingssimilariteit

In deze thesis staat de invloed van de similariteit tussen afbeeldingen op de kans dat een gemanipuleerde vraag wordt opgemerkt, of met andere woorden, dat er keuzeblindheid optreedt, centraal. Dit onderwerp is eerder aangesneden door Taya e.a. [1], waar werd gesuggereerd dat afbeeldingssimilariteit geen significante predictor is voor keuze-blindheid (coeff = 20.02, p = 0.83, t = 20.22). Het werd echter aangetoond door Hall e.a. [2] dat paren met een zeer lage similariteit aanzienlijk meer worden gedetecteerd dan paren met een vrij hoge similariteit, maar dat er geen correlatie werd gevonden voor paren in de "grijze zone" van similariteit. Een belangrijk detail in beide studies is echter dat de mate van similariteit niet werd bepaald door een onbevooroordeeld algoritme, maar eerder door de mening van de deelnemers. Om dit onderwerp verder te onderzoeken, worden in dit onderzoek de similariteitpercentages op meerdere, onbevooroordeelde manieren bepaald. Om precies te zijn worden computervisie-algoritmen gebruikt en/of gecombineerd om de similariteitpercentages tussen bepaalde beeldparen te berekenen en uiteindelijk te zoeken naar correlaties met de kans op keuzeblindheid.

Om similariteitsen in afbeeldingen te bepalen, moeten deze pixels op een of andere manier geïnterpreteerd worden om informatie van een hoger niveau te verkrijgen, ook wel **semantische informatie** genoemd. Dergelijke informatie kan gaan van objecten tot ideeën of zelfs de algemene vorm van een scène. Zij kunnen bijvoorbeeld een verband leggen tussen een geëxtraheerde kleur zoals blauw en de zee of de lucht, wit en een gebouw, enzovoort. Bij informatie op laag niveau daarentegen wordt gekeken naar patronen en waarden in de pixels, maar wordt niet geprobeerd daar semantische informatie uit te halen. In plaats daarvan worden deze waarden bijna onbewerkt geëxtraheerd en vervolgens vergeleken op basis van de nominale waarde. Bijvoorbeeld, de textuur of kleur histogram van twee afbeeldingen kunnen worden vergeleken voor similariteit, of twee afbeeldingen kunnen worden gescand voor terugkerende hoeken of randen in de pixel waarden. Aansluitend op deze noties, onderzoekt deze scriptie ook het verschil in invloed van semantische versus low-level feature similariteitsen tussen afbeeldingen op keuzeblindheid kansen.

Terwijl het onderzoeken van invloedsfactoren van keuze blindheid een interessant onderzoeksonderwerp is, kan het onderzoeken van wat mensen werkelijk denken tijdens manipulaties helpen om meer inzicht

[1] AI of kunstmatige intelligentie: de simulatie van menselijke intelligentie in machines die geprogrammeerd zijn om als mensen te denken en hun handelingen na te bootsen.

te krijgen in het proces achter keuze blindheid. Dit leidt tot het volgende onderzoeksthema van introspectie.

C. Keuzeblindheid en introspectie

In dit proefschrift wordt introspectie[2] op twee manieren onderzocht: interpretaties van **voorkeursveranderingen** en **vertrouwensniveaus**. Het eerste idee berust op de notie dat voorkeursveranderingen ten gevolge van keuzeblindheid kunnen worden toegeschreven aan twee belangrijke inter-pretaties: (i) *keuze-fout*, waarbij de deelnemer redenen geeft over de verkeerde keuze en in feite verkeerd is over wat haar keuze is, of (ii) *keuze-verandering*, waarbij de deelnemer redenen geeft en zij gelijk heeft over wat haar keuze is, maar zij zich niet realiseert dat haar keuze veranderd is [1]. Om dit begrip te onderzoeken worden gemanipuleerde vragen herhaald aan dezelfde deelnemer om na te gaan of de voorkeuren van de deelnemers zijn veranderd als gevolg van de manipulatie. Ten tweede wordt de introspectie bij de deelnemers bepaald door het meten van hun vertrouwen in termen van de tijd die per argumentatie wordt gebruikt. Als deelnemers meer tijd nodig hebben om een vraag te beantwoorden, kan worden gesteld dat hun niveau van vertrouwen lager is dan wanneer ze betrekkelijk snel antwoorden, hetgeen wijst op een mechanisme van twijfel of introspectie in hun denkproces.

D. Keuzeblindheid en verschil in optietypen

In veel van de gevonden studies worden verschillende methoden gebruikt om het keuzeblindheidseffect te analyseren. Wat echter zelden lijkt te worden bestudeerd, is het verschil tussen verschillende soorten opties en/of criteria. Om enkele voorbeelden te geven, in een studie van Hall etal. [2], werden de smaak en de geur van twee verschillende consumptiegoederen gebruikt. In een andere studie, van Hall e.a. [3], werd gebruik gemaakt van politieke toestanden, en in een studie van Steenfeldt-Kristensen en Thornton [4] werd gebruik gemaakt van tastzin, of haptische gewaarwordingen, enz. Hoewel in deze studies soms kort wordt ingegaan op het verschil met andere studies en methoden, wordt het vergelijken van meerdere optie- en/of criteriatypen in één studie echter zelden gedaan. Na onderzoek over het onderwerp werden vele vergelijkingen van optietypen voorgesteld (om enkele voorbeelden te geven: afbeeldingen met teksten of lange tekst met korte tekst). Uiteindelijk werd beslist om verschillende types beeldscènes te vergelijken op hun invloed op keuzeblindheid. Meer specifiek worden landschappen, gezichten, en "gefilterde" (grayscaled, blurred) gezichten met elkaar vergeleken.

E. Conclusie

Keuzeblindheid is een paradigma dat kan leiden tot belangrijke inzichten in de menselijke psyche en de mechanismen achter de vaak schijnbaar onlogische denkprocessen van de mens. Het doel van deze thesis is uiteindelijk om bepaalde hypothesen over keuzeblindheid te testen om zo meer inzicht te krijgen in het proces erachter. De volgende onderzoeksvragen worden gedefinieerd en getracht wordt ze te beantwoorden door middel van een online studie.

- *RQ 1: Wat is de invloed van afbeeldingssimilariteit op de kans dat keuzeblindheid optreedt?*
  - *RQ 1.1: Hoe beïnvloeden verschillende soorten afbeeldingseigenschappen of -criteria de kans op het optreden van keuzeblindheid?*
- *RQ 2: In welke mate kan keuzeblindheid resulteren in veranderingen in persoonlijke voorkeur?*
- *RQ 3:In welke mate beïnvloedt de (on)similariteit van afbeeldingen het zelfvertrouwensniveau van de deelnemers?*
- *RQ 4: Hoe verhouden verschillende soorten opties zich tot elkaar als het gaat om de kans dat er keuzeblindheid optreedt?*

[2] Het Oxford woordenboek definieert introspectie als het zorgvuldig onderzoeken van je eigen gedachten, gevoelens en redenen om je op een bepaalde manier te gedragen. Met andere woorden, het zou kunnen verwijzen naar de mate waarin iemand zich bewust is waarom hij voor een bepaalde optie kiest wanneer hem twee opties worden voorgelegd. In die zin is het sterk verwant met het keuzeblindheidsparadigma.

II. Methodologie

Om deze vragen zo goed mogelijk te kunnen beantwoorden, worden eerst meerdere beeldanalyses en similariteitsmetingen onderzocht en gedefinieerd. Ten tweede wordt een online studie ontworpen, uitgevoerd en gestart. Op basis van de genoemde maatstaven en de resultaten van het testen ervan op afbeedingsdatasets, wordt de vraagstructuur en -volgorde van deze studie opgezet.

A. Maatstaven voor afbeeldingssimilariteit

De meest gebruikte technieken om de similariteit tussen afbeeldingen te bepalen of om een classificatie uit te voeren zijn gebaseerd op beeldkenmerken. Afbeeldingskenmerken definiëren, kort gezegd, bepaalde eigenschappen van groepen pixels in een afbeelding. Eén type beeldkenmerken zijn bijvoorbeeld randen. Randen zijn punten waar er een grens (of een rand) is tussen twee beeldgebieden. Zoals eerder vermeld, kunnen beeldkenmerken in twee hoofdcategorieën worden onderverdeeld: (i) kenmerken op laag niveau en (ii) kenmerken op hoog niveau of semantische kenmerken. Low-level beeldkenmerken zijn beeldkenmerken die door computers worden vastgelegd met het oog op herkenning en classificatie (zoals pixelintensiteit, pixelgradiëntoriëntatie, kleurverdeling, enz.), terwijl semantische beeldkenmerken de kenmerken zijn die gewoonlijk door mensen worden gebruikt om beelden te beschrijven (objecten, acties, enz.), en die de grootste invloed zouden kunnen hebben op de beeldvoorkeuren van mensen (Ibarra et al. [5]).

A.1 Low-level beeldkenmerken

Kleur

Het kleurkenmerk wordt geëxtraheerd door berekening van het **kleurhistogram** van het beeld. Een kleurenhistogram is een weergave van de verdeling van kleuren in een afbeelding. In een digitaal beeld geeft dit het aantal pixels weer dat bepaalde kleuren bevat uit een lijst van kleurbereiken die de kleurruimte van het beeld (bv. RGB) beslaat. Om deze histogrammen te vergelijken werd gekozen voor de correlatiemethode.

Om beelden op basis van dit kleurenhistogram te vergelijken bestaan er vier hoofdmethoden: correlatie, chi-kwadraat, intersectie en Bhattacharyya-afstand. Uit de testresultaten blijkt dat correlatie, intersectie en Bhattacharyya-afstand relatief goede maatstaven zijn voor kleursimilariteit. Na afweging werd correlatie gekozen als similariteitsmaatstaf, waarbij vergelijking 1 werd gebruikt om twee histogrammen H1 en H2 te vergelijken in een afbeelding I.

$$d(H_1, H_2) = \frac{\sum_I (H_1(I) - \bar{H_1})(H_2(I) - \bar{H_2})}{\sqrt{\sum_I (H_1(I) - \bar{H_1})^2 \sum_I (H_2(I) - \bar{H_2})^2}} \quad (1)$$

Textuur

Om moeilijkheden zoals variaties in kleurverdeling, schaal, belichting, rotatie of affiene transformatie te overwinnen, kunnen ook textuurkenmerken worden geëxtraheerd. Textuuranalyse is een manier om de ruimtelijke verdeling van intensiteiten te beschrijven, waardoor het nuttig is bij de classificatie van soortgelijke gebieden in verschillende beelden. Om deze kenmerken te extraheren wordt het lokale binaire patroonhistogram geëxtraheerd, wat een eenvoudige maar zeer efficiënte textuuroperator is die de pixels van een beeld labelt door de omgeving van elk pixel te limiteren en het resultaat als een binair getal te beschouwen. De histogrammen worden vervolgens

vergeleken met behulp van dezelfde correlatiemethode als eerder genoemd.

Beeldmoment

Het moment van een beeld vertegenwoordigt een bepaald gewogen gemiddelde (moment) van de intensiteiten van de beeldpixels. Eenvoudige eigenschappen van het beeld die via beeldmomenten worden gevonden, zijn onder meer de oppervlakte (of de totale intensiteit), het zwaartepunt en informatie over de oriëntatie. Met andere woorden, ze geven informatie over de vorm van een object in een binair beeld, omdat ze informatie bevatten over de intensiteit I(x,y), alsnog over de positie x en y van de pixels. Om momenten te berekenen die invariant zijn voor transpositie, schaal en rotatie, moet het **Hu-moment** worden berekend. Hu-momenten (Hu moment invariants) zijn een verzameling van zeven getallen die worden berekend met behulp van centrale momenten die invariant zijn voor beeldtransformaties. Tenslotte wordt de logaritmische schaal van deze Hu momenten berekend om een vergelijkbare schaal te bereiken.

Lokale kenmerkdescriptoren

Lokale kenmerkdescriptoren kwantificeren lokale regio's van een beeld door te kijken naar bepaalde patronen, hoeken, gradiënten, etc. Hoewel dit soort kenmerken veel gebruikt wordt bij het matchen en classificeren van beelden, bleek het niet toepasbaar op het onderhavige probleem. Dit is waarschijnlijk een gevolg van de neiging van deze methode om zich te concentreren op betrekkelijk kleinschalige details in afbeeldingen, hetgeen niet noodzakelijk een goede indicator is van een menselijk gebaseerde similariteit.

Combinatie

Met uitzondering van de lokale kenmerkdescriptoren, werden alle hierboven gedefinieerde kenmerken gecombineerd tot één vlakke vector en vergeleken tussen beelden. Deze similariteitsmaatstaf wordt verder aangeduid als de **globale** similariteitsmaatstaf.

A.2 High-level of semantische beeldkenmerken

Machine learning

Het bepalen van semantische beeldsimilariteit is een grotere uitdaging. Er bestaan algoritmen voor machinaal leren die een veelheid van beeldkenmerken op laag niveau kunnen omzetten in semantische informatie. Er worden twee modellen voor machinaal leren gebruikt, namelijk het Apple TuriCreate-afbeeldingssimilariteitmodel voor landschappen en het gezichtsherkenningsmodel van de Python-bibliotheek voor gezichten. Deze modellen maken gebruik van convolutionele neurale netwerken, die de informatie over de ruwe pixels door meerdere lagen verwerken, om uiteindelijk een semantische categorie voor het gegeven invoerbeeld te verkrijgen. De afstand op basis van deze uitvoer levert dan een similariteitsscore op.

GIST features

Een andere benadering van het kwantificeren van semantische similariteit tussen beelden, zoals voorgesteld door Oliva en Torralba [6], kijkt naar een zeer laag-dimensionale voorstelling van de scène, de **Spatial Envelope** genoemd. Zij stellen voor dat een holistische representatie van de scène informatie geeft over zijn waarschijnlijke semantische categorie. Met andere woorden, dat beelden kunnen worden gecategoriseerd op basis van een vage vorm van de scène in plaats van ingewikkelde details. Het kenmerkdescriptor dat deze ruimtelijke lay-out beschrijft, wordt de GIST kenmerkdescriptor genoemd, en wordt geëxtraheerd met de Python bibliotheek Lmgist. Belangrijk is dat deze GIST-descriptor betekenisvol is voor hoe mensen beelden waarnemen: "Het spatial envelope model ordent beelden zoals mensen dat doen, en is in staat om beelden terug te vinden die dezelfde semantische categorie delen", Oliva en Torralba [6]. Het berekenen van de afstand tussen de GIST-descriptoren van twee beelden kan dus een nauwkeurig resultaat opleveren in termen van semantische similariteit, aangezien het een idee geeft van de afstand tussen hun respectieve semantische categorieën.

Andere semantische kenmerken

Om de invloed van semantische similariteit op keuzeblindheid verder te onderzoeken, worden enkele meer specifieke semantische kenmerken geëxtraheerd. Ten eerste worden de coördinaten van de herkenningspunten in het gezicht geëxtraheerd met behulp van de hierboven beschreven machine learning library. Deze coördinaten worden gebruikt om de similariteit van de **landmark-vormen** tussen de verschillende gezichten te kwantificeren door het moment van deze vormen te berekenen en de cosinussimilariteit tussen deze vormen te bepalen (bijv. het verschil in neusvorm tussen twee gezichten). Daarnaast wordt een **kleurrijkheidsmetriek**, zoals gedefinieerd door Hasler en Suesstrunk [7], geëxtraheerd als een semantisch kenmerk. Deze extractie is gebaseerd op de opponente kleurruimte representatie van het beeld samen met de gemiddeldes en standaarddeviaties van deze waarden. Tenslotte werd een **drukte-metriek** gedefinieerd, waarmee wordt getracht te kwantificeren hoe "druk" een beeld is. Dit wordt verkregen door bilateraal filteren, Otsu's thresholding, erosie en randdetectie op het beeld uit te voeren, objecten van achtergrond te scheiden en vervolgens het aantal afzonderlijke "objecten" te tellen.

B. Ontwerp van de Online Studie

Set	Globaal	Neuraal net	GIST
Landschappen	1	10	19
	2	11	20
	3	12	21
Gezichten	22	4	13
	23	5	14
	24	6	15
Gepolijste gezichten	16	25	7
	17	26	8
	18	27	9

TABLE I: De structuur van een set vragen gebruikt in de studie.

Om te trachten de voorgestelde onderzoeksvragen te beantwoorden, is een online studie ontworpen en geïmplementeerd met behulp van het PsyToolkit framework, een softwarepakket voor het uitvoeren van psychologische experimenten, dat het best kan worden geclassificeerd als software voor gedragsexperimenten [8]. Twee afbeeldingsdatasets werden van het internet gehaald. De eerste set bevat meerdere landschapsscènes, variërend van bergachtige gebieden tot stranden en woestijnen. De tweede set bevatte een groot aantal gezichten, waaruit alleen afbeeldingen werden gekozen met een uniforme, witte achtergrond, en waarbij de gezichten in dezelfde oriëntatie naar voren waren gericht. De volgorde en de structuur van de vragen waren volledig gebaseerd op de berekende similariteitsscores zoals hierboven beschreven. De drie belangrijkste similariteitsmetingen, namelijk globale, neurale net- en GIST-similariteit, worden op een round-robin manier gebruikt om te kiezen welk paar afbeeldingen moet worden gemanipuleerd. Daarnaast wordt elk van de drie beeldtypes op een round-robin manier gekozen. Met andere woorden, de beeldparen van de eerste vraag kunnen worden gekozen op basis van hun similariteit zoals berekend door de globale similariteit-maatstaf, terwijl het volgende paar zou worden gekozen op basis van hun similariteit zoals berekend door het neurale net. Dit leidt tot de vraagstructuur zoals afgebeeld in tabel I.

Om deze drie similariteitspercentages op dezelfde schaal te kunnen vergelijken, werden alle scores genormaliseerd voordat paren werden gekozen met behulp van Vergelijking 2. Hierin staat x' voor de genormaliseerde score, x voor de oude score en $max(x)$ en $min(x)$ voor respectievelijk de maximum- en minimumscore.

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{2}$$

Elke testfase bestaat uit twee afzonderlijke sets om de beste configuraties te testen. Om een voorbeeld te geven: in een eerste testfase ($N = 16$) werden manipulatieparen gehouden in het bereik van 80-100% similariteit, terwijl alle niet-gemanipuleerde vragen uniformeel gekozen werden in het bereik van 0-100%. Er werden twee reeksen gemaakt, één waarbij een tekstvak wordt gebruikt om de deelnemers om argumentatie te vragen, en één waarbij keuzerondjes worden gebruikt.

Gezien de psychologische aard van deze studie waren verschillende testruns en configuraties nodig voordat een definitieve configuratie werd gekozen voor de start van de studie. Kleine groepen deelnemers werden verzameld via sociale media om de studie te testen. Tabel IV in appendix A laat zien dat tekstboxen tot de laagste detectiegraad van manipulatie leidden. Dit kan mogelijk het gevolg zijn van een hogere mate van wantrouwen, omdat de keuzerondjes op een soort van ma-nipulatie wijzen. Na alle testfasen zijn tekstkaders gekozen als de-tectieherkenningsmethode, is een kunstmatige vertraging tussen de vragen toegevoegd om studies die niet online, maar ter plekke gebeurd zijn te simuleren, en is het gemiddelde van de similariteitsscores van gemanipuleerde vraagparen hoger gekozen. Zie Appendix A voor de resultaten van elke testfase. Deze wijzigingen resulteren in een manipulatiedetectiepercentage dat vergelijkbaar is met dat van eerdere studies over keuzeblindheid, waardoor de studie klaar is om te worden gelanceerd.

III. Resultaten

In testfase 3 werd een opsporingspercentage van 34% waargenomen bij 13 deelnemers. In de laatste studie ($N = 173$) werd dit detectiepercentage met ongeveer 6% verhoogd. Van de 881 gemanipuleerde vragen werden er 356 gedetecteerd, wat resulteerde in een detectiepercentage van 40,41%. In 293 vals-positieve gevallen werden de antwoorden van de deelnemers als detecties gecategoriseerd, terwijl de vraag in werkelijkheid niet gemanipuleerd was. Manipulatiedetecties werden herkend door de argumentaties te filteren op bepaalde sleutelwoorden die op een detectie wijzen, bijvoorbeeld 'niet', 'niet', 'herinneren', 'gekozen', enz.

A. Invloed van afbeeldingssimilariteit

De meeste paren die in het lagere detectiebereik lagen (5 tot 40 %) bleken ook de vragen te zijn die gemiddeld het meest voorkwamen in de eerste helft van het onderzoek. De relatie tussen vraagvolgorde en detectiegraad blijkt direct te zijn, wat suggereert dat naarmate paren later in de studie opduiken, de kans toeneemt dat hun manipulatie wordt gedetecteerd.

Figuur 1 toont elk van de drie maten en hun correlatie met de detectiekans voor elk van de drie beeldtypes. De resultaten tonen aan dat, terwijl voor landschappen en gepolijste gezichten alle drie de correlaties invers blijven, gezichtsbeelden resulteren in een directe correlatie voor de globale similariteitsmaat. Meer precies, GIST similariteit correleert het best met detectiepercentages in landschapsbeelden, terwijl neurale net similariteit het best correleert met detectiepercentages in gezichtsbeelden. Over het algemeen lijkt de neurale netsimilariteit de beste voorspeller te zijn. Om de sterkte van deze voorspellers te kwantificeren, wordt de R-kwadraat van de

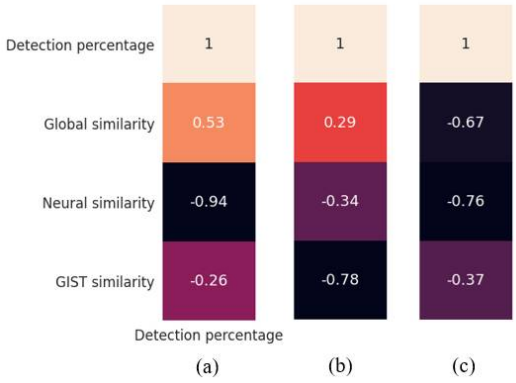


Fig. 1: Geëxtraheerde kolommen van correlatiematrices van verzamelde gegevens, voor (a) gezichtsbeelden, (b) landschapsbeeld en (c) gepolijste gezichtsbeelden.

relatie tussen deze variabelen berekend. In de statistiek is de R-kwadraatwaarde het deel van de variantie in de afhankelijke variabele dat voorspelbaar is via de onafhankelijke variabele(n) (Carpenter [9]). Deze waarde varieert van 0 tot 1, hetgeen wijst op een slechte of een zeer sterke voorspellende factor.

Tabel II toont de R-kwadraatscores voor elk van de combinaties algoritme-beeldtype. Deze resultaten suggereren dat de globale similariteit-maatstaf een goede voorspeller zou kunnen zijn voor detectiepercentages in gepolijste gezichten. Dit zou kunnen worden toegeschreven aan het feit dat de globale meting zich voor een groot deel concentreert op de textuur in het beeld. Aangezien er in het gepolijste gezicht geen kleurverschillen zijn, zouden deelnemers zich meer kunnen concentreren op de textuur van gezichtsafbeeldingen, omdat, zoals Taya et al. [10] hebben besproken, textuur een van de andere meer prominente kenmerken is die mensen neigen waar te nemen wanneer ze gezichten herkennen. De hoge R-kwadraatscore voor de GIST similariteit in landschapsbeelden is waarschijnlijk een gevolg van het feit dat het GIST-algoritme zich concentreert op de vormen en de lay-out van de scène, wat van groot belang is wanneer het gaat om de vergelijkbaarheid tussen landschappen. De neurale net similariteit levert een significant hoge R-kwadraat score op voor gezichtsbeelden (0,8848), wat wijst op een zeer hoge variantie verklaring door deze variabele.

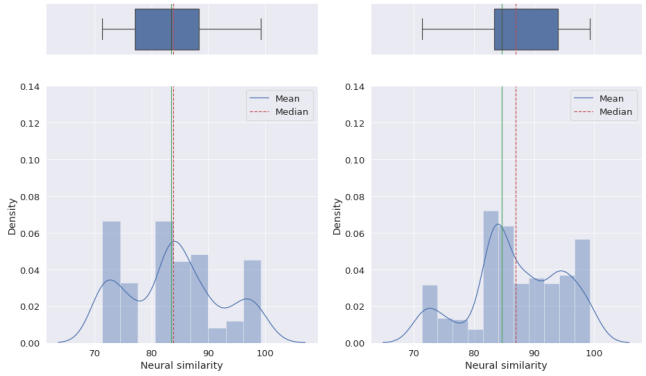
	R-kwadraatscores		
Afbeeldingstype	Globaal	Neuraal net	GIST
Landscappen	0.0822	0.1148	0.61567
Gezichten	0.2834	0.8848	0.0675
Gepolijste gezichten	0.4494	0.5829	0.1361
Alle	0.0757	0.5482	0.1630

TABLE II: R^2 waarden voor elke combinatie van algoritme en beeldtype.

Twee functies worden gedefinieerd uit de lineaire regressie modellen die op basis van deze bevindingen zijn gemaakt. Functie 3 definieert de relatie tussen detectiekansen en landschapsafbeeldingssimilariteit zoals gedefinieerd door het GIST-algoritme. Functie 4 bepaalt vervolgens het verband tussen detectiekansen en gezichtssimilariteit zoals gedefinieerd door het gebruikte neurale net.

$$y = -44.78321x + 104.72 \tag{3}$$

$$y = -47.1372x + 104.4028 \quad (4)$$



(a) Gedetecteerde manipulaties (b) Niet-gedetecteerde manipulaties

Fig. 2: Neurale net similariteit histogram- en box plot voor (on)gedetecteerde manipulaties.

Figuur 2 illustreert histogrammen en boxplots van de neurale netsimilariteitsen voor de gemanipuleerde vragen. Uit het interkwartielbereik van deze boxplots kan worden afgeleid dat precies de helft van de niet-gedetecteerde manipulaties een similariteit vertoont van 83 tot 94, terwijl dit bereik voor gedetecteerde manipulaties 77 tot 88 bedraagt. Dit wijst erop dat [83-88] kan dienen als een vaag grens**bereik** voor de detectie van manipulaties in termen van neurale netwerksimilariteit. Wanneer we een gezamenlijke plot van dezelfde vragen bekijken, zien we dat de gedetecteerde vragen in de meerderheid zijn voor similariteitsen van 79 en lager, wat wijst op een duidelijker grens**waarde** van 79 voor de similariteit.

B. Introspectie

	Percentage gewijzigde meningen [%]	
Afbeeldingstype	Niet-gedetecteerd	Gedetecteerd
Landscappen	37.73	7.87
Gezichten	31.4	7.02
Gepolijste gezichten	24.34	3.44
All	32.05	6.16

TABLE III: Percentage gewijzigde meningen voor elk beeldtype, afhankelijk van detectie.

