UNIVERSITY COLLEGE LONDON

UCL INTERACTION CENTER

Department of Computer Science

Understanding the Role of Explanations in Computer Vision Applications

by

Ahmed Alqaraawi

Thesis for the degree of Doctor of Philosophy

March 13, 2022

UNIVERSITY COLLEGE LONDON

ABSTRACT

FACULTY OF PHYSICAL SCIENCES AND ENGINEERING
Department of Computer Science

Doctor of Philosophy

UNDERSTANDING THE ROLE OF EXPLANATIONS IN COMPUTER VISION APPLICATIONS

by Ahmed Alqaraawi

Recent advancements in AI show great performance over a range of applications, but its operations are hard to interpret, even for experts. Various explanation algorithms have been proposed to address this issue, yet limited research effort has been reported concerning their user evaluation.

Against this background, this thesis reports on four user studies designed to investigate the role of explanations in helping end-users build a better functional understanding of computer vision processes. In addition, we seek to understand what features lay users attend to in order to build such functional understanding, and whether different techniques provide different gains. In particular, we begin by examining the utility of "keypoint markers"; coloured dot visualisations that correspond to patterns of interest identified by an underlying algorithm and can be seen in many computer vision applications. We then investigate the utility of saliency maps; a popular group of explanations for the operation of Convolutional Neural Networks (CNNs).

The findings indicate that keypoint markers can be helpful if they are presented in line with users' expectations. They also indicate that saliency maps can improve participants' ability to predict the outcome of a CNN, but only moderately. Overall, this thesis contributes by evaluating these explanation techniques through user studies. It also provides a number of key findings that provide helpful guidelines for practitioners on how and when to use these explanations, as well as which types of users to target. Furthermore, it proposes and evaluates two novel explanation techniques as well as a number of helpful tools that help researchers and practitioners when designing user studies around the evaluation of explanations. Finally, this thesis highlights a number of implications for the design of explanation techniques and further research in that area.

Impact Statement

Nowadays, we can see the widespread adoption and use of applications, products, and processes that leverage the latest breakthroughs in Artificial Intelligence (AI). The pervasive application of AI spans a wide variety of areas, including predictive policing, healthcare, and social services, among others. Thanks to the open-source practises, employing AI algorithms has become significantly easier each day, to the level where no technical or coding skills is necessary.

Given this growth of AI, which is expected to continue in the future, the European Parliament adopted the General Data Protection Regulation (GDPR), which includes the right to explanation when automated decision making takes place. The necessity of receiving an explanation is emphasised when the target users who are using these algorithm have no experience on the inner-workings of the AI algorithms. Moreover, previous research (Yang and Newman, 2013a) has revealed that lay users may overestimate the performance of AI systems, resulting in over-reliance and perhaps detrimental use. As a result, it is critical to assess if such explanation techniques, particularly those already present in products or claimed to be effective for lay-users, are truly useful and do not result in biased user understanding of system decisions.

By focusing on target users with no AI experience, this PhD project sought to contribute to this space by evaluating existing widely used explanation techniques and developing new ones. It is hoped that this work will influence the research community to truly design and assess novel explanation techniques that are centred around human needs and emphasise what leads to the user's functional understanding of the system. Furthermore, because of the government's regulation processes, it is anticipated to see more explanation techniques deployed in commercial applications. Therefore, we hope that the methodological contributions and design implications presented in this thesis will serve as helpful guidelines for practitioners and decision makers regarding when explanations are required, the types of explanations to use, and the types of users to whom these explanations should be directed. While this thesis focuses on two examples of computer vision explanation techniques, we believe that the guidelines presented here are applicable to other domains.

Declaration of Authorship

I, Ahmed Alqaraawi , declare that the thesis entitled *Understanding the Role of Explanations in Computer Vision Applications* and the work presented in the thesis are both my own, and have been generated by me as the result of my own original research. I confirm that:

- this work was done wholly or mainly while in candidature for a research degree at this University;
- where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated;
- where I have consulted the published work of others, this is always clearly attributed;
- where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work;
- I have acknowledged all main sources of help;
- where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself;
- parts of this work have been published, please see Section 1.4 for details.

Signed:...

Date:...

Acknowledgements

To begin, I would like to express my deepest gratitude to my supervisors, Enrico Costanza and Nadia Berthouze, for their efforts, continuous support, and constructive guidance. Their knowledge, skills, and dedication served as the most significant source of motivation throughout the course of my PhD.

I would like to thank my colleagues and friends at UCL and beyond for their valuable insights and support. I was fortunate enough to work closely with outstanding researchers from whom I learned a lot.

I would like to thank my examiners, Simone Stumpf and Yvonne Rogers, for their expert advice, encouragement, and constructive criticism. The viva was an enjoyable and memorable experience, with many excellent recommendations for improvements.

I would also like to extend my thanks to my family and friends who have supported me throughout this PhD. In particular, I would like to express my gratitude to my wife, Khawlah, and my parents for believing in me as well as for their kindness, love, and understanding.

Finally, I would like to acknowledge the financial support provided by KACST, without which my PhD would not have been possible.

Contents

List of Figures

Chapter 1

Introduction

As Artificial Intelligence (AI) increasingly becomes an integral part of many computer programs, its impact on our society spans a wide spectrum of domains. Some systems have already been shown to outperform humans at certain tasks like lung cancer screening (Ardila et al., 2019). With the ambition to increase efficiency and reduce cost, many public and private organisations are adopting "data-driven" ML systems to support or even take decisions around applications that range from predictive policing (Mohler et al., 2015), healthcare (Cai et al., 2019a), to social services (Kleinberg et al., 2018), and many others (Stone et al., 2016; Campolo et al., 2018). Therefore, there have been several calls to make such systems accountable so that even users who are not ML experts can decide when to trust their predictions (Shneiderman, 2016; The IEEE Global Initiative on Ethics of Autonomous and Intelligent Systems, 2017).

However, many ML algorithms currently operate as *opaque box systems*. When trained with large amounts of data, they may perform very well, but understanding the underlying process by which results are achieved is difficult, even for experts. In other words, interpretability is still a fundamental and open technical challenge (Lipton, 2018). This is especially the case for one of the most popular and best performing types of ML systems: Deep Neural Networks (DNN). Miller (2019a) defined interpretability as "the degree to which a human can understand the cause of a decision." Researchers from various fields have contributed to that notion. A large theme focuses on proposing algorithms that produce explanations for AI models. Others, attempt to employ the vast literature in philosophy, psychology, and cognitive science of how humans generate and present explanations to other humans and attempt to apply this knowledge to the machine-human context (Miller, 2019a). Another theme focuses on evaluating the different explanation techniques either analytically (Samek et al., 2017) or by conducting studies with target users. Although explanations are designed for humans to consume, there appear to be fewer user studies than theoretical or analytical papers in the literature. Past work has highlighted the need for further user studies to evaluate the significance of explanations in complex systems and recognised this as a research gap (Narayanan et al., 2018a; Yin et al., 2019a). Moreover, as AI applications became more widely available, users who were not necessarily AI specialists began to use them. Previous research (Yang and Newman, 2013a) has revealed that lay users may overestimate the performance of AI systems, resulting in over-reliance and perhaps detrimental use. As a result, it is critical to assess if such explanation techniques, particularly those already present in products or claimed to be effective for lay-users, are truly useful and do not result in biased user understanding of system decisions.

In light of this, this doctoral work examines the role of explanations in helping end users understand complex system decisions. We are interested in evaluating systems that process images, as this is an area for which some of the most impressive results have been reported to date, and with a broad range of applications (Pouyanfar et al., 2018). In particular, this thesis focuses on evaluating two types of explanations. The first is in the context of a classical pattern recognition system, and is referred to as "keypoint markers" - coloured dot visualisations which correspond to patterns of interest identified by an underlying algorithm. These keypoint markers are most likely derived from a keypoint matching algorithm, an intrinsic part of many computer vision applications, e.g. panorama stitching, object detection, gesture recognition, and motion tracking. The second type of explanations is saliency maps, which are a popular group of explanations in the context of Machine Learning (ML) systems and represent a visualisation that highlights which pixels were most important for the image classification of some class (e.g. the cat class). We are interested in evaluating saliency maps produced to explain the decisions of Convolutional Neural Networks (CNNs), which are currently the most prevalent ML algorithm for computer vision applications. Figure 1.1 shows examples of both techniques; keypoint markers (left) and a saliency map (right).

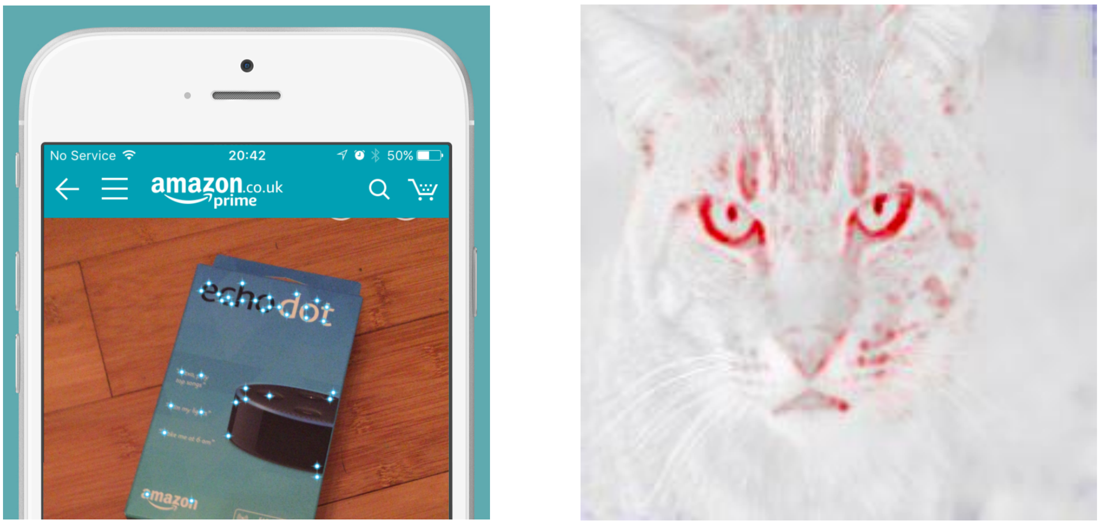


Figure 1.1: (Left): Smart Camera Apps that display keypoint markers feedback to users. (Right): a saliency map suggests that the red part of the image supports the CNN classification of this image as a cat.

1.1 Research challenges

In this section, we outline the key challenges we encountered while conducting this research:

- **The reasonable amount of data to display** –When working with complex systems, one critical design decision is deciding how much detail should be displayed to users. Previous research (Kulesza et al., 2013) explored the trade-off between showing detailed information about the underlying process of the system versus the accuracy of the explanation. In our studies, we have explored this trade-off between providing an explanation for all images in a dataset and letting the users freely navigate through the dataset or sample a few representative images that reflect the behaviour of the system on various outcomes. Both options have their own pros and cons. While the first may perhaps provide a holistic overview of the system, it does not ensure that users look at the same instances, which could weaken any conclusion regarding the evaluation of the effect of displaying saliency maps on user understanding.

- **Defining the evaluation scope** –Interpretability is a latent property that can be influenced by manipulable factors such as the number of features, model complexity, and the participants' level of expertise, and these factors influence measurable outcomes such as user trust, user ability to estimate the model's outcome, and user ability to detect biases (Poursabzi-Sangdeh et al., 2018; Chromik and Schuessler, 2020). Taking into account these elements and others results in a large design space. As a result, selecting an evaluation scope that ensures a rigorous outcome is a major challenge when evaluating complex systems (Doshi-Velez and Kim, 2017a).

- **Choosing an appropriate evaluation measure** – Another challenge was choosing an appropriate evaluation measure. Previous work used different measures to evaluate explanation techniques, including the user's ability to detect mistakes (e.g. (Kulesza et al., 2015)), user's capacity to build a better classifier (e.g. (Poursabzi-Sangdeh et al., 2018)) and the ability to choose the right model among multiple candidate options (e.g. (Krause et al., 2018a)). Other studies evaluate explanation techniques by measuring the time it takes users to complete a task (e.g. (Bussone et al., 2015)).In addition to these measures, the capacity of a user to predict the output of a model has been proposed and used as a measure of a system's transparency or explainability (Lipton, 2018; Muramatsu and Pratt, 2001). This measure may be significant since it informs the user about the classifier's generalizability to real-world data and, thus, indicates the level of trust the user should place in the classifier.

- **Study methods and constraints** –Finally, an important decision to face when evaluating systems is the choice between running lab or field studies. Generally, field studies allow users to be exposed more to the explanations and indirectly measure their value in natural settings. However, designing an experiment in such settings is challenging because of the many external factors that may influence the interaction cycle (Sharp, 2003). Studies in these settings need to be carefully controlled, preferably in a way not obvious to participants. Furthermore, in order to simulate a wide range of real world scenarios, design tasks need to lead participants to experience instances of failure and success. As an alternative to this option, lab studies have the advantage of being able to be run faster and cheaper than field studies, and the conditions can be carefully controlled. The trade-off between the two options is a known design challenge in HCI (Rogers, 2011) and represents a research challenge in the context of evaluating smart systems. Due to the COVID-19 pandemic, from March 2020 running field studies was not feasible, hence the work was constrained to use online studies.

In summary, this section highlights some challenges that a researcher may encounter while evaluating explanation techniques for complex systems. In our research, we aim to evaluate the role of explanations and how they can be used to improve user understanding of complex models. However, given the broad design area, we intend to focus our effort on a specific scope. In the following section, we explicitly define the research questions of this thesis based on the challenges outlined above and influenced by the preceding research provided in Chapter 2.

1.2 Research questions

The main topic of this thesis is to examine the role of explanation in helping end users understand complex algorithms' decisions, as well as how explanations should be designed to improve user understanding. Within this broad scope and following a survey of the literature, a number of research questions were identified and developed over the course of the PhD. The first two questions were in the context of a classical pattern recognition system. The formulation of these questions was motivated by an observation of a number of smart camera apps that have been developed to assist users in a variety of tasks, such as product searching. To simplify user interaction, these apps usually include visual feedback, overlaying the camera's viewfinder with visual aids called ”keypoint markers”. While such visualisations have long been popular as a debugging tool for software developers, to date little is known about their effect on end-user interactions. Their inclusion may simply be motivated by a need to convey background activity, however, their presence motivated us to raise the following research question:

(R1) Do keypoint markers help users in building a better functional understanding of computer vision processes?

Where by ”functional understanding”, we refer to the sort of understanding that helps users interact with the system effectively rather than understanding the inner-working processes of the model. We tackle this research question by addressing the following sub-questions (which have been investigated in Chapter 3):

- R1.1: Are keypoint markers intelligible to lay users?
- R1.2 Do they improve usability and aid users' interaction around failures?
- R1.3 Can they mislead users if misunderstood?

Furthermore, because data in computer vision is typically propagated via a pipeline with multiple stages of processing, in which keypoint markers represent information about an early stage of that pipeline, we were also interested in the following research question:

(R2): What key stages of computer vision processes need to be made visible (through keypoint markers) to improve lay users' functional understanding? (Chapter 3).

In the next studies, the thesis focus shifted to Convolutional Neural Networks (CNNs) which are currently the most prevalent algorithm for computer vision applications. Following a survey of the literature (Chapter 2), saliency maps emerged as one popular form of explanation for such algorithms. Moreover, previous work claims that they are easy to interpret by both novice and expert users (Lapuschkin et al., 2019). However, we found that a limited number of user studies have been conducted to evaluate saliency maps. Therefore, we decided to investigate the role of saliency maps in informing user understanding and pose the following research questions:

(R3) How do saliency maps help with building functional understanding, including the relation to varied system confidence? (Chapter 4).

Specifically, we seek to measure this understanding by asking participants to predict the CNN classification outcome of an image (we call it the ”*task image*”) and count the number of correct user's predictions.

In addition, we would like to understand more about the kind of features participants pay attention to with and without the presence of saliency maps. Therefore, we are interested in the following question:

(R4): What features do lay users attend to in order to build a functional understanding of computer vision processes? (Chapter 5).

Finally, we seek to understand whether the different saliency map techniques help users to varying degrees by posing the following research question:

(R5): How do different saliency map generation techniques perform to build functional understanding? (Chapter 6).

For the studies concerning saliency maps, the research questions listed above were addressed through two study designs. In the first, a number of examples along with their corresponding saliency maps were displayed, and the user task was to predict the CNN classification outcome for a task image. The second design is of lower intrinsic complexity, in which saliency maps are presented alongside the task image.

The next section describes how this thesis was structured in order to address these research questions.

1.3 Research structure

The work of this PhD involves a series of lab studies designed to investigate the role of explanation in user understanding. The thesis is divided into the following chapters:

Chapter 2: highlights significant prior work and describes how it relates to our research. The chapter also explains how the research questions emerge from the reviewed literature.

Chapter 3: documents between-groups user studies examining the role of visual feedback on informing user understanding. We examine the effect of showing keypoint markers and compare them to other feedback options that have been derived from different stages of the data processing. The study highlights a number of interesting findings and provides implications for pattern recognition feedback design.

Chapter 4 detailed our first study to examine the role of saliency maps. In this study, participants were asked to estimate the CNN outcome on images (we refer to them as "*task images*"). A few representative instances were selected and displayed as examples in the interface. These examples were the most similar ones in terms of score to the task image. Findings indicate that the presence of saliency maps did not result in a significant difference between conditions in terms of correct guessing of whether images would be correctly or incorrectly classified by the model. However, along with other interesting findings, a main theme that emerged was the mentioning of features across all conditions. Qualitative data showed that for correct answers, saliency map participants often mentioned features that could be highlighted by the saliency map, while participants in the no-saliency map condition did not. This finding drives us to look for a different sampling strategy to locate images with similar patterns, rather than selecting images with the closest score.

Chapter 5 builds on the work reported in Chapter 4, with a similar design except that the selected examples were those with similar patterns according to the CNN embeddings (a low-dimensional representation of input data learned by the CNN model). Given this new setup, when saliency maps were present, participants' ability to predict the outcome of the network for new images was improved. However, even with saliency maps present, the improvement was moderate (60.7% prediction accuracy). We further report on a number of key findings which includes the observation that saliency maps appear to prime participants to primarily focus on what saliency maps can highlight and pay less attention to other attributes that saliency maps cannot highlight.

Chapter 6 reports on a study that evaluates saliency maps with a task with a lower complexity compared to the studies reported in Chapter 4 and Chapter 5. Particularly, participants were asked to perform multiple tasks in which saliency maps were presented alongside the task image. This simplified design also allows us to compare multiple saliency map generation techniques based on multiple measures. A number of findings are provided, along with key implications for designing and using saliency map approaches, including the importance of selecting a technique in light of the intended task.

Chapter 7 provides a summary of the work presented in this thesis. It concludes with the main findings, an acknowledgement of the limitations, and a discussion of potential avenues for further research in this area.

1.4 Research contributions

This section outlines the key research contributions reported in this thesis.

In Chapter 3, we present findings from a study that investigates user interaction around pattern recognition algorithms, which addresses **R1** and **R2**. Our findings indicate that participants who received explanations derived from later stages (higher level) of the data processing demonstrated an improved understanding of the system operation compared to explanations derived from an early stage (lower level). In particular, keypoint markers can help users in building a better functional understanding of computer vision processes as long as they are derived from a stage of processing that is inline with user's expectations. From this, we suggest that the stage of processing from which feedback is derived plays an important role in users' capacity to develop coherent understandings of a system's operation and that feedback must be presented inline with user expectation, the violation of which could result in misconception.

Regarding **R3**, in the context of saliency map explanation in particular, quantitative and qualitative data in our studies highlight the following:

1. Considering participants' ability to predict the CNN classification outcome of images as one measure of users' functional understanding, our findings indicate that saliency maps could help participants predict the outcome of the model, but overall, the success rates were relatively low. Moreover, when we consider task images with different classification outcomes, we found that showing the saliency maps of the displayed examples does not appear to aid participants when compared to only showing the classification scores of these examples.

2. We report on some instances in which despite having access to the saliency maps some participants expected the system to understand human high-level concepts, where in reality, CNN learns patterns in a bottom-top hierarchy fashion in which meaningful patterns that *look like* what we humans refer to as ”semantics” may emerge in the deep layers of the network, but that is not guaranteed (Chapter 4)

3. Our data showed that when images with low CNN scores were sampled, features were mentioned a lot less frequently by our participants suggesting that the utility of saliency maps varies according to the classification score. These findings suggest that a saliency map may highlight what supports the prediction of some classes, but it will fail to provide counter-factual evidence, namely, the absence of evidence (Chapter 4 and Chapter 5).

4. Regarding **R4**, saliency maps appear to prime participants to primarily focus on what they highlight (which we called Saliency-Features), but potentially distracting them from other attributes such as colour and contrast, which saliency maps cannot highlight (Chapter 5).

5. We designed, implemented and evaluated two new saliency map techniques. The first is ”semantic occlusion” (sem-occl) which was designed to specifically focus on features that are meaningful to people (i.e., semantics). As a generalisation of this approach, we also proposed a second occlusion technique: ”multi-scale occlusion” (m-scale-occl), which uses rectangular occluding regions arranged on multi-scale grids (Chapter 6).

6. We investigated **R5**: How do different saliency map techniques perform to build functional understanding? where we measure this understanding via different tasks. Our findings indicate that the utility of the different saliency map techniques appears to vary depending on the task at hand. For example, a technique that aids users in predicting the CNN classification outcome, may not be effective in assisting users in detecting errors or biases (Chapter 6).

Based on those findings, we made a number of recommendations for using and designing explanation techniques. In addition, the work detailed in this thesis has been published (or is under review) at the following venues:

Chapter 3 is presented in the following CHI conference paper:

Jacob Kittley-Davies, Ahmed Alqaraawi, Rayoung Yang, Enrico Costanza, Alex Rogers, and Sebastian Stein. Evaluating the effect of feedback from different computer vision processing stages: A comparative lab study. In Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, CHI '19, pages 43:1–43:12, New York, NY, USA, 2019. ACM. ISBN 978-1-4503-5970-2.