Gemiddeld over alle beeldtypes resulteerde ongeveer één op vijf manipulaties in een verandering van mening over de vraag in kwestie, ongeacht de detectiestatus. Echter, zoals blijkt uit Tabel III, wanneer deze vragen verdeeld worden over de detectietoestand, resulteerde ongeveer een derde van alle niet-gedetecteerde vragen, en zelfs 6.16% van alle gedetecteerde vragen, in een verandering van voorkeur. Deze resultaten zouden kunnen impliceren dat er in heel wat gevallen sprake is van een voorkeursverandering, waarbij de deelnemers effectief van mening zijn veranderd, en dus correct zijn over wat hun keuze eigenlijk is. Bij een vergelijking van de drie beeldtypes resulteerde het landschapstype in de hoogste opiniewijziging van maar liefst 37,73%, gevolgd door gezichten met 31,4% en gepolijste gezichten met 24,34%. Deze bevindingen zouden erop kunnen wijzen dat naarmate optietypes minder aan de identiteit van een persoon gebonden zijn (wat waarschijnlijk het geval is bij landschappen in tegenstelling tot gezichten), de kans toeneemt dat hij of zij keuzeblind is.

Wat betreft de tijd die nodig is om hun keuzes te beargumenteren, lijken deelnemers er iets langer over te doen naarmate de gemanipuleerde optiesimilariteit toeneemt. Met andere woorden, voor alle drie de metingen lijkt de gemanipuleerde similariteit van de beelden direct te correleren met de gemiddelde tijd die nodig is om voor deze paren te argumenteren, wat mogelijk wijst op een soort onbewuste detectie van zelfbedrog, zoals besproken door Rieznik e.a. [11].

C. Optietypes

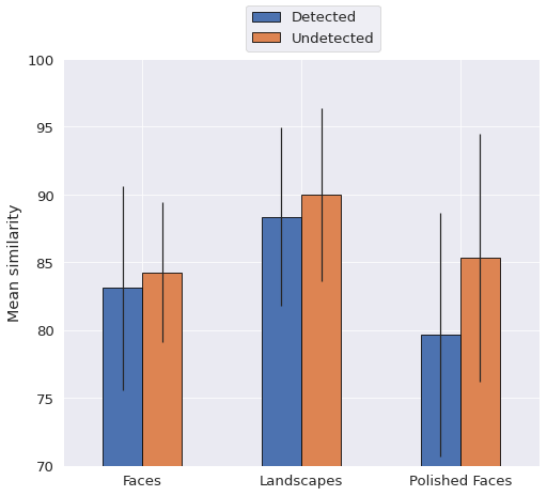


Fig. 3: Het gemiddelde van alle gelijkenissen, uitgezet voor (on)gedetecteerde manipulaties voor elk type afbeelding.

Van de drie optie- of beeldtypen leverden landschapsparen het laagste detectiepercentage op, gevolgd door gepolijste gezichten en gezichten (zie figuur 3). Ook lijkt de afwijking van de detectiepercentages betrekkelijk gering, wat wijst op een stabieler en voorspelbaarder detectiepercentage voor landschappen dan voor gezichten en gepolijste gezichten. Dit resultaat ondersteunt de conclusies van deel III-B, waar beelden van het type landschap een aanzienlijk hogere mate van verandering van mening vertoonden, wat erop wijst dat meningen over landschappen wellicht een kleinere impact hebben op iemands zelf-identiteit dan meningen over gezichten, wat leidt tot een lossere houding ten opzichte daarvan. Daarnaast kan uit deze grafiek geconcludeerd worden dat, voor deze studie, niet-gedetecteerde manipulaties inderdaad een hogere gemiddelde similariteit hadden dan gedetecteerde manipulaties. Ook voor gepolijste gezichten lijkt dit verschil het grootst te zijn.

D. Semantische eigenschappen

Tenslotte worden de bovengenoemde eigenschappen geanalyseerd op correlaties of interessante patronen in termen van invloed op detectieratio. Daarnaast worden scherpte en contrast geëxtraheerd en geanalyseerd. De resultaten geven aan dat textuur en contrast het sterkst correleren met detectiekansen. Textuurafstand correleert sterk met manipulatiedetectie in gepolijste gezichtsbeelden in tegenstelling tot gekleurde gezichtsbeelden, wat suggereert dat mensen geneigd zijn zich meer te concentreren op textuur in gepolijste gezichten vanwege het gebrek aan kleurkenmerken. De resultaten onthulden ook een gevormde cluster in kleurhistogramplots, resulterend in een gedefinieerd bereik van [0,8-1,2] voor kleurhistogramverschil waarbij alle vragen resulteren in een detectiekans van 0,3 tot 0,5 (zie figuur 4).

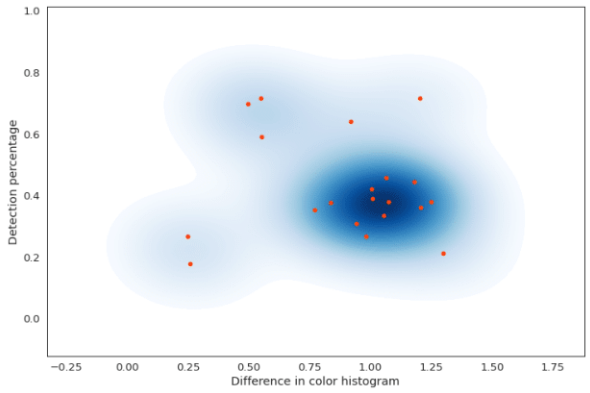
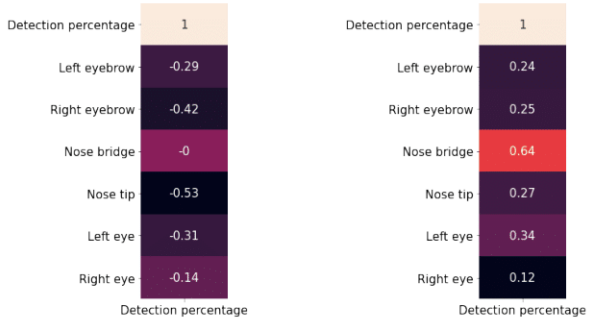


Fig. 4: Kleurenhistogram verschil uitgezet tegen detectiepercentage voor alle gemanipuleerde vragen.



(a) Face landmarks. (b) Polished face landmarks.

Fig. 5: Geëxtraheerde kolommen uit (a) normale en (b) gepolijste gezichts-featurecorrelatiematrices.

Voor gezichtskenmerken werd vastgesteld dat alle paarsgewijze verschillen tussen gezichtskenmerken omgekeerd evenredig zijn met de waarnemingspercentages bij gekleurde gezichten, terwijl ze direct correleren met de waarnemingspercentages bij grijs geschaalde gezichten. Van alle herkenningspunten correleren de afstanden van de neusbrug en het linkeroog het sterkst met de waarnemingspercentages, wat de gedachte ondersteunt dat het centrum van het gezicht van het grootste belang zou kunnen zijn voor gezichtsherkenning en dus voor keuzeblindheid. Tenslotte werd een multivariate lineaire regressie-analyse uitgevoerd om een functie te bepalen die zou kunnen dienen als voorspeller van de detectiesnelheid. Om rekening te houden met het toevoegen van variabelen van te lage statistische significantie, wordt de aangepaste R-kwadraat score berekend voor meerdere mogelijke combinaties van variabelen totdat de aangepaste R-kwadraat score niet meer daalt. Een functie die gebruik maakt van beeldtype, vraagnummer, neurale netsimilariteit, contrastverschil en textuurverschil resulteert in de hoogste aangepaste R-kwadraatscore van 0,5482 voor detectiepercentages als doelvariabele. Er werd ook een functie gedefinieerd voor gepolijste gezichtsbeelden waarin de afstandsvariabele voor de vorm van de neusbrug was opgenomen, wat resulteerde in een verbluffende aangepaste R-kwadraatscore van 0,727.

IV. Conclusie

Deze thesis heeft tot doel een aantal vragen rond het keuze-blindheidsparadigma te beantwoorden. Om deze vragen zo goed mogelijk te beantwoorden is een online studie opgezet met behulp van het PsyToolkit framework. Deze studie werd uitgevoerd door 173 ingehuurde deelnemers, waarna de verzamelde data werd geanalyseerd met behulp van de Python scripting taal. Resultaten tonen aan dat alle drie de gedefinieerde afbeeldingssimilariteit maatstaven invers correleren met de kans op het optreden van keuze-blindheid, waarbij de neurale net similariteit de sterkste correlatie vertoont. Meer specifiek blijkt uit de gegevens dat similariteit zoals gedefinieerd door het GIST algoritme de sterkste voorspeller is van keuzeblindheid in landschapsbeelden (R^2= 0.6157), terwijl het neurale netwerk het beste keuzeblindheid voorspelt voor gezichten (R^2= 0.8848), gepolijste gezichten (R^2 = 0.5829) en alle typen samen (R^2=0.5482). Ook is een neuraal net similariteit cut-off range van [83-88] en cut-off rate van 79 gedefinieerdREFERENTIESe detectie. Van alle gemanipuleerde vragen leidden 32.05% van de onopgemerkte en 6.16% van de opgemerkte manipulaties tot een verandering van voorkeur, wat er mogelijk op wijst dat er een verandering van voorkeur optreedt. Bovendien bleken deelnemers meer tijd te gebruiken om hun keuze te beargumenteren naarmate de similariteitsen tussen de beelden toenamen, wat wijst op een mogelijke vorm van onbewuste detectie van zelfbedrog. Van de drie soorten opties resulteerden landschapsfoto's in het laagste detectiepercentage van 33%, tegenover 41% voor gezichten met grijstinten en 50% voor gekleurde gezichten, wat erop wijst dat naarmate opties minder verbonden zijn met iemands identiteitsgevoel (wat waarschijnlijk het geval is voor landschappen in tegenstelling tot gezichten), de kans groter is dat deze persoon keuzeblind is. Ten slotte werden van alle geëxtraheerde eigenschappen de enige statistisch significante correlaties gevonden tussen textuur, contrast en neusbrugvorm-afstanden met detectiepercentages. Voor kleurenhistogramafstanden werd gevonden dat een [0.8-1.2] kleurenhistogramafstand hoogst-waarschijnlijk resulteert in een detectiepercentage variërend van 30 tot 50%.

Deze resultaten tonen aan dat keuzeblindheid in afbeeldingen wellicht beter voorspelbaar is en dat voorkeuren wellicht gemakkelijker te manipuleren zijn dan aanvankelijk gedacht, wanneer voldoende variabelen beschikbaar zijn. Wanneer rekening wordt gehouden met het belang van keuze en voorkeur in het dagelijks leven van mensen, kan dit begrip zeker niet genegeerd worden, en verder onderzoek is zeker van belang.

Referenties

[1] L. Bortolotti and Ema Sullivan-Bissett, "Is choice blindness a case of self-ignorance?," *Synthese*, pp. 1 – 18, 2019.

[2] Lars Hall, Petter Johansson, Betty Tärning, Sverker Sikström, and Thérèse Deutgen, "Magic at the marketplace: Choice blindness for the taste of jam and the smell of tea," *Cognition*, vol. 117, no. 1, pp. 54–61, 2010.

[3] Lars Hall, Thomas Strandberg, Philip Pärnamets, Andreas Lind, Betty Tärning, and Petter Johansson, "How the polls can be both spot on and dead wrong: using choice blindness to shift political attitudes and voter intentions," *PloS one*, vol. 8, no. 4, pp. e60554–e60554, Apr 2013, 23593244[pmid].

[4] Catherine Steenfeldt-Kristensen and Ian M. Thornton, "Haptic choice blindness," *i-Perception*, vol. 4, no. 3, pp. 207–210, 2013, PMID: 23799197.

[5] Frank F. Ibarra, Omid Kardan, MaryCarol R. Hunter, Hiroki P. Kotabe, Francisco A. C. Meyer, and Marc G. Berman, "Image feature types and their predictions of aesthetic preference and naturalness," *Frontiers in Psychology*, vol. 8, pp. 632, 2017.

[6] Aude Oliva and Antonio Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *International journal of computer vision*, vol. 42, no. 3, pp. 145–175, 2001.

[7] David Hasler and Sabine E. Suesstrunk, "Measuring colorfulness in natural images," in *Human Vision and Electronic Imaging VIII*, Bernice E. Rogowitz and Thrasyvoulos N. Pappas, Eds. International Society for Optics and Photonics, 2003, vol. 5007, pp. 87 – 95, SPIE.

[8] Gijsbert Stoet, "Psytoolkit: a software package for programming psychological experiments using linux," *Behavior research methods*, vol. 42, no. 4, pp. 1096—1104, November 2010.

[9] R. G. Carpenter, "Principles and procedures of statistics, with special reference to the biological sciences," *The Eugenics Review*, vol. 52, no. 3, pp. 172–173, Oct 1960, PMC2972823[pmcid].

[10] F. Taya, S. Gupta, Ilya Farber, and O. Mullette-Gillman, "Manipulation detection and preference alterations in a choice blindness paradigm," *PLoS ONE*, vol. 9, 2014.

[11] Andrés Rieznik, Lorena Moscovich, Alan Frieiro, Julieta Figini, Rodrigo Catalano, Juan Manuel Garrido, Facundo Álvarez Heduan, Mariano Sigman, and Pablo A Gonzalez, "A massive experiment on choice blindness in political decisions: Confidence, confabulation, and unconscious detection of self-deception," *PloS one*, vol. 12, no. 2, pp. e0171108–e0171108, 2017.

APPENDICES

APPENDIX A - RESULTATEN VAN DE TESTFASES

Herkenningsmethode	Detectiepercentages (%)
Text box	71
Radio button	86

TABLE IV: Detectiepercentages gebruikmakend van verschillende manipulatiedetectieherkenningsmethoden.

Vertraging	Detectiepercentages (%)
Ja	51
Nee	73

TABLE V: Detectiepercentages afhankelijk van artificiele vertraging.

Gemiddelde similariteit afbeeldingen (%)	Detectiepercentages (%)
78	54
87	34

TABLE VI: Detectiepercentages afhankelijk van similariteitsgemiddelde afbeeldingen.

Contents

List of Figures

List of Tables

Listings

"The most obvious suggestion to handle this problem would be to disqualify outright all opinions subject to Choice Blindness as not real. Because how can it be a 'real' attitude if we moments later are prepared to endorse the opposite?"

˜Lars, H. and Petter, J. and Thomas, S.

1

Introduction

In their day to day life, a person makes a substantial amount of choices. To a large extent, people identify themselves, and other people identify them, with their choices. These choices include the choice of their occupation, their life partner, their living area, which political party to support, and even which shoes to buy. The role of choice plays a bigger role in life than most people realise. This role is threatened to be undermined if it were shown that people attribute choices to themselves that they never made and even defend them. Concepts surrounding this theme are widely researched in the psychology field. One of these concepts is **choice blindness**, of which the thought process behind it has not yet been fully understood. Better understanding the psychological effects that underlie human reasoning, decision making and communication, which are often seemingly irrational, is also one of the most daunting challenges that lie in improving human-AI[1] interaction. In this day and age, humans increasingly interact with AI-driven systems, making this field progressively important. For this reason, the choice blindness paradigm will be the focus of this thesis.

[1] AI or Artificial Intelligence: the simulation of human intelligence in machines that are programmed to think like humans and mimic their actions.

1.1 Choice blindness

Put concisely, choice blindness is the failure to detect mismatches between intention and outcome in a simple decision task (Johansson et al. [3]). In the choice blindness paradigm, participants are presented with two choices. After choosing a certain answer, they are presented with manipulated answers. When choice blindness occurs, the participants fail to notice this manipulation and even offer thoughtful reasoning as to why they chose said (manipulated) answers. Choice blindness is often described as a form of confabulation. When people confabulate, they tell a story that they believe to be correct, for instance, a story about why they made a certain choice, but the story is not grounded in the evidence (Bortolotti and Sullivan-Bissett [4]). In recent years, this effect has received increasing attention as more studies are being performed to unravel the mysteries and scientific reasons behind it. A promising direction might be the relationship between computational features in images and choice blindness.

1.2 Choice blindness and image similarity

The main focus of this thesis will revolve around the influence of **objective** similarity between images on the chances that a manipulated question will be noticed, or in other words, that choice blindness occurs. This topic has been previously touched by Taya et al. [5], where it was suggested that similarity is no significant predictor for choice blindness (coeff = 20.02, p = 0.83, t = 20.22). It was, however, shown by Hall et al. [6] that while pairs with exceedingly low similarity get detected considerably more than pairs with fairly high similarity, no correlation was found for pairs in the "gray zone" of similarity. However, an important shortcoming for both these studies, and consequently the main motivation of this thesis, is that similarity rates were **not determined by an unbiased algorithm**, but rather by the opinions of participants. Thus, to further investigate this topic, this thesis will determine the similarity rates in multiple, **unbiased ways**. To be more precise, computer vision algorithms will be used and/or combined to calculate the similarity rates between certain image pairs and ultimately look for correlations with choice blindness chances.

Machines and humans look at images in an entirely different way. Images on a computer are stored in matrices containing intensity values, also known as pixels. Because of this constraint, recognizing shapes or finding similarities between two images is not so easily done by a machine. For example, Figure 1.1 shows the number eight as seen on a screen and as stored by a computer. A human would easily be able to recognize the number 8 and differentiate this shape from the rest of the image. For a computer to do this based on pixels, this is not an easy task.

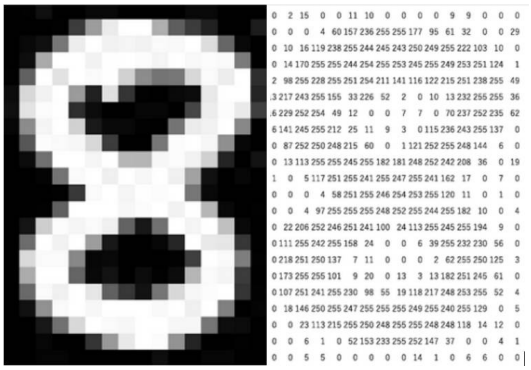


Figure 1.1: The number 8 as portrayed on a screen versus stored on a computer

To get closer to how humans perceive images, the concept of **semantic** features is needed. Semantic features describe the visual content of an image by a correlation of low-level features with the **content** of an image. For example, they correlate an extracted colour such as blue with the sea or sky, white with a building, and so on. Low-level features are acquired by extracting the pixel intensities and finding patterns or finding interesting relationships between them. Chapter 2 will discuss this further. The most prominent research topic of this thesis will revolve around this premise, leading to the following first research question:

RQ 1: Which computational features of an image correlate with choice blindness?

1.2.1 Semantic similarity versus feature-based similarity

This notion results in another viewpoint to research: the influence of **scene properties** on the chances of choice blindness. For example, might participants be more gullible when it comes to scenes that are more colourful (e.g. a pair of images of flowers as opposed to a pair of images of a rocky area)? Or might they be more reluctant to give in to their inner doubts when it comes to a scene packed with objects, as opposed to a scene with an open field? These scene properties could be classified as **semantic features**. Another semantic property, as suggested by Oliva and Torralba [7], suggests images could be reduced to a single *GIST* descriptor, capturing the general shape, or rather "gist", of the scene. This feature is related to a very low dimensional representation of the scene and is described by five descriptors: naturalness, openness, ruggedness, roughness, and expansion. It has been coined the "Spatial Envelope". Chapter 2 explains more about this feature. The key takeaway from this study, however, is that this spatial envelope organizes scene pictures roughly the same as human subjects do, and is able to retrieve images that **share the same semantic category**. This was investigated by letting participants categorize images together on a scale, and comparing this categorization with how the gist descriptor describes these images. Rogowitz et al. [8] suggest semantic information

might be of higher importance for image similarity and recognition in humans. However, in their study, contrast and colour, categorized as low-level features, also seemed to correlate with how images were categorized by the participants.

Another interesting aspect to discuss is how the different criteria of the image scenes themselves compare when it comes to chances of choice blindness occurring. For example, the question could be raised whether people tend to focus more on certain landmarks in faces when recognizing an image. One study suggests that out of all facial landmarks, eyebrows and nose shape and orientation have the biggest influence on face recognition (Sinha, [9]). These findings could be investigated for a correlation with choice blindness chances. In another study, it was shown that the central part of the face might hold the most importance: "the eyes and nose attract most visual attention", Ellis et al. [10].

This is why this thesis will attempt to investigate the difference between several different image properties on the chances of choice blindness occurring, leading to the following second research question:

> *RQ 2: How do different types of image properties or criteria influence the chances of choice blindness occurring?*

While investigating influencing factors of choice blindness is an interesting research topic, examining what humans really **think during** manipulations might help gaining more insights into the process behind choice blindness. This leads to the following research topic of introspection.

1.3 Choice blindness and introspection

The Oxford dictionary defines **introspection** as the careful examination of your own thoughts, feelings and reasons for behaving in a particular way. In other words, it could refer to how much a person is aware of why they chose a certain option when presented with two options. In that sense, it is highly related to the choice blindness paradigm.

1.3.1 Preference change interpretation

Choice blindness could be attributed to two main interpretations: either it is a consequence of (i) *choice-error*, where the participant gives reasons about the wrong choice and is in fact wrong about what her choice is, or (ii) *choice change*, when the participant gives reasons and she is right about what her choice is, but she does not realize that her choice has been changed.

As stated by Bortolotti and Sullivan-Bissett [4]: *"If we discover that being told that we chose cinnamon-apple jam makes us very likely to believe that we chose cinnamon-apple jam and to genuinely endorse that choice as ours in the future, then it is in our interests to pay more attention to, and sometimes challenge, attributions of choices to ourselves"*. To test whether it is indeed a consequence of choice change, manipulated choices could be shown again to the same participant at the end of the experiment. When the participant now chooses the choice she did not choose initially, this implies choice change has occurred.

RQ 3: How much could choice blindness result in changes in personal preference?

1.3.2 Introspection identified by confidence levels

In a study by Rieznik et al. [11], a large experiment was conducted where confidence levels were measured by simply asking the participants how confident they felt about their given argumentation afterward. The findings of this study suggest that participants who did not notice any manipulations are significantly less confident in their answering, indicating some sort of **unconscious detection of self-deception**. They propose it reflects the existence of a neural mechanism unconsciously monitoring our own thoughts. To investigate this hypothesis, the relationship between measured confidence and dissimilarity between two options could be analyzed and checked for correlations. This leads to the following research question:

RQ 4: To what degree does image (dis)similarity affect confidence levels of participants?

In the next Section, one final research topic looks deeper into the difference in influence option **types** might have on choice blindness.

1.4 Choice blindness and difference in option types

Many of the found studies use different methods to analyse the choice blindness effect. However, what seems to be rarely studied, is the difference between different types of options and/or criteria. To give some examples, in a study by Hall et al. [6], the taste and smell of two different consumer goods were used. In another study by Hall et al. [12], political statements were used, and in a study by Thornton [13], touch, or haptic sensations were used, etc. However, although these studies sometimes briefly discuss the difference with other studies and methods, actually comparing multiple option and/or criteria types in one study is rarely done. Of course, the

fact that this will be an online study is a limiting factor. Touch, taste, and smell types are not realistically testable. However, other options could be compared. A first idea might be to use pictures in certain cases and textual ideas in others. Another idea might be to compare long text options versus short text options. Table 2.2 gives an overview of option types that could be compared.

Option type 1	compared with	Option type 2
Images		Textual questions
Long textual questions		Short textual questions
Same shapes, different colours		Same colours, different shapes
Emotion-laden questions		Neutral questions

However, after analyzing existing literature on similarity measures between texts, it was concluded that this might not be a very fitting comparison to make. Firstly, it is difficult to make sense of a comparison of similarities between two texts and two images. For example, one algorithm could calculate a similarity score of 60% between two texts and the other algorithm a score of 60% between two images. While these scores might be accurate, comparing the two images with the two texts in terms of similarity and choice blindness chances might not yield very accurate results simply because they are entirely different data types. While a difference in correlation could be analyzed, it was chosen to focus on image types alone. The question is, does it really make sense to say that two texts are equally similar to each other than two images? Secondly, most accurate text similarity measures perform better as the texts' sizes increase (Gomaa et al. [14]). However, the questions that would be presented in this study would be relatively short 'texts', even when presenting longer questions. For this reason, it might be more interesting to investigate the differences between multiple **image types**. By image types are not meant different image data types, but different types of scenes or objects. After consideration, it was decided to use landscapes, faces, and "polished" faces. Polished faces represent images of faces that were converted to grayscale and have everything blurred out except for the center of the face, to remain more true to the image data set of the study by Johansson et al. [3]. See Chapter 3.1 for more information as to why this choice was made. The reasoning behind the choice of landscapes is twofold. Firstly, it allows to investigate the differences between two truly semantically different types of images. Secondly, landscape and face preferences might differ in how significantly they matter for a person's sense of identity. The study will be conducted using images of all three categories, and afterward, comparisons will be made between them.

RQ 5: How do different types of options compare when it comes to the chance of choice blindness occurring?

1.5 Thesis synopsis

Choice blindness is a paradigm that could lead to important insights into the human psyche and the mechanisms behind humans' often seemingly illogical thought processes. The goal of this thesis is ultimately to test certain hypotheses about choice blindness in order to gain more insights into the process behind it.

The main focus is set on the influence of **computational features in images** on chances of choice blindness occurring. Next to this, the difference in choice blindness chances between **varying image types** (landscapes, faces, and filtered faces) and the difference between **low-level and semantic similarity** are researched. For this reason, similarity between multiple option image types will be calculated using multiple similarity algorithms. Thirdly, **introspection** in the form of lower confidence levels and preference changes due to choice blindness is examined. In addition to this, an analysis will be performed on the correlation between varying **image properties** such as colourfulness, busyness, texture, colour histogram, sharpness, contrast and facial feature shapes, and the chances of choice blindness occurring. This thesis aims to build an online study and perform an analysis of the collected data to test these hypotheses. Said study's question structure will be designed based on multiple defined image similarity measures. In this study, participants will have to decide between two images after which they will motivate their choices. Some of these choices will be manipulated, leading to either a case of choice blindness or a manipulation detection.

The thesis is divided as follows. First, Chapter 2 examines the variety of image analysis measures and why the implemented measures were chosen in detail. Next, in Chapter 3, firstly, the design of the study is explained in detail, with each design choice and testing phase thoroughly discussed. Section 2 of this Chapter then introduces and reviews the different possible frameworks for online experimental studies and their respective (dis)advantages, after which the chosen framework and implementation will be presented in more detail. Afterward, in Chapter 4, the collected data is analyzed and results are discussed. Finally, in Chapter 5, the conclusions and reflections are formulated.