My role in relation to this paper included designing and conducting the study as well as analysing the results.

Chapter 5 is presented in the following IUI conference paper:

Ahmed Alqaraawi, Martin Schuessler, Philipp Weiß, Enrico Costanza, and Nadia Berthouze. 2020. Evaluating Saliency Map Explanations for Convolutional Neural Networks: A User Study. In Proceedings of the 25th International Conference on Intelligent User Interfaces (Cagliari, Italy) (IUI'20). Association for Computing Machinery, New York, NY, USA, 275–285.

Chapter 4 and 6 form the biases for the following paper in submission:

Ahmed Alqaraawi, Enrico Costanza, Nadia Berthouze and Emma Holliday. Evaluating and Improving Heatmap Explanations for CNNs through to online user studies. ACM Transactions on Computer-Human Interaction (TOCHI).

For this paper, the development of the tree structure and its processing (which will be explained in detail in that chapter) was performed by Emma Holiday as part of her Master's thesis.

1.5 Acknowledgements

This work was supported by the Engineering and Physical Sciences Research Council Aperio (EP/L024608/1) and A- IoT (EP/N014243/1) projects and was approved by the Ethics Committees of UCLIC.

Chapter 2

Literature Review

In this chapter, a review of the literature on the role of explanations in informing users of complex models is presented. We begin by providing a brief overview of some of the concepts covered in this thesis. Then, we examine research that argues for the importance of users developing sound mental models of complex systems. Following that, we provide a general background on pattern recognition and machine learning algorithms, explore why they are more difficult to interpret, and compare them to other modeling techniques. Next, we detail works that are more relevant to our studies. In particular, we review works that propose or evaluate methods that derive a feedback from the processing pipeline as a basic type of explanation. Then we discuss how visualisation and interaction are utilised for the purpose of model understanding, where we place more emphasis on saliency maps as popular group of explanation techniques typically employed to provide instance-level explanations. Finally, we present an overview of several works that focus on evaluating ML post-hoc explanation techniques, with a focus on relevant research that evaluates saliency maps through user studies.

2.1 Background

Before going over the existing literature that is relevant to our work, and because some terms are defined differently by different authors, in this section, we briefly define some concepts as they are used in this thesis.

Data processing pipeline –Machines are designed to process data starting from low level features, propagating the data in a series of processing stages until it reaches a final decision. For example, in computer vision, images are passed through various stages of processing, starting from a low level of processing that identifies distinctive points of interest in an image (e.g. keypoints features), to other stages that map these features to a domain that is invariant to geometric and photometric variations. Figure 2.1 shows an example of a data processing pipeline that we have used in Chapter 3. Depending on the algorithm and the application, the pipeline can be structured in a variety of ways.

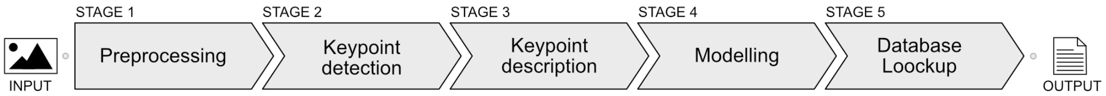


Figure 2.1: An example of a data processing pipeline that we have used in Chapter 3

Keypoint matching algorithm processes – The keypoint marker feedback seen in many consumer applications is most likely derived from a keypoint matching algorithm, an intrinsic part of many smart camera apps, e.g. panorama stitching, object detection, gesture recognition and motion tracking. Most keypoint matching algorithms involve three stages of processing: (i) identify distinctive points of interest in an image (the keypoints), (ii) programmatically describe them, so that the description is resilient to geometric variations e.g. rotation, scale and perspective, and photometric variants e.g. contrast and brightness, and (iii) compare the descriptions with those of another image. How the results of this comparison process are used is application specific. In panorama stitching for example, the closest matching descriptions between images are assumed to represent the same point in the physical world. Using their relative changes in position the images can be transformed such that the keypoints overlap creating a new combined image with a wider field of view.

Machine learning (ML) –refers to the set of algorithms that allow a computer to learn and discover patterns in data without having to be explicitly programmed (Samuel, 1959). Computer vision, natural language processing (NLP), and bioinformatics are examples of machine learning's sub-fields.

Neural Networks (NNs) –are the group of ML algorithms that represent a mathematical framework for learning patterns from data. The network consists of a collection of connected units structured in multiple layers. Each unit applies a simple data transformation function. An optimisation algorithm (called backpropagation) adjusts the parameters of these functions in an iterative form by observing and learning from many input examples (often called the training set). Figure 2.2 depicts a basic demonstrative example, where for state of the art architecture, the number of nodes can reach millions. It is worth noting that although the name (Neural Networks) has a reference to the neurobiology term, there is no evidence that the brain processes information the same way as NNs (Chollet, 2017).

Convolutional Neural Networks (CNN) –is a subset of the NNs which includes different types of layers such as densely connected layer but mainly it involves what is called *convolutional layers*. When compared to densely layers, convolutional layers have number of key advantages. First, the learned patterns (or features) have the characteristics of being translation-invariant, which means that if the model learns a

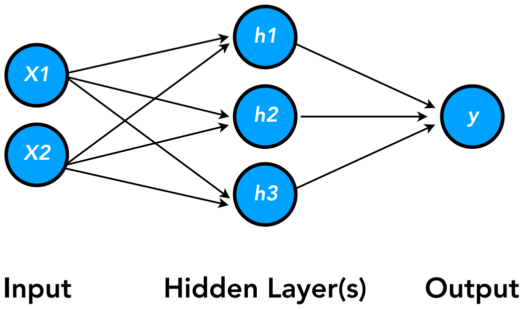


Figure 2.2: A demonstration of a basic NN with one hidden layer.

pattern at the left lower corner of an image, then this pattern would be recognised at other locations such as the centre. Second, that they can learn spatial hierarchies of patterns in which basic local patterns such as edges and corners are learned in the first few layers, while more complex representations (such as dog faces) emerge in deeper layers (Zeiler and Fergus, 2014). Third, they require fewer parameters when compared to dense architectures. These characteristics make CNNs a powerful algorithm for learning from image data (Zhang et al., 2020). Figure 2.3 shows one types of CNN architectures called VGG16, which is the one we used in our work.

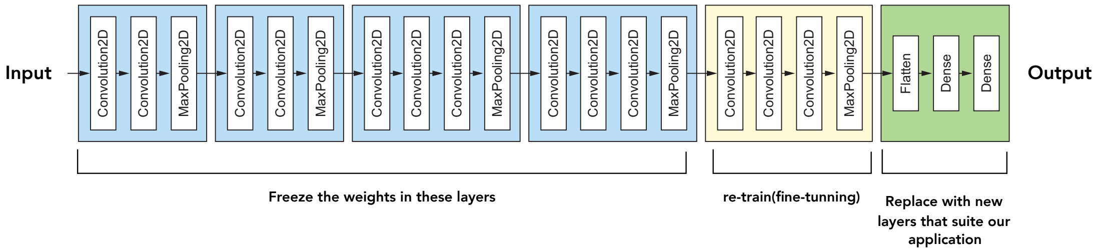


Figure 2.3: VGG16 architecture. Adapted from (Chollet, 2017)-Figure 5.19

CNN features (Embeddings) –Previous work (Zeiler and Fergus, 2014) showed that when CNNs are trained on a large and diverse dataset, they learn a powerful and compressed representation of the data (sometimes called feature vectors) in a hierarchical fashion. They show that the network's first few layers are typically responsive to low-level features (such as corners and shapes), and that as you move deeper into the network, more complex representations (such as dog faces) emerge. Such representation could be helpful in variety of tasks such as instance retrieval (Sharif Razavian et al., 2014). Chollet (2017) demonstrated this learning process in Figure 2.4.

Fine-tuning –Training a NN for complex dataset such ImageNet (Deng et al., 2009) from scratch require a lot of computational resources. Fortunately, it has been shown that one can utilise the parameters (i.e. embeddings) that have been learnt for one dataset and use them for others. Typically, a pre-trained network is a network that was trained on a large and general enough dataset, on which generic representations were learned and can prove useful for many different computer vision problems, even if these new problems involve classes that are completely different from those of the original task.

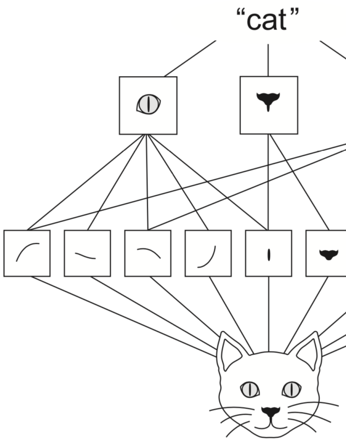


Figure 2.4: A demonstration of how CNN features emerge. Early layers learns basic patterns, while next layers build on that and learn more complex patterns. From (Chollet, 2017)

Fine-tuning is a widely used approach to achieve this task, which consists of freezing some of the network's early layers and substituting and training the remaining layers with new ones that suite the new problem (Chollet, 2017). For example, in our work, we have utilised a model that has been pre-trained to classify the ImageNet dataset (1000 classes), and replaced the last few layers with new fully connected layers that classify the Pascal dataset (with only 20 classes) (Figure 2.3).

Multi-label problems – In the multi-label image classification problem, an image can contain multiple objects. For example, the assignment of the labels "*horse, train*" is considered correct if both, a horse and a train are visible in the image. This problem has been considered in our studies because in this context, saliency maps have the potential to highlight specific parts of the image that correspond to one label, as well as parts that correspond to alternative labels.

The definition of TP, TN, FN and FP for multi-label classification – In the multi-label image classification problem, for each object class, the CNN computes a classification score between 0 and 1. Hence, a criterion needs to be defined so that the score can be translated into an outcome: TP, TN, FN and FP, where these outcomes are defined as follows:

- True Positives (TP), where a label had been correctly assigned;
- True Negatives (TN), where the CNN had been correctly excluded a label.
- False Negatives (FN), where the CNN had failed to assign the label;
- False Positives (FP), where the CNN had incorrectly assigned the label.

In Chapter 4, the system accepts the predicted labels that satisfy one of two criteria: (1) the predicted label is among the top three highest scores (By inspecting the PASCAL dataset, we found that it is very rare for an image to contain more than 3 objects of interest). This criterion was motivated by the common measure used for evaluating datasets by considering the top-n labels. However, we believe that this measure is appropriate for datasets with a large number of labels, such as ImageNet-1000, therefore we chose to include another criterion that takes into account the prediction score value, which is (2) the predicted category is higher than a pre-defined threshold (we used here 0.1).

In Chapter 5, we revised this metric to account for CNN's various performances across classes. In particular, we calculated threshold values for each class (e.g. horse, cat) where each threshold was obtained by maximising the F1-score (a common statistical measure that takes into account the number of TPs, FNs, and FPs) for the class on the dataset.

Interpretability –There seem to be no universal agreement on the definition of interpretability. The reference to this notion is diverse and distorted and often depends on the context in which it was used (Lipton, 2018). Miller (2019a) attempts to provide a general definition of interpretability as "the degree to which a human can understand the cause of a decision". Miller uses the notions of interpretability and explainability interchangeably. Montavon et al. (2017) differentiate between interpretations and explanations. In their definition, an interpretation is the mapping between an abstract concept to a human understandable domain. Images and text are examples of interpretable domains, while the collection of abstract model weights is not. An explanation is the collection of features or examples in that interpretable domain that support the model's outcome.

Post-hoc explanation –given a pre-trained model, A post-hoc explanation is a representation that explains predictions ("functional understanding") without clarifying how models work (Lipton, 2018). Examples of post-hoc explanations are natural language explanations, prototypical examples and saliency maps (which are the ones we focus on in our work).

2.2 The importance of system Intelligibility

Before reviewing the literature that are relevant to our research, we seek to demonstrate the importance of system intelligibility for effective user interaction with smart systems. The HCI community is particularly interested in how users understand systems. Consequently, a large body of relevant literature exists. For example, Dix (1992) discussed the potential impact of deep neural networks (DNN) on interpretability. If mental models are sufficiently accurate, they enable an interaction with a system that is more efficient. However, when flawed they may cause confusion, misconceptions, dissatisfaction and erroneous interactions (Kulesza et al., 2015). Similarly, the overestimation of a system's intelligence or capabilities has been shown to impact user interaction negatively (Alan et al., 2016). This may lead to over-reliance on a system, less vigilance towards system failures and unrealistic expectations (Yang and Newman, 2013b). Explanations for better system understanding have been investigated in several field including the context of information retrieval (Koenemann and Belkin, 1996), recommender systems (Kulesza et al., 2012) and context-aware systems (Lim et al., 2009).

Our work contributes to this space, investigating the importance of user understanding on effective interaction with smart systems. We present observations of misconceptions and their consequences for user interaction, as well as examine the implications for designing effective visual feedback for such systems. In the next section, we discuss the primary benefits of employing pattern recognition and machine learning systems, which justify their widespread use despite being more difficult to interpret.

2.3 Are pattern recognition and ML algorithms less Intelligible?

Following our discussion in the preceding section of why it is necessary to make systems understandable, we now present a general background on pattern recognition and ML algorithms, discuss why they are more difficult to interpret, and contrast them with other modeling techniques.

By the definition of Breiman et al. (2001), there are two main approaches of statistical modeling. The first method is known as ”data modeling,” in which data are generated by a known data model and the model is frequently validated using goodness-of-fit. Examples of this category are the linear regression and logistic regression models. The second is called **algorithmic modeling** where the model is considered unknown (black box). The main aim of these algorithmic models is to automatically find the optimum underlying structure that leads to the highest predictive power. The majority of ML learning algorithms, including k-nearest-neighbors, Neural Networks (NN), and Random Forest, fall into this second category.

The question therefore becomes, what is the point of using algorithmic modeling if it leads to black-box systems? According to Breiman et al. (2001), in addition to reaching state of the art accuracies, algorithmic techniques have a number of other advantages. For example, with the availability of data, algorithmic models (e.g. NN) may highlight novel connections or potential causal relationships between variables, leading to new hypotheses in that field. When a mathematical model cannot reach a comparable performance of a state-of-the-art algorithmic model, this hypothesis and suggests that further understanding is still missing in that field (Shmueli et al., 2010).

Algorithmic modeling, on the other hand, should be used with caution. Because what the algorithm learns is constrained by the data provided, it may not reflect the underlying model, posing the risk of arriving at a biased model. The medical case study mentioned in (Caruana et al., 2015) is a good example in this context. The algorithmic model (which was a NN in that work) learns a counter-intuitive rule from a pneumonia dataset, concluding that asthmatic pneumonia patients have a lower chance of dying from pneumonia than those with general pneumonia. However, it turns out that pneumonia patients with a history of asthma are more likely to be admitted to the Intensive Care Unit (ICU) and receive better care, which helps. The data-driven algorithm will not be able to spot this unless the dataset reflects the link between asthma and ICU admission. In (Lapuschkin et al., 2016), two algorithmic models, Fisher vector (FV) and CNN, were trained to predict a ”horse” class (Figure 2.5). The accuracy of the two models was comparable. Despite the high accuracy of the FV model, the authors applied an explanation algorithm, which highlighted the importance of a copyright tag (which is often found in horse photos from that dataset). After removing the copyright tag, the FV model’s accuracy dropped substantially. These two examples demonstrate that attaining a high level of accuracy is not a sufficient indicator that the algorithm is learning the intended model.

ML systems are currently widely employed by users of varying levels of expertise. Because there are many different ways to improve the intelligibility of machine learning models, in this thesis, we focus on ML explanation techniques that are suitable for users that don’t necessarily have knowledge of machine learning. In the next section, we review works that are relevant to our research.

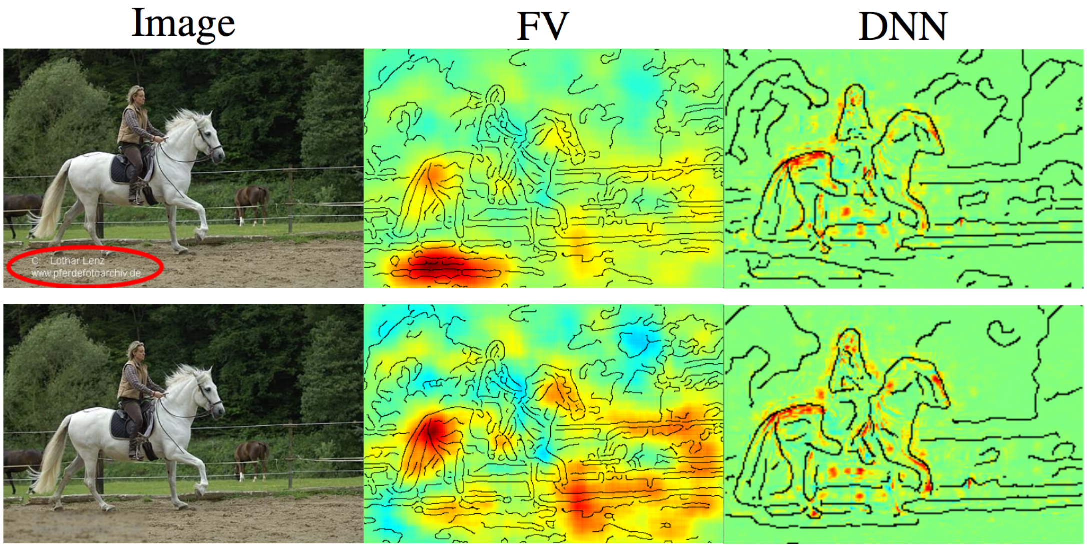


Figure 2.5: An explanation presented as a saliency map for FV and DNN models. Before and after removing the copyright tag (Lapuschkin et al. (2016))

2.4 Explanations in computer vision systems

A large body of literature proposes a variety of different solutions to improve the intelligibility of machine learning models. For literature reviews, we refer the interested reader to (Lipton, 2018; Guidotti et al., 2018; Adadi and Berrada, 2018).

One stream in this research field seeks to understand how a model works. For example, Strobelt et al. (2016) proposed a visual analysis tool to support the understanding of the hidden state dynamics of Recurrent Neural Networks (RNN). Kahng et al. (2018) presents *ActiVis*; an interactive visualisation system designed for large-scale deep learning models. One component of their system displays neuron activation, which aids practitioners in identifying and comparing patterns in the models' underlying processes. Such visualisations are useful for debugging complex models, but they demand a high level of ML knowledge. In our research, we target users that don't necessarily have such a background.

In this thesis, we are interested in the research field that seeks to explain computer vision models predictions with post-hoc explanations without uncovering the mechanisms behind them. Common explanation techniques in that space include:

- Deriving feedback from different stages of the data processing pipeline (Patel et al., 2010; Krause et al., 2016): which is a basic type of explanation that visualises the processed data (often called feature vectors) as it progresses through the various phases of processing. In Chapter 3, we designed a study around one type of this class of explanations called keypoints (Section 2.1).
- Saliency maps: which is a popular group of post-hoc explanations that assign a score to each input feature, determining the importance of this feature to the classification of some class. This score is then visualised as a saliency map that highlights the importance of such a feature. For example, the input data could be an image, and a saliency map would highlight the pixels that support the prediction of some label (e.g. car). In Section 2.4.2, we mention more details about the mechanism of generating saliency maps and a description of some popular techniques.
- Textual explanations: instead of highlighting the relevant part of the image that support the prediction, another explanation techniques is to produce a natural language sentences that describes the image. For example, for an image, the system produces the following sentence: ”A group of people sitting on a boat in the water” (Xu et al., 2015). Commonly this task is achieved by using a combination of CNN to extract an image embeddings and another network such as recurrent neural networks (RNN) to decode those embeddings into a meaningful text (Xu et al., 2015). Because the text is created by two different models (i.e., CNN and RNN),

it is difficult to determine which of these two models is at fault if the explanation is incorrect. In fact, some works showed that such textual output is not reliable and proposed techniques to explain them (Han et al., 2020).

- Example-based explanations: present subset of instances from the dataset that describe the behaviour of the model (Molnar, 2020). This type of explanations can be represented in multiple ways which includes: (1) counterfactual instances, which indicate which parts of the input would have the most impact on classifier output if plausible alternative values were substituted (Chang et al., 2018), (2) Prototypes which are representative data points sampled from the data (Molnar, 2020; Kim et al., 2016), (3) Influential instances represents the data points most responsible for a given prediction (Koh and Liang, 2017).

Montavon et al. (2017) defined an explanation as ”the collection of features of an interpretable domain (e.g. images or text, where a human can look at them and read them), that have contributed for a given example to produce a decision”. According to this definition, the techniques listed above are considered explanations. In our work, we focused on the first two explanation techniques: keypoints and saliency maps. The choice of keypoints was motivated by the observation that a number of commercial smart camera apps that target lay users include these keypoints. In regards to saliency maps, we observed that, in addition to the argument that such explanations can aid non-expert users (Ribeiro et al., 2016a), several techniques that are used to explain CNNs have been proposed in the literature, but they are rarely evaluated with users. Furthermore, previous work showed that when a constant shift is introduced to the input, various saliency map approaches fail to attribute appropriately (Kindermans et al., 2019), suggesting that they are not completely reliable. We seek to investigate this argument by evaluating several saliency map techniques through user studies. In the following subsections, we detail works that are more relevant to our studies.

2.4.1 Deriving feedback from the processing pipeline

Machines are designed to process data starting from low level features, propagating the data in a series of processing stages until it reaches a final decision. A basic type of explanation is to visualise the processed data (often called feature vectors) as it progresses through the various phases of processing.

Software platforms such as Crayons (Fails et al., 2003) and Eyepatch (Maynes-Aminzade et al., 2007) were specifically developed to insulate users from the complexities of computer vision and pattern recognition technologies. They theorise that by providing users with interfaces that facilitate ”rapid trial-and-error”, the most effective solutions to classification problems can be found.