2

Image analysis measures

As it stands, there exist many measures to determine the similarity between two images. From a basic viewpoint, two images cannot be directly compared because the structural units of an image (pixels) do not contain sufficient information about its content, making images a rather complex data type to measure. Hence, the general approach is to represent image content in terms of mathematical features (n-dimensional vectors) and then use classifiers (euclidean distance, neural networks, etc.) to compare these features in order to get a measure regarding their similarities.

The most commonly used techniques to determine image similarity or perform classification is based on **image features**. Concisely put, image features define certain properties of groups of pixels in an image. For example, one type of image feature is **edges**. Edges are points where there is a boundary (or an edge) between two image regions.
Image features can be divided into two main categories [15]: (i) **low-level features** and (ii) **high-level or semantic features**. Low-level image features are image characteristics that are captured by computers for the purpose of recognition and classification (such as pixel intensity, pixel gradient orientation, colour distribution, etc.), while semantic image features are the features that are commonly used by humans to describe images (objects, actions, etc.), and might have the most substantial impact when it comes to image preferences for humans (Ibarra et al. [16]). The following Sections will explain more about how and why certain features are extracted

from images to be compared for similarity.

2.1 Low-level image features

The following subsections explain the low-level image features that will be used to compare the images used in the experiment in detail.

2.1.1 Colour

Colour is one of the most widely used low-level image features. The first attempt towards digital image recognition was a colour-based algorithm (El-gayar et al. [17]). This algorithm works by first calculating a **colour histogram**. A colour histogram is a representation of the distribution of colours in an image. In a digital image, this represents the number of pixels containing certain colours from a list of colour ranges that span the image's colour space. A common example of such a colour space is RGB (red, green and blue). The main drawback of this similarity measure is that colour histograms are dependent of the colour of the image being studied, while being invariant of shape and texture. For two different objects with identical colour distribution, a histogram would yield entirely wrong results in terms of similarity.

To calculate the colour histograms of two images, many methods exist. OpenCV's `calcHist(image, channels, mask, bins, ranges)` is an efficient, straightforward function usable in Python [1]. Here, the `channel` argument represents which channel to calculate a histogram for. It is possible to pass [0] for gray-scale images, and [0], [1], or [2] for colour images if it is necessary to consider the channel green, blue or red respectively. The `mask` argument can be used to only consider a specific region of the image. In this case, the entire image needs to be considered. The `bins` argument creates the possibility to count certain ranges of pixel intensity together. The `range` argument is used to determine the range of intensities considered, usually [0-255]. Below is an example using bins of size 15 and a range of size 256:

$$[0, 255] = [0, 15] \cup [16, 31] \cup .... \cup [240, 255]$$
$$range = bin\_1 \cup bin\_2 \cup .... \cup bin\_n$$

In order to compare the extracted colour histograms of two images, it is possible to use OpenCV's `cv.compareHist` function. This function takes as arguments the two histograms to compare and a `method` flag to establish the comparison algorithm. There exist four main comparison

[1] Documentation available at https://docs.opencv.org/3.4/d6/dc7/group__imgproc__hist.html

algorithms. A first method computes the correlation between the two histograms H_1 and H_2 computed on an image I:

$$d(H_1, H_2) = \frac{\sum_I (H_1(I) - \bar{H_1})(H_2(I) - \bar{H_2})}{\sqrt{\sum_I (H_1(I) - \bar{H_1})^2 \sum_I (H_2(I) - \bar{H_2})^2}} \tag{2.1}$$

Where $\bar{H_k} = \frac{1}{N} \sum_J H_k(J)$ and N is the total number of histogram bins. The polarity of the output of this method indicates what type of correlation is found: either a direct correlation, an inverse one or no correlation at all. The output of this method is either a positive number, a negative number, or zero, indicating a positive, negative, or none-existing correlation respectively. A second method applies the Chi-Squared distance to the histograms:

$$d(H_1, H_2) = \sum_I \frac{(H_1(I) - H_2(I))^2}{H_1(I)} \tag{2.2}$$

The output of this method is zero when the two histograms are identical, and a higher number the less similar they are. A third method calculates the intersection between two images, using

$$d(H_1, H_2) = \sum_I \min(H_1(I), H_2(I)) \tag{2.3}$$

The higher the output of this method, the more similar the two histograms are. And finally, the last measure uses the Bhattacharyya distance, used to measure the "overlap" between the two histograms, using

$$d(H_1, H_2) = \sqrt{1 - \frac{1}{\sqrt{\bar{H_1}\bar{H_2}N^2}} \sum_I \sqrt{H_1(I) \cdot H_2(I)}} \tag{2.4}$$

This method returns a lower number the more similar the two histograms are. Table 2.1 shows the result for each method tested on a combination of the three images from Figure 2.1. The correlation method seemed to perform best for finding semantic similarities, meaning, finding similarity that is more in line with how a human would see similarity, rather than based on some mathematical reasoning. However, except for intersection, all three methods yielded relatively accurate results. For example, the results showed that the two road images correlate positively with each other, while they both correlate negatively with the image of a lavender field. All other methods consistently yield likewise results, except for the intersection and the Chi-squared method. For a code implementation of the extraction and comparison of colour histograms, please refer to Appendix A-1.

	Compared images		
Comparison method	(a) & (b)	(b) & (c)	(a) & (c)
Correlation	0.42	-0.022	-0.018
Chi-Squared	36.18	9.32	18243.17
Intersection	1.25	0.11	0.19
Bhattacharyya distance	0.62	0.96	0.93

Table 2.1: Histogram method comparison on images of Figure 2.1.

(a) Road one (b) Road two (c) Lavender field

Figure 2.1: Three images to be compared based on colour features.

2.1.2 Texture

Most low-level feature-based algorithms do not perform as well as expected when images are subjected to variations in colour distribution, scale, illumination, rotation or affine transform. To overcome these limitations, a new class of image matching algorithms was developed: texture-based algorithms. Texture analysis was developed in the 1970's as a method for image analysis and classification. It is a way of describing the spatial distribution of intensities, which makes it useful in classification of similar regions in different images (Yonggang et al. [18]).

Local binary pattern histogram

Local Binary Pattern (LBP) is a simple yet very efficient texture operator which labels the pixels of an image by thresholding the neighbourhood of each pixel and considers the result as a binary number. Through LBP combined with histograms, images can be represented with a simple data vector. First, the examined image is divided into cells. For each pixel in a cell, the pixels are compared to each of its 8 neighbouring pixels (see Figure 2.2). Now, the pixels are followed along a circle, and where the center pixel's value is greater than the neighbour's value, "0" is written. Otherwise, "1" is written. This gives an 8-digit binary number. Next, the histogram is computed over the cell, of the frequency of each "number" occurring (i.e., each

combination of which pixels are smaller and which are greater than the center). This histogram can be seen as a 256-dimensional feature vector. After concatenating the histograms of all cells, this yields a feature vector for the entire image. This feature vector can be seen as a representation of the window's texture.

Now, all that remains is to compute the distance between these two histograms. This happens in the same fashion as the colour histogram comparison. For a code implementation of the texture feature extraction and comparison, please refer to Appendix A-2.

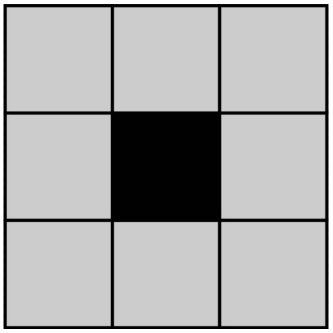


Figure 2.2: A pixel and its eight neighbouring pixels

2.1.3 Image moments

Although not a feature that can be interpreted easily in a non-mathematical way, moments are an important feature for image classification (Hu [19]). It represents a particular weighted average (moment) of the image pixels' intensities. Simple properties of the image which are found via image moments include area (or total intensity), its centroid, and information about its orientation. In other words, they capture information about the shape of an object in a binary image because they contain information about the intensity $I(x, y)$, as well as position x and y of the pixels. To calculate moments that are invariant to translation, scale, and rotation, the **Hu moment** needs to be calculated. Hu Moments (Hu moment invariants) are a set of seven numbers calculated using central moments that are invariant to image transformations. The first six moments have been proven to be invariant to translation, scale, rotation, and reflection, while the seventh moment's sign changes for image reflection. To calculate the Hu moments, first, the **raw moments** M_{ij} of the images are calculated using the formula of 2.5, where i and j are integers (e.g. 0, 1, 2 ....).

$$M_{ij} = \sum_{x} \sum_{y} x^i y^j I(x, y) \tag{2.5}$$

$$\bar{x} = \frac{M_{10}}{M_{00}}, \bar{y} = \frac{M_{01}}{M_{00}} \tag{2.6}$$

Note that these moments depend on the intensity of the pixels and their location in the image. So intuitively these moments are capturing some notion of shape. Here, $\bar{x}$ and $\bar{y}$ represent the center of mass of the image, calculated by equation 2.6. Next, the central moment is calculated using equation 2.7, where the centroid is subtracted off from x and y.

$$\eta_{pq} = \sum_x \sum_y (x - \bar{x})^p (y - \bar{y})^q I(x, y) \tag{2.7}$$

Notice that the central moments are translation invariant. In other words, no matter where the object resides in the image, if the shape is the same, the moments will be the same. Lastly, for scale invariance, the central moments are normalized by the following equation:

$$\mu_{pq} = \frac{\eta_{pq}}{\eta_{00}^{\gamma}}, \gamma = \frac{p+q}{2} \tag{2.8}$$

Next, the output of these calculations is used to calculate the seven Hu moment invariant numbers, using the formulas below:

$\mathrm{H}_1 = \mu_{20} + \mu_{02}$
$H_2 = (\mu_{20} - \mu_{02})^2 + 4(\mu_{11})^2$
$H_3 = (\mu_{30} - 3\mu_{12})^2 + (\mu_{03} - 3\mu_{21})^2$
$H_4 = (\mu_{30} + \mu_{12})^2 + (\mu_{03} + \mu_{21})^2$
$H_5 = (\mu_{30} - 3\mu_{12})(\mu_{30} + \mu_{12})((\mu_{30} + \mu_{12})^2 - 3(\mu_{21} + \mu_{03})^2) + (3\mu_{21} - \mu_{03})(\mu_{21} + \mu_{03})(3(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2)$
$H_6 = (\mu_{20} - \mu_{02})((\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2) + 4\mu_{11}(\mu_{30} + \mu_{12})(\mu_{21} + \mu_{03})$
$H_7 = (3\mu_{21} - \mu_{03})(\mu_{30} + \mu_{12})((\mu_{30} + \mu_{12})^2 - 3(\mu_{21} + \mu_{03})^2) + (\mu_{30} - 3\mu_{12})(\mu_{21} + \mu_{03})(3(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2)$

The image moments alone won't be used to compare image similarities, but they are an important factor for the measure used in Section 2.1.5. For a code implementation of the moment

feature extraction, please refer to Appendix A-3.

2.1.4 Local feature descriptors

Feature descriptors quantify local regions of an image by looking at certain patterns, corners, gradients, etc. While these types of features are widely used in image matching and classification, they might not be a good fit for the problem at hand. This is a consequence of the fact that these features look at patterns unique to the image or object they are calculated on. In other words, an image of the same scene (containing the same mountains or buildings), taken from a different angle or with different lighting would yield almost equal feature descriptors, and thus result in a good match. However, when comparing two scenes of different mountains that, to the human eye, would be perceived as highly similar, feature descriptors could potentially yield next to no matches simply because the details in the pixels do not match adequately.

To test this method as a similarity measure, the two most widely used feature extraction algorithms were tested on a small subset of images. Respectively, ORB (Oriented FAST and Rotated BRIEF) and SIFT (Scale Invariant Feature Transform) were used to test performance differences. SIFT was one of the first widely accepted feature extraction algorithms, as it performed with high accuracy and scale-invariance (Karami et al. [20]). However, SIFT has as a downside that it is relatively computationally expensive. To try and overcome this issue, ORB was created. ORB was designed purely to perform faster than existing algorithms, while trying to remain as accurate as possible. Figure 2.3 shows the ORB feature descriptors matched between two semantically similar images of mountains. Immediately it becomes clear that these matches do not describe similarity in the way necessary for this thesis: clouds or parts of the lake get matched with snow on the mountains, trees barely get matched with each other, etc. SIFT, while being more accurate for image matching in general, performed even worse in finding similarity between two images of different mountains, because it looks in even more detail (see Figure 2.4).

Figure 2.3: ORB feature descriptors matched between two images of mountains

Figure 2.4: SIFT feature descriptors matched between two images of mountains

To define what counts as a 'match' when comparing image descriptors between images, a threshold is given to the algorithm. Any two descriptors whose distance does not exceed this threshold get categorized as a match. To then quantify the similarity between the images, the total number of matched key points is divided by the number of key points found in the image with the lowest number of key points. To further show that these algorithms are not fit for this problem, the SIFT method resulted in a 2.5% similarity score between the two showcased images.

2.1.5 Combination

While most of these low-level features are interesting on their own and may be used to draw interesting conclusions in the data analysis, a general measure that looks at all these aforementioned low-level features is needed. It is possible to use a combined vector that represents all of these features in one single vector by concatenating them using Python library **NumPy**'s **hpstack** function[2]. Before combining the separate vectors into one, a normalization to the same scale is of the order. If the vectors are not normalized in advance, some features might weigh down more on the global similarity than others, depending on their respective scales. This normalization is performed by using the Python sklearn library's normalize function[3]. By calculating the distance between these global feature vectors, it becomes possible to get a notion of the similarity-based on multiple low-level features combined. For a code implementation of this measure, please refer to Appendix A-4.

2.2 High-level or semantic image features

While low-level features are what most modern supervised image classification algorithms are based on, high-level or semantic feature similarities lie significantly closer to what humans perceive to be similar, because the brain tends to search for patterns and global features rather than detailed low-level pixel patterns (Palumbo and Allasia [21]). Smeulders et al. [22] define the semantic gap as "the lack of coincidence between the information that one can extract from

[2]Documentation available at: https://numpy.org/doc/stable/reference/generated/numpy.hstack.html
[3]Documentation available at: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.normalize.html

the visual data and the interpretation that the same data have for a user in a given situation". Image annotation or labeling attempts to fill the semantic gap by mapping low-level visual features into high-level concepts, either manually or through machine learning algorithms (see 2.2.1). In this section, two main similarity measures are introduced, namely machine learning and the GIST algorithm. Next to this, some semantic properties that are interesting for the data analysis are introduced and discussed.

2.2.1 Machine learning

As mentioned before, low-level features on their own might give a strong notion of how similar two images are on pixel-level, but this does not always mean humans will perceive them as similar images. This is where **machine learning** comes in. A sophisticated and mathematically complex evolution of machine learning algorithms, also known as **deep learning**, provides a method for giving semantic meaning to low-level features. Deep learning is a methodology for learning high-level concepts about data, frequently through models that have multiple layers of non-linear transformations. Learning high-level concepts about data means that deep learning models take data, for instance raw pixel values of an image, and learn abstract ideas like 'this is an animal' or 'this is a tree' about that data.

The name deep learning originates from the fact that it functions with multiple layers or *neurons*, where "raw" data arrives at the first layer and gets processed through all the next layers in a certain fixed order, until it is refined and processed enough to classify it. Figure 2.5 shows a convolutional neural network, where each layer performs some elementary calculation on the data. For example, one layer could search for certain patterns in the edges of the images, and give the image a value based on these patterns. The next layer could then look at the colours of the image, and so on. To get the processed data from one neuron to the next, it is rectified through some non-linear transformation. Sometimes, in-between the neurons, max pooling is used to reduce the spatial resolution. This is visualized in Figure 2.5 by the height of the neurons getting smaller and smaller. The width, on the other hand, becomes larger and larger after each iteration, visualizing the growth of the semantic information [23]. A convolutional neural network is only one type of neural network. However, its importance in this thesis comes from the fact that it is specifically used for image-based classification. To determine the similarity between two images based on a CNN, a library created by Apple, called TuriCreate[4] is used. This library makes it possible to create an image similarity model that can give scores to the similarity between two images. This happens in three main stages:

1. Uses a pre-trained CNN classifier on a large, general data set (e.g ImageNet, with 1000 categories and 1.2 million images).

[4]https://apple.github.io/turicreate/docs/userguide/image_similarity/

2. The outputs of each layer in the CNN can be viewed as a meaningful vector representation of each image. Extract these feature vectors from the layer prior to the output layer on each image of your task.

3. Create a nearest neighbors model with those feature vectors as input.

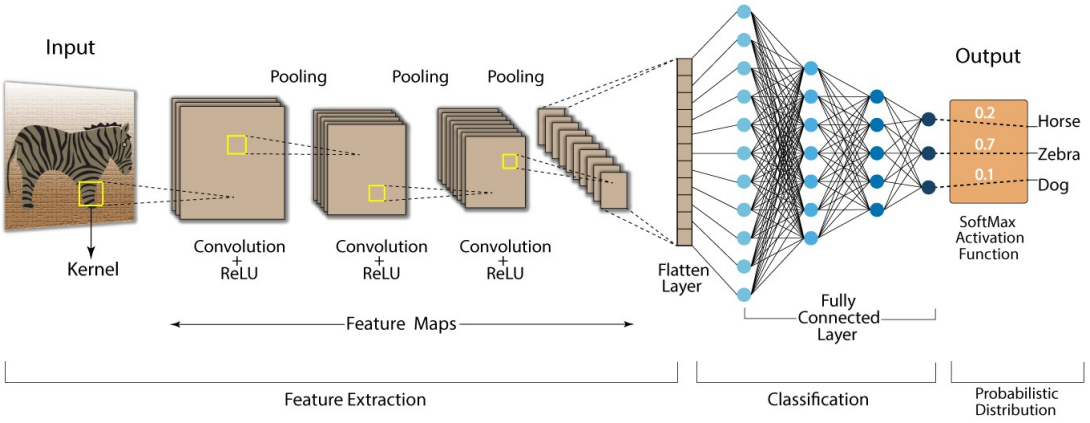


Figure 2.5: Convolutional neural network [1]

.

For images of type face, another Python library was used, called face-recognition[5]. Built using dlib's state-of-the-art face recognition, this model has an accuracy of 99.38% on the Labeled Faces in the Wild benchmark [24]. While this library was created with the intent of serving as a face recognition tool, this tool is based on a similarity score. When this score is high enough, two faces are recognized as the same. In the case of this thesis, these scores can be used as a measure of similarity.

2.2.2 GIST features

While features such as texture, shape, colour, etc. catch details of the scene and calculating the distance between these details might give a notion of similarity, semantic similarity might be better approached another way. Oliva and Torralba [7] propose a computational model of the recognition of real-world scenes that bypasses the segmentation and the processing of individual objects or regions. As discussed in 1.2.1, this procedure is based on a very low dimensional representation of the scene, called the Spatial Envelope. The study shows that specific information about object shape or identity is not a requirement for scene categorization and that a holistic representation of the scene informs about its probable semantic category.

[5]https://pypi.org/project/face-recognition/

In other words, it shows that images could be categorized based on a vague shape of the scene rather than intricate details. Figure 2.6 illustrates how a scene with a coarse layout could easily be misinterpreted as something it is in fact not. In a study by Navon [25], most people were confident in describing the spatial layout of a street when shown the image on the left. However, the high-resolution image reveals that the buildings are in fact furniture. By considering a scene like an individual object with a unitary shape rather than looking at a scene as a configuration of objects, a new representation becomes available: the **GIST feature**. The spatial envelope is determined by extracting five semantic properties from the image:

- Degree of Naturalness
- Degree of Openness,
- Degree of Roughness, i.e. the size of its major components,
- Degree of Expansion, i.e. how many lines converge and diverge in the image,
- Degree of Ruggedness, i.e. the deviation of the ground with respect to the horizon.

Figure 2.6: Illustration of the effect of a coarse layout on scene identification and object recognition, illustrating the strength of the global spatial layout in constraining the identities of the local image structure.

The key conclusion from this study for this thesis is that a GIST descriptor is meaningful to human observers: "the spatial envelope model organizes scene pictures as human subjects do, and is able to retrieve images that share the same semantic category", Oliva and Torralba [7].

Thus, computing a distance between the GIST descriptors of two images could yield an accurate result in terms of semantic similarity, seeing as it gives a notion of the distance between their respective semantic categories.

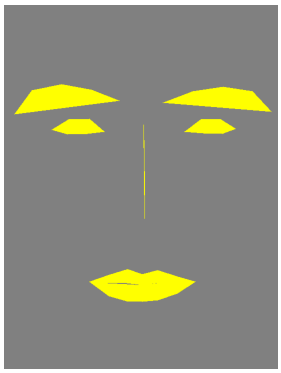
2.2.3 Semantic properties

As previously mentioned, image features alone do not usually represent how humans process images. To test correlations with more semantic features, some semantic properties are extracted for analysis. The following subsections will introduce these properties and explain how they are extracted.

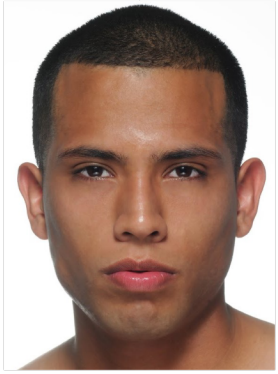
2.2.3.1 Shape matching

To compare separate features in images of faces (as a means of partially answering the research question of Section 1.4), such as the mouth, the nose or the eye shapes, shape **contours** can be used as a similarity measure. First, the coordinates of the different face 'landmarks' are determined using the discussed neural network. These coordinates are converted to contours and compared based on their contour moments (see Section 2.1.3). This way, it is possible to determine whether two faces differ more or less in certain landmarks (e.g., two faces with drastically different noses will have a high contour moment distance for the nose coordinates).

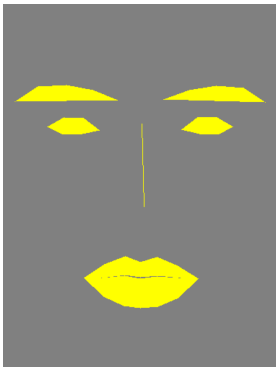
(a) Face one



(b) Drawn face landmarks of face one



(c) Face two



(d) Drawn face landmarks of face two

Figure 2.7: Two faces and their respective drawn facial landmarks.

For a code implementation of this extraction, please refer to Appendix A-5.

2.2.3.2 Colourfulness

While a colour histogram yields a good representation of the colour differences between two images, image **colourfulness** might also be of interest for image recognition. For example, two images could differ drastically in terms of colour histogram if, for example, one image is very green and the other is very blue. However, the colourfulness of both these images would be substantial. Could the difference in colourfulness be of significance for the chances of choice blindness? To test this hypothesis, colourfulness scores are calculated for each image and the distance between these scores is used as a similarity measure. Hasler and Suesstrunk [26] performed a series of experimental calculations through which they derived a metric that correlated with what viewers perceived as a certain level of colourfulness. It was found that an opponent colour space representation along with the mean and standard deviations of these values correlates to 95.3% of the survey data. Equations 2.9 and 2.10 respectively represent the opponent colour space representation where R is Red, G is Green, and B is Blue. In the first equation, rg is the difference between the Red channel and the Green channel values. In the second equation, yb represents half of the sum of the Red and Green channel values minus the Blue channel values.

$$rg = R - G \tag{2.9}$$

$$yb = \frac{1}{2}(R + G) - B \tag{2.10}$$

Next, the standard deviation and mean of the opponent colour space representation are computed using equations 2.11 and 2.12. Lastly, using these values, the colourfulness metric C is computed by adding the standard deviation to one-third of the mean.

$$\sigma_{rg,yb} = \sqrt{\sigma_{rg}^2 + \sigma_{yb}^2} \tag{2.11}$$

$$\mu_{rgyb} = \sqrt{\mu_{rg}^2 + \mu_{yb}^2} \tag{2.12}$$

$$C = \sigma_{rgyb} + 0.3 * \mu_{rgyb} \tag{2.13}$$

The resulting metric is then used to compare images on the basis of colourfulness. Please refer to Appendix A-6 for a code implementation.

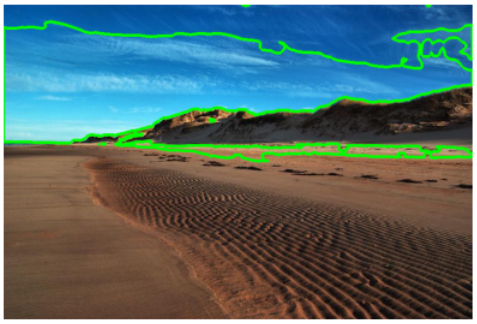
2.2.3.3 Busyness

In addition, an interesting semantic idea is to not look at what the objects or shapes in the image represent, but to look at how **many** objects or defined shapes exist. For this reason, an algorithm was derived that is able to quantify the **busyness** of an image to a certain degree. Here, busyness is defined as the amount of closed-boundary contours found in an image after a thresholding is performed. To do so, first, the image is converted to grayscale. Next, bilateral filtering is performed on the image. A bilateral filter is a non-linear, edge-preserving, and noise-reducing smoothing filter for images. As such, it preserves all pixels with more intense gradient changes (meaning pixels where the difference in intensity with their neighbouring pixels is high) and blurs the others (Aswatha et al. [27]). This leads to an image where the edges are still seen, but the rest is blurred out. After this, Otsu's thresholding is applied to the image. Otsu's thresholding is a thresholding technique as defined by Otsu [28] that can perform automatic image thresholding. In the simplest form, the algorithm returns a single intensity threshold that separates pixels into two classes: foreground and background. It determines these classes by classifying pixels as part of a class if its intra-class intensity variance is minimized. This way, "objects" or shapes that strike the eye are classified and coloured together, while the background is classified and coloured differently. Lastly, very small objects are eliminated by applying an erosion kernel to the image. Erosion works by looking at each pixel's neighbouring pixels and replacing its value with the minimum value of the surrounding pixels. This way, small objects that should not be counted as objects are eliminated. Now, contours can be sought after in the resulting image, leading to an outlining of objects. This happens by looking for significant changes in gradient in the image and then tracing this border along its path, and attempt to find a closed border in this manner (Suzuki and Be [29]). All that is left now is to count the amount of outlined objects, and a metric for busyness has been acquired. Figure 2.8 shows this process for two images. Figure 2.8a shows an image of a rocky seaside after Otsu's thresholding. Because of the many rocks in the foreground, this leads to a detection of a relatively high amount of "objects" or a rather high busyness of the image, namely 131. The outlines are drawn over the image in Figure 2.8b. Figure 2.8c then shows the same filtering done on a lesser busy image of a plain of sand with a hill in the background. This leads to a small amount of separated "objects" (the sky, the hill and the plain get separated), and thus ultimately to a lower busyness metric of merely seven. For a full implementation of this algorithm, please refer to Appendix A-7.

(a) Otsu's thresholding on image of rocky beach.

(b) Image of rocky beach with drawn-over contours.

(c) Otsu's thresholding on image of a sandy plain.