However, as machine learning technologies become increasingly complex, Patel et al. (2008) have suggested that successful implementation can only be achieved with a deeper understanding of the inner- workings of the processes. DejaVu (Kato et al., 2012) was developed to expose domain-expert programmers to computer vision technologies, with the ambition of aiding code debugging. The system allows images passing through the various stages of processing to be inspected and an interactive timeline interface lets users record and examine data flow temporally. Although a small user study was conducted, the focus of that paper was demonstrating system functionality rather than the assessing user understanding. Zhao et al. (2016) conducted a study examining lay-users' interactions with an augmented reality pattern recognition system on a head-mounted display (HMD) designed to assist users with low vision in a product search task by recognising the product and utilising visual feedback to guide the user's attention to the product. The feedback in this study was derived from the output of a pattern recognition processing pipeline. Exposing the underlying data processing is an idea which has been explored in the domain of machine-learning. The creators of Gestalt (Patel et al., 2010) an integrated development environment (IDE) designed specifically to assist programmers creating software which makes use of machine learning technologies, demonstrated through lab studies, that exposing data at various stages of a process significantly improves programmers' ability to identify and correct errors in their code.

While these studies employ some sort of visual feedback, they do not compare feedback derived from multiple stages of the pipeline in an interactive computer vision application. The closest work in that space is (Zhao et al., 2016), however, the feedback was derived from one stage of the pipeline, namely the output. Our work in chapter 3 builds on the reported studies with the aim of investigating the capacity of algorithmic feedback to support user understanding, but also how it can lead to misconceptions if poorly designed.

2.4.2 Saliency map as an explanation technique

A popular group of post-hoc explanation techniques is feature-attribution or saliency map. In this section, we provide more details about saliency maps, how they are generated and key differences between several saliency map techniques. Although saliency maps can be classified as visualisation techniques, we have chosen to devote this section to them because this type of explanation covers up a large portion of this thesis.

For a given input, a relevance score is calculated for each individual input feature (e.g. pixel) then rendered as a saliency map (see for example Figure 2.6). Several techniques in the literature have been proposed to produce these saliency maps. In NN, the back propagation step involves calculating the gradient, which can be readily used to assess the relevance of the input features (Montavon et al., 2017). This process can be efficient since it often only requires one forward and backward pass through the network. The

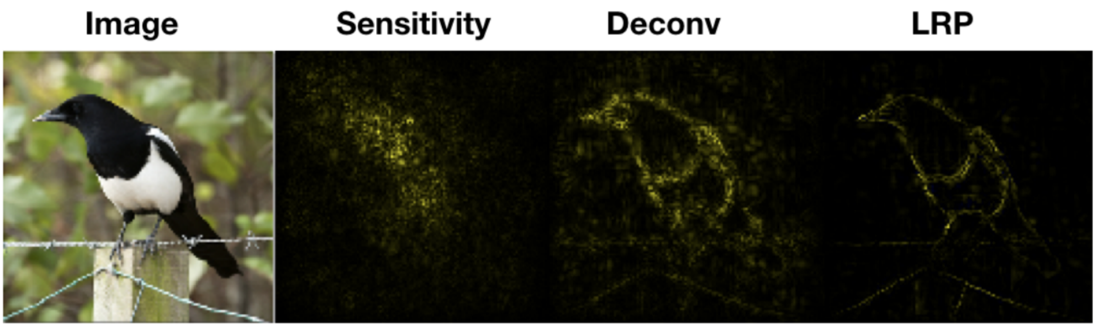


Figure 2.6: An explanation presented as a saliency map using Sensitivity Deconvolution and LRP approaches (Samek et al., 2017)

basic form of this process is the sensitivity analysis (Baehrens et al., 2010; Simonyan et al., 2013), where the saliency map intensities are calculated by taking the gradient (partial derivative) of the output score for a specific class with respect to the input.

To understand how a saliency map is constructed, it is helpful to recall how the back-propagation steps work. In NN, learning is achieved by minimising the loss (prediction error) with iterative updates of the network parameters. This process is often performed efficiently by the backpropagation algorithm, which involves updating the parameters in the opposite direction from the gradient, which decreases the loss. For demonstration, Figure 2.7 explains the process in a simple 1D continuous function that maps an input x to an output y.

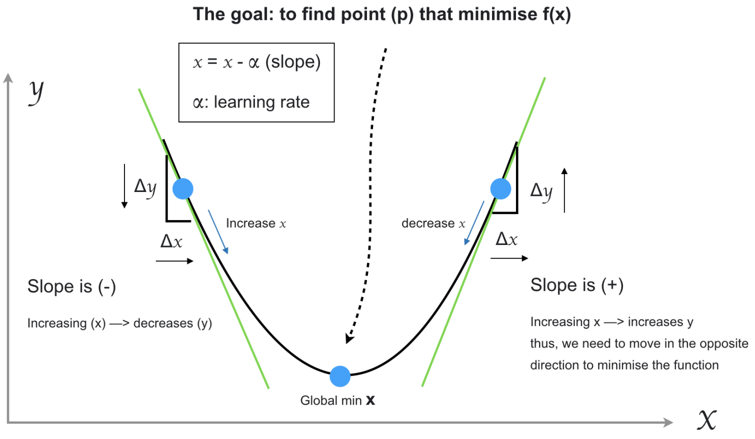

The goal: to find point (p) that minimise f(x)
x = x - α (slope)
α: learning rate
y
Δy
Δx
Increase x
decrease x
Δy
Δx
Slope is (-)
Increasing (x) —> decreases (y)
Slope is (+)
Increasing x —> increases y
thus, we need to move in the opposite
direction to minimise the function
Global min **x**
x

Figure 2.7: a simple demonstration of the optimisation process in the back-propagation algorithm

In the context of saliency maps, we utilise the gradients not to minimise the loss, but to have an indication of the contribution of each pixel. In particular, considering the most basic forms of saliency maps, in the forward pass step, we get a probability score for each class. Then we apply one backpropagation step for a specific class (e.g. cat), which yields a gradient for each pixel. Each gradient has a magnitude that indicates the importance of this pixel, and a sign to indicate that changing this pixel would contribute to the increase or decrease of the model's output.

Modern NNs architecture often involves multiple rectified linear units (ReLU) which serve as activation functions. In the back propagation step, the sensitivity analysis approach (Simonyan et al., 2014) sets the negative values to zero (i.e. multiplying by the indicator function) when mapping the signal from one layer to the previous one, which makes the backward mapping discontinuous (Montavon et al., 2017). To solve this limitation, Zeiler and Fergus (2014) proposed the deconvolution method where the back-propagated signal is passed through a ReLU similar to the one used in the forward-pass, which makes the mapping continuous. In addition, for pooling layers, this approach also records the location of the maxima to use it in the backward mapping. Bach et al. (2015a) argues that this approach, on the other hand, has two shortcomings; first, the negative relevance will be discarded by the ReLU in the backward pass, and second, because the back-propagated signal is not layer-wise normalised, the final saliency map may be primarily determined by a few dominating relevance scores. Thus, Bach et al. (2015a) attempts to solve these shortcomings by introducing the Layer-wise Relevance Propagation (LRP) algorithm. In contrast to the previous techniques, in the backward pass, the ReLU step is skipped to preserve the negative evidence. In addition, the prediction score is distributed to each node in that layer subject to how much that node has contributed in the feed-forward phase. To further provide an insightful saliency map, when propagating back across the different layers, scores have to satisfy a local relevance conservation principle to ensure meaningful mapping between the final prediction scores and the produced saliency map.

Alternative to the gradient-based approaches is a set of techniques that rely on the relationship between the input and the output. Saliency maps are formed by occluding parts of the input and observing how that affects the output. These are often referred to as perturbation-based (or occlusion-based) techniques, and they have the advantage of being model-agnostic. Solutions in that space include the occlusion of an image with a fixed-size grey square, and monitoring how that affects the output of the classifier (Zeiler and Fergus, 2014). Petsiuk et al. (2018) proposed RISE a technique that forms a saliency map by combining multiple random masks, where these masks are weighted by the score of the target class. Ribeiro et al. (2016b) presented LIME an algorithm that explains the prediction of any classifier (model-agnostic) by drawing samples around an instance x (super-pixels) and learning a model that is locally faithful by performing perturbations.

In Chapter 4 and the ones that follow, we focus on evaluating saliency maps as an explanation technique. In addition, in Chapter 6, we introduce and evaluate two novel occlusion-based saliency map generating approaches, extending previous work in this space.

2.5 The presentation of explanation

Given the complexity of current ML models, it is essential to study the level of detail that should be presented to users. To that end, Kulesza et al. (2013) examined the impact of soundness (the accuracy with which the feedback accurately reflects the underlying processes of the system), and completeness (To what extent the feedback describes all of the underlying processes of the system). They evaluated the impact of these two notions on the user's mental model by using a music recommender system as a vehicle. Their findings imply that completeness, rather than soundness, is more significant.

Bussone et al. (2015) raised the question of when explanations could be considered harmful. In a study that targeted primary care physicians to diagnose and treat balance disorders, the system showed two versions to the users: a comprehensive version, which provides an explanation that shows inputs associated with the diagnosis, and a second version, which shows fewer details. Their findings indicate that users who received a rich explanation from the system developed an over-reliance bias, which led them to accept results from the system despite knowing the possibility of error.

More relevant to our studies, some popular techniques produce instance-level explanations. Building a coherent understanding of the underlying model by examining these individual instances can exceed users' cognitive load. Therefore, selecting and displaying representative data points or finding ways to summarise these individual instances is crucial (Krause et al., 2018a). In our work, we plan to explore multiple options that enable users to examine large datasets.

2.6 Evaluating ML post-hoc explanation techniques

Subjective evaluation on aesthetic appearance of saliency maps (e.g. how the saliency map accurately highlights the parts of interest) is not accurate, yet some proposed techniques are guided accordingly (Adebayo et al., 2018). Samek et al. (2017) argue that explanations do not have to correspond to human intuition or focus on the item of interest, but rather on what the classifier has learned from the provided data. Moreover, it is important to define the scope of the evaluation; claiming that one approach is more interpretable than the other should be within the scope of the study design, which includes but is not limited to: data type, model, user expertise level, and designed task.

Chromik and Schuessler (2020) proposed a taxonomy for rigorously evaluating XAI, which was backed up by a thorough analysis of the literature from several disciplines involved in XAI. They grouped the requirements of XAI evaluation into three main groups: task related, participant-related and study design related dimensions. In the task dimensions, the authors distinguish between multiple intended explanation goals such as transparency, trust, debugging and education. In addition, they stated a number of user tasks that have been proposed or used in the literature to evaluate the quality of the provided explanation, and grouped them according to the information provided to participants and the defined task. Examples of these user tasks are forward simulation tasks (or simulatability as defined by Lipton (2018)) and Counterfactual simulation tasks. In the Participant Dimensions, when evaluating XAI, it is necessary to take into account the participant expertise level: AI novices, domain experts, or AI experts, which may also determine the number of participants to be able to recruit (i.e. novices are often easier to recruit than domain experts). The last group is the study design dimensions in which the study method (i.e. a qualitative, quantitative, or mixed and whether it is between or within-subject) is also a factor that may determine the appropriate choice of treatments in terms of the provided types of explanations.

Given an explanation, Hoffman et al. (2018) attempt to understand whether this explanation has provided the user with a functional understanding of the AI system. For this goal, they proposed multiple levels of XAI evaluation which include: (1) the goodness of explanations which could be represented by clarity and precision often evaluated by researchers. (2) The satisfaction of the given explanations, which defined by the degree to which users feel that they understand the AI system or process being explained to them. (3) Their understanding of the AI system (i.e. accuracy of their mental model). A number of methods from the literature were proposed to elicit users' mental models which involves: Think-Aloud Problem Solving Task, the Nearest Neighbour Task, in which participants choose the explanation or diagram that best matches their views. (4) and how to measure the performance in terms of the users' success in conducting the intended task for which the system is designed. The authors noted that this measure will be a function of the previous levels: user satisfaction of the explanation, and their accuracy of their mental model. It will also be a function of their trust on the system.

2.6.1 Analytical evaluation of techniques

For saliency maps in particular, Samek et al. (2017) proposed an objective measure for evaluating different techniques that is based on a region perturbation process where an algorithm progressively removes relevant regions highlighted by the saliency map. They showed that LRP (which was proposed by the same authors) outperforms deconvolution and sensitivity analysis techniques (Section 2.4.2) when considering this measure.

Adebayo et al. (2018) suggests that relying solely on the visual appearance of saliency maps could be misleading. They further back up this argument with experiments that show that for some techniques, saliency maps appear to be unconnected to the model or the data generation process. In addition, they propose an evaluation framework that consists of a model parameter randomisation test and a data randomisation test. Gradients and GradCAM passed their sanity check among the saliency maps they tested, whereas Guided BackProp and Guided GradCAM failed. Similarly, Sixt et al. (2020) proposed a metric (cosine similarity convergence (CSC)) that can be used to assess the faithfulness of a heamap technique by tracing the lost information during the back-propagation step. The above mentioned work proposed analytical methods to evaluate saliency maps. Despite the fact that explanations are designed for humans to consume, there appear to be fewer user studies than theoretical or analytical papers in the literature. Therefore, in this thesis, we focus on evaluating saliency maps through user studies. In the next section, we put more emphasis on user studies that are relevant to our work.

2.6.2 User evaluation of techniques

Some studies in the literature use approaches that explain individual data points (i.e. in contrast to a global explanation that provide a summary of the model). In that sense, they are comparable to saliency maps, therefore, we believe that reporting some of these studies in this section will be helpful. Figure 2.9 provides a summary of the studies reported in this section, where the measures (i.e. the green columns) are the ones defined in Figure 2.8.

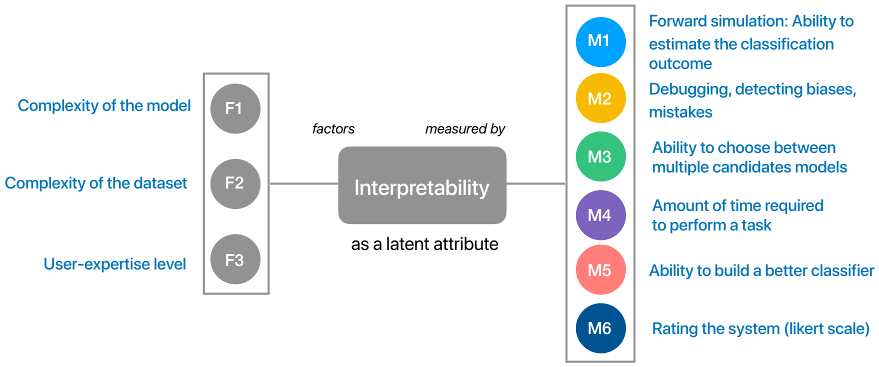

Complexity of the model
Complexity of the dataset
User-expertise level
F1
F2
F3
factors
measured by
Interpretability
as a latent attribute
M1
M2
M3
M4
M5
M6
Forward simulation: Ability to estimate the classification outcome
Debugging, detecting biases, mistakes
Ability to choose between multiple candidates models
Amount of time required to perform a task
Ability to build a better classifier
Rating the system (likert scale)

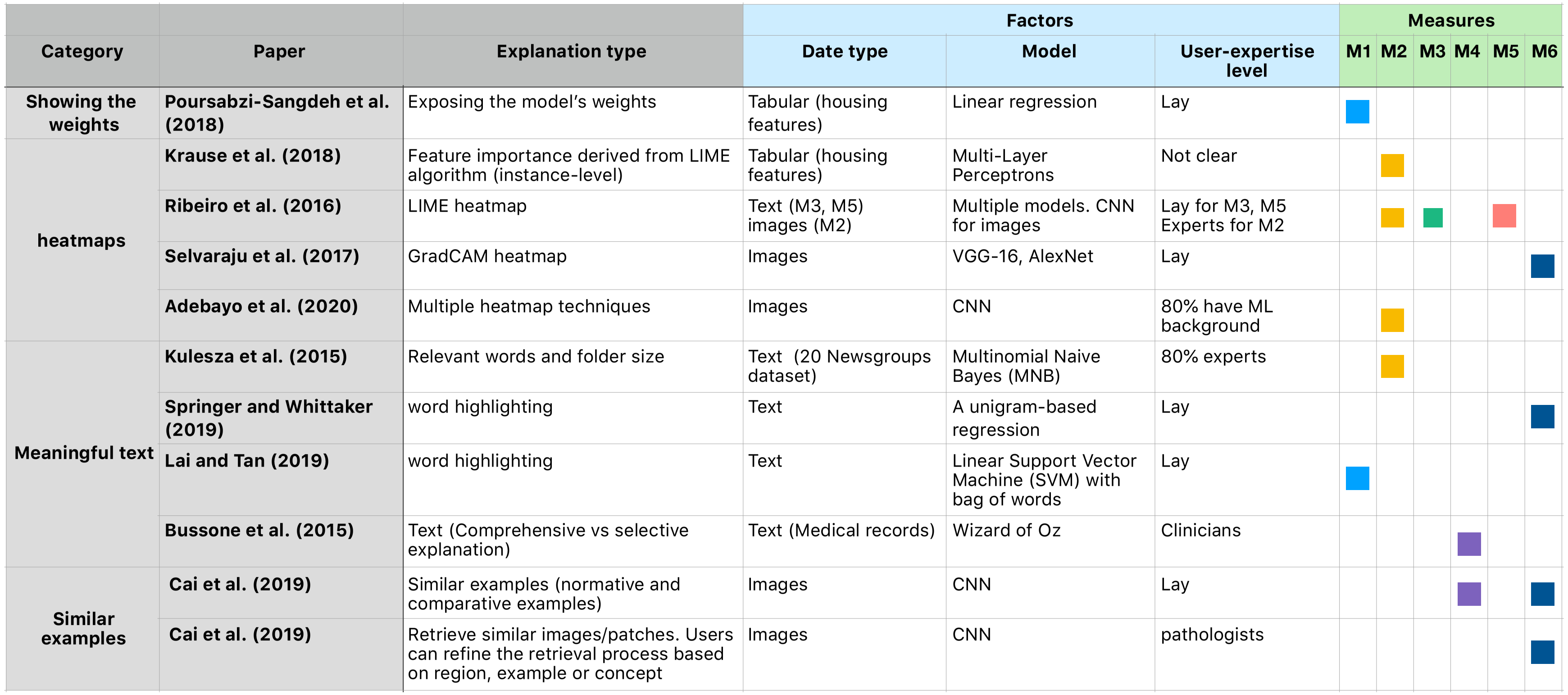
Figure 2.8: How can we measure interpretability?

Poursabzi-Sangdeh et al. (2018) conducted experiments with 1250 lay-users. They found that participants performed better at estimating the outcome of their model if there were fewer input features (2 vs. 8), and feature weights were revealed (transparent condition). However, such results cannot be generalised because predicting the outcome of such a simple linear model is a matter of performing a simple multiplication, which does not reflect the complexity of current machine learning models.

Explaining individual data points (sometimes called instance-level explanations) can be misleading if a user is exposed to a few instances that do not reflect the actual distribution of the data. Building patterns from such individual instances can exceed people's cognitive capacity. To that end, Krause et al. (2018a) proposed a visualisation that displays such instance-level explanations in aggregate format and showed that such visualisations improve users' capacity to spot data biases when compared to inspecting individual instances. Their work uses a housing price dataset with 10 features. In comparison to working with tabular data, creating a suitable aggregate representation of more complex data types such as images, with their approach, would not be applicable.

Several studies (Kulesza et al., 2015; Ribeiro et al., 2016a; Lai and Tan, 2019; Springer and Whittaker, 2019) have demonstrated the benefits of techniques that explain the importance of individual words for text-based classifiers. More relevant to our work are the studies conducted in the context of image classification. For example, Ribeiro et al. (2016a) proposed the LIME saliency map technique, and conducted a within-subject study in which they trained a biased classifier to distinguish between wolves and huskies. All the images of wolves in this set have snow in the background, whereas the images of Huskies do not. In the first phase, participants were shown ten images in which the model incorrectly identified two images, and they were asked to identify how the algorithm distinguishes between wolves and huskies. In the following phase, an explanation (LIME) for these ten images was presented. Their findings reveal that explanations help participants identify (snow) as a feature used by the algorithm to classify images. The study has two major limitations. First, because the sample size of the participants was small (the authors did not specify the number), no statistical analyses were performed. Second, the study was conducted on a simple binary classifier, and it's not clear whether increasing the dataset's complexity, and hence the model's complexity, will yield comparable results.

Selvaraju et al. (2017) proposed the GradCAM and Guided-GradCAM saliency map techniques and conducted two studies to evaluate their utility. In the first study, the aim was to compare four types of saliency map techniques in terms of their ability to discriminate between classes. For example, if an image contains both a human and a horse, a good visual explanation should be able to distinguish between the pixels that support the horse prediction and the pixels that support the person prediction. A total of 43 people were asked to look at a saliency map and select the category (e.g., horse or person) that was best depicted in the saliency map. They found that Guided-GradCAM outperforms all other techniques. In the second study, participants were given two explanations produced by two different models with varying accuracy (VGG16= 79.09 mAP versus AlexNet= 69.20 mAP), and they were asked to rate which model seemed more trustworthy using the provided saliency map techniques. With Guided-GradCAM, participants achieved a higher score in identifying that VGG16 is more accurate than AlexNet. Similar to (Ribeiro et al., 2016a), the study lacks any statistical analysis. More importantly, there is a strong assumption that a better saliency map is the one that highlights what we as humans expect, assuming that CNNs process and classify images the same way as humans. We argue that a reliable saliency map should highlight what CNN actually learns, even if it does not align with our expectations. For example, the saliency map presented in Figure 2.5 highlights the copyright tag rather than the



			Factors			Measures					
Category	**Paper**	**Explanation type**	**Date type**	**Model**	**User-expertise level**	**M1**	**M2**	**M3**	**M4**	**M5**	**M6**
Showing the weights	**Poursabzi-Sangdeh et al. (2018)**	Exposing the model's weights	Tabular (housing features)	Linear regression	Lay	🟦					
heatmaps	**Krause et al. (2018)**	Feature importance derived from LIME algorithm (instance-level)	Tabular (housing features)	Multi-Layer Perceptrons	Not clear		🟨				
	Ribeiro et al. (2016)	LIME heatmap	Text (M3, M5) images (M2)	Multiple models. CNN for images	Lay for M3, M5 Experts for M2		🟨	🟩		🟥	
	Selvaraju et al. (2017)	GradCAM heatmap	Images	VGG-16, AlexNet	Lay						🟦
	Adebayo et al. (2020)	Multiple heatmap techniques	Images	CNN	80% have ML background		🟨				
Meaningful text	**Kulesza et al. (2015)**	Relevant words and folder size	Text (20 Newsgroups dataset)	Multinomial Naive Bayes (MNB)	80% experts		🟨				
	Springer and Whittaker (2019)	word highlighting	Text	A unigram-based regression	Lay						🟦
	Lai and Tan (2019)	word highlighting	Text	Linear Support Vector Machine (SVM) with bag of words	Lay	🟦					
	Bussone et al. (2015)	Text (Comprehensive vs selective explanation)	Text (Medical records)	Wizard of Oz	Clinicians				🟪		
Similar examples	**Cai et al. (2019)**	Similar examples (normative and comparative examples)	Images	CNN	Lay				🟪		🟦
	Cai et al. (2019)	Retrieve similar images/patches. Users can refine the retrieval process based on region, example or concept	Images	CNN	pathologists						🟦

Figure 2.9: Examples of works that have evaluated some post-hoc explanations through user studies. Measurements (M1,M2, .., etc) are cross-referenced with figure 2.8

horse. Relying on the author's measure, we would reject this explanation because it does not agree with our expectations. The copyright tag example might be obvious, and one can reason and figure out such a bias; however, we should keep in mind that CNN may learn other unexpected patterns that do not correspond with human conceptions.

Adebayo et al. (2020) examined the efficacy of saliency maps in detecting artefacts generated at different stages of the development pipeline: input, model, and test-time. They discovered that while the investigated explanation approaches were capable of diagnosing a spurious background artefact, they were unable to detect mis-labeled data points. They conducted a user study and discovered that participants relied mainly on model prediction, but not on the saliency map, to identify flawed models.

Rendering prototypical examples proved to be valuable feedback (Kim et al., 2014). To that end, Cai et al. (2019a) evaluated two forms of example-based explanations: normative (similar examples from the same class) and comparative (most similar example from the training set regardless of the class). Participants are assigned to one of these two forms and asked to draw images, which are then passed to the NN algorithm. When the algorithm is unable to recognise the sketch, it provides examples from either form, depending on the condition. When the drawing was not recognised, participants who were given Normative explanations rated the system capability and their understanding higher. While relevant to our studies (i.e., later in our studies, we display examples to inform participants), this study did not evaluate saliency maps, and has the limitation of basing the evaluation on users' subjective ratings. In another work, Cai et al. (2019b) proposed an interactive tool that helps pathologists search for similar images or patches. Users can direct the retrieval process of the algorithm by instructing the tool to search for a similar region, example or concept. The tool was evaluated with pathologists with multiple metrics (rated on a 7-point scale) related to its utility.

To date, CNNs are becoming the default approach for many computer vision problems (Pouyanfar et al., 2018). While numerous post-hoc explanations for CNNs exist, they are rarely evaluated with users, presenting an opportunity to contribute to this space.

2.7 Summary and Discussion

As a summary, previous work shows that building sound mental models impacts the capacity of users to interact effectively with a system (section 2.2). The complexity often increases when specifically considering systems that employ pattern recognition or machine learning technologies. Several studies have demonstrated the benefits of making the motivations behind automated decisions salient to users.

Deriving feedback from the processing pipeline is one technique to inform users about system behaviour (section 2.4.1). Systems such as (Patel et al., 2008; Kato et al., 2012; Patel et al., 2010) were developed to reveal the underlying processes to achieve effective interaction. This can be reasonable as the target users in these studies were developers who perhaps need to understand and debug these systems. In contrast, systems such as (Fails et al., 2003; Maynes-Aminzade et al., 2007) which target lay-users, choose to distance the user from the internals of the system. This design choice can be feasible for some systems, however, as some new smart systems become increasingly complex and prevalent, there is a need to make these systems intelligible enough for users to support effective interaction. However, what information should be presented to users? Moreover, as data in these systems is typically propagated via multiple processing stages (low-level to high-level), from which stage should we derive the explanation? In Chapter 3 We aim to investigate these questions in the context of ”keypoint markers” because this basic type of explanation, which is typically derived from an early stage of the processing pipeline, can be seen in a variety of smart camera apps, but the literature lacks studies that investigate their utility. This observation motivated us to raise the research questions previously presented in Section 1.2.

The study in chapter 3 explored the research question in the setting of a classical CV algorithm (i.e. pattern recognition that employs a keypoint matching algorithm). To date, CNNs are becoming the default approach for many computer vision problems (Pouyanfar et al., 2018). Given its better performance and widespread use in many applications, it would seem reasonable to investigate explanation strategies in this domain influenced by earlier research findings. CNNs, unlike classic dense neural networks, preserve image spatial structure, allowing them to create more powerful models more efficiently. Furthermore, CNN operations can be easily parallelized across GPU units (Zhang et al., 2020).However, the number of parameters that construct CNN models is large, making interpretation difficult. Therefore, in section 2.4, we reviewed works that focus on explaining ML models in comparison to other data models, emphasising their positive and challenging aspects and reporting on how other fields of study contribute to this space by introducing explanation techniques or investigating interaction around them.

For pattern recognition algorithms, the findings of the study reported in Chapter 3 suggests that deriving feedback (explanation) from the later stages of the processing pipeline can be an effective means of informing lay-user understanding. Does that conclusion hold in the context of ML systems? And what would be an equivalent explanation that could be derived from a later stage of the ML pipeline, especially for computer vision applications?

As reported in Section 2.4, several explanation techniques have been proposed in the literature to explain ML models. Previous work claims that saliency maps (or heatmaps) as a form of explanation are easy to interpret by both novice and expert users, and that they can help to detect unexpected behaviour (Lapuschkin et al., 2019), and develop appropriate trust towards the system (Ribeiro et al., 2016a). Although many saliency map generation algorithms have been proposed and analytically examined (Doshi-Velez and Kim, 2017b), they are rarely evaluated with users (Abdul et al., 2018a; Chromik and Schuessler, 2020). Indeed, calls have been made for careful evaluation of ML explanations (Doshi-Velez and Kim, 2017b; Chromik and Schuessler, 2020). As reviewed in Section 2.6.2, and demonstrated in Table 2.9, there appears to be a limited number of user studies that have evaluated the utility of saliency maps using CNNs or models of equivalent complexity. We see this as a research gap and an opportunity to contribute to this space. As a result, we raise a number of research questions in Section 1.2, which we aim to address in a series of user studies reported in Chapter 4, Chapter 5 and Chapter 6.

To accurately assess the utility of saliency maps, one must validate a method while taking into account a number of elements, such as the complexity of the model and the dataset, as well as how these factors affect measurable outcomes, such as their ability to estimate the model's outcome. The mind map in figure 2.8 is an attempt to demonstrate some of the factors and measures (which are not comprehensive) that can be useful in identifying research gaps and serving as a guideline for future study design. In section 2.6.2, we reviewed the literature on the lens of the factors and measures reported in figure 2.8. Informed by the suggestion of prior work, in the first two saliency map studies (Chapter 4 and Chapter 5), we considered the participants' ability to predict the outcome of a ML classifier (M1) as a measure of their understanding of how such a system works (Lipton, 2018; Muramatsu and Pratt, 2001). This measure may be important since it informs the user about the classifier's generalisability to real-world data and, thus, indicates the level of trust the user should place in the classifier. In the next chapter, we begin by reporting our first study, which investigates user interaction around pattern recognition algorithms.

Chapter 3

Evaluating the Effect of Feedback from Different Computer Vision Processing Stages

In this chapter, we examine the role of visual feedback in informing end-users in the context of pattern recognition systems. The work detailed in this chapter has been published at the following CHI conference paper:

> Jacob Kittley-Davies, Ahmed Alqaraawi, Rayoung Yang, Enrico Costanza, Alex Rogers, and Sebastian Stein. Evaluating the effect of feedback from different computer vision processing stages: A comparative lab study. In Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, CHI '19, pages 43:1–43:12, New York, NY, USA, 2019. ACM. ISBN 978-1-4503-5970-2.

For the purpose of aiding user interaction, a number of commercial smart camera apps include visual feedback, over-laying the camera's viewfinder with visual aids. Two notable examples are Amazon app's "search by image" feature and Samsung's Bixby, a camera-based search tool[1] (Figure 3.1). Both display feedback in the form of "keypoint markers" - coloured dot visualisations which correspond to features of interest identified by an underlying algorithm. When a user points the camera at an object, the app starts searching for similar images and displays keypoint markers over-laying the camera's viewfinder to reflect the identified features. The placements of the keypoint markers vary interactively as the user moves and tilts their phone. While such visualisations have long been popular as a debugging tool for software developers[2], to date

[1] which tries to find matching images from an internet search

[2] e.g. OpenCV https://goo.gl/bX4XEM

little is known about their effect on end-user interactions. Their inclusion may simply be motivated by a need to convey that some background activity is taking place, however, their presence raises some interesting questions:

(R1) Do keypoint markers help users in building a better functional understanding of computer vision processes?

We tackle this research question by addressing the following sub-questions:

- R1.1: Are keypoint markers intelligible to lay users?
- R1.2 Do they improve usability and aid users' interaction around failures?
- R1.3 Can they mislead users if misunderstood?

In addition, because keypoint markers represent information about the early stage of data processing, we were also interested in the following research question:

(R2): What key stages of computer vision processes need to be made visible (through keypoint markers) to improve lay users' functional understanding?

Addressing these questions through a controlled yet ecologically valid study is particularly challenging, because it requires observing interactions around failures of the pattern recognition system. Such failures need to be controlled and repeatable, but their causes should not be obvious to participants. Moreover, the experimental tasks need to be engaging and enjoyable to motivate participants, have a clear goal and provide discussion points. Through experimentation, we found the task of creating stop-motion animations (described below) is a good choice that satisfies these criteria.

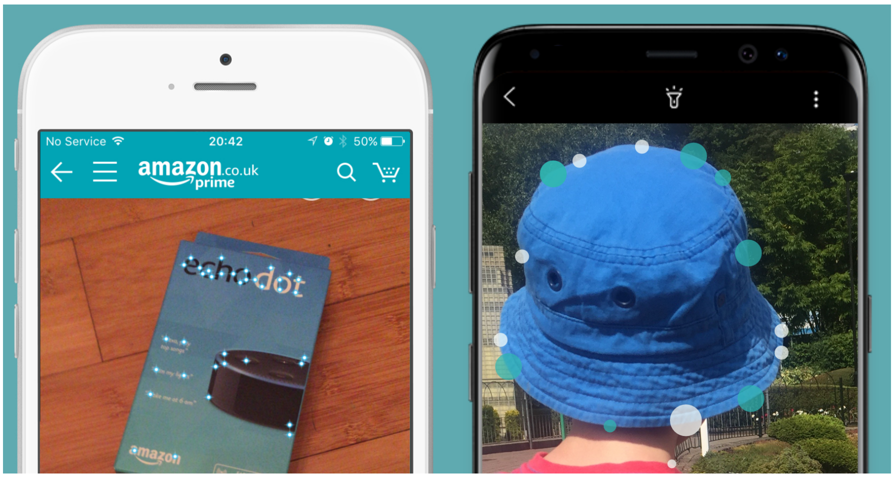


Figure 3.1: Smart Camera Apps that display keypoint markers feedback to users: left, Amazon and right, Samsung's Bixby.

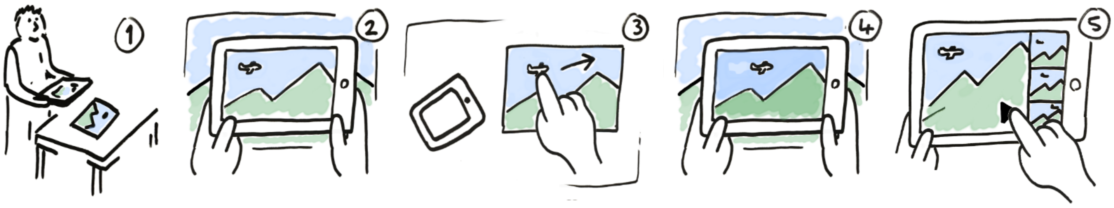


Figure 3.2: Illustration of procedure for creating an animation with Anim8 - (1) Set up the background scene with the character in its starting position and hold the tablet such that the camera's viewfinder encapsulates the scene and the character. (2) Capture a frame. (3) Place the tablet aside and manipulate the character in some way e.g. reposition or rotate. (4) Reposition the tablet and capture another frame. (5) Preview the captured frames / playback the animation. If the result is not acceptable then the frame can be deleted at this stage (or at any time later). (6) Repeat stages 3 to 5 until the animation is complete.

3.1 Study Design

To address the research questions presented above, we designed and conducted a between-groups study with four conditions with the aim of seeing if visual feedback obtained from various stages of the processing pipeline can assist a user in a task of automatic alignment of frames that results in a stop-motion animation.

3.1.1 Creating a stop-motion animation

To create a stop-motion animation, an animator must capture a series of still images (frames) of a given scene. By incrementally moving artifacts (characters) between frames the illusion of animation can be achieved (i.e. when the frames are played back in order the characters appear to move autonomously in relation to the static elements of the scene (e.g. the background)). Figure 3.2 demonstrates the process. Traditionally stop motion animations are created using cameras where the position and angle are strictly controlled e.g. held in a tripod. To incorporate pattern recognition technologies into our study design we replaced the controlled camera with a handheld tablet computer and bespoke app (Anim8[3]) which employs a keypoint matching algorithm[4] to align each frame to its predecessor - a process of stabilisation. This process makes all frames appear to have been captured from the same physical location even though the camera's position and angle vary. The keypoints with the closest descriptions are matched and assumed to point to the same physical feature in both frames. The most recently captured image can then be transformed so that its keypoints overlap its predecessors. Characters which have been moved between frames will create erroneous mappings, however if enough matches are found for the elements of the scene which have remained static (e.g. the

[3] For more information about the Anim8 app visit: http://anim8.space/

[4] Through experimentation the ORB algorithm Rublee et al. (2011) proved to offer the best compromise of performance, speed and control for our study.

background) then the matches associated with the moving characters will be treated as outliers and ignored.

In order for the stabilisation process to work effectively it is critical that the static elements of the scene are “feature rich”, i.e. the algorithm can identify many keypoints. If there are too few then the transformation process may output an image where the background is distorted and the character remains stationary (Figure 3.3). Leveraging this limitation, the likelihood of whether the stabilisation process will succeed or fail can be controlled - by providing “feature rich” and “feature poor” backgrounds participants of the study can be exposed to situations where the stabilisation process succeeds and fails respectively. Factors such as lighting conditions, shadows and camera angle make this form of manipulation not immediately obvious to study participants.

3.1.2 Tasks

Through pilot studies we concluded that four animation tasks with 4 to 5 frames per task provides sufficient exposure. We designed the tasks to assess whether feedback derived from the stabilisation process can help participants develop better understandings of the systems’ needs. To create discussion points and elicit user understanding we ask

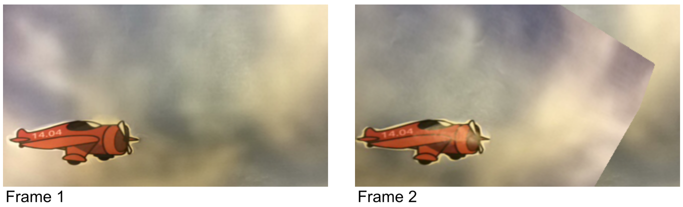


Figure 3.3: When too few matching keypoints are identified in the background, the stabilisation process can result in an image transformed such that the character appears to remain stationary and the background becomes distorted.

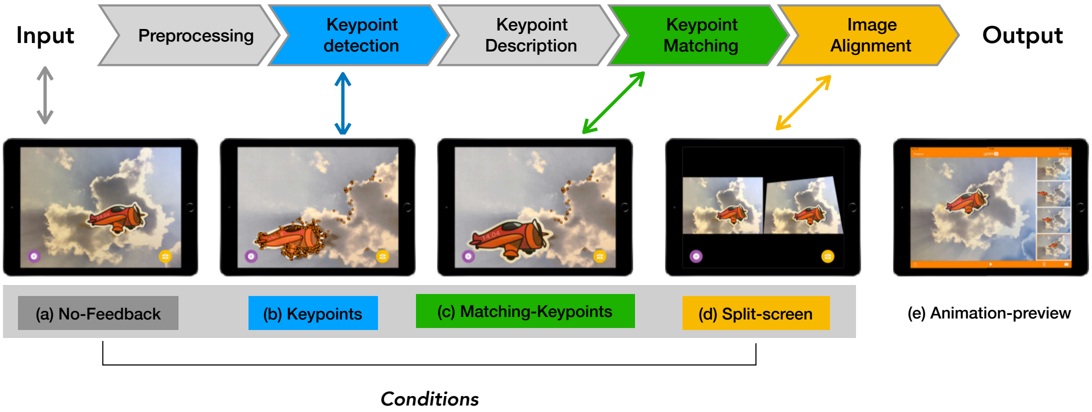
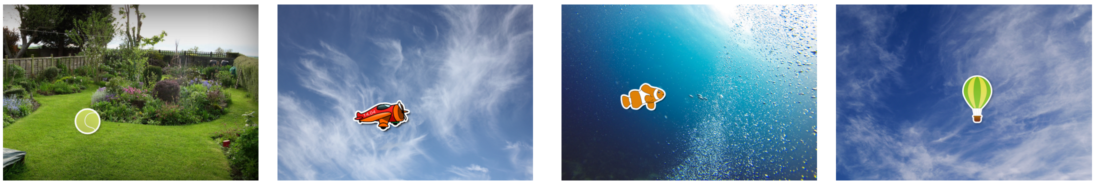


Figure 3.4: Examples of the feedback conditions presented by the Anim8 application and their relationship to the processing pipeline (a, b, c, d). Also the preview interface (e).



(a) Task 1 (b) Task 2 (c) Task 3 (d) Task 4

Figure 3.5: Example frame for each of the animation tasks.

participants to choose one of three background options in the last two animation tasks (3 options per task). The feature richness of the three background options varied and thus the likelihood of the stabilisation process succeeding varied (Figure 3.6). The tasks were structured as follows:

Task 1: This task was designed to allow participants to familiarise themselves with the UI and reassure them that the app works as described. To this end, a feature rich background (Figure 3.5a) which proved in testing to work with almost no failures was provided, making the task easy to succeed. On completion, the experimenter asked how the participants found using the app and if they had any queries.

Task 2: This task was designed to highlight the limitations of the system. The background in this task (Figure 3.5b) proved in testing to always fail. As it was impossible to complete this task, the experimenter would intervene after a time limit of 2 minutes, if the participant had not already raised concerns. The experimenter would ask the participants to explain what was happening and if they knew why it did not work, before suggesting that they proceed to the next task for brevity.

Task 3: This task was designed to assess users' understanding and create a point of discussion in the interview. Participants were asked to choose the background they felt would work best for the app from a selection of 3 backgrounds (see Figure 3.6). Participants were advised that they could preview them through the application's viewfinder if they wished. The backgrounds offered had previously been assessed and ranked according to the algorithm's ability to effectively identify features within them. One of the backgrounds consistently failed in testing and the remaining two consistently worked well, although one was more visibly "feature rich" than the other. The motivation for presenting users with this range of background options was to make the different levels of detail between the backgrounds less obvious. Once the participant completed this animation task, they were asked why they had selected that specific background.

Task 4: This task followed the same structure as Task 3, with a new character and set of 3 backgrounds (see Figure 3.6). This last task was designed to sustain participant interaction with the application, collect an additional data point and further assess user understanding (i.e. what, if anything, had been learned in Task 3.)

3.1.3 Conditions

We designed and conducted a between-groups study with four conditions. To explain the study conditions, we describe them in relation to the computer vision pipeline employed by Anim8 (Figure 3.4). It should be noted that we did not explain the feedback nor point out its presence to participants. This was done to mirror the experiences of current consumer smart camera app users.

No-Feedback (Figure 3.4 (a))

This condition was included as a baseline. The input images to the pipeline were presented back to participants without any additional feedback.

Keypoints (Figure 3.4 (b))

The camera's viewfinder was augmented with keypoint markers which indicate the locations at which keypoints had been detected in stage 2. It is important to note that not all the identified keypoints will be matched. Matches where the descriptions are considered too dissimilar are deemed outliers and are ignored by the stabilisation process. Despite this, the location, distribution and volume of identified keypoints are good indicators for the potential success of the stabilisation process.

Matching-Keypoints (Figure 3.4 (c))

Again the viewfinder was augmented with keypoint markers, however in this case only those which have been successfully paired with keypoints in the previous frame were displayed (Stage 4).

Split-Screen (Figure 3.4 (d))

This condition represents the final stage of processing. The viewfinder was divided into two equal halves. On the left: the input image updated in real-time (as per No-Feedback condition). On the right: the image outputted by the processing pipeline (update every ˜120ms).

The No-Feedback and Keypoints conditions were compared first, while the Matching-Keypoints and Split-Screen conditions were included at a later stage.

3.1.4 Procedure

All studies were conducted in the same empty windowless meeting room (so lighting conditions could be controlled) on the UCL campus. Two experimenters were present at all times - one to conduct the experiment and the other to observe, take notes and make audio recordings.

At the start of the study participants received written instructions (Section B.1) detailing: (i) the procedure necessary to create stop-motion animations, (ii) how Anim8 uses computer vision technologies to remove the need for a tripod, and (iii) a high level explanation of the image processing operations - that Anim8 tries to align images “by looking for things in each image which are not supposed to have moved, for example the background”. After reading the instructions participants were asked to stand up while performing the animation tasks.