(d) Image of a sandy plain with drawn-over contours.

Figure 2.8: Two images after busyness algorithm filtering.

2.2.3.4 Image properties

Lastly, some easy-to-extract image properties are defined and used as analysis measures. Namely, the properties sharpness and contrast are determined using functions from the OpenCV library. The number of edges found in an image inversely correlates with the blurriness of that image. Finding the number of edge pixels is possible using OpenCV's Canny[6] function, which returns the image array where all edges received the value 255 and all other pixels receive the value 0. This function's algorithm works much like how contours are found in 2.2.3.3. To find the sharpness, all that is left to do is simply calculate the mean of this edged array. As a contrast measure, the Root mean square (RMS) contrast is calculated. RMS contrast does not depend on the angular frequency content or the spatial distribution of contrast in the image. It is defined as the standard deviation of the pixel intensities (Peli [30]). Please refer to Appendix A-8 for code implementations of sharpness and contrast extraction.

[6]Documentation available at: https://docs.opencv.org/master/dd/d1a/group$_{imgproc}{}_{feature.html}$

2.3 Conclusion

Out of all the above image analysis measures, the global, GIST, and neural network similarity measures will be used as a means to attempt answering the first and fourth research questions as defined in Chapter 1. Aside from the local feature descriptors measure, all other measures will be used to attempt to answer the second research question. The question structure and order will be based on the three main similarity measures, as will be discussed in the next Chapter.

3

Design of the Online Study

This Chapter first covers the design choices and test phases that were required to fine-tune the study to a state where it is ready to answer the research questions at hand. Afterward, the framework choice is explained, followed by some details of the implementation.

3.1 Study design

In this Section, the study's design is illustrated, and choices made are explained in detail. Due to the psychological nature of this study, several test phases were necessary before a final design was decided upon. For this reason, design choices will be explicated in a fluent manner, rather than e.g. explaining image choices, question order, etc. in different Sections.

3.1.1 General design and structure

In this Subsection, the general design and structure of the study are discussed. Firstly, the used image data sets and the corresponding pre-processing will be introduced. Next, the question structure and order are explained, followed by an explication of each testing phase and their corresponding conclusions.

3.1.1.1 Image data sets (and pre-processing)

As discussed in Chapter 1, one of the research goals of this study revolves around the difference in choice blindness chances between different types of images. After consideration, it was decided that images of landscapes and faces would be the types in question. As such, a landscape and face data set was retrieved from the internet (respectively from sources [31] and [32]). To minimize any influencing factors apart from facial features, the data set of faces was cleaned out to only contain images with uniform backgrounds and faces facing forward. Next to this, all face images were resized to 210 x 280 pixels, and all landscape pictures were resized to 340 x 290 pixels. In the study by Johansson et al. [3], a previous study on choice blindness, faces were also used as images presented to the participants. However, in this case, they were presented in grayscale, and with blurred edges, as to not let the hair or clothing influence too much. This polishing was also performed on the face data set of this thesis. Figure 3.1 shows two faces with uniform background and orientation, taken from the used data set. Figure 3.2 shows the same two faces after polishing.

Figure 3.1: Two faces used in the study, with uniform backgrounds and face orientation.

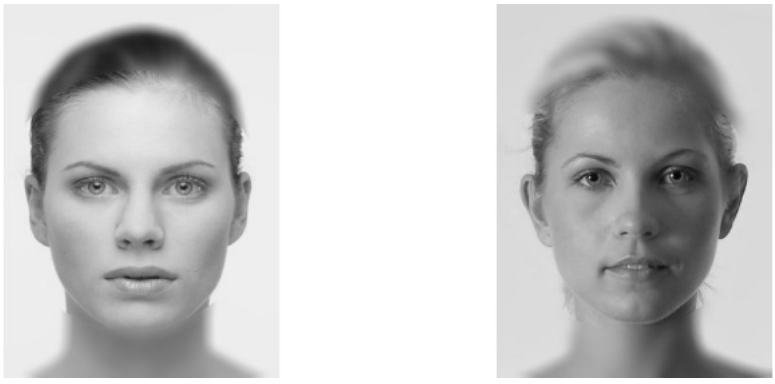


Figure 3.2: Two faces used in the study, polished.

Landscapes were chosen from multiple categories, going from mountains, to deserts, to beach scenes. Figure 3.3 shows an example of two landscape images from the data set.

Figure 3.3: Two landscapes used in the study.

3.1.1.2 Question structure and order

The next step in building a study is to set up the question structure and order. To answer the research questions as effectively as possible, the study is structured in a way such that each image type gets represented equally. In addition to this, images get manipulated based on three different similarity algorithms, as discussed in Chapter 2. This happens in round-robin style, where every three questions are of the next image type, and chosen based on the next algorithm's similarity. In other words, if questions one to three are of type landscape, and are chosen based on the neural network's similarity rate, questions four to six will be of type face, and chosen based on the GIST descriptor similarity. To not deviate too greatly from the study by Johansson et al. [3], where each participant receives 15 pairs of faces, a total of 27 questions are presented. This choice is based upon the fact that in the case of this study, not only images of faces will be researched, but also images of landscapes and even images of polished faces. Thus, at least double the amount of questions will be necessary. Using this system, the study is given a rainbow-like structure, which is illustrated in Table 3.1. In the study by Johansson et al. [3], three out of the 15 pairs were manipulated, leading to the decision of a total of six manipulations in this study. In the study by Johansson et al. [3], participants were either asked to give a long or a short report or argumentation on their choices. The study contained 111 short and 117 long non-manipulated, and 81 short and 82 long manipulated reports. Long reports were up to one minute long, and short reports were roughly five times shorter, or 12 seconds long: "the average length of the reports was 20 words for the short ones and 97 words for the long ones". Based on this, the mean of long and short reports was used leading to roughly 30 seconds of allowed deliberation time per argumentation.

Set	*Global similarity*	*Neural net similarity*	*GIST similarity*
Landscapes	1	10	19
	2	11	20
	3	12	21
Faces	22	4	13
	23	5	14
	24	6	15
Polished faces	16	25	7
	17	26	8
	18	27	9

Table 3.1: The structure of a set in the used study.

Image choices

To choose an image based on a certain algorithm, similarity scores have to be extracted into JSON files for analysis. Listing 3.1.1.2 shows an extract of the code used to calculate and extract GIST similarity scores. Similarity scores are calculated between all of the images by comparing their GIST descriptors one by one. Then, for each image, the most similar images are stored as pairs into JSON files. To inspect the full code, please refer to Appendix A-9.

```
...

# Calculates similarities between all GIST scores of each image, adding them to a nested dictionary
# Returns said dictionary containing 'image1: image2: similarity' values
def calculate_similarities(images, scores):
    gist_distances = defaultdict(dict)
    for image1 in images:
        for image2 in images:
            if image1 != image2:
                distance = spatial.distance.cosine(scores[image1], scores[image2])
                gist_distances[image1][image2] = 1 - distance
                print('Distance between ', image1, ' and ', image2, ': ', distance)
    return gist_distances

```

```
15
16  # Calculate scores, distances and write them to json file
17
18  scores = calculate_gist_scores(landscape_images)
19  distances = calculate_similarities(landscape_images, scores)
20
21  with open('gist_landscape_scores.json', 'w') as fp:
22      json.dump(scores, fp)
23  with open('gist_landscape_distances.json', 'w') as fp:
24      json.dump(distances, fp)
25
26  # Finds most similar other image for each image, based on previously
    ↪  calculated similarity scores
27  def find_most_similar_pairs(distance_json_name, score_json_name,
    ↪  output_json_name):
28
29  # ------------------------------------------------------------ #
30  # -------- Open JSON files containing necessary values ------- #
31  # ------------------------------------------------------------ #
32
33  with open(distance_json_name) as json_file:
34      distances = json.load(json_file)
35  with open(score_json_name) as json_file:
36      scores = json.load(json_file)
37
38  # ------------------------------------------------------------ #
39  # ------------------ Find most similar pairs ----------------- #
40  # ------------------------------------------------------------ #
41
42  # Sort nested dictionary on similarity score, so each image's dictionary is
    ↪  sorted separately
43  sort_scores = {key: dict(sorted(val.items(), key=lambda ele: ele[1],
    ↪  reverse=False)) for key, val in distances.items()}
44
45  # Keep a dictionary to keep track of already used images, for efficiency
    ↪  purposes.
46  used = {}
47  for image in sort_scores.keys():
48      used[image] = False
```

```
49
50  # Iterate over each image and take first image from dictionary, as this will
    ↪  be the most similar image
51  pairs = []
52  for image in enumerate(sort_scores.keys()):
53      if used[image[1]] is not True:
54          best_image = list(sort_scores[image[1]].keys())[0]
55          if best_image in used and used[best_image] is not True and best_image
            ↪  != image[1] and sort_scores[image[1]][best_image] >= 0:
56              used[image[1]] = True
57              used[best_image] = True
58
59              pair = (image[1], best_image, round(100 * (1 -
                ↪  round(sort_scores[image[1]][best_image], 3)), 2))
60              pairs.append(pair)
61
62  # Write pairs to JSON file
63  with open(output_json_name, 'w') as fp:
64      json.dump(pairs, fp)
65
66  ...
```

Listing 3.1: Calculation and storage of GIST descriptors

The same process is repeated for neural net and global similarity scores. Please refer to Appendix A-4 for the code implementation of this global similarity score extraction.

In order to successfully compare similarity rates for different algorithms, the scores should be normalized to yield a number in the same range. First, the highest and lowest distance scores are calculated. Next, each score is replaced by a normalized score as given by Equation 3.1. Here, x' represents the normalized score, x represents the previous score, and $max(x)$ and $min(x)$ represent the maximum and minimum scores respectively.

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{3.1}$$

Listing 3.1.1.2 shows the implementation of this normalization.

Now, the scores are successfully extracted and the images are ready to be chosen. For this part, the structure of 3.1 was used, where each three questions where chosen based on the next

algorithm type. Initially, the idea was as follows. Two main question sets were to be created, with the first and the second set respectively containing manipulations of relatively higher and lower similarities. This way, the two sets could be compared for higher or lower manipulation detection rates. These sets were achieved by consulting the created JSON files and randomly selecting image pairs containing high or low similarity scores. For set one, image pairs were chosen in the 80-100 range, while for set two, images were selected in the 60-80 range. So, if for example, two images have a similarity score of 88.3%, they could be chosen for set one. They then become one of the six question pairs that are to be manipulated. In other words, if a participant were to choose the first image out of this pair, they would get asked to give arguments as to why they chose the second image. All of the non-manipulated pairs are uniformly selected in the 0-100% range. The question number these manipulated questions get assigned is also chosen randomly. The only rule that was followed, is that no two manipulations are ever performed back to back. The six manipulated pairs were always chosen such that two images of each image category get included.

3.1.1.3 Detection recognition

Because of the psychological nature of this study, multiple challenges were encountered in the design of this study. The most prominent challenge was to capture the participant's awareness of the manipulation process. In other words, how can the program confirm whether a participant has realized the fact that their question was manipulated? In the study by Johansson et al. [3], which was not an online, but rather a tangible study, the participants simply told the interviewers, who then wrote down that the manipulation was detected. Because the study at hand is conducted online, this is no longer possible. Thus, a method needs to be derived to automatically collect detections.
A first idea is to not only ask the participants for their argumentation through the means of a text box, but also ask them using multiple radio buttons. These radio buttons could then serve as a subtle way to literally ask the participants whether they noticed the manipulations. The participants are asked to further argue their choice by picking their main reason. These radio buttons then consist of multiple reasons as to why they chose this image. The key option here would be the last option, "I did not choose this image / I chose this image accidentally". If the participant were to choose this option, it can be safely assumed they have realized the manipulation.

- This image really stood out
- This image appeared more colourful
- This image spoke to me personally

- I did not choose this image / I chose this image accidentally

Without these radio buttons, concluding a detection would be based purely on the textual answer of the participant. This is possible by performing a text analysis on the answers, and determining whether the manipulation has been detected or not by means of a dictionary. For further details on the implementations of this analysis, please refer to Chapter 4. This method has the advantage that none of the premises of the study are being given away. The danger that using radio buttons drags with it, is that participants might start having inner doubts about the true intention of the study, which influences their future answers. This, of course, is sub-optimal. For this reason, two extra sets were created, identical to set one and two, except for the argumentation. These two new sets would use radio buttons, while the first two sets use text boxes. Section 3.1.2 shows how both systems were tested before one was decided upon.

After setting up the question structure, the study is ready to be implemented. For technical details of how this was done, please refer to Chapter 3.2.

3.1.2 Testing

After designing and implementing the study, testing can begin. For this stage, the study was already deployed and functionally tested. The following testing phases thus purely exist for the purpose of testing whether the necessary data was collected and/or any changes would need to be made to the design in order to make choice blindness occur. For the purpose of testing, the study link was shared through social media with multiple participants who filled it in for free.

In the first testing phase, the main goal was to determine the best manipulation detection recognition technique. A small group of people ($N = 16$) was asked to participate and was randomly assigned one of the two testing sets. By splitting the questions into two separate testing sets, the detection rates could be compared between the two methods. Results seemed to follow the presumption that the radio button method too heavily implied a second, hidden intention behind the study. Manipulation detection rates were significantly higher for the radio button test set than for the text box test set (see Table 3.2).

Recognition method	Detection rates (%)
Text box	71
Radio button	86

Table 3.2: Detection rates using different manipulation detection recognition methods

Concluding from test phase one, the text box seems to be the superior recognition method, as

detection rates lie lower while remaining above zero. However, detection rates still lie significantly too high (detection rates lie around 25% in the study by Johansson et al. [3]). To accommodate this issue, some more changes needed to be made. To more accurately simulate the effect of choice blindness from previous studies, where the manipulations happened physically, behind the participants' back, an artificial delay is inserted between each question and its argumentation. This way, participants won't see the switch happen on the screen, but will instead look at a placeholder text asking them to wait a moment for the next question to load. These changes were implemented to test in test phase two.

In the second testing phase, a new group of people was asked to fill in the updated study ($N = 13$). Again, two testing sets were created to compare results with the updated script. Results showed a significant lowering of the detection rates, although still too high (see Table 3.3).

Delay	Detection rates (%)
Yes	51
No	73

Table 3.3: Detection rates depending on delay usage.

To lower the detection rates even further, further measures needed to be taken. After closer inspection of the chosen images, it could be concluded that the similarity rates still fared too high. To accommodate for this, new sets of questions were created, where the mean similarity was kept at a higher rate than the previous sets. A third and final testing phase ($N = 17$) using these new question sets resulted in a detection percentage of 34% for the set with elevated similarity rates (see Table 3.4).

Similarity rate mean (%)	Detection rates (%)
78	52
87	34

Table 3.4: Detection rates depending on image similarity mean.

As this is an acceptable detection rate, the study is now ready to be deployed and participants are ready to be hired. The next Chapter will now introduce the possible frameworks that could be used to implement this study, and then explain why the used framework was chosen.

3.2 Implementation Framework

In order to collect data to test the hypotheses, a framework is required to perform the online study. The rise in popularity of web-based data collection in both experimental and survey-based research has its reasons. It is a flexible, efficient and location-independent approach (Felix et al. [2]). Many software packets for this purpose exist. However, the majority of them are proprietary, and thus most likely too expensive and not extensible enough for this thesis. In this Chapter, firstly, two of the most prominent possible frameworks for running / building experiments in a web browser are discussed. Afterward, the chosen framework and the implementation will be discussed in more detail.

3.2.1 An overview

Many different frameworks exist for this purpose, all with their own differences, but also many matching properties. In the coming subsections, two of the most noteworthy frameworks will be discussed.

3.2.1.1 Lab.js

Lab.js is a **free, open, online** study builder for the behavioural and cognitive sciences (Henninger [33]). One of the most important aspects of this framework is that it relies on a GUI, and requires no code to build experiments. This could be seen as an advantage, because the process of building becomes simpler and more intuitive. However, this could also mean that the building becomes less flexible and more rigid, which is seldom a good attribute. To make up for this, the possibility to add HTML, CSS, and JavaScript adds more to the control over the studies' presentation and behaviour. Another downside is that as of the date of writing (feb. 2021), lab.js does not seem to be fully documented yet. Only the basics have been covered, which might make it more of a challenge to build a working experiment with this framework. On the other hand, lab.js could enable more efficient data collection. In addition to this, lab.js supposedly makes it easy to export parts of studies in an editable format for sharing and re-use, facilitating collaboration and cumulative science (Felix et al. [2]).

Workflow

The main editor interface can be seen in Figure 3.4. Lab.js works with four different **building blocks**: Canvas and HTML **screens** and sequence and loop **flows**, as can be seen in Figure 3.5. Screens are used to show participants the necessary information e.g. the two options from

which they get to choose. Multiple screens can be added and can be switched between. In order to combine screens into one, where e.g. a certain user interaction could cause a switch between them, a sequence flow is used. To repeat one component multiple times, a loop flow is used. In order to create variation each time a component is activated, parameters can be defined that change between repetitions, and set the respective levels across the loop iterations [34].

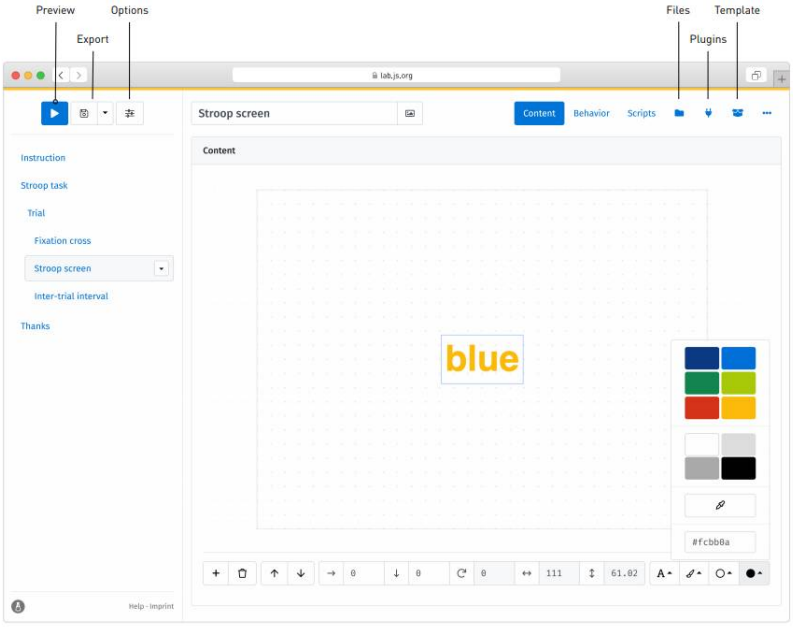


Figure 3.4: The visual editor in action, assigning a colour to a piece of text [2].

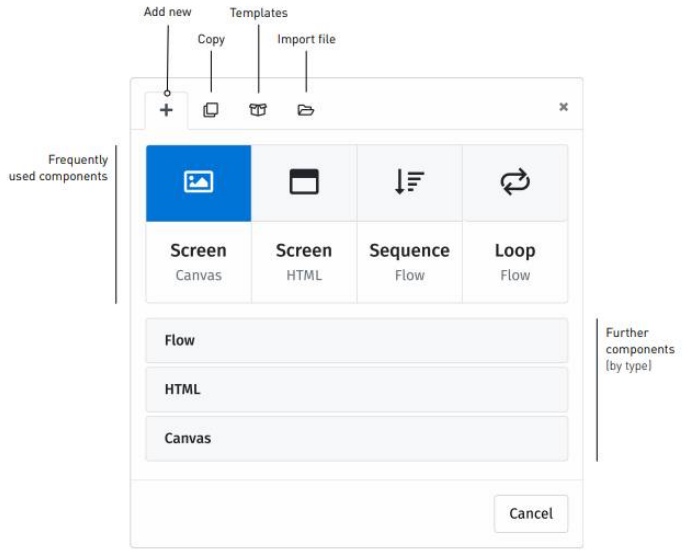


Figure 3.5: Lab.js interface for adding new building blocks [2].

Data collection

Data are collected automatically as the study progresses (Felix et al. [2]). There exist many options in lab.js to collect data. Data can be collected *offline*, where it is exported into a zip archive as a comma-separated values (CSV) file. This enables easier sharing and local testing and collection. Another, more controllable option is to use a *PHP backend* bundle. Through this, data collection can be fully automated by sending it from the client and gathering it in a database.

3.2.1.2 PsyToolkit

PsyToolkit is a software package for running psychological experiments, best classified as behaviour experiment software. It was developed by Dr. Gijsbert Stoet as a software project at Washington University in St. Louis in 2005 (Stoet [35]), and also comes free of charge. One of the biggest advantages of this framework is the ease of getting the experiment working online. After scripting a survey and compiling it using an online interface, the survey can immediately be made available (Stoet [36]). Adding to this, it has been extensively documented online using a documentation web page as well as multiple YouTube tutorial videos [37]. However, its scripting language is rather rigid, making it not very scalable and not easy to debug.

Workflow

To create surveys, PsyToolkit uses its own high-level scripting language. Among the most useful functions this language provides, is **conditional jumping** and **randomization of question or question set order**. Surveys are created as text in a text box in the browser. Scripts consist of one or more blocks of text, with each block representing either a question or a jumping statement. To differentiate between different types (e.g. radio button question, normal question, jumping statement, etc.), a line starting with 't:' is added to the block, indicating the type. Each block can optionally be assigned a label, indicated with an 'l:', for future reference or for use in the data analysis. To add a question text, a 'q:' line is used. In the case of this thesis, it is desired to give the participants a maximum amount of time to complete each question, e.g. 20 seconds. Adding the line 'o: maxtime 20s' provides this functionality. To show images, it is simply possible to add a 'i:' line, followed by the name of the images that need to be shown. The two possible options to be displayed are then defined by writing '- Option' at the bottom of the block. An example of a question is seen in Listing 3.1.

```
1 # question 1
2
```

```
3 l: 1_landscape_q_nm
4 t: radio
5 o: maxtime 20s
6 i: landscape_image_1_l.jpg landscape_image_1_r.jpg
7 q: Which of the following images do you like the best?
8 - Image 1
9 - Image 2
```

Listing 3.1: PsyToolkit question script

To ask the participants for an argumentation, it is possible to use a jump statement to lead the participant to the desired question based on their previous answer. When the question is to be manipulated, simply lead the participant to the argumentation of question two if they answered with option one. For example, in Listing 3.2, first, the answer to the question is checked with an if statement on lines three and four. Based on these statements, a jump is performed to either the argumentation of the right option (the second option), or the left option (the first one).

```
1 # jump to argumentation question 1 based on answer
2
3 l: set1_jump2
4 t: jump
5 - if $1_2_q = 2 then goto 1_2_landscape_a_m_glob_r
6 - if $1_2_q = 1 then goto 1_2_landscape_a_m_glob_l
7
8 l:
9 t: jump
10 - goto 1_3_q
11
12 l: 1_2_landscape_a_m_glob_l
13 t: textbox
14 o: maxtime 30s
15 i: landscape__85_.jpg
16 q: On question 2, you chose this image. Can you tell us why?
17 - {w=100,h=5}
```

Listing 3.2: PsyToolkit argumentation jump script

Counterbalancing

Counterbalancing is an important feature of carrying out psychological experiments. In short, it means that different participants carry out different elements of an experiment in a different

order. To implement this, PsyToolkit provides the possibility to generate a random number. Based on this number, the participant jumps to a (set of) question(s). An example of such a counterbalancing technique is showcased in Listing 3.3.

```
1  # first , we randomly jump to one of six question sets for
       counterbalancing
2
3  l: chooserandom
4  t: set
5  - random 1 10
6
7  l:
8  t: jump
9  - if $chooserandom == 1 then goto 1_1_q
10 - if $chooserandom == 2 then goto 1_1_q
11 - if $chooserandom == 3 then goto 2_1_q
12 - if $chooserandom == 4 then goto 3_1_q
13 - if $chooserandom == 5 then goto 3_1_q
14 - if $chooserandom == 6 then goto 4_1_q
15 - if $chooserandom == 7 then goto 4_1_q
16 - if $chooserandom == 8 then goto 5_1_q
17 - if $chooserandom == 9 then goto 5_1_q
18 - if $chooserandom == 10 then goto 6_1_q
```

Listing 3.3: PsyToolkit counterbalancing script

Data collection

Data are stored on PsyToolkit web servers. After each completion of the survey, the new data is appended to a **data.csv** file. This spreadsheet file contains all the answers of each participant (one row per participant). For each participant, a raw data survey-question file is available (as a text file). This file is also named in the spreadsheet file. For each action performed by a participant, the timing is also added to this data, with **millisecond precision** (Kim et al. [38]). In order to analyze the data, it is possible to simply download this file from the servers after all the participants have performed the survey.

3.2.2 Conclusion

Initially, the idea was to build the experiment using the Lab.js framework. This decision was mostly based on the fact that the framework had a clean, seemingly easy-to-use user interface,

multiple templates online, and made use of Javascript. However, soon after starting to implement the experiment in this framework, issues started to arise. As expected, the graphical user interface made the framework quite rigid and offered less space for creativity and/or freedom. As it turns out, the framework was not necessarily intended to use for more complex experiments where intermediary data is to be temporarily stored and reused later on in the experiment. For example, due to the GUI-oriented layout, it is not possible to simply store data in a variable and use this data in another iteration of the experiment. None of the offered templates had a need for such functionality, and as such, no examples were found online. However, there exists a way to build an experiment from the ground up, still using the Lab.js library for functionality such as loops, but building all the rest using simple Javascript, HTML, and css. While this might seem like a solution to the data problem, where now intermediary data could simply be stored in global variables throughout the experiment, this also takes away one of the only advantages that seemed to make Lab.js stand out. For this reason, it was decided to drop Lab.js as the experiment-building framework, and **PsyToolkit would be the framework of choice moving on**.

However, while PsyToolkit offered the most important functionality to perform this study, it also had its shortcomings, as already discussed. Its rather rigid scripting language results in very limited scalability. Due to this, adding sets, questions or new functionality can be a time-consuming process. For-loops, functions or other repeatable scripting structures are not included in the language. Thus, any new functionality must be hard-coded into the script, resulting in a substantially large script. Another difficulty with this framework lies in its disability to debug the code easily. When a compilation error occurs, no line number is given. Combining this with the inevitably large script, debugging can become quite strenuous. Nevertheless, PsyToolkit scripting allows to

- save older participant answers for later use (necessary for question manipulation),
- use older answers for conditional jumps (necessary for question manipulation),
- add an artificial delay between questions,
- show multiple images on one question,
- ask for argumentation through text boxes,
- ask for argumentation through radio buttons (necessary for test phases),
- set a time limit per question.