Participants were tasked with creating 4 stop motion animations. Animating a two dimensional cardboard character (approximately 8cm by 5cm in size) moving across an A3 printed background (see Figure 3.5 for examples). To ensure that all participants had a good understanding of how to use the Anim8 application, the experimenter demonstrated the capture, playback and delete operations prior to the first task commencing. Whilst demonstrating the capturing of a frame, the participants were advised to ensure the printed background scene was fully encapsulated in the camera’s viewfinder and that the desk should not be visible. This was done to prevent features other than those in the scene impacting the outcome of the experiment (this was not explained to the participant). The participants were also advised that if they needed any assistance regarding the operation of the application during the study, then they could ask at any time.

Prior to each animation task, the experimenter provided each participant with the necessary materials (i.e. a character to animate and static background scene / scenes) and an instruction sheet detailing an example path for the character to follow, along with the number of frames expected (4 to 5). On completion of the task, the participant was asked to play back the animation they had created to the experimenter. The tasks were conducted in the same order for all participants to ensure that they experienced both successful and unsuccessful attempts.

At the end of the study a semi-structured interview was conducted. The interview began by asking participants if their experience in Task 3 and Task 4 had given them a better understanding of why the animation in Task 2 resulted in failure. Using this as a starting point, the experimenter asked further questions to assess the participants’ understanding of the algorithm and their motivations for selecting the backgrounds in Task 3 and Task 4. For the participants of conditions where feedback was presented in the viewfinder, the experimenter also asked what they thought it represented and if they used it in their decision making.

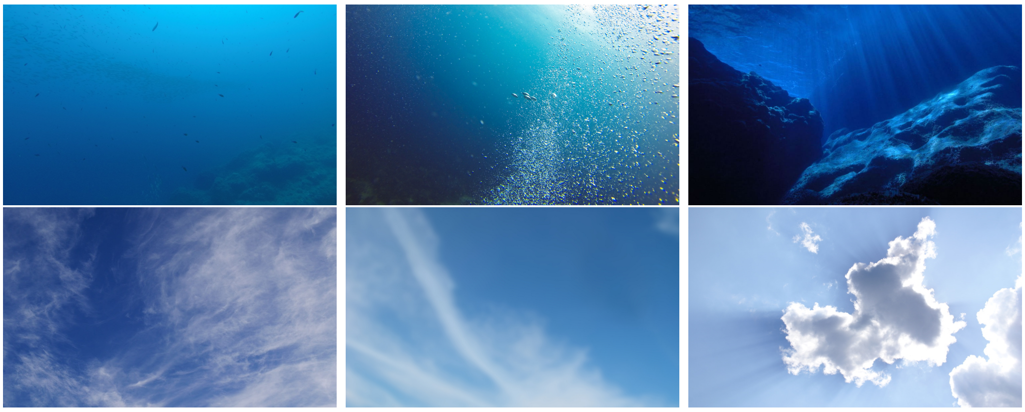


Figure 3.6: Background options presented to participants in Tasks 3 (Top row) and Task 4 (Bottom row). Left: Likely to fail, Center and Right: Likely to succeed.

3.1.5 Participants

We recruited 40 participants (15F, 25M) from the university participant pool which includes university staff, students and the general public. Anyone who expressed interest was allowed to participate in the study, so long as they did not identify as having technical hobbies or interests (e.g. computer programming), were not in technical employment (e.g. lab assistant) and were not technically educated (e.g. no degree in computing or engineering related subjects). Participants were also required to have normal or corrected to normal vision. Each participant received a £10 payment for their participation. Of the 40 participants 29 reported to be in education and 11 in full time employment. Participants' backgrounds were diverse with the most common being Business & Economics (13) followed by Social Sciences (9) Law (5), Languages (5), Art (4), Accountancy (2), Medicine (1) and Geography (1). One participant was aged between 40 and 49 years, 6 between 30-39 and 33 between 20-29.

Ten participants were randomly assigned to each condition. For conciseness, we will refer to participants by condition and subject number, for example, K7 was subject number 7 of the Keypoints condition. The other prefixes "N", "M" and "S" were refer to the No-Feedback, Matching-Keypoint and Split Screen conditions respectively.

3.2 Results

3.2.1 Data analysis: choices and processes

We analysed data through a combination of quantitative and qualitative methods.

In the **quantitative findings** section below, we assess the effect of feedback across the conditions, where three researchers independently coded participants' responses to questions (taken from researcher notes and transcripts of audio recordings) pertaining to their (1) background selections and (2) whether this choice was based on a participants' understanding of how the system works. In particular, a participant's response was coded as "correct understanding" if they described how the presence of distinctive shapes and features in the background positively impacted the app's ability to align frames. For example, the following statements were coded as demonstrating a correct understanding: "*I think it picks up the shapes on the picture and it [. . .] then compares the position of the dots on the other one [. . .] the next picture? So it can tilt the frame accordingly*" (K9) or "*because the background is distinct enough*" (N6). If a participant reported motives not connected to the requirements of the app or their understanding of what is significant was incorrect they were coded as "incorrect understanding". For example, the following statements were coded as demonstrating an incorrect understanding: "*Because it's nice and colourful*" (N8) or "*[. . .]it looked more homogeneous than the other ones. So I thought [. . .] it would be easier to take the photos like this*" (K2). To compare participants' understanding between the conditions we consider the total number of answers. Because the collected data represents a count, we found the chi-square to be an appropriate statistical test for analysing the data.

We report a further analysis of the data through broader, more general coding in the **qualitative findings** section, where transcripts of all audio recordings and researchers' notes collected during the studies were also independently coded by three researchers in a second round of analysis. Codes were initially drawn from research questions and then supplemented with those that emerged from the interviews before being grouped by consensus. In the subsequent subsections we report on the quantitative and qualitative findings.

3.2.2 Quantitative Findings

Table 3.1 summarizes the background selections made by participants in Task 3 and Task 4 and Figure 3.7 shows whether their selection was based on a correct understanding of the stabilisation processes.

	Task 3 (max=10)	Task 4 (max=10)
No-Feedback	10	10
Keypoints	7	10
Matching-Keypoints	10	10
Split-Screen	9	10

Table 3.1: No. Participants who selected a "correct background" i.e. suited to the needs of the app.

To compare participants' understanding between the conditions we consider the total number of answers which demonstrated a correct understanding in Task 3 and Task 4 (Figure 3.7). For example, 7 of the 10 participants in the Split-Screen condition demonstrated a correct understanding in Task 3 and 9 participants in Task 4, giving a summed value of 16. A chi-square test of the summed values revealed a statistically significant difference (chi-square=8.33, p=.040, df=3, Cramer's V=0.323). To better understand the differences between the conditions, we analysed the chi-squared standardized residuals (presented in Table 3.2). It can be noticed that the standardized residuals are larger (in absolute value) for the Keypoints and Split-Screen conditions, suggesting that these two conditions explain the significance of the chi-square test. A chi-square test also shows no statistically significant differences for correct background selections (chi-square=6.316,p=.097,df=3), nor when testing the tasks individually[5]. It should be noted that participants sometimes selected a 'feature-rich' background for aesthetic reasons rather than because it would make the app work better (as instructed), failing to demonstrate correct understanding. In the next section we discuss our qualitative findings and the role of background selection further.

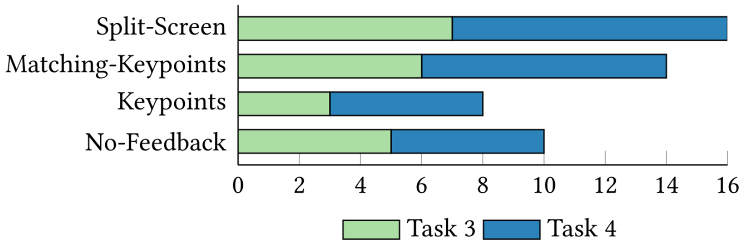


Figure 3.7: No. Participant responses coded as "correct understanding" when reporting their motivation for background selection in task 3 and task 4.

	Count (max=20)	Expected	Std Residual
No-Feedback	10	12	-0.6
Keypoints	8	12	-1.2
Matching-Keypoints	14	12	0.6
Split-Screen	16	12	1.2

Table 3.2: Standard residual results of the No. participants who demonstrated a "correct understanding".

3.2.3 Qualitative Findings

Codes that emerged from the interviews and grouped by consensus. In the subsequent subsections we detail these groups and give example quotations. First however, we would like to note that overwhelmingly participants reported the task to be interesting and entertaining. This suggests that the experimental task was sufficiently engaging and participants were invested in creating animations successfully.

[5] understanding on Task 3: chi-square=3.509, p=.320, df=3, Cramer's V=0.296; understanding on Task 4: chi-square=5.812, p=.121, df=3, Cramer's V=0.381; correct selections on Task 3: chi-square=6.667,p=.083,df=3; all selections were correct in Task 4, so no statistical test needed

3.2.3.1 Participants drew from their existing knowledge

First we note, that when asked about previous experience with computer vision applications, participants mentioned QR Code scanning, Facebook and Instagram (none of which provide visual feedback). No participants reported using Amazon or Bixby's search by image, or any other application which provides keypoint feedback.

In the No-Feedback condition, half of the participants demonstrated a correct understanding. These participants explained that having elements in the background which were "*more detailed*" (N1), "*most defined*" (N7), "*distinct*" (N6) or "*prominent*" (N2) would help the app because they were good reference points for alignment. The remaining five participants had an incorrect understanding and in the main focused on the aesthetics, e.g "*I thought the clouds would go really well with [. . .] the hot air balloon*" (N9).

Interestingly, participants in the No-Feedback condition selected a correct background more often than participants in the Keypoints condition (Table 3.1). Participants K2, K4 and K8 of the Keypoints condition made associations between the keypoint markers and their experience of other applications, suggesting that the keypoint markers functioned in much the same way as the auto-focus on digital cameras, in that they highlight regions on which the camera is focusing. Whether these analogies are helpful is not clear. One of the participants who drew such parallels made good choices when selecting backgrounds, while the remaining two were misled by their assumptions - K2 for example, chose a feature poor background for Task 3, expecting that a plain background would make it easier for the app to identify the character.

3.2.3.2 Early stage keypoint marker feedback is not easy to understand

Participants of the Keypoints condition broadly failed to understand the meaning of keypoint markers and how it related to low-level features of interest to the algorithm (30% demonstrated a correct understanding in Task 3 and 50% Task 4). Participants K1, K2 and K3 incorrectly thought that the keypoint markers were highlighting regions where the algorithm had identified a moving object, something the user intended to animate. These participants theorised that if the algorithm succeeds in finding the objects which are meant to move, then the algorithm will be able to successfully transform the captured images to create animations e.g. K2 said "*these dots might help show that the focus of the photo is the [character] [. . .] if I have these dots around the [character] then the image will be clearer*". K2 and K3 both selected the worst background option for Task 3. They justified their choice by saying that among the three options the plainest background would work best because it would make the identification of the character easier for the algorithm e.g. K3, when asked why they chose a plain background in Task 3, said it was "*because [the app] could be confused about the subject of the picture*". Both K2 and K3 expressed confusion when keypoint markers appeared in locations which did not fit their understanding of how the system works i.e. on the background instead of the character. K2 remarking: "*[keypoint markers] try to capture the [character] in the photo, a balloon, [. . .], but it's not on the balloon*" and K3, "*[if keypoint markers] mean the [character] is moving, [. . .] I don't understand why [keypoint] markers are showing up on the cloud, not the [character]*". Despite witnessing evidence to the contrary both participants failed to correct their misunderstanding, a behaviour pattern previously reported in work on intelligent system Tullio et al. (2007).

3.2.3.3 When keypoint marker feedback was helpful

The quantity of the keypoint markers was the most commonly reported explanation of how participants took into account Keypoint feedback. For example, K1 explained that if "*[. . .] in background, [I] see a lot of dots. I can tell that background is definite. When I did the [animation of the] plane [for which the app failed], there were only 1 or 2 dots*". K6 stated that "*if there is nothing [in the background], it's not going to work. [If] something is there it's going to work*". However, only four participants demonstrated a better understanding which was consistent with the workings of the stabilisation process. These participants noticed how and where the keypoint markers appeared and were able to develop more specific theories of how the algorithm identifies keypoint markers within an image. For example, K10 correctly speculated that the algorithm "*pick[s] up the shape*" and "*areas of heavy contrast*".

In the Matching-Keypoint condition, six of the ten participants reported the feedback to be helpful. Of these participants, three described the keypoint markers as indicators, reporting what the algorithm was doing: "*I can see what the dots are surrounding. [...] I know what it's doing*" (M10), "*when I saw [keypoints markers], it was more reassuring [...] saying you're doing it right*" (M7), and "*the app is trying to match between images [...] things which the app sees in this image which it also saw in the previous image*" (M1). The other three participants explained that they saw the keypoint markers as guides, that the keypoint markers were designed to help them test if the background image would work or not: "*the dots showed if the picture would work out*" (M6), "*I can tell what's the problem of the image*" (M8) and "*[the keypoints] might help you pick a background*" (M5).

Participants in the Keypoints condition tended to overestimate the meaning of the Keypoint feedback and relate the meaning to higher level concepts, such as the separation of background and foreground objects. In this regard Matching-Keypoints appeared to be more intuitive as its meaning is more inline with user expectation. M1 for example, reported that when the app didn't work in Task 2 he did not know why. During Task 3, he speculated that the colour might have an effect (lighter or darker colour), but found through experimentation that this was not the case. He then correctly theorised that the app needed distinct features. He explained, "*The dots meant like it's picking distinct points throughout the image. [...] I think [the app is] re-mapping the points that [it had] taken in an image before. I think that's what it's trying to do*".

3.2.3.4 Split Screen feedback was helpful, but not in the way we expected

Seven participants in the Split-Screen condition also reported the feedback to be helpful. Four participants suggested that it acted as a cue, indicating when best to capture a frame e.g. "The preview helped me decide when to take a picture" (S7) or "*I [wait] for the preview to stabilize before taking the picture*" (S3). An artifact of the stabilisation processes implementation is a "flickering effect" which occurs when the system is rapidly toggling between a successful transform and a failure. This strictly speaking is a usability "bug" which participants reappropriated, using it as a means of gauging the likelihood of a successful transform e.g. "If it was flickering I wouldn't take the picture" (S7), and "*I waited for a clear picture [. . .] then hit capture*" (S4).

Another unexpected way of using Split-Screen feedback was described by two participants (S7 and S2). They used the feedback to position the camera in the same place as the previous image, S7 commenting "*the preview tells me what angle to take the picture from*". Both participants would keep moving the camera until the left and right images matched in the preview i.e. the alignment transformation was minimal. This approach does in fact help make better quality animations, however it is not how the app was intended to be used and this process of positioning was very time consuming for the participants.

3.2.3.5 When feedback was unhelpful

Five participants in the Split-Screen condition and three in the Matching-Keypoints condition reported the feedback to be distracting or unhelpful. For example, "*I found the split screen very distracting and would rather not see it*" (S4), "*I found the dots distracting because it ruined the focus at times*" (M4), "*They were a bit annoying, they get in the way*" (M1) and "*they could be obstructive*" (M6). Interestingly, S6 described the feedback as unhelpful because they prefered to frame the photo from memory, using the viewfinder to align the camera with features they had identified in the background. To this end the preview was unhelpful because the split screen design reduced the size of the viewfinder. These comments illustrate the risk that feedback visualisations can be distracting.

3.2.3.6 Background selection motivation

Although all participants selected a correct background in Task 4, not all provided a correct explanation. Participants responses when asked why they chose the background image they selected in Task 3 and Task 4 were coded into one of two categories: aesthetic - they were motivated by how the image looked, and detail - where they stated in some way that the level of detail was important (including incorrect understandings). Aesthetics was the primary motivation for 27 selections out of 80 (10 No-Feedback, 9 Keypoints, 5 Matching-Keypoints and 3 Split-Screen), with detail accounting for the remaining 53 selections (10 No Feedback, 11 Keypoints, 15 Matching-Keypoints and 17 Split-Screen). It should be noted that it is by chance that some of our participants considered the correct background to be more aesthetically pleasing.

3.3 Discussion

At the beginning of this chapter, we set out a series of questions. In this section we discuss the outcomes of our study in light of these questions.

3.3.1 Does the processing stage from which feedback is derived impact user understanding?

Our results indicate that feedback derived from the later stages of the processing pipeline (Matching-Keypoints and Split-Screen) are more effective at informing users' understanding. The chi-square test of "user understanding" reveals a significant difference between conditions, with the standard residuals indicating the Keypoints and Split-Screen are responsible. More participants of the Split-Screen condition demonstrated a correct understanding of how the system works than participants of any other condition (Figure 3.7), with Matching-Keypoints second. In contrast, participants in the Keypoints condition performed worse than participants who received no feedback at all.

Despite users understandings varying between conditions, most participants across all conditions were successful in selecting a correct background (see Figure 3.1). As mentioned above, participants sometimes selected the correct background for aesthetic reasons, rather than to make the algorithm work (as requested by the study instructions). As a consequence, instead of using selection as a measure of understanding, we rely only on the participants' explanations of *why* they selected a specific background.

3.3.2 Is keypoint marker feedback intelligible to lay-users?

More participants in the Matching-Keypoints condition were able to correctly describe the input requirements of the system in comparison with those who received no additional information in the form of feedback (No-Feedback). Interview responses indicate that users have a tendency to interpret feedback as an outcome rather than a progress notification of an intermediary stage. In this regard Matching-Keypoints appeared to be more intuitive, as their meaning is more inline with user expectation. We tentatively propose that keypoint markers can be used to inform user understanding, so long as the meaning being conveyed is inline with user expectations.

3.3.3 Can keypoint markers mislead if misunderstood?

Given that the Keypoints and Matching-Keypoints conditions utilise exactly the same feedback visualisation (keypoint markers), the result showing that Keypoints condition participants were least able to understand the needs of the algorithm (Figure 3.7) suggests that they may have been detrimental to user understanding. While the keypoint markers are a good indicator of the future stabilisation processes success, participants commonly understood them to represent the final output, that they represented regions where the stabilisation process had identified matches. It is feasible that this misconception could result in users using the markers in ways which inhibit their interactions. Indeed, Keypoints condition participants' interview responses indicate a disconnect between their interpretation of feedback and the actual information conveyed e.g. K3, "*[if keypoints] mean the [character] is moving, [. . .] I don't understand why keypoints are showing up on the cloud, not the [character]*".

3.3.4 Can keypoint markers improve usability and aid users' interaction?

The inherently visual nature of computer vision processes, both in their input and also the intermediate stages, makes visual feedback the logical medium through which to deliver feedback Kato et al. (2012). However, participants in our studies, at times reported the feedback to be distracting or obtrusive (e.g. M1 "*They were a bit annoying, they get in the way*"). This highlights a design tension between attracting attention and causing distraction, and between being informative and not overwhelming. These tensions are well understood in graphic design, particularly around the design of interactive visualisations. However, the situation here is more complex. Some aspects of algorithm design are conceptually simple and naturally map to visual representations. Keypoints for example, are a concept that lend themselves to being represented pictorially e.g. by marking their physical location with geometric points. It could at first be tempting to see this as an example of “form follows function” Sullivan (1896), however when dealing with the design of feedback for systems which employ pattern matching algorithms, we argue that the “form follows function” principle requires careful interpretation. What is “function” in this case? At first, it may seem to be the “technical” function of the algorithm, but this is not the case. We need to remind ourselves that the “function” is instead the function to help users understand what the system does. One implication then, is that to design feedback, it may be beneficial to distance oneself from the question of how algorithmic steps and internal states map to form, and instead think about the end result of the system and how it will be used. Moreover, in some cases, it may be challenging, or even impossible, to map the function of the algorithm to form.

3.4 Summary

This chapter reported a comparative between-groups lab study examining the role of visual feedback in smart camera apps. Leveraging a novel experimental design centred on the creation of stop-motion animations, 40 participants were exposed to four different levels of feedback. Through a combination of quantitative and qualitative methods, our findings indicate a disconnect between user expectations and the information actually represented by the feedback. In particular, they show that keypoint markers can help users in building a better functional understanding of computer vision processes as long as they are derived from a stage of processing that is inline with user’s expectations (R1), where in our study, participants who received keypoint marker feedback derived from later stages of processing demonstrated an improved understanding of the system operation (R2). Conversely, participants exposed to keypoint marker feedback derived from early stages of processing showed a tendency to misunderstand it and overall they performed worse than participants who received no feedback at all.

Because CNNs became the most common algorithm for Computer Vision applications, the thesis focus shifted to that class of algorithms. As discussed in the literature review chapter, saliency maps emerged as one popular form of explanation for such algorithms. Therefore, in the next chapter, we start by discussing our rationale in choosing saliency maps to study. We then report on our first study examining the role of saliency maps in informing user understanding.

Chapter 4

Evaluating the role of saliency maps

In the previous chapter, we looked at how the stage of the processing pipeline from which feedback is obtained affects users' capacity to form a coherent and correct understanding of how systems work. In particular, one implication of this study is that when designing an explanation for users who are not experts in complex algorithms, we should seek an explanation that provides a "functional understanding" of the system and how it will be utilised rather than describing the system's inner workings.

The prior study explored the research question in the setting of a classical CV algorithm (i.e., pattern recognition based on keypoints matching). In Chapter 2, we mentioned how Convolutional Neural Networks (CNNs) are now the most widely used algorithm in computer vision. Thanks to their ability to process spatial structures of the images, CNNs can be efficiently trained to recognise images without explicitly instructing the algorithm on what input parameters to use; a processing stage often called "feature engineering" which involves a careful selection of best input parameters that helps the algorithm to achieve its goal (e.g. classification). In other words, in CNN, these parameters are learned directly from the data.

While the superiority of CNNs makes them the most popular technique, the large number of parameters used to build their models makes them challenging to interpret. Thus, many techniques have been proposed to help explaining these complex models. However, what would be an explanation that provides end-users with a "functional understanding" that may be helpful?

As we mentioned in Section 2.4.2, a popular approach to trying to make CNNs explainable is to produce “saliency maps” (also called “heatmaps”) that highlight which pixels were most important for the image classification algorithm. The claim is that such explanations are easy to interpret by both novice and expert users (Ribeiro et al., 2016a).

Despite the fact that numerous algorithms have been published to generate such saliency maps from CNNs (Linardatos et al., 2021), little research effort has been reported regarding their assessment with users (Narayanan et al., 2018b; Yin et al., 2019b). This observation led us to run a series of user studies that investigated the role of saliency maps in informing user understanding of the CNN models. In section 2.6.2, we reviewed the works that evaluated saliency maps through user studies, highlighting the research gaps and formulated the following research question:

(R3) How do saliency maps help with building functional understanding, including the relation to varied system confidence?

In this chapter, we report on two user studies to investigate this main research question. In both studies, we seek to measure the impact of saliency maps on users' functional understanding by asking them to predict the CNN classification outcome of an image (we call it the "*task image*") and count the number of correct user's predictions.

The first is a pilot study: *Study 2-pilot* (Section 4.2), where for each task image, participants were encouraged to use an interactive browser that allowed them to explore a large dataset of examples and examine how CNN classified the different images. Findings of this pilot study indicate that there were wide variations in data exploration strategies across participants. Such variation points us to the need to design a study with more constraints to ensure that the effect of showing the saliency map is genuinely evaluated. Therefore, we designed another study: *Study 2* (Section 4.3), where instead, for each task image, we chose to present a few examples to participants to reduce the possible noise that may emerge from users' diverse data exploration strategies.

In addition, because task images can be sampled from various areas of the input space where the CNN performs differently, in the study reported in Section 4.3, we would also like to investigate whether saliency maps help with building functional understanding, in light of varying CNN's classification outcomes?. In other words, are higher scores easier to predict than lower scores?. And how do saliency maps contribute to this?.

In the sections that follow, we begin by describing the materials utilised in both studies, then report on the *Study 2-pilot* in Section 4.2, and discuss *Study 2* in Section 4.3.

The work detailed in this chapter and Chapter 6 form the biases for the following paper in submission:

Ahmed Alqaraawi, Enrico Costanza, Nadia Berthouze and Emma Holliday. Evaluating and Improving Heatmap Explanations for CNNs through to online user studies. ACM Transactions on Computer-Human Interaction (TOCHI).

4.1 Materials

4.1.1 Dataset, CNN Model Architecture and Training

Various public datasets, algorithms and configuration options exist for the multi-class image classification problem. In both studies, we used the PASCAL Visual Object Classes dataset (19714 images where each may contain around 1-3 objects), because of its popularity, and its limited number of classes (20). Additionally, we used the Keras library for Python, starting from an existing Keras model trained on the ImageNet dataset Deng et al. (2009), utilizing the VGG16 architecture Simonyan and Zisserman (2014)[1]. We then fine-tuned the model on the train-val part of the PASCAL VOC 2012 dataset Everingham et al. (2012), achieving an Average Precision (AP) score of 0.7 on the validation-set. On a hold-out test-set (the PASCAL VOC 2007 test data Everingham et al. (2007)), the AP was also 0.7 [2]. This performance could have been improved further by several techniques such as augmentation. However, we chose to work with an off-the-shelf model simulating a realistic case where a domain expert (with minimal ML expertise) would use explanation techniques. In addition, such a model should produce a sufficient number of cases where the model struggles to recognise an object in an image, providing us with the enough false positive and false negative examples for the user study.

4.1.2 Saliency maps Generation

A variety of algorithms have been proposed for generating saliency maps. In our pilot studies, we investigated two popular implementations: LIME (Ribeiro et al., 2016a) and LRP (Bach et al., 2015b). Unlike LIME, with LRP, saliency maps are not restricted to a super-pixel (neighbouring patch of similar pixels) but highlight contours of objects, which was preferred by most of our pilot study participants (perhaps because they do not occlude the objects). For this reason and to simplify our study design, we chose to focus on the LRP algorithm only to create saliency maps. Concretely, we used the α-β propagation rule (Bach et al., 2015b) with $\alpha = 2$ and $\beta = 1$.

Figure 4.1 shows a **true positive (TP)** example, where the model correctly predicts a train. The saliency map suggests that the red part of the image containing the rail supports the classification of this image as a train. Figure 4.2 shows what we may refer to as **false positive (FP)** example where the system incorrectly assigned a train label. The red part of the image contains what *looks like* a rail is what supports the classification of this image as a train. The blue parts are against this classification.

[1] https://keras.io/applications/#vgg16

[2] For a reference, in the Visual Object Classes Challenge 2012 [3], the AP for the winner team was around 0.82 on a hold-out set.

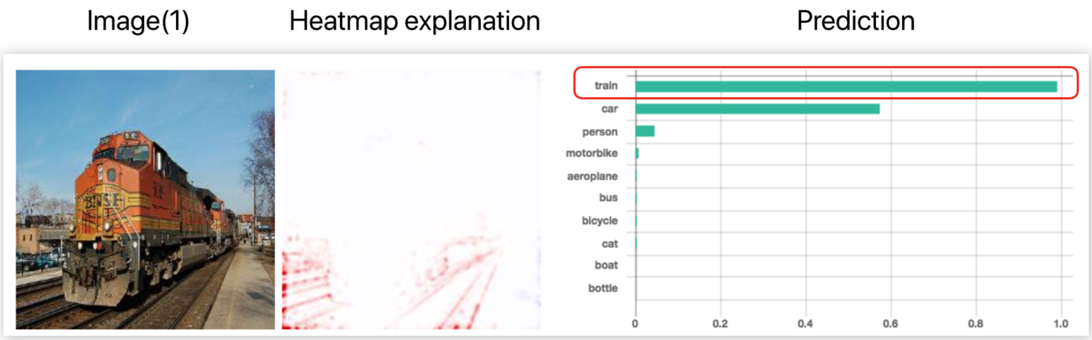


Figure 4.1: Example of a saliency map explanation of a True Positive (TP) image for the label *“train”*. It highlights the contours of the lines below the train. A possible interpretation is that the CNN has learned to recognise trains when rails are present.

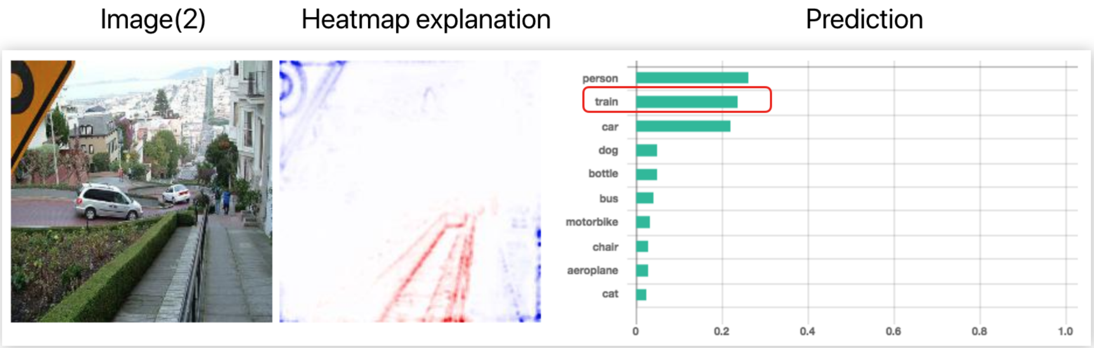


Figure 4.2: Example of a saliency map explanation of what we may refer to as a False Positive (FP) image for the label *“train”*. A possible interpretation is that edges in the lower part appeared similar to rails, which could explain this error.

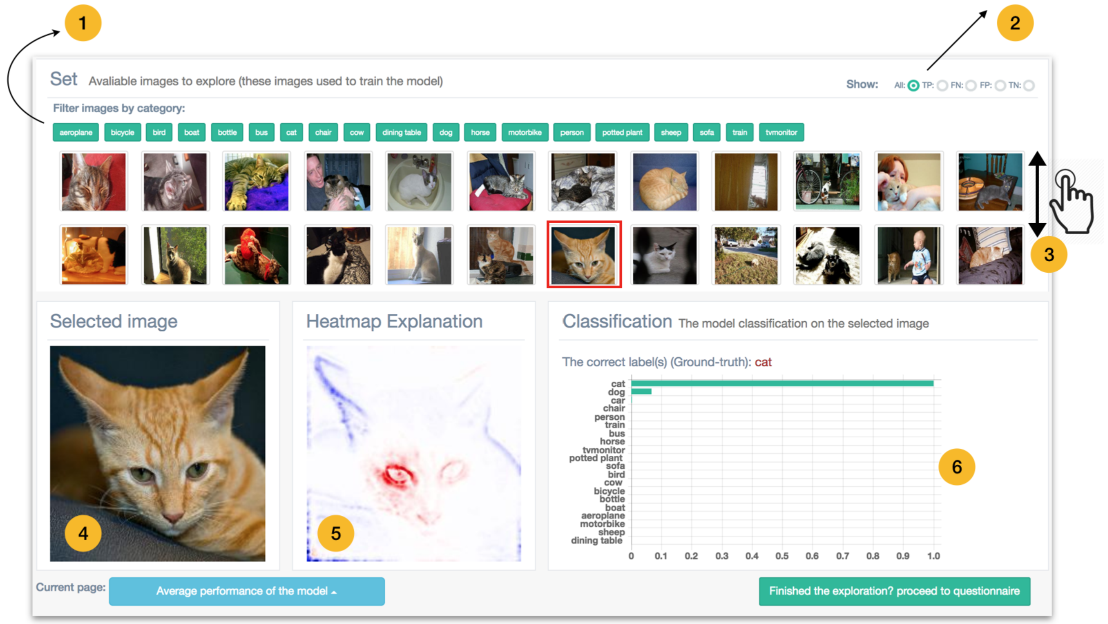
4.1.3 Presentation

The interfaces of the study (Figure 4.6 was implemented as a Web application, using HTML5 and Python with the Django framework. We served the application from a standard Web server. The view-port of the participant browser window was required to be at least a 1000px wide and 600px high, in order to take part in the study.

4.2 Study 2-pilot: A preliminary investigation of the role of saliency maps

In this pilot study, we designed a between-group study to evaluate whether saliency maps can help users understanding of a highly complex CNN used for multi-label image classification. Participants were presented with a pre-trained CNN and asked to estimate the outcome of the model for 10 *task* images. For each question, participants were encouraged to use an interactive browser (details below) to examine how the CNN classifies a large set of training images.

The study included one independent variable that varied between groups which is the *presence of saliency maps*. A screenshot of the experimental setup is shown in Figures 4.3 and 4.4.


Set Avaliable images to explore (these images used to train the model)

Show: All: ● TP: ○ FN: ○ FP: ○ TN: ○

Filter images by category:

Selected image

Heatmap Explanation

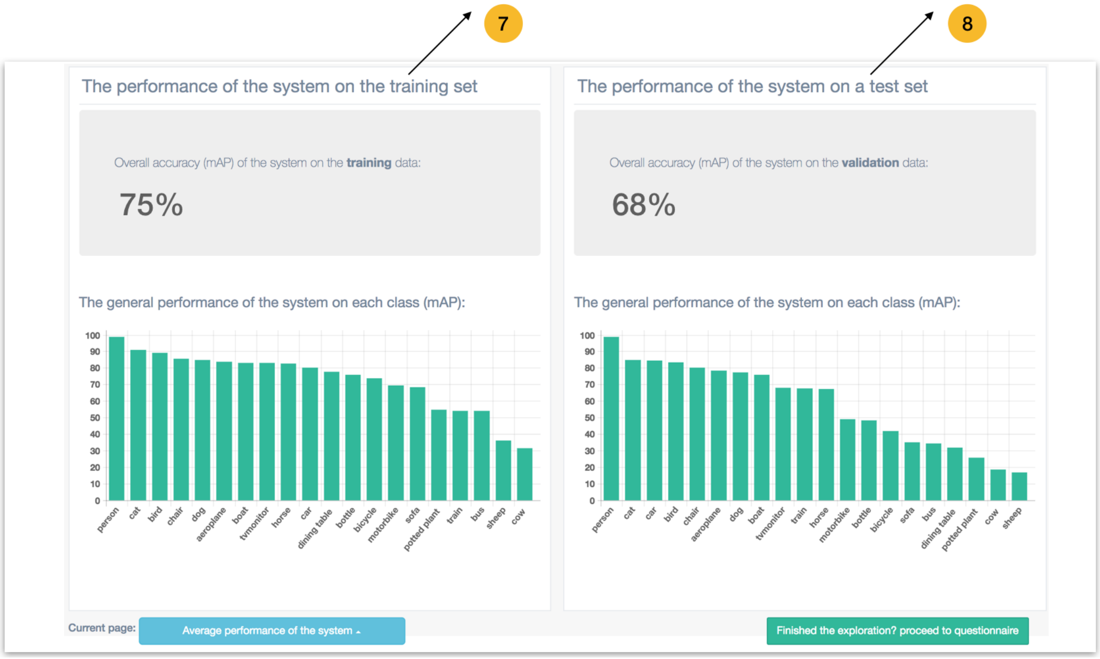
Classification The model classification on the selected image

The correct label(s) (Ground-truth): cat

Current page: Average performance of the model

Finished the exploration? proceed to questionnaire

Figure 4.3: Exploration page 1: a tool to navigate the dataset


The performance of the system on the training set

Overall accuracy (mAP) of the system on the **training** data:

75%

The general performance of the system on each class (mAP):

The performance of the system on a test set

Overall accuracy (mAP) of the system on the **validation** data:

68%

The general performance of the system on each class (mAP):

Current page: Average performance of the system

Finished the exploration? proceed to questionnaire

Figure 4.4: Exploration page 2: the overall performance of the model

4.2.1 Conditions

Each participant was assigned to one of the following two conditions: **No-saliency map:** each training image was displayed along with a bar chart showing the probability score of each category. **saliency map:** An explanation (a saliency map) for each training image is displayed in addition to the bar chart.

4.2.2 Participants

20 participants were recruited from the university participants pool which includes university staff, students and the general public. Each received a £10 payment for their time, as well as an additional performance-based bonus of £0.5 for each correct answer as an incentive. Anyone above 18 years of age who expressed interest was included in the study, so long as they have a technical background (i.e. degree in computing or engineering), had normal or corrected to normal vision and fluent in English.

4.2.3 Procedure

All studies were conducted in person in the same empty meeting room on the UCL campus. At the beginning of the study, all participants went through a brief tutorial that included background information on the experiment as well as step-by-step instructions on how to utilise the system. The tutorial includes a demonstration of how the system classifies a specific image, as well as how the classification scores should be interpreted. The definitions of TP, FN, and FP are also provided. The saliency map group is given extra information in the tutorial that describes (with examples) the saliency map explanation and how it should be interpreted (see for example the saliency map displayed in figure 4.3). In both conditions, participants spent a few minutes to familiarise themselves with the interactive browser. They were then asked to answer ten questions. Each question asks if the model will recognise an object (for example, a horse) in a new image. Participants were encouraged to utilise the interactive browser as a tool to assist them answer the question accurately.

In addition to being asked if the system would correctly recognise an object (e.g. horse) in a task image, participants are verbally asked to justify their choices at the end of the study (i.e. why did you pick this choice for this particular question?). This question allows us to get more specific information on their decision-making strategy. It also helps us understand whether exposing the saliency map to users affect their interpretation of the model performance.

The set of images used in the questionnaire are only sampled from two classes: horses and monitors, where the first 5 images are sampled from the ”horse” category, while the last 5 are sampled from the ”monitors” category.

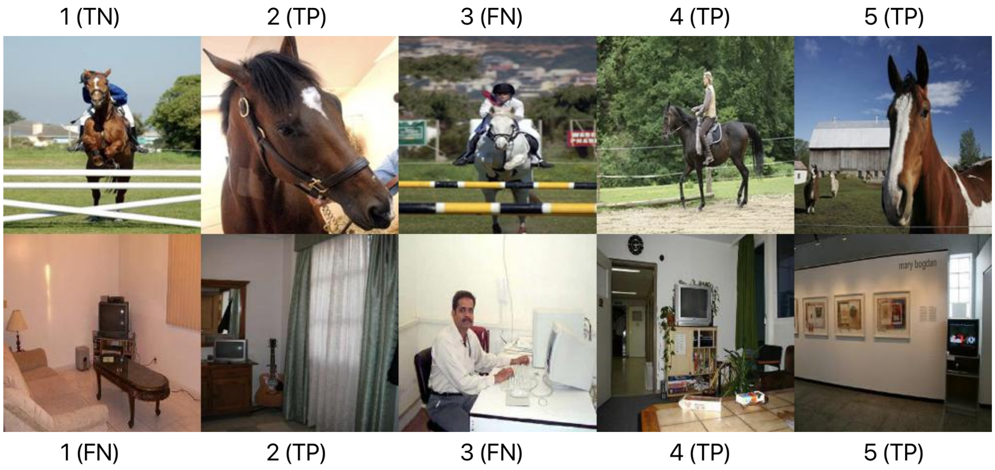


Figure 4.5: Query images used for the study

4.2.4 Results

The quantitative assessment is based on counting the total number of correct answers out of 10. A chi-square test of the overall scores revealed no statistically significant differences between conditions (chi-square=0.526, p=0.46, df=1). The average score for participants in the saliency map condition was ($\mu = 0.67$, $\sigma = 0.24$) and for No-Saliency map ($\mu = 0.65$, $\sigma = 0.22$).

Further to the quantitative results, a thematic analysis was performed on the collected qualitative data. Four themes emerged from this process. Each theme is described in one of the following subsections. In what follows, we use the letter S to indicate that a participant was in the Saliency map condition, and the letter N to indicate the No-Saliency map condition.

4.2.4.1 Variations of exploration strategies

In both conditions, participants' comments reflect a variety of exploration strategies. several participants gave more weight to where images belong to in terms of performance measures (e.g., TP, FN, or FP) (S2, S6, S8, N1, N5, N8, N9, N11).

Some participants dismissed The FN examples, believing that such a metric is irrelevant or ineffective in making a decision. Following a discussion, those participants revisited this metric (N7, S2). The FP examples were ignored by the other participants (N10). In certain circumstances, a participant would disregard the saliency map for some questions (S5) or rely solely on the visuals while ignoring the metrics (N8, N10).

In other cases, the overall performance of the system played an important role in participants' answers (S8, S13, N9, N12). For example, S10 reported: "*I put (No) for Q10* *when I realised that I have many Yes's for monitors, and the model is not perfect*". Many participants considered distractions from other objects (S6, S14, N6, N8, N9, N11). For example: "*for Q7, no, because it's distracted by other objects, guitar, the mirror which has a similar shape*" (N8).

Some comments showed that participants did not always notice certain details: "*what about this one?, oh .. seems that I didn't notice it*"(S2). Or that properly answering some questions is due to chance since there are similar images in the dataset that are classified differently (i.e. TP, FN, or FP), but they were not seen to that particular participant (S10, S2, S5, N1, N7, N8).

4.2.4.2 Building a pattern

When exploring the examples, a number of participants commented on how difficult it was to establish a pattern. Some participants stated that they became confused as a result of finding identical images that were categorised differently (N3). In some extreme cases, participants expressed their dissatisfaction and viewed the exploration process as a total guessing game (N5, N8, N10)."*[...] I can't work out why it's not picking up on the horse in the FN, to me it seems very random. I can't seem to determine, what features of the horse, the system is using to categories the horse*"(N10).

In other comments, some participants explained how challenging it is to build patterns (N1, N3, N11). For example, a participant reported that in order to answer a question, one must go back and forth between TP, FN and FP examples in order to build a pattern: "*I'll check TP and try to find the most important feature for horse (e.g. brown colour), and validate that with FN, as you can see the system misses some black and white horse images*" (N1).

4.2.4.3 Attention and understanding the saliency map

Quantitative results show a comparable user performance for both groups (i.e. with and without saliency map). A cause for this effect could be that some participants did not pay attention to the saliency map. Several comments seem to highlight this limitation in design. For example, one participant ignored the saliency map from the beginning and argued that the role of saliency map is minor given that the metrics are provided (S12). In other cases, participants stopped relying on the saliency map after noticing that the saliency map has picked up things that do not belong to the object of interest (S5, S14). One participant reported that he was only using the saliency map for FN examples (S8).

4.2.4.4 Confusion and other biases

Comments in this category relate to some wrong assumption about the system. Sometimes the comments highlight that some participants overestimated the system capabilities: "*[...] did you also check FN for that image? No, I also had high confidence in the system, [...] usually the system learns from its mistakes*"(S8). Similarly, a participant assumed that the system has an online learning feature which can learn from a few new data points (S3). One participant thought that the saliency map "*has also its own negatives, it can identify things that are not relevant*" (S5). Although saliency maps' purpose is to reflect what the model has learned from the data (i.e. to serve as a diagnostic tool), the last comment demonstrates that some participants may overlook the fact that the model can also learn incorrect features, which the saliency map would then "illogically" highlight. To put it another way, the saliency map can be "truthful" by highlighting the wrong part of the image to reflect the model's flaws.

4.2.5 Discussion

4.2.5.1 Guiding users' interpretation of Saliency maps

The system provides users with a number of features, including overall scores, probability scores for particular cases (bar chart), and a saliency map. This diversity has a cost, which includes inconsistencies among individuals during the exploring phases. To accurately assess the impact of a saliency map as an additional feature, it is necessary to ensure that participants are paying attention to the saliency map and are aware of how to utilise it.

An implication for design is to consider techniques that draw users' attention to the details of the saliency map. Previous research, for example, shows that individuals typically describe one event in relation to another event; a strategy based on contrasting Miller (2019b). In the context of saliency map explanation, users' attention to the saliency map may be directed by contrasting two comparable examples that have been classified differently by the model, such as two similar horse pictures, one correctly identified (TP) and one misclassified by the model (FN). Users may learn more by comparing the two images than by looking at each saliency map individually.

4.2.5.2 Designing more constrained studies

The explorer enables users to easily navigate the dataset by category (for example, cat or horse) and filter out based on classification criteria (i.e. TP, FN and FP). Despite these characteristics, we observed that participants use a variety of exploratory strategies to explore examples. Due to this inconsistent nature of explorations, each participant was exposed to a different set of examples, which appears to have influenced their judgement.