The first two points can be solved using conditional jumps as explained above. E.g., when the participant chooses the first image on a pair that will be manipulated, the program jumps to the argumentation of the second image using a conditional jump depending on the answer. The third, fourth, fifth, and sixth points are also implemented using the above-mentioned functionality. As

for the artificial delay, this was possible by adding a question of type 'info' in between questions and their argumentation. By adding a 'maxtime' and 'mintime' of two seconds to this question and hiding this timer, the program appears to be loading, as it were. Listing 3.4 shows the scripting block that achieves this functionality.

```
1 l:
2 o: maxtime 2s hide mintime 2s hide
3 t: info
4 q: Please wait one moment for the next question to load...
```

Listing 3.4: PsyToolkit artificial delay script

Combining all these implementations, the script performs all the necessary functionalities to attempt to answer the research questions. The next step is data collection and analysis.

```
# Open file containing scores
with open('glob/glob_sim_faces_polished.json') as json_file:
        scores = json.load(json_file)

# Search for maximum and minimum among scores
max = -1
min = sys.maxsize
for image in scores.keys():
    for other_image in scores[image].keys():
        if max < scores[image][other_image]:
            max = scores[image][other_image]
        if min > scores[image][other_image]:
            min = scores[image][other_image]

# Normalize scores using (value-min)/(max-min)
for image in scores.keys():
    for other_image in scores[image].keys():
        scores[image][other_image] = (scores[image][other_image] - min) / (max
        ↪  - min)

# Similarity = 1 - distance
for image in scores.keys():
    for other_image in scores[image].keys():
        scores[image][other_image] = 1 - scores[image][other_image]

# Save the normalized scores to a new JSON file
with open('glob/glob_sim_faces_polished_normalized.json', 'w') as fp:
    json.dump(scores, fp)
```

Listing 3.2: Normalization of global similarity scores

4 Results

In this Chapter, pre-processing and analysis of the data is elucidated. The first Section will look closer at how the data was collected and prepared for analysis, after which the second Section will go deeper into the analysis itself, and the conclusions that could be drawn from it.

4.1 Data pre-processing

Initially, the data consists of two main components: the question configurations, which include all questions with their respective number, set number, image names, similarity rates, etc. And the collected data, which consists of all the data collected from the hired participants ($N = 173$) having performed the study. As discussed in Chapter 3.2, collected data is downloaded in CSV files. Converting CSV files to in-memory data structures is done easily using the Pandas library (Python Data Analysis Library)[1]. Pandas is an open-source library that helps organize data across various parameters, depending upon requirements. This makes it one of the best libraries for this purpose [39]. In order to draw meaningful conclusions and information from the collected data, it will need to be merged with the pre-known data.

4.1.1 Merging pre-known data with collected data

Table 4.1 shows the first three rows of pre-known data. The collected data contains one row for each submission. Each row in turn contains all answers of this participant, the participant's ID, and some more unused columns. In order to concatenate collected data for a certain question

[1]Documentation available at: https://pandas.pydata.org/docs/

from a certain participant, both DataFrames need to share some column(s).

number	*set*	*img_type*	*img_1*	*img_2*	*manipulated*	*glob_sim*	*neural_sim*	*gist_sim*
1	1	*landscape*	59	15	*False*	99	79.5	66.7
2	1	*landscape*	85	49	*False*	99.5	84.2	80.2
3	1	*landscape*	88	105	*False*	98.8	97.2	85.2

Table 4.1: First three rows of the pre-known data

In this case, both data sets can be merged on their question number and set number. However, adding the question and set number to the collected data is not done automatically. As seen in Listing 3.2, it is possible to give each question a unique label. By parsing out the question and set numbers from these labels, it becomes possible to extract the correct row from the pre-known data. Adding the collected data to this row and appending this row to a new DataFrame yields a DataFrame containing all the necessary data to begin analysis. Listing 4.1 demonstrates the code that performs this merge.

```
# Make new DataFrame to save all values, merged with hardcoded values
merged = pd.DataFrame()

# Loop over all participants
for participant in tqdm(participants):

    # Take row of data of this participant
    copy_of_participant = data[ data['participant'] == participant]

    # If this participant actually participated
    if not copy_of_participant.empty:

        # Transpose this row as to get all columns (questions) as rows
        trans = copy_of_participant.transpose()

        # Now, iterate over all these rows (questions) to add necessary data
        ↪  to new DataFrame
        for index, row in trans.iterrows():

            # We skip over the participant row
            if index != 'participant':

```

```
22                  # Match and capture the set and question number from the
                    ↪  question labels
23                  m = re.match('([0-9][0-9]?)_([0-9][0-9]?).*', row.name)
24
25                  # Take only question name from row name (e.g. 1_2_q:1 becomes
                    ↪  1_1_q)
26                  name = row.name[:-2]
27
28                  # If a match is found
29                  if m:
30
31                      # Create new row from hardcoded template (with unvariant
                        ↪  data: algo type, image type,...)
32                      # E.g. if we are currently on question 1_3_q, we take the
                        ↪  hardcoded row of question 3, set 1
33                      new_row = pre_known[ ( pre_known['number'] ==
                        ↪  int(m.group(2))) & (pre_known['set'] ==
                        ↪  int(m.group(1))) ]
34
35                      numeric = True
36
37                      # Check whether choice, argumentation or requestion using
                        ↪  RegEx, and add to answer type
38                      if re.match('.*_a_.*', name):
39                          new_row['answer_type'] = 'argumentation'
40                          numeric = False
41                      if re.match('.*_q', name):
42                          new_row['answer_type'] = 'choice'
43                      if re.match('.*_r', name) and len(name) < 8:
44                          new_row['answer_type'] = 'requestion'
45                      if re.match('.*_radio_.*', name):
46                          numeric = True
47                          new_row['answer_type'] = 'arg_choice'
48
49                      # Check total time used for this question using
                        ↪  data_times.csv and add to row
50                      new_row['time_ms'] = data_times_all[
                        ↪  data_times_all['participant'] == participant
                        ↪  ][name].values[0]
```

```
51
52                      # Add actual answer of participant to row
53                      if numeric:
54                          new_row['num_answer'] = row.values[0]
55                          new_row['answer'] = row.values[0]
56                          new_row['str_answer'] = 'N/A'
57                          new_row['numeric'] = True
58                      else:
59                          new_row['str_answer'] = row.values[0]
60                          new_row['answer'] = row.values[0]
61                          new_row['num_answer'] = 5
62                          new_row['numeric'] = False
63
64                      # Add participant ID to row
65                      new_row['participant'] = participant
66
67                      new_row['label'] = row.name
68
69                      # Finally, add new row to new DataFrame
70                      new = new.append( new_row )
71
72  # As a last step, we drop all rows containing NaN datatypes, as these are the
73  # questions that were not (or could not be) filled in by the participant and
    ↪  are thus rendered useless
74  new = new[new['answer'].notna()]
75  new
```

Listing 4.1: Merging of the pre-known data with the collected data

4.1.2 Manipulation detection recognition

Lastly, all argumentative answers need to be processed in order to determine whether a question's manipulation was detected or not. For this purpose, each of the argumentative answers was analyzed in each of the previously discussed test phases in search for recurring words that were typical for detections. A dictionary containing these words was set up, after which each of the answers was filtered for these words.

```
detection_words = ['no', 'wait', 'didn\'t', 'didnt', 'not', 'remember',
↪  'chose', 'wait', 'don\'t', 'dont','wrong','\\?']

df.loc[ (manip_df['str_answer'].str.contains('|'.join(detection_words),
↪  flags=re.IGNORECASE)), 'detected'] = 1
```

Listing 4.2: Recognition of answers implying manipulation detection.

After this initial filtering, samples of answers that were labeled as detected were analyzed to search for special cases or new words to append to the dictionary.

4.2 Data analysis

Now, the data is fully organized in one Data Frame, where each row represents one answer of one participant, and contains all the necessary data. Analysis can begin. In this Section, each subsection will cover a different research question and try to answer these questions as closely as possible.

4.2.1 General analysis

In test phase 3, a detection rate of 34% was observed for 13 participants. This study saw an increase of about 6% to this detection rate. Out of 881 manipulated questions, 356 were detected, resulting in a 40.41% detection rate. In 293 false-positive cases, the participants' answers were categorized as detections while the question was not actually manipulated. Some examples of these answers are "Not much of a thug-like appearance" or "Water is the source of life, the desert is not, the choice was easy". Because the word "not" is an exceptionally good indicator of a detection, these false positives can not be flushed by deleting this word from the dictionary. A possible solution to this problem would be to add phrases like 'did not choose' to the dictionary instead of simply 'not'. However, this would require knowing each of the phrases participants use to indicate they have detected a manipulation, and could result in too many false negatives. Other examples of false positives include participants who honestly believe the question was manipulated, while it was in fact not. This could be the case because they have already been manipulated on previous questions, causing them to be overly cautious on the next ones, leading to false-positive detections. After inspecting each pair of manipulated questions' detection rate, another interesting pattern was found. It was found that most pairs that lied in the lower detection range (5 to 40 %) seemed to also be the questions that, on average, appeared the most in the **first half** of the study. Concurrently, questions with relatively high detection rates appeared more in the second half of the study. To look closer into this notion, the detection rate was plotted against the question number (indicating the order that the question appeared in the study). Figure 4.1 shows how the relationship between these variables indeed

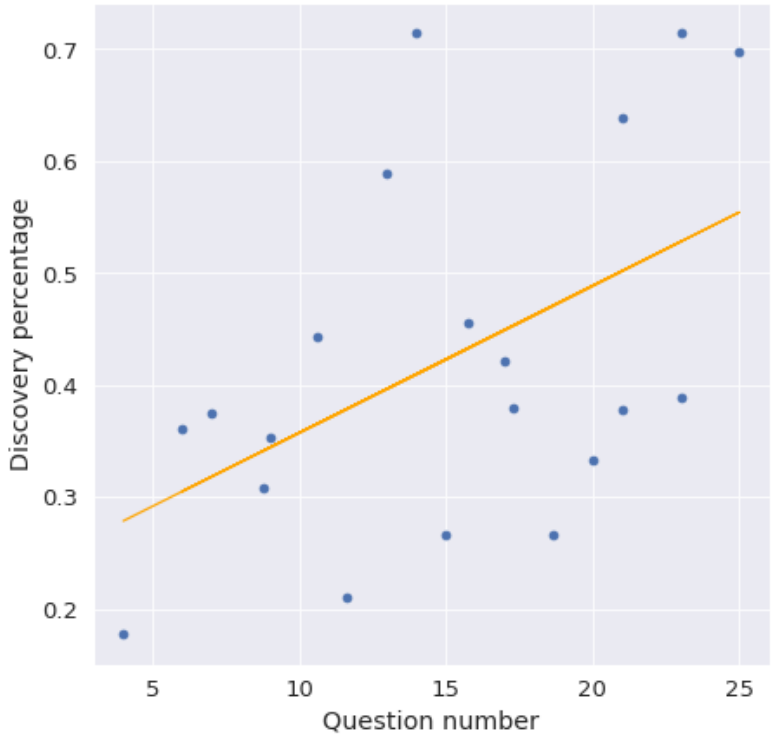


Figure 4.1: Scatter plot of detection rate against question number.

turns out to be direct, suggesting that as pairs turn up later in the study, the chances of their manipulation being detected rise. This connection could be explained by a rise in suspicion by the participants after initial manipulations. As a participant gets manipulated a first time, their suspicion and guard might still be low. As more manipulations happen, newer ones are likely to be detected more easily. A study by Sagana et al. [40] suggests that participants are not more likely to notice manipulations when warned about technical difficulties. However, after an initial manipulation detection, chances start to rise.

4.2.2 Which computational features of an image correlate with choice blindness?

The main focus of this study lies around this research point. For this purpose, three similarity measures were used as discussed in Chapter 2. As discussed in Section 1.2, previous studies on the relationship between choice blindness chances and similarity suggest similarity is no significant predictor for choice blindness (coeff = 20.02, p= 0.83, t= 20.22) (Taya et al. [5]). While it was shown by Hall et al. [6] that pairs with exceedingly low similarity get detected considerably more than pairs with fairly high similarity, no correlation was found for pairs in the "gray zone". As discussed in 1, another important aspect is that for these studies, the similarity rates were determined by asking participants to rate how similar they found these pairs. Naturally, similarity between two options could be regarded as a subjective measure

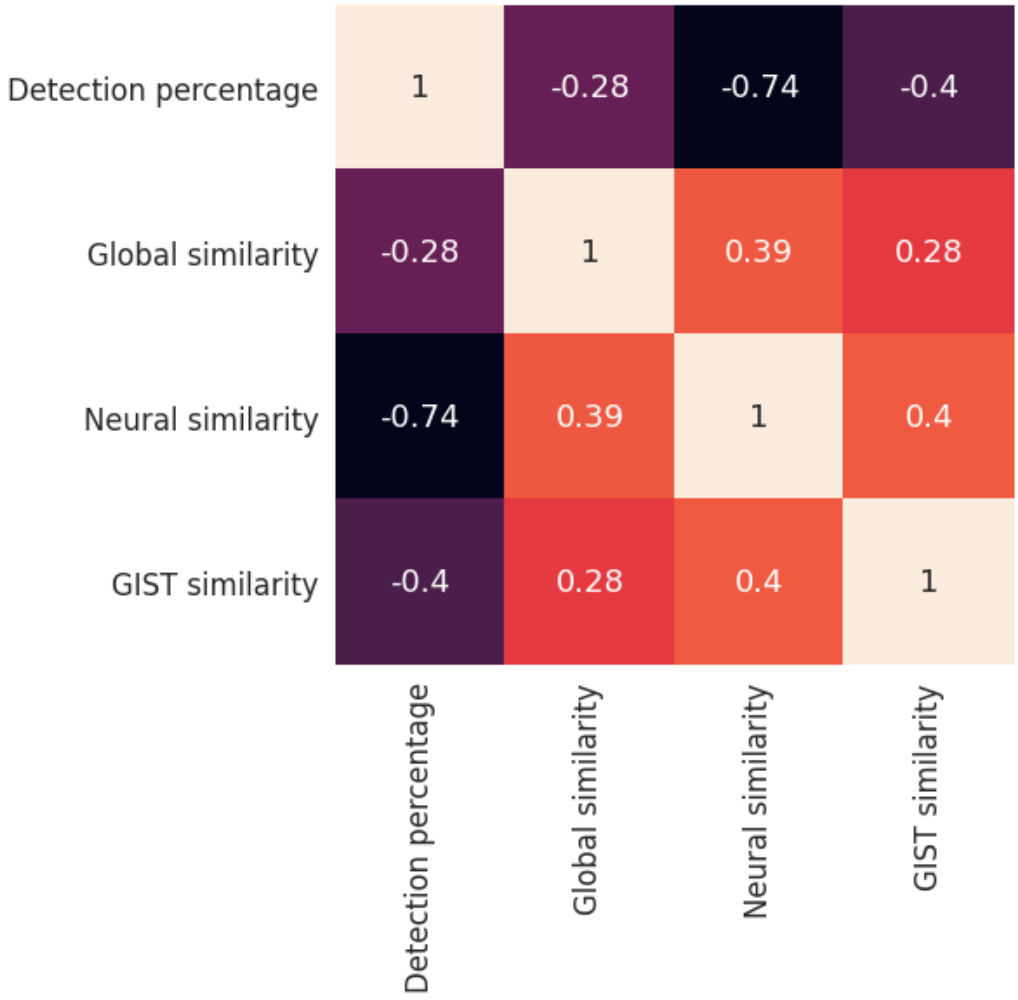
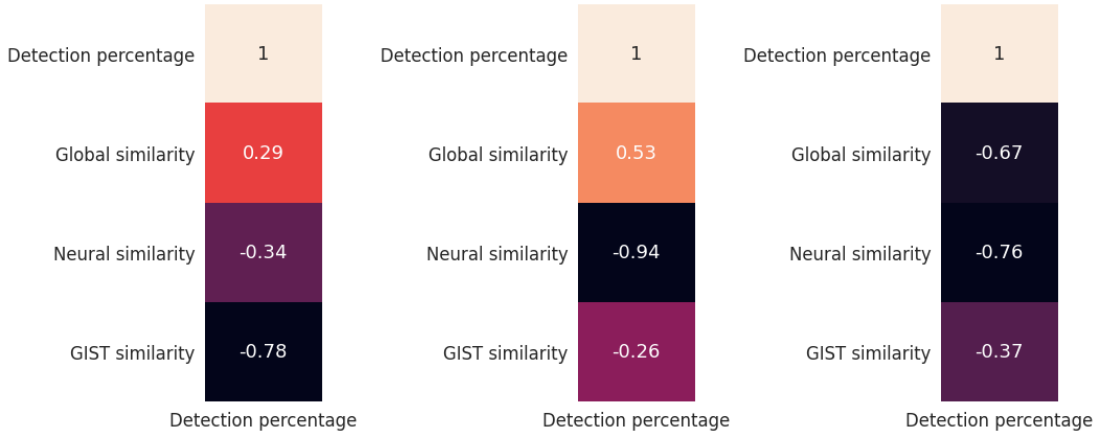


Figure 4.2: Correlation matrix of collected data.

rather than an unbiased one. Thus, in this subsection, the correlation between choice blindness chances and three **computer-based** similarity measures is investigated.
In order to gain a first, general insight into the relationship between these properties, a correlation matrix is created (Figure 4.2). The most crucial column of this matrix is the column of detection rates. When inspecting this row, it becomes clear that all three similarity measures seem to have an inverse correlation with the detection rate. This means that, in this study, the images with a higher similarity showed fewer detections and vice versa. However, this is a general result, calculated over all questions. To refine these results, a grouping on image type was done and correlation matrices were calculated on these groups (see Figure 4.3). Refined results show that while for landscapes (Subfigure 4.3a) and polished faces (Subfigure 4.3c), all three correlations remain inverse, face images result in a direct correlation for the global similarity measure (Subfigure 4.3b).

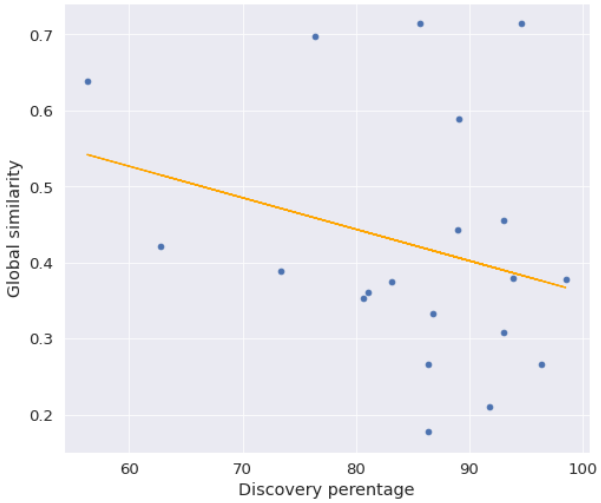


(a) Landscape images (b) Face images (c) Polished face images

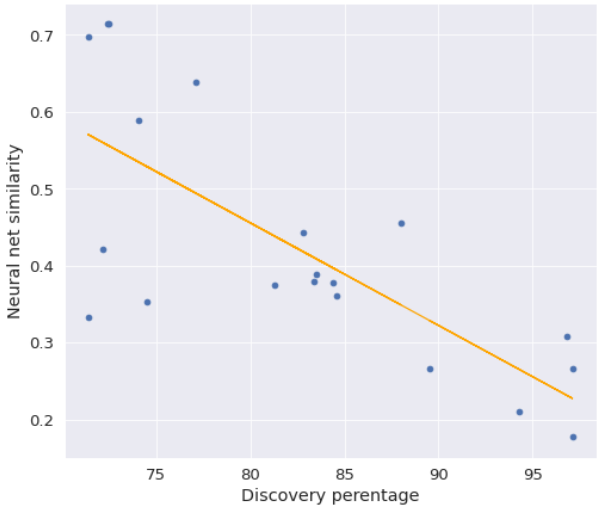
Figure 4.3: Extracted columns of correlation matrices of collected data, grouped per image type.

As discussed in Chapter 2, each measure concentrates on different aspects of an image. In light of the fact that GIST descriptors are mainly calculated based on the general shape and spatial properties of the image, it might seem logical that it does not serve as a good indicator for semantic similarity in regard to faces. The resulting direct correlation with detection rates could then be attributed to the lack of feature extraction performed by the GIST algorithm. As suggested by Taya et al. [5], people tend to recognize faces based on facial landmarks or features rather than colour or shape. While extracting features is easily done using today's frameworks, calculating some accurate similarity rate based on these features is where the difficulty lies (it is straightforward to determine where certain edges in an image lie, and which edges or corners are similar to those in another image, but rather difficult to calculate some similarity rate on this). This is why a neural network would most likely calculate more accurate scores for the task at hand. This prospect also clarifies the direct correlation with the global similarity measure. Because this measure is based on colour, shape, and texture, the same problems arise. However, as discussed by Taya et al. [5], texture is one of the other more prominent features people tend to observe when recognizing faces. This might explain the slightly higher correlation than GIST similarity. Because the neural network extracts a significant amount of features, processes them using its layers, and calculates a final similarity score based on many of these features combined, it results in a better predictor. These points also give a possible explanation for the rather significant inverse correlation between GIST-based similarity and detection rates.

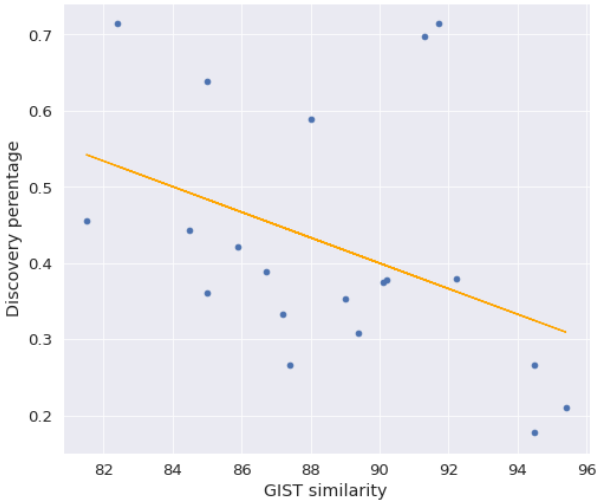
To look in more detail into this prospect, the detection rate is plotted against similarity rates for each of the three measures. For all three measures, an inverse relationship exists between the similarity rate and the detection rates, indicating that as the similarity decreases, the odds of detection increase. Figure 4.4 shows that a neural net seems to have the foremost inverse relationship with detection rates. (x- and y-axes labels are incorrectly placed, this will be fixed)



(a) Global similarity



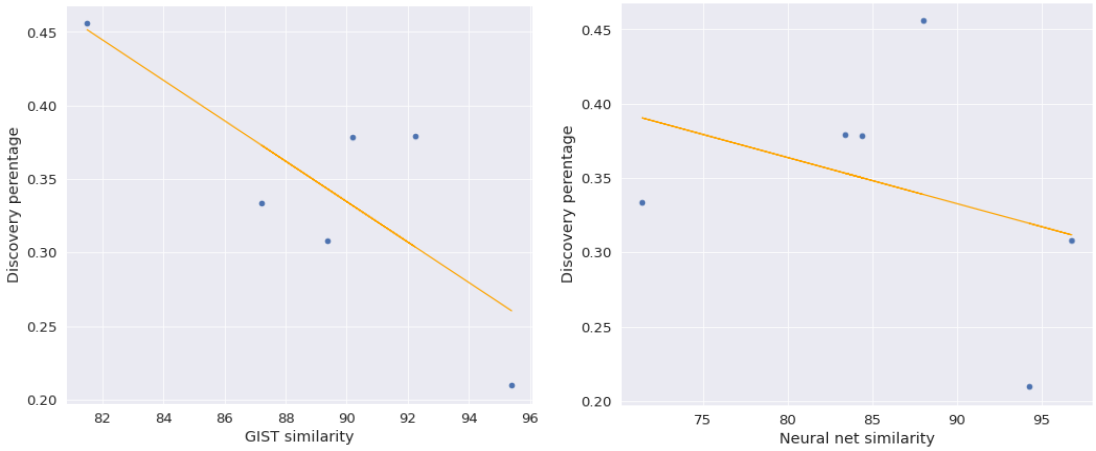
(b) Neural net similarity



(c) GIST similarity

Figure 4.4: Similarities plotted against detection rates for all three similarity measures.

However, grouping these questions by image type reveals that this does not count for each of these types separately. As Figure 4.5a suggests, similarity as the GIST descriptors define it appears to be an excellent predictor for choice blindness detection in landscape images.



(a) GIST similarity

(b) Neural net similarity

Figure 4.5: Plotting of similarities against detection rates for landscape type images.

To quantify how strong this relationship could work as a predictor, the Coefficient of determination or R-squared value is calculated. In statistics, the R-squared value is the proportion of the variance in the dependent variable that is predictable from the independent variable(s) (Carpenter [41]). This value ranges from 0 to 1, representing a model that explains respectively none to all the variation in the response variable around its mean. In order to calculate the R-squared score, first, the coefficients of this linear regression are determined using the Numpy library's **polyfit**[2] function. R-squared compares the fitted regression model to the detection percentage's mean values (the baseline model), which is the worst possible model. In this way, the R-squared value denotes whether this regression model improves upon simply predicting based on the mean value (Miles [42]). To calculate this value, the total and unexplained variation are calculated and divided from each other, and this quotient is subtracted from one (Equation 4.1). The total variation is equal to the residual sum, which is the sum of squared errors of the regression model. The unexplained variation is the total sum of squared differences between the observed values and their mean. This comes down to comparing the observed y-values to the baseline model: the mean of the observed y-values.

$$R^2 = 1 - \frac{Total\ Variation}{Unexplained\ Variation} = 1 - \frac{SS_{RES}}{SS_{TOT}} = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \overline{y})^2} \tag{4.1}$$

Applying this formula to the linear function of 4.5a results in an R-squared score of 0.6156. This is a relatively high value for a study in behavioral sciences [43], indicating that GIST similarity is indeed a strong manipulation detection predictor for landscape type images. To define a mathematical function describing this relationship, the coefficients calculated earlier using the

[2]Documentation available at: https://numpy.org/doc/stable/reference/generated/numpy.polyfit.html

polyfit function can be utilized. This results in the following function:

$$y = -44.78321x + 104.72 \tag{4.2}$$

Here, y refers to the predicted detection rate and x refers to the similarity rate as calculated by GIST. Continuing this practice of linear regression on the other relationships, it is possible to derive each of the relationships' R-squared scores to determine other good predictors. A summary of these R-squared values is given in Table 4.2. In behavioral studies, an R-squared score of 0.50 or higher is already deemed significant [44]. Notable R-squared values are indicated with gray cells. This further strengthens the previous assumption that the GIST algorithm might the superior algorithm for manipulation detection prediction in landscape images.

	R-squared value		
Image type	Global	Neural net	GIST
Landscapes	0.0822	0.1148	0.61567
Faces	0.2834	0.8848	0.0675
Polished faces	0.4494	0.5829	0.1361
All	0.0757	0.5482	0.1630

Table 4.2: R-squared values for each algorithm-image type combination.

From this table, it is clear that the neural net indeed might be the best predictor overall. Figure 4.7 demonstrates this inverse relationship with faces and polished faces respectively, exhibiting clearly how the errors are relatively small. Again, functions can be derived that explain these relationships and attempts to predict detection percentages. After performing the same fitting technique as previously explained, the following functions were defined. Function 4.3 defines the relationship between detection chances and face similarity as determined by the used neural net. These functions were all derived on the presumption that a linear model best captures the relationship between the variables at hand. However, in some cases, it might be better to fit other functions such as exponentials or higher-order polynomials [45]. This assumption of which model best fits the data is called **inductive bias**.

$$y = -47.1372x + 104.4028 \tag{4.3}$$

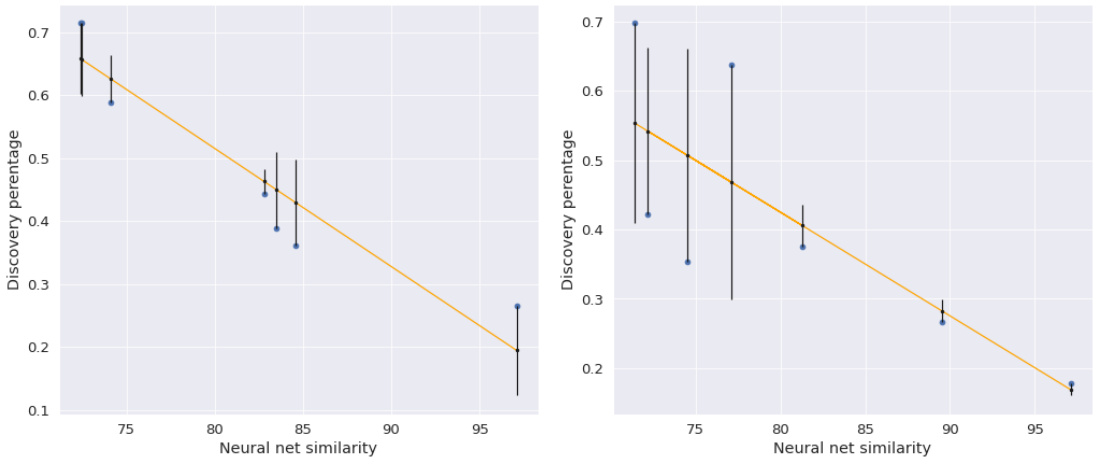
Function 4.4 then defines the relationship between detection chances and polished face similarity as determined by the used neural net.

$$y = -38.9270x + 96.7244 \tag{4.4}$$

Lastly, a function is defined that describes the relationship between the chances of manipulation

detection in polished faces and the global similarity metric:

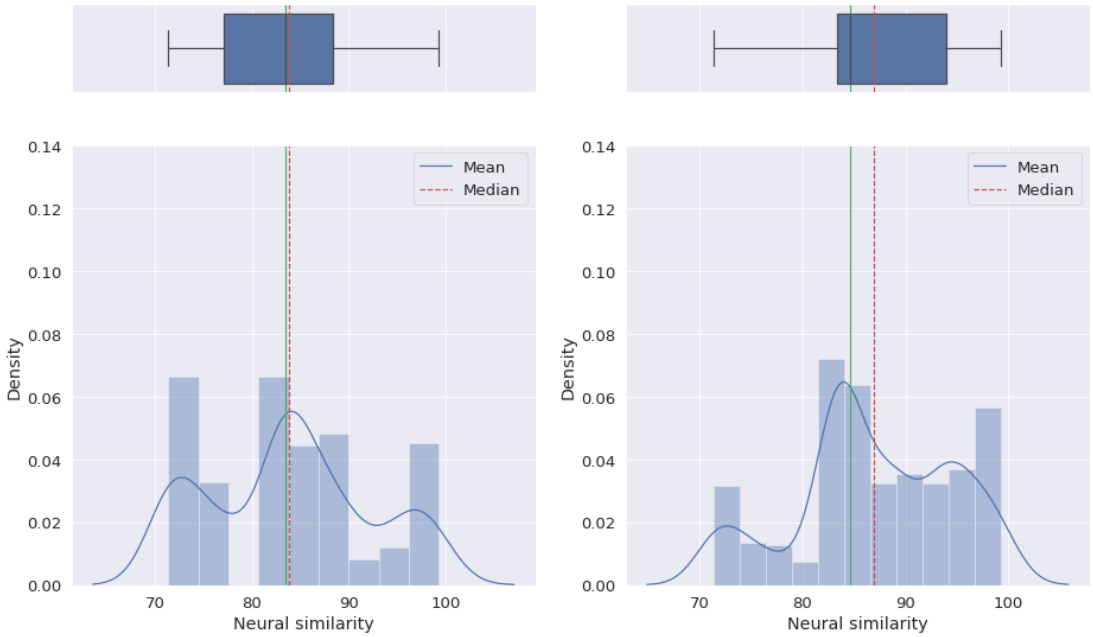
$$y = -49.0523x + 97.9737 \tag{4.5}$$



(a) Faces (b) Polished faces

Figure 4.6: Plotting of neural net similarities against detection rates for images of (a) faces and (b) polished faces, showing errors against linear regression model.

In order to further investigate the distribution of similarities and define a cut-off range of similarities where detection rates start to rise, detection histograms are plotted for the neural network similarity measure. This measure is chosen because it has the highest correlation with detection chances overall. Figure 4.7 illustrates histograms and box plots of neural net similarities for manipulated questions. The left graph only shows detected manipulations, while the right graph shows undetected manipulations. As expected, distribution appears to be more skewed to the right for undetected manipulations. The middle 50% of the population, as represented by the blue box in the box plot graphs, also sees a significant shift towards the higher similarity rates for undetected manipulations. While the mode of both distributions remains around a similarity rate of 82, both the mean and median switch over to the other side of the mode. Concluding from the inter-quartile range of these box plots, exactly half of the undetected manipulations have a similarity rate of 83 to 94, while for detected manipulations, this range is 77 to 88. This indicates that $[83-88]$ might serve as a vague cut-off **range** for manipulation detection in terms of neural network similarity.



(a) Detected manipulations (b) Undetected manipulations

Figure 4.7: Neural net similarity histogram- and box plot for (un)detected manipulations.

Figure 4.8 shows the joint-plot of neural network similarity rates and the question number, where each axis' histogram is plotted on the edge. This Figure makes it clear that starting from a similarity rate of 83, detected manipulations are higher in numbers than undetected manipulations. Concurrently, detected questions are in the majority for similarity rates of 79 and under, indicating a more clear cut-off similarity **rate** of 79.

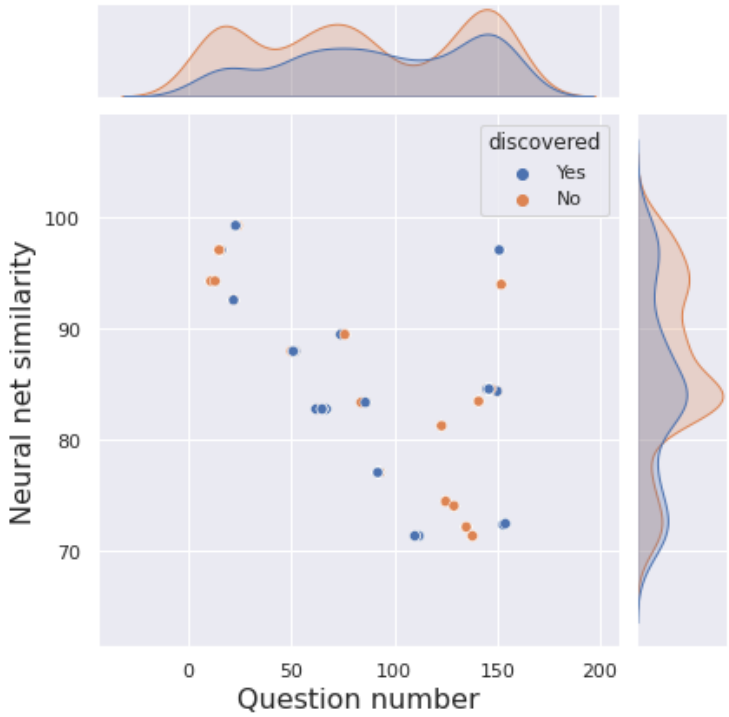


Figure 4.8: Joint-plot of neural network similarity rates and detection rates.

4.2.3 How much could choice blindness result in a change in personal preference?

This question is rather straightforward to analyze. For the purpose of this question, at the end of each study session, the participant was asked to choose between some options a **second time**. Except that this time, only the options which were previously manipulated were asked. This way, a simple comparison of the newer answer with the previous answer would suffice. For example, if the participant now chooses for the second face instead of the first one, it can be stated that their opinion has, in fact, changed after the manipulation. Table 4.3 shows the different rates of opinion shifts for each image type and for all questions combined. It may be concluded that about one in five manipulations result in a change of opinion of the question at hand.

Image type	Rate of changed opinions [%]
Landscapes	26.78
Faces	19.32
Polished faces	15.33
All	21.09

Table 4.3: Rate of changed opinions for each image type

However, these results might be deceiving. As discussed in subsection 1.3.1, choice blindness may be attributed to either *choice-error* or *choice-change*, where a shift in opinion would imply choice-change has indeed occurred. In order to analyze this premise more accurately, it is necessary to compare detected and undetected questions for opinion change. It would be expected that images of which the manipulation has been detected result in a lower opinion change rate. Table 4.4 confirms this hypothesis: a staggering one in three manipulations led to opinion change if they were undetected. However, 6% of detected manipulations resulted in a change of opinion, which is still a relatively high rate, considering this comes from participants who had noticed the manipulation for this pair of images. This rather high rate of opinion change for undetected manipulations could implicate that choice blindness might indeed be attributed to choice-change.

	Rate of changed opinions [%]	
Image type	Undetected	Detected
Landscapes	37.73	7.87
Faces	31.4	7.02
Polished faces	24.34	3.44
All	32.05	6.16

Table 4.4: Rate of changed opinions for each image type, depending on detection

Another interesting finding is that opinion change rates for landscapes are distinctly higher than for faces. This implies that people could have stronger opinions about which face they prefer. As implied by Bortolotti and Sullivan-Bissett [4], people identify themselves strongly with their choices. One possible explanation might thus be that preferring a certain face might add more value to a person's identity and/or personality than a landscape might. However, polished faces seem to result in an even lower opinion change rate. This goes against what would be expected from previous research on face recognition by humans: "...recognition performance with colour images is significantly better than with gray-scale images...", Sinha [9]. One feasible clarification for this seemingly illogical result could be that humans more quickly tend to search for special characteristics in grayscale images, because it is harder to make a choice based on the gist of the image. For coloured images, the brain is significantly better (and likely faster) at recognizing known faces (Sinha [9]). While this still seems to contradict the results shown above, a factor that needs to be taken into account here is the relatively short deliberation time of six seconds each participant receives to make a choice. It might be so that a participant glosses more quickly over the more detailed features of a coloured face, simply because they have already made up their mind as to which face they prefer. Thus, in the case of a grayscale face, they might search more thoroughly for more identifying features in the face, making recognizing these faces a second time a simpler task.

4.2.4 To what degree does option (dis)similarity affect confidence levels of participants?

For this subject, each participant's time used per question is required. As mentioned in Chapter 3.2, PsyToolkit automatically measures this time accurately in milliseconds (Kim et al. [38]). The correlation between this measured time and option (dis)similarity is investigated. As a first insight, a correlation matrix is created for all manipulated questions (see Figure 4.9). It immediately becomes apparent that as similarities increase, so does the time used by participants. This could suggest that participants do indeed act less confident when faced with similar image pairs as opposed to lesser similar image pairs.

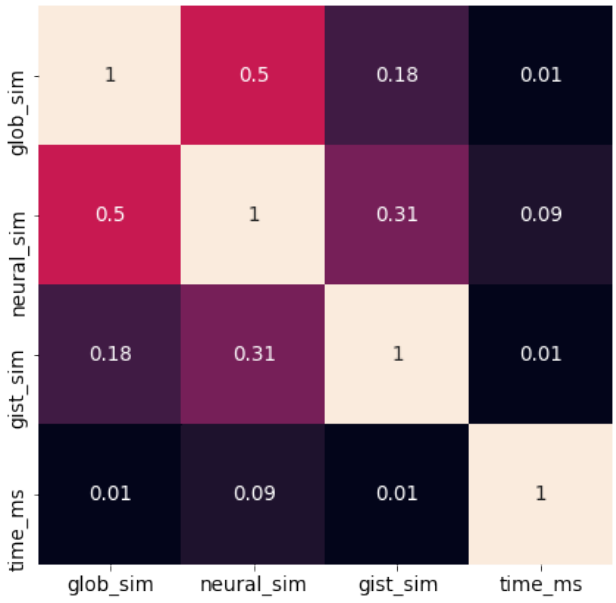
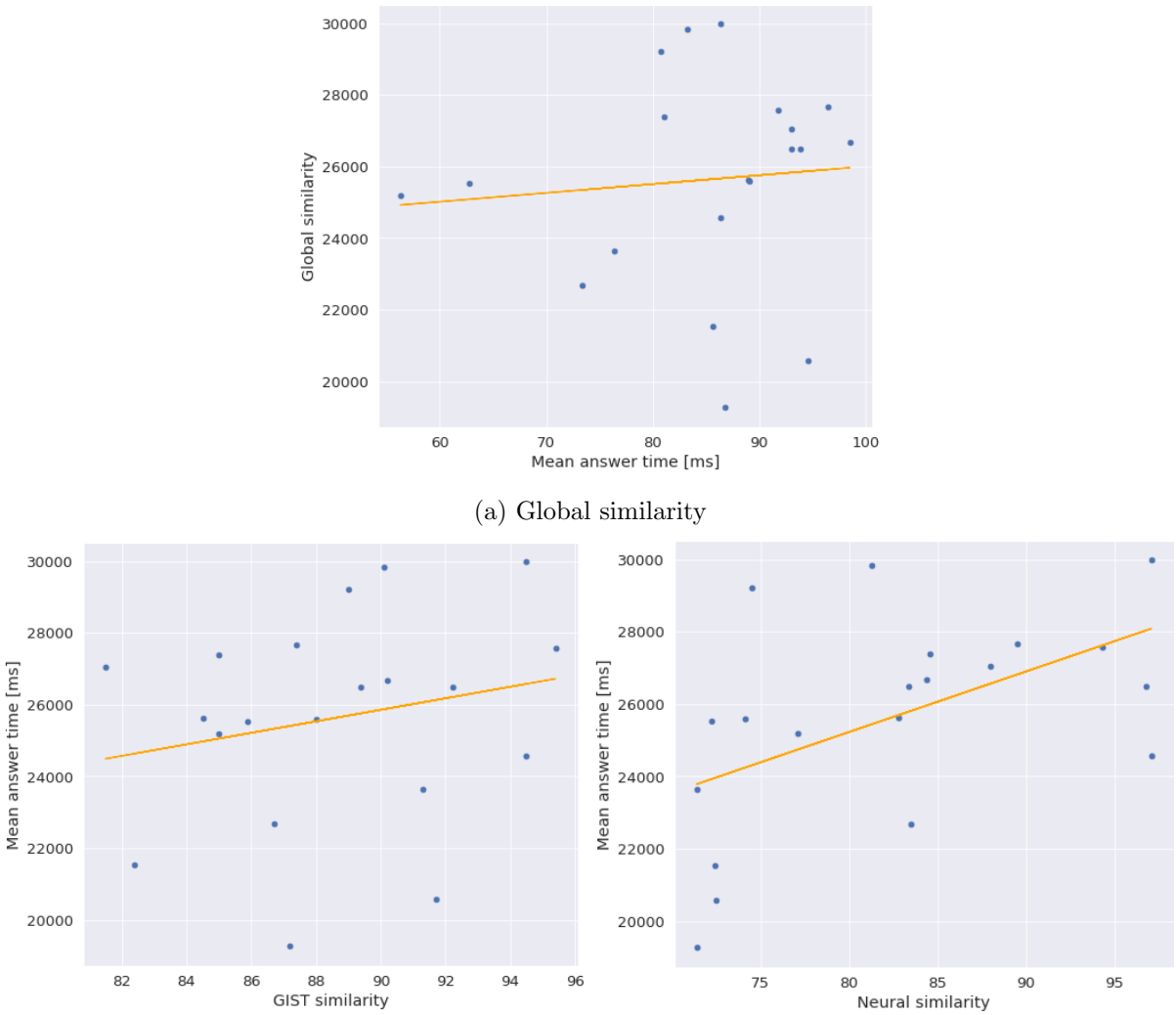


Figure 4.9: Correlation matrix of similarity measures and measured time per question.

To further investigate this notion, time is plotted against each of the manipulated questions' similarities. As expected, neural net and global similarity seem to have a linear relationship with the time taken to answer the question (as can be seen in Subfigures 4.10c and 4.10b). This further reinforces the hypothesis that as images become more similar, participants might doubt their answers more when a manipulation occurs.



(a) Global similarity

(b) GIST similarity

(c) Neural net similarity

Figure 4.10: Plotting of time in milliseconds against similarity rates.

4.2.5 How do different types of options compare when it comes to the chance of choice blindness occurring?

This question is partly answered in subsection 4.2.2. In this subsection, a more general comparison is made between the three used types of images. Firstly, a bar plot is made to analyze the means of the detection percentages of all manipulated questions for each type (Figure 4.11). The black lines indicate the standard deviations.

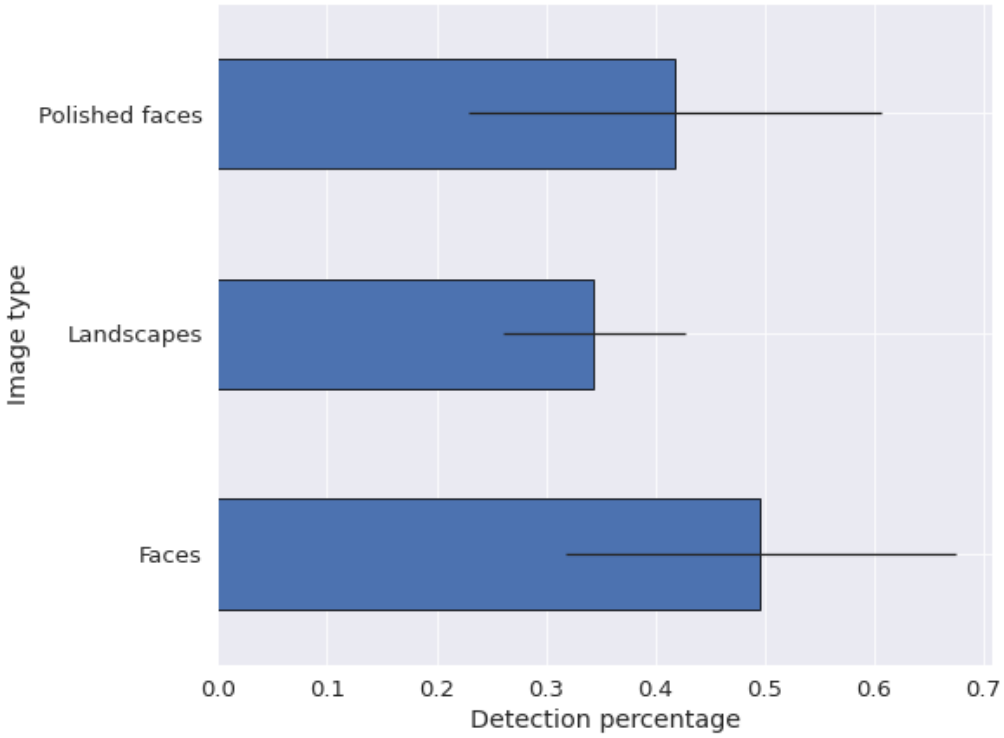


Figure 4.11: Detection percentages for each image type.

It immediately becomes evident that images of type landscape seem to result in a significantly lower detection percentage than both other types. Also, the deviation of the detection percentages seems to be relatively low, indicating a more steady and predictable detection rate for landscapes than for faces and polished faces. This result supports the conclusions drawn from subsection 4.2.3, where images of type landscape showed a significantly higher rate of opinion alteration, indicating that opinions on landscapes might have a smaller impact on a person's self-identity than opinions on faces, resulting in a more loose attitude towards them. However, now, face images turn out to result in the highest mean detection rate (about 50% as opposed to the landscape types' 33%). As previously discussed, for coloured images, the brain is significantly better (and likely faster) at recognizing known faces. This directly supports the results shown, as coloured face manipulations were detected more than grayscaled faces. Next, for each image type, two bars are plotted against the mean of all three similarity measures: one bar for detected images, and one for undetected images. This graph is showcased in Figure 4.12. Firstly, from this graph can be concluded that, for this study, undetected manipulations indeed had a higher mean similarity than detected manipulations. Also, for polished faces, this difference seems to be the highest. However, this difference could simply be attributed to the higher variance in similarity rates seen in polished face image pairs.

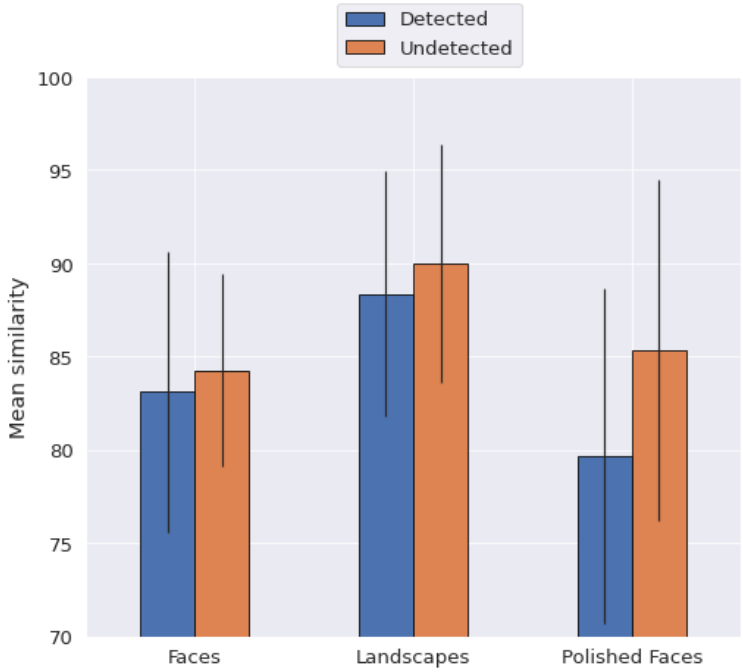


Figure 4.12: Mean across all similarity measures bar plotted for (un)detected manipulations for each type.

4.2.6 How do different types of image properties or criteria influence the chances of choice blindness occurring?

For this final research question, the correlation between choice blindness chances and multiple image properties is analyzed. These properties, as introduced in Chapter 2, include colourfulness, contrast, sharpness, and busyness. Figure 4.13 shows the correlation matrix for these properties. Please note that these properties are calculated based on the difference or **distance** between the two images of a manipulation, as opposed to the **similarity** between the images. The results imply that colourfulness, contrast and busyness difference all correlate relatively strongly with the detection percentage, while sharpness does not seem to have such a strong correlation. More precisely, both colourfulness and busyness difference inversely correlate with the detection percentage relatively strongly. In other words, as these properties differ more between images, the detection percentage decreases. Also, when inspecting the correlation between these properties and the similarity measures, a direct correlation is found, indicating that as the difference in these properties goes up, the calculated similarity also rises, which is counter-intuitive. The busyness result could be attributed to the imperfectness of the busyness metric, which simply searches for bounded contours in the thresholded images and counts them. One possible explanation for the colourfulness result is derived from the scatter plot on Figure 4.14. Firstly, as polished faces are grayscaled, their colourfulness difference will always yield zero, as both images in a pair have no colourfulness. Consequently, these results are not shown on the

plot. Secondly, it becomes clear that landscapes are more likely to yield a high difference in colourfulness (which is rather logical, because faces do not usually differ much in terms of colourfulness). Combining this with the fact that for this study, landscape image pairs resulted in a significantly lower detection rate, the inverse relationship becomes more clear: landscape pairs are most likely influencing the results overall. However, when looking at image types separately, no relationship seems to be present between colourfulness and detection rates.

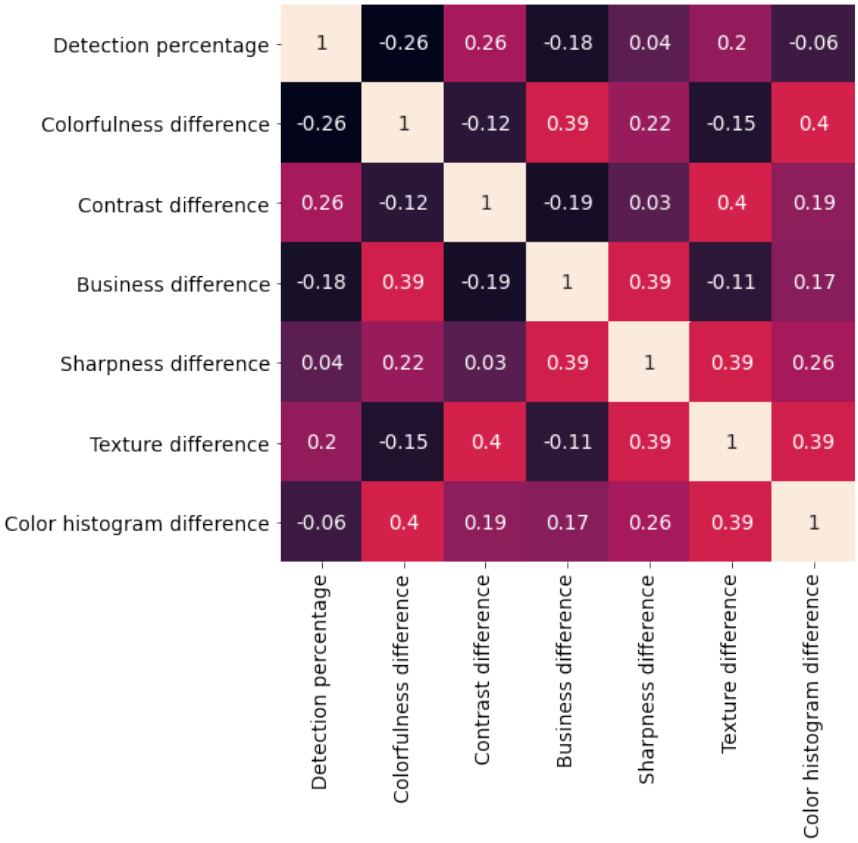


Figure 4.13: Correlation matrix of all properties and the detection percentage.