For example, if we knew that the model correctly classified an image, individuals who discovered a similar example from the TP category during the exploration phase would be considered favoured. This observation holds true for both conditions. Being exposed to saliency maps would also be ineffective, as one acknowledged limitations of saliency maps is that they are regarded instance-level explanation Krause et al. (2018b). An implication for design then is to find strategies to reduce this potential noise by limiting the possible exploratory variances between participants. One method is to sample a small portion of the dataset that represents the most useful images from which to learn for a specific task image. This perhaps may assist improve the efficiency of saliency maps as well as the design of user studies based on them.

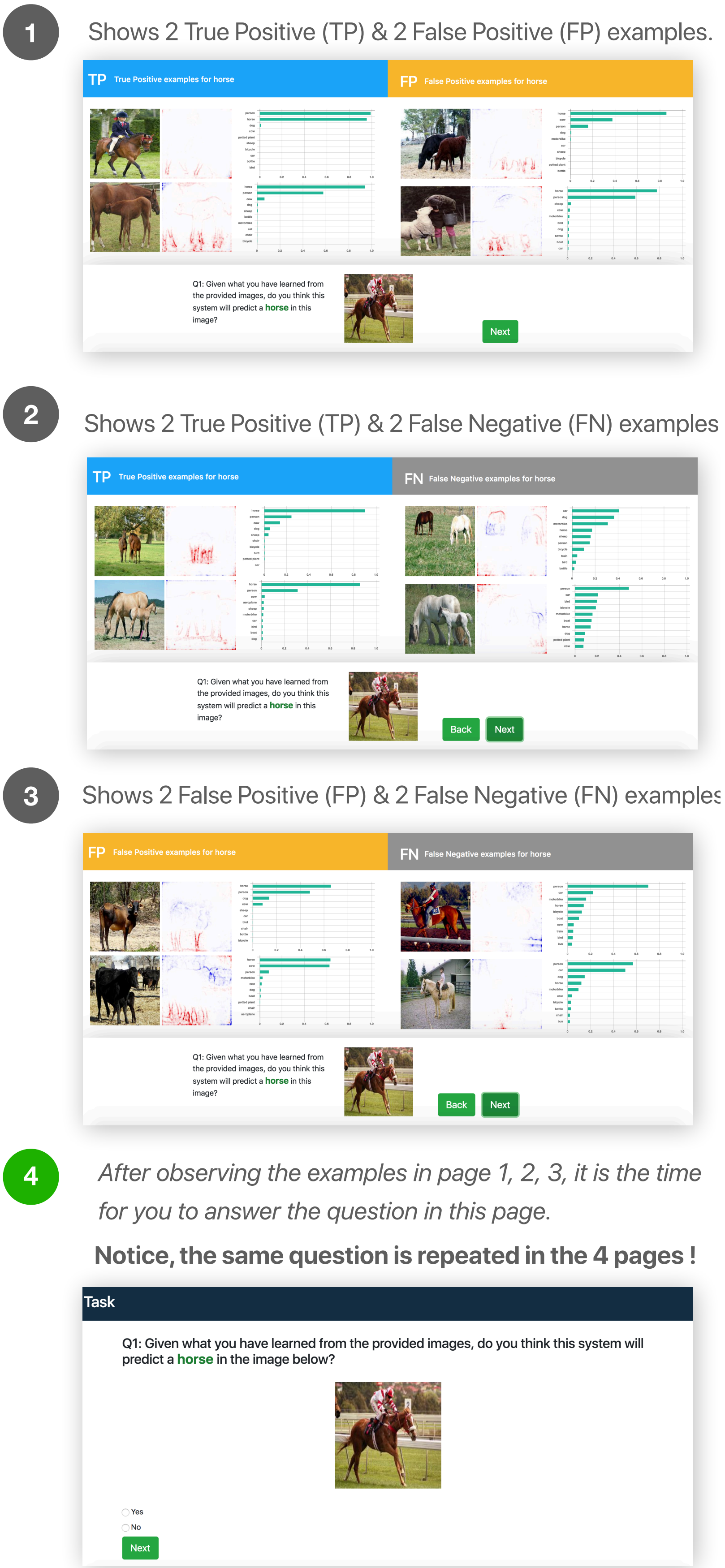
4.3 Study 2: Evaluating the role of saliency maps

The results of the pilot study reported above showed that participants' data exploration strategies differed greatly. Such variations suggests that a study with more constraints is required to ensure that the effect of displaying the saliency map is evaluated correctly. Therefore, we designed another a between-group online study in which we sampled and presented a few examples for each task instead of allowing participants to explore the whole dataset. A screenshot of the experimental setup is shown in Figure 4.6. In the following sections, we lay out a more elaborate description of the study.

4.3.1 Tasks

The main task was to predict the classification outcome of a CNN for a set of 12 task images. This task has been proposed in prior work to evaluate the explainability of a system (Lipton, 2018). In addition, for each task, participants were asked to justify their choice (i.e. why did you pick this choice). This question helps us gather more detailed data about whether exposing the saliency map to users affect their interpretation of the system and how they use the saliency map. Figure 4.6 shows the interface for one task image.

To increase participant engagement in the study, participants received a performance-based bonus of £0.50 for each correct answer in addition to an £8 payment for their time,



1
Shows 2 True Positive (TP) & 2 False Positive (FP) examples.
TP True Positive examples for horse
FP False Positive examples for horse
Next
2
Shows 2 True Positive (TP) & 2 False Negative (FN) examples
TP True Positive examples for horse
FN False Negative examples for horse
Back
Next
3
Shows 2 False Positive (FP) & 2 False Negative (FN) examples
FP False Positive examples for horse
FN False Negative examples for horse
Back
Next
4
Task
○ Yes
○ No
Next

Q1: Given what you have learned from the provided images, do you think this system will predict a **horse** in this image?

After observing the examples in page 1, 2, 3, it is the time for you to answer the question in this page.

Notice, the same question is repeated in the 4 pages !

Q1: Given what you have learned from the provided images, do you think this system will predict a **horse** in the image below?

Figure 4.6: A demonstration of how examples and question are displayed to a saliency map participants. The system will show few examples of how it classifies some images, where for each task, the system shows the depicted 4 pages. The same information, excluding saliency maps, are presented to no-saliency map participants.

4.3.1.1 Selection of task images

We intended our study to be no longer than 40 minutes to avoid fatigue effects. This design choice limited the possible number of task images. Consequently, we had to choose between sampling from a variety of classes or sampling from a subset of classes. In our pilot studies, participants found predicting model behaviour very confusing when the class in question was continually switching. Furthermore, the more classes they had to reason about the more challenging the tasks became, because they were not able to "learn" much about the model's behaviour regarding a specific class. We also wanted to capture a variety of cases where the model had given correct as well as incorrect output. For these reasons, we decided to limit our experiment to four classes but included one TP, one FN and one FP for each class.

Task images were evenly sampled from the following four categories: *horse, cat, car, bus*, where the order of these categories is counterbalanced between participants to ensure there is no order effect. For each task image, participants were shown 12 example images from the CNN training set to inform their judgement.

The study also aimed to examine if there is a difference in the participants' performance depending on the CNN classification outcomes (i.e., are task images with high CNN scores easier to predict than the ones with lower scores?). Therefore, we set the sampling strategy of task images as an independent variable in the study with two levels: either task images were randomly sampled with a high CNN classification score or with a with a medium CNN classification score. For the high CNN classification score sampling strategy, the image with highest score was taken from each of the TP, FN, and FP clusters. To sample images with a medium CNN classification score, the midpoint was found within the positive certainty range and images with the closest score were sampled. This midpoint is calculated by finding the half-way point between 1 and the acceptance threshold (in this study, the threshold is 0.1, thus, the midpoint is 0.1 + (1 - 0.1)/2 = 0.55).

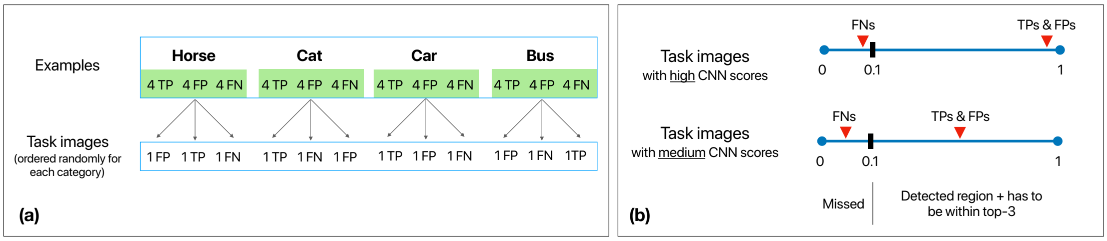


Figure 4.7: A demonstration of the sampling procedure. (a) How examples and task images were sampled. A total of 15 images were sampled from each category, 12 were used as examples, while the rest 3 were used as task images, this results in 48 example images (12 x 4 categories) when images were sampled with a high CNN score and another 48 images when example images were sampled with a medium CNN score. (b) High and medium CNN score images are sampled around these locations.

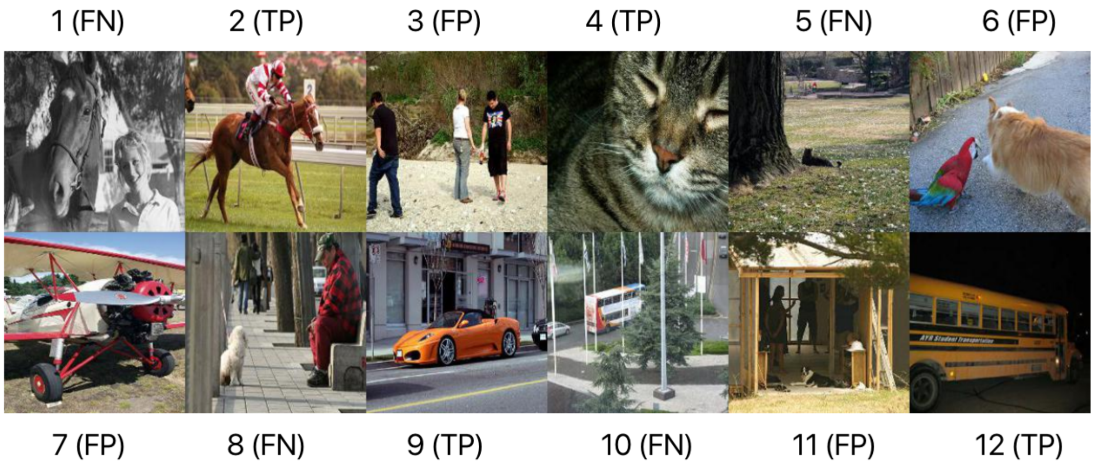


Figure 4.8: The 12 task images used in the study when they were sampled with a high CNN score

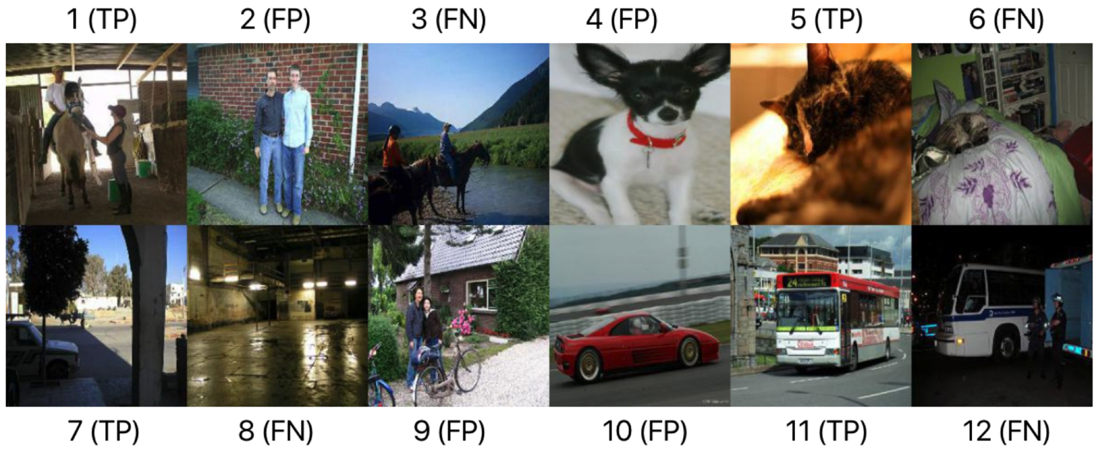


Figure 4.9: The 12 task images used in the study when they were sampled with a medium CNN score

4.3.1.2 Selection of example images

Example images were sampled for every task image from the PASCAL dataset, based on their score similarity to the task image. In particular, a total of 15 images were sampled from each category, 12 were used as examples, while the rest 3 were used as task images (questions). Figure 4.7 demonstrates how examples and task images were selected with a high CNN score sampling strategy, resulting in a total of 48 example images, and 12 task images. Similarly, a different 48 example images, and 12 task images were used with the medium CNN score sampling strategy. The assumption was that user understanding might benefit from looking at instances where the model assign a similar score. Moreover, showing the outcome of the classifier (i.e. TP, FN and FP) for the examples has been found to be important for the utility of explanation techniques (Lai and Tan, 2019). For this reason, we sampled the most similar examples (in terms of score) of different outcomes for each task image as follows:

- 4 examples of True Positives (TP), where a label had been correctly assigned;
- 4 examples of False Negatives (FN), where the CNN had failed to assign the label;
- 4 examples of False Positives (FP), where the CNN had incorrectly assigned the label.

We also based our decision, regarding the number of shown examples, on experience from pilot studies. We had noticed that if we presented too many examples, participants were likely to only look at a random subset of them. At the same time, if the number was too low, there was a risk that not enough information was made available to participants. For this study, we selected 12 as a compromise. The definition of how the system accepts the predicted labels (as a binary outcome) are clearly explained in the tutorial before the tasks begin (please refer to Section 2.1), therefore, although examples show continuous scores, participants should understand how these scores maps to the binary outcome: accepted or not accepted.

4.3.2 Conditions

The study included the following two independent variables:

Presence of saliency maps This factor had two levels: shown or omitted. When shown, the saliency map for the relevant class was displayed next to each example image. It is important to note that saliency maps were not shown for the task image but only for the examples.

Sampling strategy This factor also had two levels: sampling task images with a *high* CNN score and sampling task images with a *medium* CNN score.

These two variables bring out the following four conditions, where each participant was exposed to only one of these conditions (i.e. between subject study design):

- Saliency maps not shown and images were sampled with a high CNN score.
- **Saliency maps shown** and images were sampled with a high CNN score.
- Saliency maps not shown and images were sampled with a medium CNN score.
- **Saliency maps shown** and images were sampled with a medium CNN score.

Because classification scores produced by the CNN are the default sources of explanatory information on the instance level, a bar chart of the top 10 classification scores was displayed next to each example image for all conditions.

4.3.3 Participants

64 participants (16 per condition) were recruited through Prolific [4], an online crowdsourcing platform. For the sake of data quality, we required participants to have an approval rate above 95% on the Prolific Academic platform, have normal or corrected to normal vision, and to be fluent in English. Moreover, we also made it mandatory for participants to be above 18 years of age and to have a technical background (i.e. a degree in computing or engineering), because of the technical concepts used in our study (i.e. neural networks, classification outcomes, scores, image pixels).

4.3.4 Procedure

After providing informed consent, each participant went through a short tutorial (Section B.2) providing the necessary background about the experiment as well as clear instructions for using the system. The tutorial included examples of how the model classified a specific image. Further, we provided participants an information about the explanation technique and how they can be interpreted.

Upon completion of the introduction, participants commenced completing their 12 tasks. Participants may navigate between examples linked with each task, but once they reach page 4 (Figure 4.6), they cannot return. The rationale of this design choice was to control the time each participant spend on examples (i.e., avoiding that one participant (out of attitude) would spend lots of time going back and forth and another would not). On page 4, participants chose whether they thought the image would or would not be classified with the given label. Although there was no time restriction for viewing examples, participants were told that the study should take no longer than 40 minutes. Because prior work (Lascau et al., 2019) pointed out that participants in online studies tend to multi-task, working in parallel, which suggests that task completion time may not be a reliable measure, so we do not report it in our online studies. At the end of the study, we gave them feedback for each task images and showed them the bonus earned.

4.3.5 Results

4.3.6 Data analysis: choices and processes

We analysed data through a combination of quantitative and qualitative methods.

In **Section 4.3.6.1** below, we first evaluated participants ability to predict the CNN score by counting the total number of correct answers per condition. Because the collected data represents a count, we found the chi-square to be an appropriate statistical

[4] https://prolific.ac/

test for analysing the data. We further applied a post-hoc analysis (with a Bonferroni adjustment) to understand the pairwise effect between conditions.

Besides making a prediction, for each task image, we asked participants why they expected the system to succeed or fail in recognising an image. Therefore, in **Section 4.3.6.2**, we carried out a qualitative content analysis (Braun and Clarke, 2006) on the free text replies. A main theme that emerged from the data is the reference to features. Therefore, we focused our analysis on coding participants' answers in respect to the mentioned features.

In particular, in the first pass, two researchers coded the answers inductively. Each response could be assigned several open codes (e.g. nose, low contrast) based on the features or concepts it addressed. Subsequently, coders discussed their individually established codes and agreed on a shared and simplified codebook. We decided to assign each code to one of two code groups: **Saliency-Features** and **General-Attributes**.

The **Saliency-Features** group included codes referring to features which could be localised to pixels in the proximity of the object of interest and that saliency maps *could* highlight. The rationale for this was that we aimed to compare how frequently participants mentioned concepts related features that saliency maps *could potentially* highlight. Besides the somewhat obvious feature codes such as *Ears* and *Legs*, this group also included: *Outline* which applied to answers referring to the *"shape"* or *"contour"* of the object of interest and *"Fur"* which was used for utterances referring explicitly to the *"fur", "skin"* or texture pattern *on* the animal.

The *General-Attributes* group included codes that refer to utterances of generic properties. An example is the code *Background* - which applied to answers referring generically to *"surroundings"* or *"context"*. Another example is *Image Quality* which was used for replies addressing issues of *"contrast"*, *"blur"* or *"lighting condition"*. Finally, *Generic* was assigned when the code refers to other concepts that are not directly related to the image (i.e. *threshold*, *number*, *easy*).

To compare participants' answers between conditions, we also applied the chi-square test on the total number of times (i.e., count) a participant mentioned Saliency-Features (F) or General-Attributes (G) and whether the provided answer was correct (C) or wrong (W) for all conditions.

Finally and further to the focused coding process, in **Section 4.3.6.3**, a thematic analysis (Braun and Clarke, 2006) was performed on the qualitative data collected in the study.

In the subsequent subsections we report on our findings.

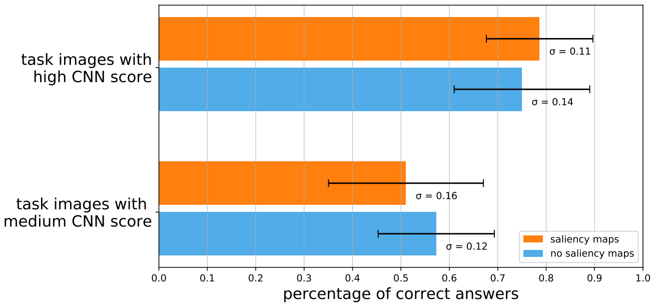


Figure 4.10: Percentage of correct answers for each condition. For both sampling strategies (i.e high CNN scores and medium CNN scores), the presence of saliency map showed no statistically significant difference in terms of the outcome prediction accuracy

4.3.6.1 Participants ability to predict the CNN score

A summary of the analysis is shown in Figure 4.10. A chi-square test of the overall scores revealed statistically significant differences between conditions (chi-square=45.8, p=06.15e-10, df=3). We ran a post-hoc analysis to understand this significance, where in that case, the Bonferroni-adjusted p-value is equal to 0.05/6 or 0.0083. In terms of the sampling strategy, when the sampling was for images with a high CNN score, participants were more accurate in predicting the outcome of the classifier (Saliencymap-top vs Saliencymap-middle: chi-square=30.9, p=**2.7e-08**, df=1) and (No-Saliency map-top vs No-Saliency map-middle: chi-square=12.7, p=**0.0004**, df=1). However, the test revealed no statistically significant differences between conditions within either sampling strategy; neither when images were sampled with a high CNN score (Saliency map-top vs No-Saliency map-top: chi-square=0.526, p=0.46, df=1), nor when sampled with a medium CNN score (Saliency map-middle vs No-Saliency map-middle: chi-square=1.27, p=0.26, df=1).

When images with a high CNN score were sampled, the overall average number (regardless of the explanation condition) of correct answers was 9.15 out of 12, while it was 6.47 when images with a medium CNN score were sampled. As the only difference between the two variations is the sampling strategy, this difference indicates that different parts of the input space are more easily predictable than others. Furthermore, the saliency maps did not seem to aid performance. In fact, the performance of Saliency map participants was worse when images with medium CNN scores were sampled (Figure 4.10). However, no statistical significance exists.

4.3.6.2 Mentioned saliency maps features

Figure 4.11 shows the percentage of different codes mentioned by participants when task images were sampled with top CNN scores, while Figure 4.12 shows them when task images were sampled with medium CNN scores.