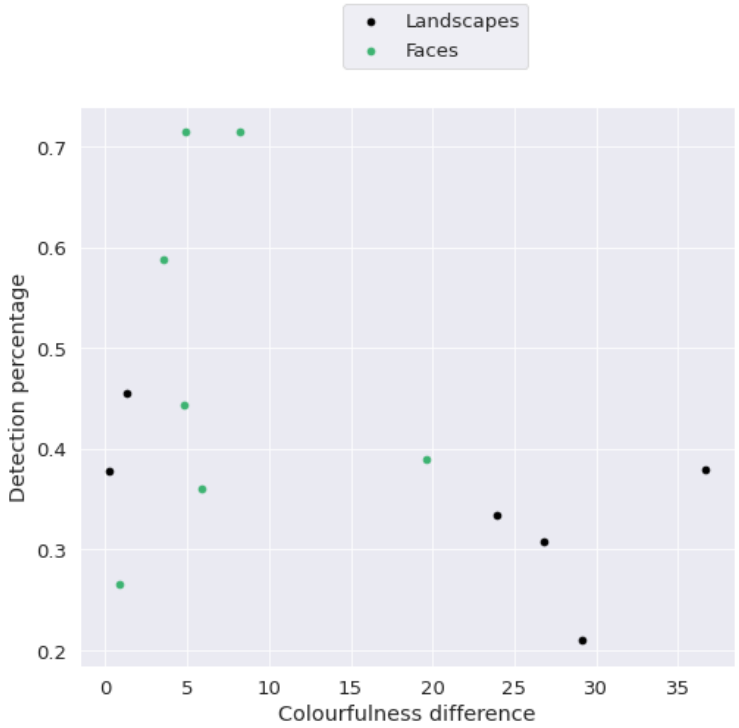


Figure 4.14: Scatter plot of colourfulness difference with detection percentage.

The contrast difference, however, holds a relatively strong direct correlation with the detection percentage. This indicates that as the contrast between two images differs more, manipulation detection chances rise. This could suggest that contrast is a rather significant factor in regard to choice blindness. Out of 4053 argumentations, 57 mentioned the word 'contrast', further strengthening this notion. Although none of these argumentations were detection cases, this still implies that contrast might have an impact on the decision-making process of the participants. The same can be said about the difference in texture, which also resulted in a direct correlation with the detection rate. Figure 4.15 shows a scatter plot of this texture difference with detection percentage, grouped by image type. While it is clear here that this linear relationship could be more attributed to chance than anything else, one interesting thing becomes clear: for polished faces, texture difference seems to correlate more with detection chances than for coloured faces. This joins nicely with the hypothesis that participants might tend to focus more on features such as textures in the absence of colour features.

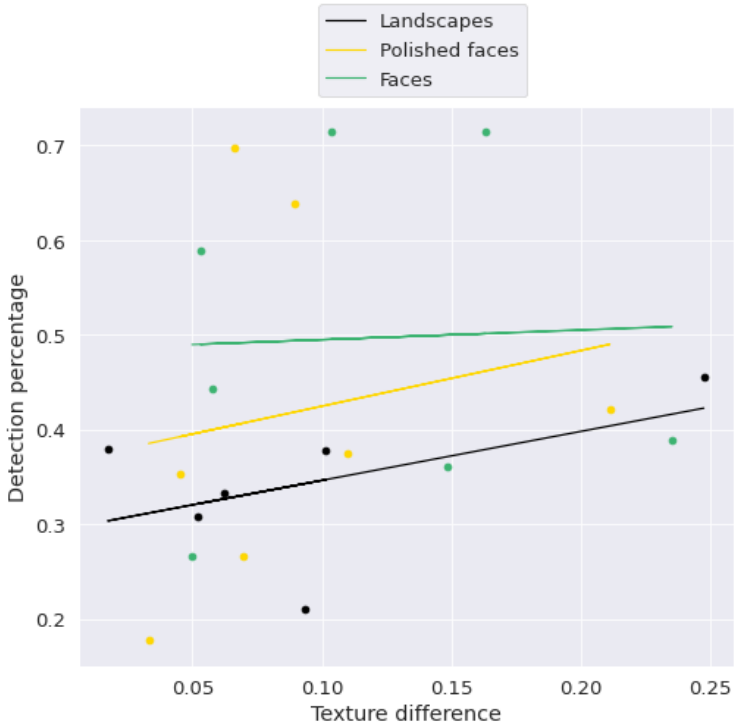
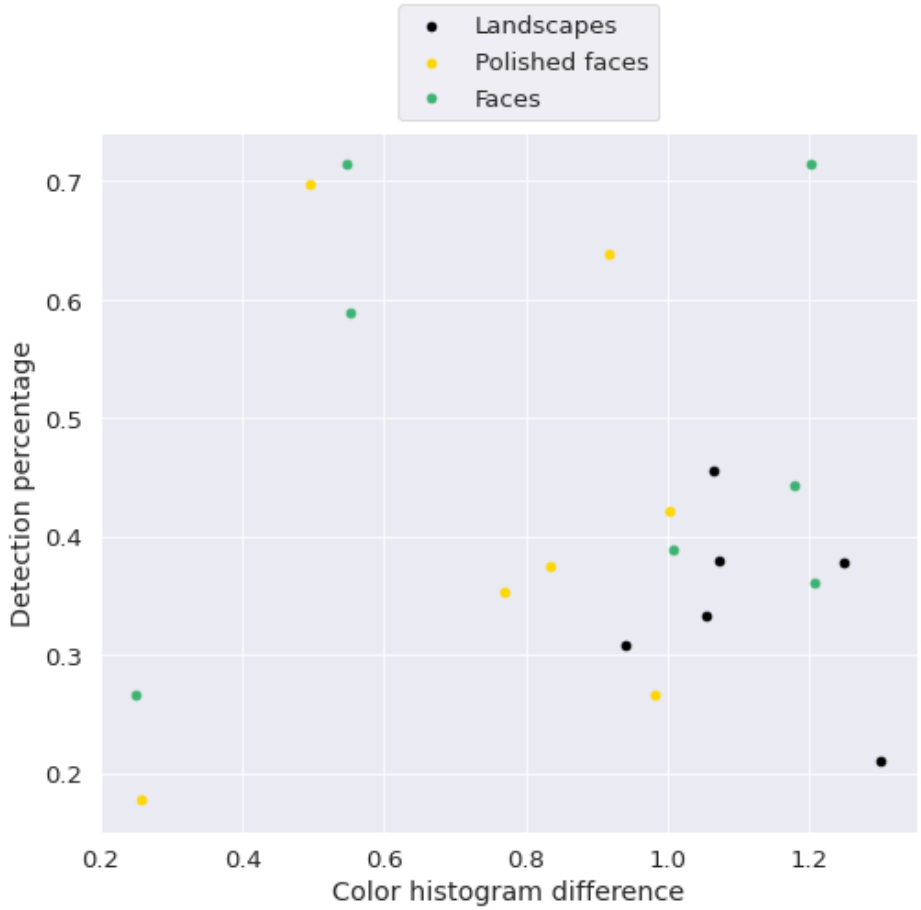
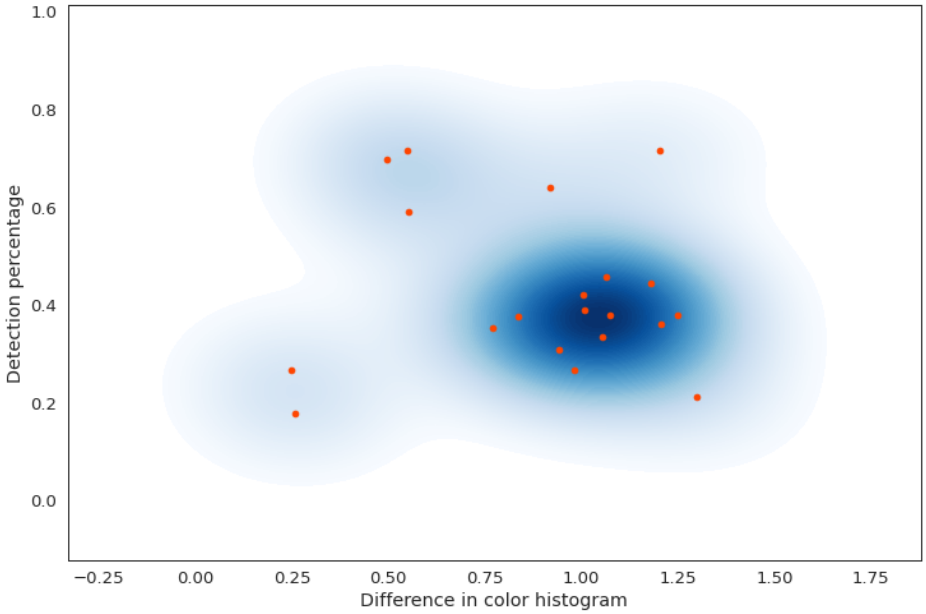


Figure 4.15: Scatter plot of texture difference with detection percentage.

While no clear relationship was found between colour histogram differences and detection rates, scatter plotting the data shows an interesting pattern. Figure 4.16 shows the colour histogram difference scatter plotted against detection percentages for all manipulated questions. Figure 4.16a shows that for faces and polished faces, outliers seem to predict detection changes excellently, with low colour histogram differences usually resulting in low detection rates, and vice versa. All none-outlier question pairs seem to cluster together in the $[0.8 - 1.2]$ range. This clustering is visualized in 4.16b, on which it is clear that a $[0.8 - 1.2]$ colour histogram difference most likely results in a detection percentage ranging from 0.3 to 0.5.



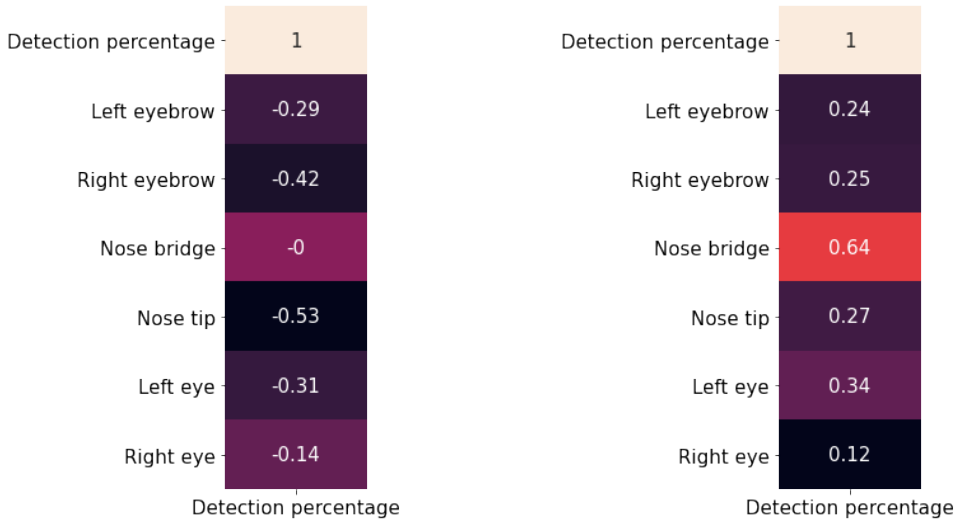
(a) Plotted per image type.



(b) Plotted with clustering.

Figure 4.16: Colour histogram difference scatter plotted against detection percentage for all manipulated questions.

Lastly, landmarks shape dissimilarities in faces get analyzed for correlations with the detection rate as well. Figure 4.17 shows extracted columns from (polished) face landmark dissimilarity correlation matrices. It becomes clear immediately that these landmark shapes correlate with detection percentages very differently depending on them being polished or not. More precisely, in unpolished faces, each landmark shape dissimilarity metric has an inverse relationship with the detection rates, while for polished faces, the relationship is direct. This could again be attributed to the hypothesis that participants look more into details like texture and facial features (such as the eyebrows) when more obvious features such as colour difference are not present. As such, each of these landmark dissimilarities bears a strong direct correlation with the chances of manipulation detection in grayscaled face images The most notable value here is the correlation with the difference in nose bridges. Please note that the nose bridge is not the outline of the nose but rather a single line indicating the position, length and angle of the bridge. Another notable remark is that the left eye shape difference seems to correlate more with detection chances than the right eye shape difference, suggesting that people might tend to subconsciously focus more on the right eye in the image. However, Sinha [9] suggests that the eyebrows should be among the most important facial features humans turn to for facial recognition. In this study, for both faces and polished faces, this is not the case, as eyebrow differences hold the lowest correlation overall. According to Ellis et al. [10], however, the central part of the face holds the most importance: "the eyes and nose attract most visual attention". This notion seems to be strengthened in this study.



(a) Face landmarks.

(b) Polished face landmarks.

Figure 4.17: Extracted columns from (polished) face landmark correlation matrices.

4.2.6.1 Multivariate linear regression analysis

All of the above-mentioned properties or features can now be used to create a linear regression model, as already introduced in subsection 4.2.2. However, where the previous models contained only one data variable for detection prediction, here, multiple variables will be used. This concept is called **multivariate linear regression**. Because of the increase of variables, the R-squared score is expected to rise, as more variables should be able to explain more of the variance. However, only a subset of the properties is used, as some might not explain the variance correctly. As was discussed above, busyness, colour histogram and colourfulness differences are not likely to be good predictors for detection rates, so these properties are dropped from this analysis. More precisely, two regression functions are created: one general function for all image types, and one for both face type images. For the general function, all relevant features will be used. For the face type function, the face landmark differences will also be included.
When constructing a multivariate function, it is important to analyze the possibility that the used variables might be correlated **with each other**. This concept is called **multicollinearity**, and might temper with the effectiveness of the model. Consider the simplest case where Y is regressed against X and Z and where X and Z are highly positively correlated. Then the effect of X on Y is hard to distinguish from the effect of Z on Y because any increase in X tends to be associated with an increase in Z. In the case of this thesis, this problem is almost sure to arise, because most measures used are bound to be correlated in some way (e.g. if the images differ in contrast, their similarity will likely drop). To solve this issue, a Principle Component Analysis (PCA) could be performed on the data set. A PCA finds a list of the principal axes in the data and uses those axes to describe the data set. By determining the most important axes, it is possible to zero out the lesser important ones and, in turn, reduce the collinearity of the variables. However, importantly, PCA also causes a difference in the meaning of the new variables, resulting in a significantly harder interpretation, which is of high importance for this model. For this reason, the function will be defined using the untouched variables. The general function is then defined using the following variables: global similarity, neural net similarity, GIST similarity, contrast difference, sharpness difference and texture difference. However, as discussed in 4.2.1, the question number also correlates significantly with the detection rates. For this reason, another general function will be designed where this variable is included. Equation 4.6 shows this relationship, where Y represents the expected detection percentage, and each X represents the image type (X_{it}), global similarity (X_{gl}), neural net similarity (X_{nn}), GIST similarity (X_{gi}), contrast difference (X_{cd}), sharpness difference (X_{sd}) and texture difference (X_{td}) respectively.

$$\begin{aligned} y(x) = 2.2657 - 0.0520\ x_{it} - 0.0020\ x_{gl} - 0.0139\ x_{nn} - 0.0049\ x_{gi} - 0.0056\ x_{cd} \\ -0.00215\ x_{sd} + 0.1656\ x_{td} \end{aligned} \tag{4.6}$$

This model yields an R-squared score of 0.652, which is relatively high. When the question number is included, this score rises to 0.685 and this results in equation 4.7. Here, again, Y

represents the expected detection percentage, and X_{qn} represents the question number.

$$y(x) = 2.1704 - 0.0522\ x_{it} + 0.0068\ x_{qn} - 0.0018\ x_{gl} - 0.0109\ x_{nn} - 0.0078\ x_{gi} \\ -0.0037\ x_{cd} - 0.0044\ x_{sd} + 0.0098\ x_{td} \tag{4.7}$$

A problem with the R-squared metric is that there is no penalty for adding additional input features. A model with more input features might result in a higher R2 value simply by chance instead of due to a statistical significance of its variables. To accommodate for this, it is possible to calculate an **adjusted R-squared score**, which penalties the score for each added variable. Adding too many variables with no statistical significance in predicting the target variable (detection chances) could lead to an over-fitting of the model, causing a lower accuracy (Miles [42]). While a subset of variables was chosen by hand based on their apparent correlation with the target variable, this adjusted R-squared score might still turn out lower than an R-squared score having neural similarity as the sole variable. The adjusted R-squared score is calculated as follows:

$$R^2_{adj} = 1 - (1 - R^2)\frac{n-1}{n-p-1} \tag{4.8}$$

Where p is the total number of explanatory variables in the model (not including the constant term), and n is the sample size. After inspecting this adjusted R-squared score for the created models, it becomes clear that the model might be over-fitted. The adjusted R-squared score was calculated to be 0.457. To conclude whether this model is better than models containing fewer variables, models were fitted for each combination of variables, while the adjusted R-squared score was analyzed for each combination. Concluding from this analysis, the combination of image type, question number, neural similarity, contrast difference and (adding a slight increase) texture difference resulted in the highest adjusted R-squared score (0.5590 to be precise, which is higher than the neural net's R-squared score of 0.5482). In other words, all other variables sparked a rise in the R-squared score but lowered the adjusted R-squared score due to them being not statistically significant enough for the target variable. Using these five variables, a new model is fitted, of which the function can be seen in equation 4.9.

$$y(x) = 1.4945 - 0.0410\ x_{it} + 0.0037\ x_{qn} - 0.0134\ x_{nn} - 0.0030\ x_{cd} + 0.1881\ x_{td} \tag{4.9}$$

Lastly, the same process is repeated for images of type polished faces. This image type was chosen due to the significant correlations found between grayscale face landmarks and detection chances. For the grayscaled images, the adjusted R-squared score resulted in a staggering 0.727 when combining the variables nose bridge difference (X_{nb}), neural similarity, global similarity, and sharpness difference. The following function can then be defined:

$$y(x) = 0.5720 + 3.1767\ x_{nb} + 0.0027\ x_{nn} - 0.0065\ x_{gl} + 0.0190\ x_{sd} \tag{4.10}$$

5

Conclusions and discussions

In this final Chapter, firstly, the conclusions of this thesis are summarized. Then, reflections and future work are discussed, followed by some final words.

5.1 Conclusions

This thesis sets out to answer several questions surrounding the choice blindness paradigm. To answer these questions as sufficiently as possible, a study was built up using the PsyToolkit framework. This study was performed by 173 hired participants, after which the collected data was analyzed using the Python scripting language.

The most prominent of these questions focuses on computational features in images and their influence on the chances of choice blindness occurring: *"Which computational features of an image correlate with choice blindness?"*. To answer this question adequately, three similarity measures were defined and used: global similarity, neural net similarity and GIST similarity. It was found that all three defined similarity measures inversely correlate with chances of choice blindness occurring, with neural net similarity showing the strongest correlation. More specifically, data suggests that similarity as defined by the GIST algorithm is the strongest predictor for choice blindness in landscape images ($R^2 = 0.6157$), while the neural network best predicts choice blindness for faces ($R^2 = 0.8848$) and polished faces ($R^2 = 0.5829$). Overall, the neural network seems to be the best predictor for all three image types ($R^2 = 0.5482$). It can be stated that for this study, the neural net similarity range $[83 - 88]$ might serve as a cut-off

range for manipulation detection, while a similarity rate of 79 might function as a cut-off rate. Furthermore, introspection in the form of preference change was investigated under the research question *"How much could choice blindness result in changes in personal preference?"*. In total, about one in five manipulated questions resulted in a change of preference when the same question was repeated to the same participant. Of all undetected manipulations, 32.05% resulted in a change of preference. Interestingly, 6.16% of detected manipulation also resulted in a change of preference. For all questions, landscape type images result in the highest preference change rate (26.78% to be precise), suggesting that landscape images might be the most loosely bound with a person's sense of identity. These relatively high percentages strengthen the hypothesis that choice blindness might be a result of choice-change as opposed to choice error.
A third research question reads "To what degree does image (dis)similarity affect confidence levels of participants?". As such, introspection identified by confidence levels was measured by the amount of time participants used to argue their answers. It was found that participants use more time to argue their choice as similarities increase, suggesting that participants might indeed act less confident when faced with similar image pairs as opposed to lesser similar image pairs. This in turn strengthens the notion, as proposed by Rieznik et al. [11], of a possible existence of a neural mechanism unconsciously monitoring our own thoughts, possibly indicating some sort of unconscious detection of self-deception.
In addition, the following question was defined: *"How do different types of options and criteria compare when it comes to the chance of choice blindness occurring?"*. For this purpose, three separate image data sets were retrieved and used in the online study. Namely, landscapes, faces, and "polished" faces were set up. Results show that landscape image manipulations got detected significantly less than the other two types, with a detection percentage of 33%, as opposed to 41% for polished faces and 50% for unpolished faces, indicating participants are most easily manipulated when it comes to landscapes, and least easily for coloured faces. These results might lead to the notion that humans are more liable to choice blindness when presented with images or ideas that have a weaker link with their self-identity.
Lastly, separate image properties or features and their influence on choice blindness chances were investigated to attempt answering the following question: *"How do different types of image properties or criteria influence the chances of choice blindness occurring?"*. For this purpose, colour, texture, colourfulness, sharpness, contrast, busyness and shape features were all defined and extracted. Results imply that texture and contrast most strongly correlate with detection chances. Texture distance correlates highly with manipulation detection in polished face images as opposed to coloured face images, suggesting humans tend to focus more on texture in polished faces due to the lack of colour features. Results also revealed a formed cluster in colour histogram plots, resulting in a defined range of $[0.8 - 1.2]$ for colour histogram difference in which all questions result in a detection rate ranging from 0.3 to 0.5. For facial landmarks, it was found that all facial feature shape distances in pairs inversely correlate with detection rates in coloured faces, while they directly correlate with detection rates for

grayscaled faces. Out of all landmarks, the nose bridge and left eye shape distances correlate the strongest with detection rates, supporting the notion that the center of the face might be of most importance when it comes to face recognition and thus choice blindness. Lastly, a multivariate linear regression analysis was performed to define a function that might serve as a detection rate predictor. A function using image type, question number, neural net similarity, contrast difference and texture difference results in the highest adjusted R-squared score of 0.5482 for detection rates as the target variable. A function was also defined for polished face images including the nose bridge shape distance variable, resulting in a staggering adjusted R-squared score of 0.727.
In general, 59.59% of manipulated questions remained undetected. Detection rates also correlate with the question number, or the order they appear in the study, possibly indicating that participants are quicker to raise suspicion as more manipulations have already happened.

These results show that choice blindness in images might be more predictable and preferences might be more easily manipulated than initially thought, when enough variables are available. When taking into account the importance of choice and preference in the daily lives of people, this notion is certainly not to be ignored, and further research is definitely of interest.

5.2 Reflections

The process of building an online psychological study around a paradigm of which the participants need to be oblivious of the premise behind it, while collecting the necessary data to attempt answering all research questions, turned out to be more challenging than initially thought. Nonetheless, this research can be called a success, as all main goals of the working plan have been, at the least, reached. This Section covers what could be improved upon in future work and lists chosen libraries, limitations, difficulties, ethical reflections, etc.

5.2.1 Future work

For future research, some points could be improved upon for more accurate results. One limitation of the performed research is the relatively small amount of data points, or rather, manipulated image pairs. Only a handful amount of questions (six per participant in this study) can be manipulated before participants would simply notice each subsequent manipulation due to heightened suspicions. However, the image pairs could still be diversified more. In other words, for each image type, more image pairs could be used instead of repeating the same pairs more times. This, of course, increases the variance for each separate image pair. In this study, one pair would appear multiple times, leading to more accurate mean detection rates. But when more pairs would be used, pairs would appear less in total, resulting in lesser accurate mean detection rates. To solve both problems, more pairs could be used, and pairs could be used multiple times, but the number of participants would need to increase.

Another limitation that was discussed in the data analysis, is that manipulation detection might be partly correlated with the question number. As such, the order of manipulated pairs shown to participants could be of importance. Because of the chosen framework in this research, namely PsyToolkit, randomization of specific questions is not possible (only an entire set of questions can be randomized, not each question and its argumentation part as one. In other words, when question randomization would be used, everything would be scrambled, resulting in questions and their argumentation part ending up separated). To accommodate for this, in this study, it was made sure that each image pair appears at least once as the first manipulated question. However, another framework that is capable of randomizing this order each time a participant starts the study would be better in this case.
Adding to this, another possible improvement would be to add certain weights to the manipulated questions based on their order of appearance in the study. As such, manipulations appearing later in the study could receive a weight that lowers the observed detection rate, and vice versa.
One of the main results of this study is the set of models that were fitted to predict the target variable of manipulation detection. After a limited search, these models were chosen to be linear models, also called the inductive bias. However, it could be that other, more complicated models fit the data better. More models could be researched to define a better predictive function.
As for the linear model itself, this kind of model is quite sensitive to outliers [45]. To refine it more, outlier detection could be performed and said outliers could be made to be ignored by the model, or receive lower weights.
One last improvement for future research lies in the similarity algorithms. While one of the main research topics of this thesis, the self-defined global similarity metric was far from perfect. More features could be extracted, and a better method could be found for the concatenation of these features.

5.2.2 Libraries and choice of technology

This thesis makes use of multiple Python libraries for the code implementations. This subsection serves as an inventory of the most prominent used libraries, frameworks and programs, and explains why some of them were used over others. Table 5.1 lists this inventory.
The Python library OpenCV was used for most computer vision functions because it is one of the largest, best supported, and widely documented computer vision libraries [46]. Data was collected in CSV files and also kept in CSV files in memory. For this reason, Pandas was chosen as the data storage and analysis framework, because it works with Data Frames, which are trivially converted to and from CSV files. For the same reason, hardcoded, pre-known question data was stored in Excel spreadsheets, which provide helpful editing functions and are also trivially converted to and from CSV files. The installation of the similarity algorithm libraries resulted in many difficulties and unnecessary time-loss. The GIST algorithm implementation

library, LMgist [1], for example, did not provide a step-by-step tutorial for installation on Windows systems. Because of this, many dependencies had to be installed manually, causing problems of their own. As for the Apple TuriCreate library, this library was entirely unavailable on Windows. This limitation led to the decision to use the Windows Subsystem for Linux[2] (WSL). Using WSL, it is possible to use programs and libraries available on Linux on a Windows system. To edit and run code easily and dynamically on Windows through WSL, Jupyter Notebook[3] was used on WSL [47].
Similarity rates were stored in JSON files after extraction due to their easy conversion from and to Python dictionaries. However, in retrospect, it would have been more efficient and less time-consuming to store these in Pandas Data Frames as well, making it easier to merge them into the pre-known data (which happened by hand in this case).

Libraries	
OpenCV	Computer vision functions
Pandas	Data organisation and visualization
Seaborn	Data visualization
LMgist	GIST algorithm implementation
Apple TuriCreate	Neural network landscape image similarity model
face-recognition	Neural network face image similarity model
Frameworks	
PsyToolkit	Study implementation and data collection
Programs	
WSL	Access to Linux and MacOS libraries on Windows
Jupyter Notebook	Combine software code and computational output, dynamic outputs, etc.
Microsoft Excel	Editing and storage of question data structure

Table 5.1: Inventory of most prominently used libraries, frameworks and programs

[1]Documentation available at: https://github.com/Kalafinaian/python-img_gist_feature
[2]Website: https://ubuntu.com/wsl
[3]Website: https://jupyter.org/

5.2.3 Sustainable Development Goals

Many of the Sustainable Development Goals (SDG) goals can be connected to both the science and profession of psychology - such as ending poverty (SDG 1), working towards zero hunger (SDG 2), improving good health and well-being (SDG 3), ensuring quality education (SDG 4), improving gender equality (SDG 5), providing decent work opportunities and economic growth (SDG 8), reducing inequality (SDG 9), developing sustainable cities and communities (SDG 11), building peace (SDG 16), and also strengthening partnerships to achieve the goals (SDG 17) (Eloff [48]). Understanding the human thought process and behaviour is an important aspect of science of which the importance has not yet been made substantively visible and prominent in scientific research and academic literature. A step towards a more deep understanding of this aspect is a step towards solving each of the aforementioned goals. Additionally, better comprehension of the human thought and decision-making process is an important step into further improving human-AI interactions. This, in turn, will help in reaching multiple SDGs, such as quality education, good health and well-being, innovation, sustainable cities and communities, and decent work and economic growth. These goals will be partly supported as AI technologies are used to push boundaries in almost all of these fields [49]. Next to this, the goal of responsible consumption and production (SDG 12) is partly tackled by better understanding how humans might be subtly manipulated into certain buying tendencies or other harmful practices without their own knowing.

5.2.4 Ethical and societal impact

This thesis made use of human participants to acquire data surrounding the choice blindness paradigm in a psychological study. **Informed consent was obtained from all individual participants included in the study**. Apart from IP addresses, no identifying information was collected from the participants.

The concept of choice blindness implies the possibility of molding people's preferences or making them believe they prefer one thing when, before manipulations occurred, they preferred the other. This paradigm could be used for better or worse. Further research of choice blindness could lead the way to a better understanding of the inner workings of the human brain. Among others, decision-making processes and self-identification through preferences could become better understood through choice blindness research. To give one example, for artificial intelligence agents, it is desirable to be capable of cooperating effectively with humans. However, humans are known to not preserve their rationality at all times and suffer different biases that affect their decision processes. This makes inferring their behavior and collaborating with them a daunting challenge for AI practitioners. A better understanding of said decision process might help in overcoming this challenge. However, the paradigm can most certainly be applied in a malicious manner. To give an example, take the industry of online

shopping. Webshops with malicious intend could switch people's chosen items out with highly similar, higher-priced items after navigating to the final payment screen. Practices like this already exist to a certain extend, where only the price is changed to a higher price once the person navigates to the final payment screen. Another situation where choice blindness might be used malignantly, is political campaigns. As discussed by Hall et al. [12], a psychological experiment on choice blindness resulted in 48% of participants considering a left-right coalition switch after manipulations of questions on political issues. As such, the phenomenon could be used to switch people's political preferences weeks before an election. The prediction models created in this thesis, although imperfect, should never be used for such intends.

Bibliography

[1] S. K. E, "Convolutional Neural Network — Deep Learning." [Online]. Available: https://developersbreach.com/convolution-neural-network-deep-learning/

[2] H. Felix, S. Yury, K. M. Ulf, J. K. Pascal, and E. H. Benjamin, "lab.js: a free, open, online study builder," 2019.

[3] P. Johansson, L. Hall, S. Sikstro¨m, B. Ta¨rning, and A. Lind, "How something can be said about telling more than we can know: On choice blindness and introspection," *Consciousness and Cognition*, vol. 15, pp. 673–692, 2006.

[4] L. Bortolotti and E. Sullivan-Bissett, "Is choice blindness a case of self-ignorance?" *Synthese*, Sep. 2019.

[5] F. Taya, S. Gupta, I. Farber, and O. Mullette-Gillman, "Manipulation detection and preference alterations in a choice blindness paradigm," *PLoS ONE*, vol. 9, 2014.

[6] L. Hall, P. Johansson, B. Tärning, S. Sikström, and T. Deutgen, "Magic at the marketplace: Choice blindness for the taste of jam and the smell of tea," *Cognition*, vol. 117, no. 1, pp. 54–61, 2010. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0010027710001381

[7] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *International journal of computer vision*, vol. 42, no. 3, pp. 145–175, 2001.

[8] B. E. Rogowitz, T. Frese, J. R. Smith, C. A. Bouman, and E. B. Kalin, "Perceptual image similarity experiments," in *Human Vision and Electronic Imaging III*, B. E. Rogowitz and T. N. Pappas, Eds., vol. 3299, International Society for Optics and Photonics. SPIE, 1998, pp. 576 – 590. [Online]. Available: https://doi.org/10.1117/12.320148

[9] P. Sinha, "Face recognition by humans: Nineteen results all computer vision researchers should know about," in *Proceedings of the IEEE*, 2006, pp. 1948–1962.

[10] H. D. Ellis, J. W. Shepherd, and G. M. Davies, "Identification of familiar and unfamiliar faces from internal and external features: Some implications for theories of

face recognition," *Perception*, vol. 8, no. 4, pp. 431–439, 1979, pMID: 503774. [Online]. Available: https://doi.org/10.1068/p080431

[11] A. Rieznik, L. Moscovich, A. Frieiro, J. Figini, R. Catalano, J. M. Garrido, F. Álvarez Heduan, M. Sigman, and P. A. Gonzalez, "A massive experiment on choice blindness in political decisions: Confidence, confabulation, and unconscious detection of self-deception," *PloS one*, vol. 12, no. 2, pp. e0 171 108–e0 171 108, 2017.

[12] L. Hall, T. Strandberg, P. Pärnamets, A. Lind, B. Tärning, and P. Johansson, "How the polls can be both spot on and dead wrong: using choice blindness to shift political attitudes and voter intentions," *PloS one*, vol. 8, no. 4, pp. e60 554–e60 554, Apr 2013, 23593244[pmid]. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/23593244

[13] C. Steenfeldt-Kristensen and I. M. Thornton, "Haptic choice blindness," *i-Perception*, vol. 4, no. 3, pp. 207–210, 2013, pMID: 23799197. [Online]. Available: https://doi.org/10.1068/i0581sas

[14] W. H. Gomaa, A. A. Fahmy *et al.*, "A survey of text similarity approaches," *International Journal of Computer Applications*, vol. 68, no. 13, pp. 13–18, 2013.

[15] B. Marr, "What is the difference between low image features and semantic image features?" 2015. [Online]. Available: https://www.researchgate.net/post/What-is-the-difference-between-low-image-features-and-semantic-image-features

[16] F. F. Ibarra, O. Kardan, M. R. Hunter, H. P. Kotabe, F. A. C. Meyer, and M. G. Berman, "Image feature types and their predictions of aesthetic preference and naturalness," *Frontiers in Psychology*, vol. 8, p. 632, 2017. [Online]. Available: https://www.frontiersin.org/article/10.3389/fpsyg.2017.00632

[17] M. El-gayar, H. Soliman, and N. meky, "A comparative study of image low level feature extraction algorithms," *Egyptian Informatics Journal*, vol. 14, no. 2, pp. 175–181, 2013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1110866513000248

[18] Y. He, N. Sang, and R. Huang, "Local binary pattern histogram based texton learning for texture classification," in *2011 18th IEEE International Conference on Image Processing*, 2011, pp. 841–844.

[19] M.-K. Hu, "Visual pattern recognition by moment invariants," *IRE Transactions on Information Theory*, vol. 8, no. 2, pp. 179–187, 1962.

[20] E. Karami, S. Prasad, and M. S. Shehata, "Image matching using sift, surf, BRIEF and ORB: performance comparison for distorted images," *CoRR*, vol. abs/1710.02726, 2017. [Online]. Available: http://arxiv.org/abs/1710.02726

[21] E. Palumbo and W. Allasia, "Semantic similarity between images: A novel approach based on a complex network of free word associations," in *Proceedings of the 8th International Conference on Similarity Search and Applications - Volume 9371*, ser. SISAP 2015. Berlin, Heidelberg: Springer-Verlag, 2015, p. 170–175. [Online]. Available: https://doi.org/10.1007/978-3-319-25087-8_16

[22] A. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, "Content-based image retrieval at the end of the early years," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 12, pp. 1349–1380, 2000.

[23] P. Veelaert, "Computer vision for autonomous vehicles."

[24] A. Daly, "Face Recognition - Project description," 2021. [Online]. Available: https://pypi.org/project/face-recognition/

[25] D. Navon, "Forest before trees: The precedence of global features in visual perception," *Cognitive Psychology*, vol. 9, no. 3, pp. 353–383, 1977. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0010028577900123

[26] D. Hasler and S. E. Suesstrunk, "Measuring colorfulness in natural images," in *Human Vision and Electronic Imaging VIII*, B. E. Rogowitz and T. N. Pappas, Eds., vol. 5007, International Society for Optics and Photonics. SPIE, 2003, pp. 87 – 95. [Online]. Available: https://doi.org/10.1117/12.477378

[27] S. M. Aswatha, J. Mukhopadhyay, and P. Bhowmick, "Image denoising by scaled bilateral filtering," in *2011 Third National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics*, 2011, pp. 122–125.

[28] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[29] S. Suzuki and K. be, "Topological structural analysis of digitized binary images by border following," *Computer Vision, Graphics, and Image Processing*, vol. 30, no. 1, pp. 32–46, 1985. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0734189X85900167

[30] E. Peli, "Contrast in complex images," *J. Opt. Soc. Am. A*, vol. 7, no. 10, pp. 2032–2040, Oct 1990. [Online]. Available: http://josaa.osa.org/abstract.cfm?URI=josaa-7-10-2032

[31] A. Rougetet, "Landscape Pictures," 2019. [Online]. Available: https://www.kaggle.com/arnaud58/landscape-pictures

[32] A. Gupta, "Human Faces," 2020. [Online]. Available: https://www.kaggle.com/ashwingupta3012/human-faces/

[33] F. Henninger, "lab.js," 2021. [Online]. Available: https://lab.js.org/

[34] ——, "Get started building studies," 2021. [Online]. Available: https://labjs.readthedocs.io/en/latest/learn/builder/

[35] G. Stoet, "Psytoolkit: a software package for programming psychological experiments using linux," *Behavior research methods*, vol. 42, no. 4, p. 1096—1104, November 2010. [Online]. Available: https://doi.org/10.3758/BRM.42.4.1096

[36] ——, "Psytoolkit: A novel web-based method for running online questionnaires and reaction-time experiments," *Teaching of Psychology*, vol. 44, no. 1, pp. 24–31, 2017. [Online]. Available: https://doi.org/10.1177/0098628316677643

[37] PsyToolkit, "PsyToolkit," 2021. [Online]. Available: https://www.psytoolkit.org/

[38] J. Kim, U. Gabriel, and P. Gygax, "Testing the effectiveness of the internet-based instrument psytoolkit: A comparison between web-based (psytoolkit) and lab-based (e-prime 3.0) measurements of response choice and response time in a complex psycholinguistic task," Sep. 2019.

[39] O. Pathak, "The Best Data Science Libraries in Python." [Online]. Available: https://stackabuse.com/the-best-data-science-libraries-in-python/

[40] A. Sagana, M. Sauerland, and H. Merckelbach, "Warnings to counter choice blindness for identification decisions: Warnings offer an advantage in time but not in rate of detection," *Frontiers in Psychology*, vol. 9, p. 981, 2018. [Online]. Available: https://www.frontiersin.org/article/10.3389/fpsyg.2018.00981

[41] R. G. Carpenter, "Principles and procedures of statistics, with special reference to the biological sciences," *The Eugenics Review*, vol. 52, no. 3, pp. 172–173, Oct 1960, pMC2972823[pmcid]. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2972823/

[42] J. Miles, "R squared, adjusted r squared†," 2005.

[43] "Regression Analysis: How Do I Interpret R-squared and Assess the Goodness-of-Fit?" [Online]. Available: https://blog.minitab.com/en/adventures-in-statistics-2/regression-analysis-how-do-i-interpret-r-squared-and-assess-the-goodness-of-fit

[44] J. Frost, "Regression Analysis: How Do I Interpret R-squared and Assess the Goodness-of-Fit?" [Online]. Available: https://statisticsbyjim.com/regression/interpret-r-squared-regression/

[45] P. Simoens, "Toegepast machinaal leren."

[46] "About." [Online]. Available: https://opencv.org/about/

[47] Jupyter, "About," 2021. [Online]. Available: https://jupyter.org/about

[48] I. Eloff, "Psychology and the sustainable development goals," *Journal of Psychology in Africa*, vol. 30, no. 1, pp. 86–87, 2020. [Online]. Available: https://doi.org/10.1080/14330237.2020.1712810

[49] B. Marr, "What Is The Importance Of Artificial Intelligence (AI)," 2020. [Online]. Available: https://bernardmarr.com/default.asp?contentID=1829

Appendices

Appendix A - Code implementations of features extractions

A-1 - Extraction, calculation and comparison of color histogram

```
# global feature descriptor 1: Color Histogram extraction
bins = 8
def fd_histogram(image):

    # convert the image to HSV color-space
    image_conv = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)

    # compute the color histogram
    hist  = cv2.calcHist([image_conv], [0, 1, 2], None, [bins, bins, bins],
    ↪  [0, 256, 0, 256, 0, 256])

    # normalize the histogram
    cv2.normalize(hist, hist)

    # return the histogram as a 1-d array
    return hist.flatten()

hist_1 = fd_histogram(image_1)
hist_2 = fd_histogram(image_2)

# calculating the distance between two histograms based on the correlation
↪  method
d = cv2.compareHist(hist_1, hist_2,  cv2.HISTCMP_CORREL)
```

Listing 1: Code implementation for extraction and comparison of color histograms.

A-2 - Extraction, calculation and comparison of texture feature vectors

```
from skimage.feature import local_binary_pattern

# global feature descriptor 2: Texture feature vector extraction
def lbp_histogram(image):

    # convert image to gray scale
    img = col.rgb2gray(image)
```

```
8
9       # calculation of the LBP
10      patterns = local_binary_pattern(img, 8, 1)
11
12      # extracting texture histogram for comparison
13      hist, _ = np.histogram(patterns, bins=np.arange(2**8 + 1), density=True)
14      return hist
15
16  hist_1 = lbp_histogram(image_1)
17  hist_2 = lbp_histogram(image_2)
18
19  # calculating the distance between two histograms based on the correlation
    ↪  method
20  d = cv2.compareHist(hist_1, hist_2,  cv2.HISTCMP_CORREL)
```

Listing 2: Code implementation for extraction and comparison of texture feature vectors.

A-3 - Extraction, calculation and comparison of moment feature vectors

```
1   # global feature descriptor 1: Hu Moment extraction
2   def fd_hu_moments(image):
3
4       # convert image to gray scale
5       image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
6
7       # extract and flatten hu moments
8       hu_moments = cv2.HuMoments(cv2.moments(image)).flatten()
9
10      # Log scale hu moments to achieve comparable scale
11      for i in range(0,7):
12          hu_moments[i] = -1* math.copysign(1.0, hu_moments[i]) *
            ↪  math.log10(abs(hu_moments[i]))
13
14      return hu_moments
```

Listing 3: Code implementation for extraction and comparison of moment feature vectors.

A-4 - Extraction, calculation and comparison of global feature vectors

```
# Calculates a concatenated vector of multiple low-level feature vectors and adds this to nested dictionary of all images
# Returns said vector
def calculate_low_level_global_sim_scores(images):

    sim_scores = defaultdict(dict)
    for image_name1 in images:

        # Load image 1
        image = cv2.imread(image_name1)

        # Extract features of image 1
        texture = lbp_histogram(image)
        hist = lbp_histogram(image)
        hu = fd_hu_moments(image)

        # Concatenate features into global feature descriptor
        global_feature = np.hstack([texture, hist, hu])

        for image_name2 in images:

            # Skip this image if it has already been handled or if it is the same as image 1
            if image_name2 not in sim_scores.keys() and image_name2 != image_name1:

                    # Load image 2
                    image_2 = cv2.imread(image_name2)

                    # Extract features of image 2
                    texture_2 = lbp_histogram(image_2)
                    hist_2 = lbp_histogram(image_2)
                    hu_2 = fd_hu_moments(image_2)

                    # Concatenate features into global feature descriptor
                    global_feature_2 = np.hstack([texture_2, hist_2, hu_2])

```

```
                    # Calculate euclidian distance between these features
                    glob_eucl = (numpy.linalg.norm(global_feature -
                    ↪  global_feature_2))

                    # Add global feature vector to dictionary
                    sim_scores[image_name1][image_name2] = glob_eucl
    return sim_scores
```

Listing 4: Calculation and storage of global feature descriptors

A-5 - Extraction and calculation of facial landmark differences

```
face_image_names = glob.glob(FACE_IMAGES_LOCATION)

shapes = {}

for image_name in face_image_names:

        # load image
        face = face_recognition.load_image_file(image_name)

        # resize all faces to same size
        face = cv2.resize(face, (500,550))

        # compute face landmark coordinates
        face_shapes = face_recognition.face_landmarks(face)

        # store in memory
        shapes[image_name] = list(face_shapes)

# store coordinates to JSON
with open('Datasets/shape_data.json', 'w') as fp:
    json.dump(shapes, fp)

# initialize Data Frame for storage
# each column represents one face landmark
shapes = pd.DataFrame(columns=['img_type','image1', 'image2', 'left_eyebrow',
↪  'right_eyebrow','nose_bridge','nose_tip', 'left_eye', 'right_eye'])
```

```
26
27  # calculates the difference in shape between each face's landmarks and each
    ↪  other face's landmarks
28  def calculate_face_shape_distances():
29
30      with open('Datasets/shape_data.json') as json_file:
31
32          data = json.load(json_file)
33          for image_name in tqdm(data.keys()):
34
35              if data[image_name]:
36                  for image_name2 in data.keys():
37
38                      if image_name != image_name2 and data[image_name2]:
39
40                          # keep dictionary for all landmark differences between
                            ↪  current two images
41                          ret={}
42
43                          # For each landmark, calculate distance between faces
                            ↪  and store in dictionary
44                          for landmark in data[image_name][0]:
45                              face = cv2.imread(image_name)
46                              face2 = cv2.imread(image_name2)
47                              face_landmarks_list1 =
                                ↪  face_recognition.face_landmarks(face)
48                              face_landmarks_list2 =
                                ↪  face_recognition.face_landmarks(face2)
49                              landmark1 = face_landmarks_list1[0][landmark]
50                              landmark2 = face_landmarks_list2[0][landmark]
51
52                              # actual comparison of shapes using
                                ↪  cv2.matchShapes
53                              ret[landmark] =
                                ↪  cv2.matchShapes(np.array(landmark1).astype(np.uint8),np.arr
54
55                          # extract image numbers for storage
56                          number1 = re.match('.*\(([0-9][0-9]?)\).*',
                            ↪  image_name)
```

```
                        number2 = re.match('.*\(([0-9][0-9]?)\).*',
                        ↪  image_name2)

                        # add data to Data Frame
                        new_row = ['polished',number1.group(1),
                        ↪  number2.group(1), ret['chin'],
                        ↪  ret['left_eyebrow'], ret['right_eyebrow'],
                                ret['nose_bridge'], ret['nose_tip'],
                                ↪  ret['left_eye'], ret['right_eye']]
                        shapes.loc[len(shapes)] = new_row
```

Listing 5: Code implementation for extraction and calculation of facial landmark differences.

A-6 - Extraction and calculation of colourfulness from an image

```
# returns colourfulness of given image
def image_colorfulness(image):

        # split the image into its respective RGB components
        (B, G, R) = cv2.split(image.astype("float"))

        # compute rg = R - G
        rg = np.absolute(R - G)

        # compute yb = 0.5 * (R + G) - B
        yb = np.absolute(0.5 * (R + G) - B)

        # compute the mean and standard deviation of both `rg` and `yb`
        (rbMean, rbStd) = (np.mean(rg), np.std(rg))
        (ybMean, ybStd) = (np.mean(yb), np.std(yb))

        # combine the mean and standard deviations
        stdRoot = np.sqrt((rbStd ** 2) + (ybStd ** 2))
        meanRoot = np.sqrt((rbMean ** 2) + (ybMean ** 2))

        # derive the "colorfulness" metric and return it
        return stdRoot + (0.3 * meanRoot)
```

Listing 6: Code implementation of the colourfulness metric extraction.

A-7 - Busyness metric algorithm implementation

```
def sort_contours(cnts, method="left-to-right"):
    # initialize the reverse flag and sort index
    reverse = False
    i = 0

    # handle if we need to sort in reverse
    if method == "right-to-left" or method == "bottom-to-top":
        reverse = True

    # handle if we are sorting against the y-coordinate rather than
    # the x-coordinate of the bounding box
    if method == "top-to-bottom" or method == "bottom-to-top":
        i = 1

    # construct the list of bounding boxes and sort them from top to
    # bottom
    boundingBoxes = [cv2.boundingRect(c) for c in cnts]
    (cnts, boundingBoxes) = zip(*sorted(zip(cnts, boundingBoxes),
                                        key=lambda b: b[1][i],
                                        ↪  reverse=reverse))

    # return the list of sorted contours and bounding boxes
    return cnts, boundingBoxes

# function to calculate busyness of image
def calculate_busyness(image):

    original = image.copy()
    gray = cv2.cvtColor(image,cv2.COLOR_BGR2GRAY)

    # bilateral filtering
    blur = cv2.bilateralFilter(gray, 15, 40, 40)

    # apply Otsu's thresholding
    thresh = cv2.threshold(blur, 0, 255, cv2.THRESH_BINARY +
    ↪  cv2.THRESH_OTSU)[1]
```

```
35
36      # erode image to eliminate very small objects
37      dilation_kernel = cv2.getStructuringElement(cv2.MORPH_RECT, ksize=(5, 5))
38      thresh = cv2.erode(thresh, dilation_kernel, 3)
39
40      # search for contours in filtered image
41      cnts = cv2.findContours(thresh, cv2.RETR_EXTERNAL,
        ↪  cv2.CHAIN_APPROX_SIMPLE)
42      cnts = cnts[0] if len(cnts) == 2 else cnts[1]
43      (cnts, _) = contours.sort_contours(cnts, method="left-to-right")
44
45      # the number of closed contours is the busyness of the image
46      number_of_objects_in_image = len(cnts)
47
48      # visualization
49      ret = cv2.drawContours(image, cnts, -1, (0,255,0), 3)
50
51      return number_of_objects_in_image
```

Listing 7: Code implementation of the busyness metric algorithm.

A-8 - Extraction and calculation of sharpness and contrast

```
1   def get_sharpness_value(image):
2       canny = cv2.Canny(image, 50,250)
3       sharpness = np.mean(canny)
4       return sharpness
5
6   def get_contrast_value(img):
7       img_grey = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
8       contrast = img_grey.std()
9       return contrast
```

Listing 8: Code implementation for extraction and calculation of sharpness and contrast.

A-9 - Extraction, calculation and comparison of GIST feature vectors

```
# Create object to extract GIST descriptors
gist_helper = GistUtils()

# Load image names
landscape_images = glob.glob("../Datasets/Landscapes/*.jpg")
face_images = glob.glob("../Datasets/Faces/*.jpg")
polished_face_images = glob.glob("../Datasets/Faces_polished/*.jpg")

# Function to calculate GIST scores of all given images, and add them to dictionary
# Returns said dictionary containing 'image: score' pairs
def calculate_gist_scores(images):
    gist_scores = defaultdict(dict)
    counter_1 = 0
    for image_name in images:
        image = cv2.imread(image_name)
        gist = gist_helper.get_gist_vec(image)
        gist_scores[image_name] = gist.tolist()
        print('Calculating gist for ', image_name, '...')
    return gist_scores

# Calculates similarities between all GIST scores of each image, adding them to a nested dictionary
# Returns said dictionary containing 'image1: image2: similarity' values
def calculate_similarities(images, scores):
    gist_distances = defaultdict(dict)
    for image1 in images:
        for image2 in images:
            if image1 != image2:
                distance = spatial.distance.cosine(scores[image1], scores[image2])
                gist_distances[image1][image2] = 1 - distance
                print('Distance between ', image1, ' and ', image2, ': ', distance)
    return gist_distances

# Calculate scores, distances and write them to json file
```

```
34
35  scores = calculate_gist_scores(landscape_images)
36  distances = calculate_similarities(landscape_images, scores)
37
38  with open('gist_landscape_scores.json', 'w') as fp:
39      json.dump(scores, fp)
40  with open('gist_landscape_distances.json', 'w') as fp:
41      json.dump(distances, fp)
42
43
44  # Finds most similar other image for each image, based on previously
    ↪  calculated similarity scores
45  def find_most_similar_pairs(distance_json_name, score_json_name,
    ↪  output_json_name):
46
47      # ------------------------------------------------------------ #
48      # -------- Open JSON files containing necessary values ------- #
49      # ------------------------------------------------------------ #
50
51      with open(distance_json_name) as json_file:
52          distances = json.load(json_file)
53      with open(score_json_name) as json_file:
54          scores = json.load(json_file)
55
56      # ---------------------------------------- #
57      # -------- Find most similar pairs ------- #
58      # ---------------------------------------- #
59
60      # Sort nested dictionary on similarity score, so each image's dictionary
        ↪  is sorted separately
61      sort_scores = {key: dict(sorted(val.items(), key=lambda ele: ele[1],
        ↪  reverse=False)) for key, val in distances.items()}
62
63      # Keep a dictionary to keep track of already used images, for efficiency
        ↪  purposes.
64      used = {}
65      for image in sort_scores.keys():
66          used[image] = False
67
```

```
    # Iterate over each image and take first image from dictionary, as this will be the most similar image
    pairs = []
    for image in enumerate(sort_scores.keys()):
        if used[image[1]] is not True:
            best_image = list(sort_scores[image[1]].keys())[0]
            if best_image in used and used[best_image] is not True and best_image != image[1] and sort_scores[image[1]][best_image] >= 0:
                used[image[1]] = True
                used[best_image] = True

                pair = (image[1], best_image, round(100 * (1 - round(sort_scores[image[1]][best_image], 3)), 2))
                pairs.append(pair)

    # Write pairs to JSON file
    with open(output_json_name, 'w') as fp:
        json.dump(pairs, fp)
```

Listing 9: Calculation and storage of GIST descriptors