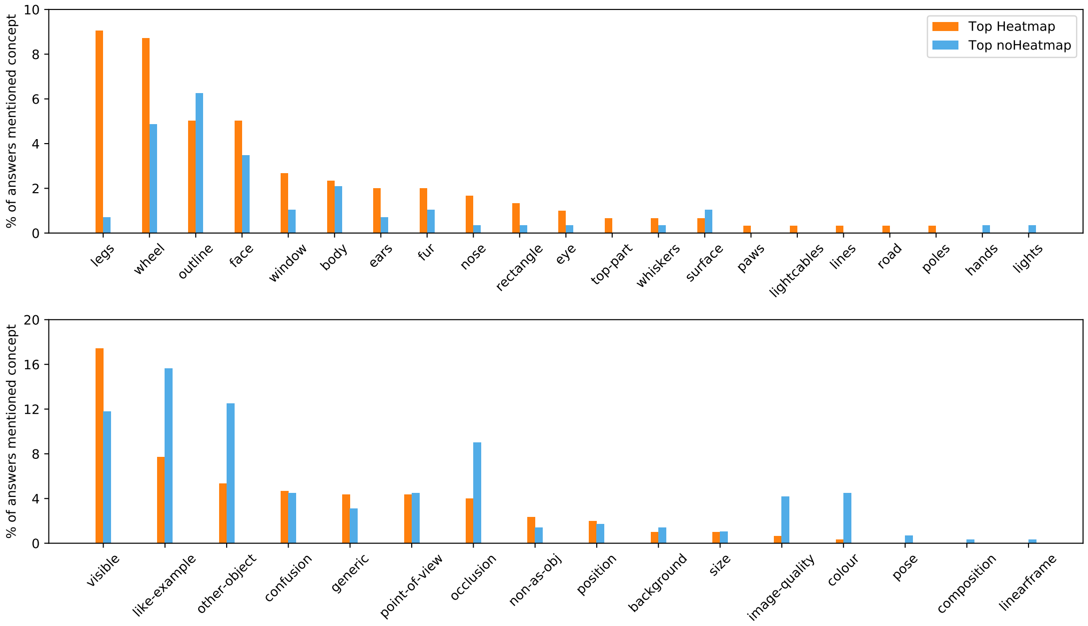


Figure 4.11: Frequencies of individual attributes mentioned by participants when task images were sampled with **high** CNN scores. Top: Features belonging to the Saliency-Features. Bottom: belonging to the General-Attributes (frequencies were averaged over all participants in that condition).

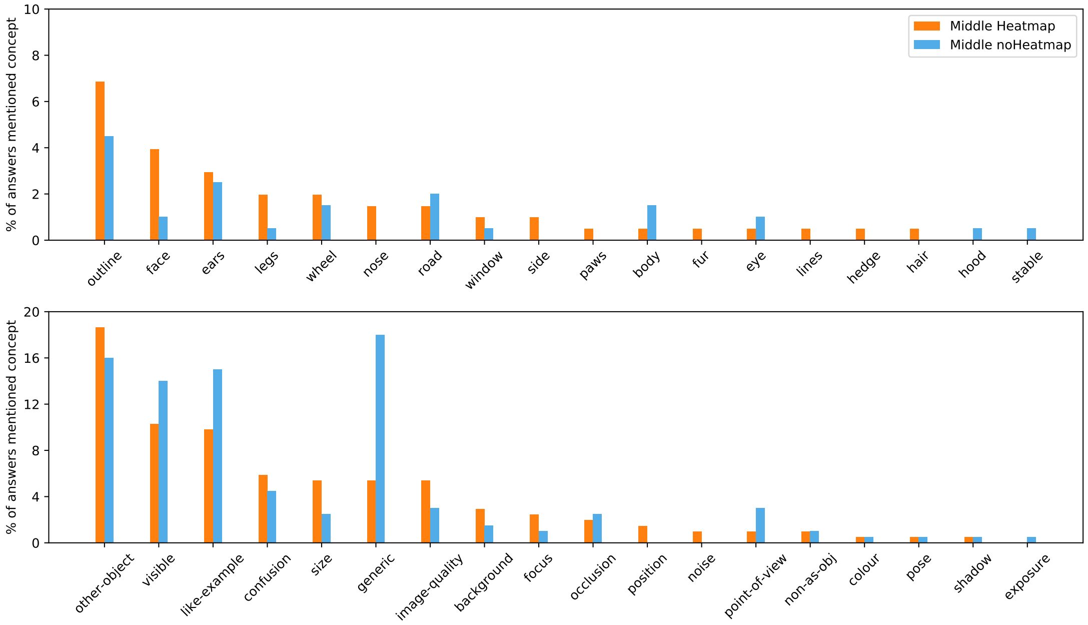


Figure 4.12: Frequencies of individual attributes mentioned by participants when task images were sampled with **medium** CNN scores. Top: Features belonging to the Saliency-Features. Bottom: belonging to the General-Attributes (frequencies were averaged over all participants in that condition).

To compare participants' performance between conditions, we consider three variables: Condition (Saliency map, No-Saliency map), Answer (Correct, Wrong), and Code (Saliency-Features, General-Attributes). For conciseness, we will refer to these sets by the following: "S", "N" refer to "Saliency map" and "No-Saliency map", "C", "W" refers to "Correct" and "Wrong", and "F", "G" refers to "Saliency-Features" and "General-Attributes". Also, we will refer to participants by condition and subject number, for example, S7 was subject number 7 of the Saliency map condition.

Images with a high CNN score Table 4.1 summarises the scores for each of these sets, the same data is visualised (as a ratio) in figure 4.13a. A chi-square test of the scores revealed a statistically significant difference (chi-square=20.86, p=0.00011, df=3). Results suggest that there is an association between the distribution of correct/wrong answers for Saliency map participants and the distribution of correct/wrong answers for No-Saliency map participants. This significance can be explained by the standardised residuals (presented in Table 4.1). It can be noticed that the standardised residuals are larger (in absolute value) when considering the correct answers only, (*i.e. F-C and G-C*). For our analysis, the Bonferroni-adjusted p-value is equal to 0.05/6 or 0.0083, which **clearly suggests that for correct answers, Saliency map participants often mention a feature, while No-Saliency map participants do not** (chi-square=18.4, p=.0000176, df=1).

Condition	F-C	G-C	F-W	G-W
Saliency map	101	50	16	23
Std Residual	4.40	-3.61	-0.167	-1.15
no-Saliency map	59	84	17	31
Std Residual	-4.40	3.61	0.167	1.15

Table 4.1: The number of answers (freq) of Saliency-Features (F) or General-Attributes (G) and whether the provided answer was correct (C) or wrong (W) when task images were sampled with high CNN scores

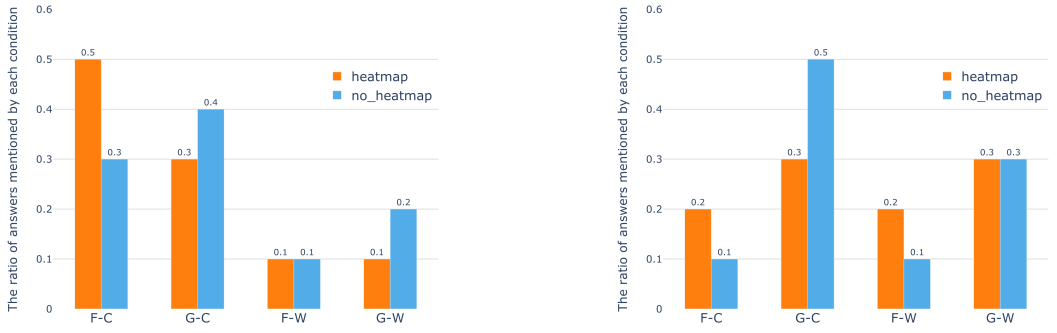
Images with a medium CNN score The same analysis was performed on the data collected from the two conditions showing images with a medium CNN score. Table 4.2 summarises the qualitative scores for each condition, the same data is visualised in figure 4.13b. A chi-square test of the scores revealed a statistically significant difference (chi-square=11.299, p=0.0102, df=3). Results suggest that there is an association between the distribution of correct/wrong answers for Saliency map participants and the distribution of correct/wrong answers for No-Saliency map participants.

This significance can be explained by the standardised residuals (presented in Table 4.2). It can be noticed that the standardised residuals are larger (in absolute value) when considering the correct answers only, *i.e. F-C and G-C.* For our analysis, the Bonferroni-adjusted p-value is equal to 0.05/6 or 0.0083, which **suggests that for correct answers, Saliency map participants tend to mention a feature more than No-Saliency map participants** (chi-square=8.42, p=0.0037, df=1).

Condition	F-C	G-C	F-W	G-W
Saliency map	37	59	31	65
Std Residual	2.25	-3.087	0.846	0.8424
no-Saliency map	21	88	25	57
Std Residual	-2.25	3.087	-0.846	-0.8424

Table 4.2: The number of answers (freq) of Saliency-Features (F) or General-Attributes (G) and whether the provided answer was correct (C) or wrong (W) when task images were sampled with medium CNN scores

Overall, the data indicates that for correct answers, saliency map participants often rely on Saliency-Features, while participants in the no saliency map condition do not. However, the effect of the saliency maps falls short of causing a difference in the number of correct answers between conditions.



(a) Sampling task images with high CNN scores

(b) Sampling task images with medium CNN scores

Figure 4.13: The ratio of Saliency-Features (F) or General-Attributes (G) and whether the provided answer was correct (C) or wrong (W) for both conditions

4.3.6.3 Qualitative Analysis

In what follows we use the letter S to indicate that a participant was in the Saliency map condition, and the letter N to indicate the No-Saliency map condition. For images we use the letter T to indicate that the image was sampled from an area of the input space with high CNN scores (Figure 4.8), and the letter M to indicate that the image was sampled an area of the input space with medium CNN scores (Figure 4.9). Six themes emerged from the thematic analysis process, each described in one of the following subsections.

Mis-classification Comments in this category relate to the expectation that the system may mis-classify, or confuse an object for another. Sometimes the comments refer to specific features that may cause mis-classification such as mentioning that the squares in the image form a shape similar to a bus shape (S4, image T11), or that the system may incorrectly classifies a photo of a plane as a car because of the appearance of wheels (S2, image T7). Other times they refer to the number of items in the photo: *"Since there are two elements (even if they're humans), the system recognises them as horses"* (N18, image M2). Some comments are more generic such as expecting the system to see dogs as cats as in (N2, image T6 and N17, image M4).

Other objects The presence of objects other than the one being classified positive is sometimes reported as potentially helping the recognition, in terms of context. For example a stable was mentioned as context for a horse, a house as context for a cat, a road as context for a car or a bus, or a "car workshop" as context for a car: "*Looks like a car workshop which can be lead to car recognition.*" (N22, image M8).

Other times the presence of other objects was reported instead as a cause for failing to recognize an object, beyond the issue of mis-classification. For example, S24 expected the system to incorrectly see a car if there are people in the image, but assume that is not the case with the presence of a bicycle. Along the same lines, the absence of other objects was also sometimes put forward as a reason for the system to successfully recognize an object: "*Because a bus is similar to a car and there aren't other things on the picture*" (S21, image M10). When referring to T3, N10 thought there is nothing in this image that might look like a horse to the system, and justified this assumption by the observation that all of the false positive images contained at least a non-human animal. Similarly, for the same image, N5 thinks the system needs to see some kind of animals in the picture before classifying it as a horse.

Image Qualities A number of comments referred to the quality of the image. Sometimes participants referred the entire image being clear (S11, image T9) or unclear (S8, image M12). Other comments in this category referred to specific characteristics, such as brightness (N16, image T2 and S25, image M8), blurriness (N11, image T8), colour contrast (N30, image M3) or "*Because of the shape of the bus is very clear with the black background*" (N31, image M12).

Some of the comments referred to specific portions of the image: "*[...] this picture has a car in a shadowed area so I think the system will struggle.*" (S26, image M7). In some of the comments image quality is provided as the only reason why participants expected the system to recognize, or fail to recognize the object. Other times, image quality is mentioned in reference to specific features such as mentioning the clear appearance of the shape and the wheels of the car (S14, image T9). Finally, sometimes poor image quality is mentioned as a challenge that the system can overcome: "*Although the image is dark, the contrast between the bus and the background makes the bus stand out, making it easier to be identified*" (N14, image T12).

View The distance of objects ("*Cat is too far away*" S10, image T5) and the zoom level ("*It appears to be at the right zoom level to be recognised. Seems to have an issue with those too close or too far away*" - S32, image M11) were reported as factors that would determine the success or failure of the system. Similarly, occlusion was mentioned as a reason for the system to fail such as the insufficient visibility of the car (S16, image T8), and conversely participants refer to entire object being visible as a reason for success (N26, image M3). Also, the viewpoint from which the object was captured such as having the horse facing the camera (N32, image M6), and the posture (e.g. a "*cat lying down*") were also reported as influential.

Like-example In the study, examples were the main source of information for our participants (in addition to the preliminary information provided in the introduction). So it is perhaps not surprising that a number of comments referred explicitly to the image provided as examples. In some instances, participants referred to general similarity of a photo to a specific examples (N11, image T10 and S21, image M8). In other instances, it was reported that a photo was similar to an entire group of examples: "*this image is similar to all TPs examples*" (N17, image M5).

Sometimes, the comments referred to specific aspects of the examples, such as colour and viewpoint. For example, the majority of the true positives, according to N15, were yellow buses, similar to the one in T12, and one of these buses was practically in the same sideways posture. Similarly, for T8, N6 referred to a previous case in which the system failed to recognise a car because the shape was not properly displayed, or the object (cat) being on the foreground (S24, image M4). Sometimes the similarity to examples was weighted in the judgement, in conjunction with other factors, for example: "*Horse if fully visible with good lighting on it, also it reminds me of the first TP example.*" (N7, image T2). To a more extreme extent, occasionally participants refer to the example, but offer contrasting conclusion: "*it is similar to some false negatives but I think the shape is not as much hidden.*" (N31, image M7).

Generic Comments Under this category we grouped comments that quite generically state that the object to be classified appears in the image. For example, S31 and N5 referred to the appearance of an object (M11 and T4), while S31 and N17 mentioned the absence of an object (M2 and M9). N24 expressed uncertainty on the recognition of a car in M8.

Sometimes participants mentioned aspects of the system, such as the threshold, which was mentioned in the introduction and visible in the histogram examples: "*I think it resembles a bus enough to be above the threshold, but not very high*" (N3, image T10). Other times the probability and ranking of the images were mentioned ("*The system will recognise the car in this image and probably the bus category would be in the top three*" - N27, image M10, or "*The probability of identifying the bus will be second after the probability of identifying a person.*" - N19, image M12), or the saliency map ("*Because the saliency map of the first example is very similar to the photo*" - S30, image M11).

4.3.6.4 Contrasting Images

In addition to the thematic analysis reported above, we also looked for *contrasting* points in the data, by considering the following question: are there images, where most of the participants in one condition provide the correct answer, while most of the participants in the other condition offer the wrong answer?

When images with a high CNN score were sampled, the first to observe are the answers where most of H participants mentioned a feature and provide a correct answer, while most of the N participants mentioned a feature but provide a wrong answer (i.e. the intersection between *H-F-C and N-F-W*). T1 (figure 4.8) is an example that demonstrates this observation. For that image, 12 H participants answered this question correctly, mentioning the reason for the system to mis-recognise the horse is that *legs* are not visible. On the other hand, 4 N-participants (N1, N6, N7, N8) incorrectly answered *Yes*, making a reference to the visibility of the *horse face*. It is worth mentioning that the saliency map for horse images seems to highlight that the model gives more weight to the legs of the horse (figure 4.6) which is missing in T1. Similarly, in T6, The saliency map suggests that *fur* is a feature that is considered by the model. Some N-participants (N6, N14) rely on *face* or *ears*, but did not consider *fur* as a feature for cats. 7 H participants mentioned *ears* or *fur* as features and answered the questions correctly. On the other hand, 2 participants incorrectly answered the question, referring to the existence of another object, e.g. "*There is more than one item in the image, and that might confuse the system*" (S4).

It is also worth observing the answers where most of H participants mentioned a feature and provide a correct answer, while most of the N participants does not mention a feature but provide a wrong answer (i.e. the intersection between *H-F-C and N-nF-W*). In T2 (figure 4.8), 12 H participants answered this question correctly, mentioning the reason for the system to recognise the horse is that *legs* are visible. On the other hand, 4 N-participants (N4, N9, N15, N12,) incorrectly answered *No* making a reference to the existence of a *person* in that image. Moreover, the 4 remaining H participants who answered this question incorrectly did not mention a feature, e.g. "*because the system fails to recognise other horses like this one*" (S48).

When images with a medium CNN score were sampled, there seems to be no clear pattern when we start looking closely into each question individually. This observation is inline with (figure 4.13b), which shows that there is less contrast between the two conditions compared to (figure 4.13a) .

4.3.7 Discussion

At the beginning of this chapter, we posed the following research question: **(R3) How do saliency maps help with building functional understanding?**. In this section, we will discuss the findings of our research in light of this question. We also reflect on the key issues, highlighting the implications for design and further research.

4.3.7.1 Can saliency maps improve users' understanding?

The saliency maps did not seem to aid performance. In fact, the performance of Saliency map participants was worse when images with medium CNN scores were sampled (Figure 4.10). However, no statistical significance exists. As a result, more studies are needed to properly characterise the impact of various sampling procedures on users' understanding of system operation. Furthermore, more research and analysis are needed to determine whether the existence of a saliency map has a negative impact on user performance when task images are sampled from portions of the input space with lower CNN scores.

In terms of information gained by reasoning on examples, many of our participants' comments reveal that through the examples, they could infer features of the objects that were used for the classification (even without the presence of saliency maps). For example, the presence of wheels seems to be correctly associated with the prediction of the 'car' or 'bus' classes. In this context, N18 suggests that the system would recognise the bus in image (M12) because of the "sideview and the wheels." However, in this photo, the wheel contrast with the black background is quite poor. While a person can infer wheels from the image context, it is less likely that a network would (indeed, this image is a false negative). One possible interpretation of this finding is that participants expect the CNN to recognise a horse or cat whenever they can recognise one. This interpretation could be an instance of the cognitive psychology notion of "attribute substitution," which is considered the basis for a number of cognitive heuristics (Kahneman and Frederick, 2002): participants might be "replacing" the difficult question "does the CNN recognise a horse/cat in this image?" with the easier question "do I recognise a horse/cat in this image?". Also, human expectation can give emphasis to some factors, such as the brightness of the image, even if it is not a major cause of failure for the model.

It is worth noting that if a specific model is resilient to the brightness of an image being dark, then the saliency map will highlight edges even if the contrast is not high (notice how the saliency map highlights the rail of the train in Figure 4.1). An implication for design is to run more user studies to assess whether saliency maps can spot these possible biases or ideally shift user expectations to the correct understanding, complementing the large spectrum of literature that supports this idea, as we mentioned in Section 2.2.

4.3.7.2 Saliency maps can help participants notice Saliency-Features

In this study, examples with the highest similarity score to the task image were selected. Given the observation that for correct answers, saliency map participants often mentioned Saliency-Features, then would selecting examples that share similar Saliency-Features help participants answer questions more correctly? As an implication of these findings, in the next study (Chapter 5), we utilise a unique property of CNNs which enables them to learn powerful descriptors of the data represented as a vector (often called embeddings), which can be used to find images that share similar patterns (that may resemble Saliency-Features).

Another point to make regarding features is the observation that when images were sampled from an area of the input space where they have a lower classification score, we still found a statistically significant difference (in terms of the number of mentioned features) between the Saliency map and No-Saliency map conditions, but overall, features are mentioned a lot less frequently than when images were sampled with a high CNN score (across both conditions). This finding suggests that the utility of saliency maps varies according to the classification score which is novel compared to other prior works, where the usefulness of the saliency maps is often demonstrated through visual inspection, as argued by (Adebayo et al., 2018), rather than through systematic sampling.

4.3.7.3 Tensions between local and global feedback

The saliency maps can highlight portions of the image that act as context for the object being detected. For example, referring to the same example we mentioned previously, the pixels corresponding to the train tracks in Figure 4.1 are highlighted as supporting the classification of this image as 'train'. The qualitative analysis revealed instances where our participants applied the same type of reasoning, suggesting an effect of a specific feature (see for example Section 4.3.6.3).

In Section 2.1, we demonstrated how CNNs respond to low-level patterns in the first layers and how more complex representations resembling concepts like "cat's eyes" emerge in deeper layers. This finding does not necessarily, however, imply that the network has learnt the specific concept of "cat's eyes." This discussion remind us of the tension between *local* and *global* AI explanations (Ribeiro et al., 2016b; Lipton, 2018). For example, a participant comment such as "*I think the system needs some kind of animal in the picture before it can say it is a horse*" (N5, image T3) suggests that this participant may expect the system to have a global label of "animal", yet we know that it is not necessary that CNN has an "animal" label. In fact, CNN may have actually learned a high-level representation of features that *looks like* "animal".

The complexity increases when we consider saliency maps. The generation process of an explanation represents a different process (that is related but not identical) to the forward pass process. In particular, saliency maps are local explanations in the sense that changing some pixels in the input image may result in a very different saliency map (Lipton, 2018). Yet a comment such as "Whenever there's a person in the image, it recognises the person before the horse." (S4) points to participants' expectations of 'concepts' understood and used by the system. An implication for design is how to develop explanations that convey the right expectation to users. When considering explanations generated for CNNs in particular, works such as (Kim et al., 2017) and (Hamidi-Haines et al., 2018) proposed techniques that attempt to bridge the gap between concepts defined by users and how these concepts are represented by the model.

The tension between local and global also exists at the image level (rather than in terms of the classifier input space). Saliency maps are defined on individual pixels. In contrast, some of the properties mentioned by our participants in relation to "Image Qualities" (see for example, Section 4.3.6.3) are global properties of the images, such as contrast and overall brightness. An implication for the design of explanation systems, then, is that saliency maps should be complemented by more global representations. For example, saliency information could be related to global descriptors of the images, such as overall contrast or brightness measures, as well as histograms.

4.4 Summary

In this chapter, we report on two between-subjects lab studies designed to investigate the role of saliency maps in informing technical users making sense of Convolutional Neural Networks (CNN). The first is Study 2-pilot and was conducted in person with 20 participants. They were asked to estimate the outcome of an object recognition algorithm, half of them were shown a saliency map of each image. No statistically significant differences were found between groups in terms of user prediction accuracy. Participants observations and post-study interviews point to the local nature of saliency maps as a key limitation, but they also indicate that there were wide variations in data exploration strategies across participants. Such variation points us to the need to design a study with more constraints to ensure that the effect of showing the saliency maps is genuinely evaluated.

Study 2 builds on the results of the pilot study by sampling a few examples to reduce the possible noise that emerges from users' variation in data exploration strategies. The study was conducted online with a total of 64 participants. They were asked to estimate the CNN outcome on task images, with 12 examples (each with a similar score) shown to help them make a decision. The study included four (two-by-two) conditions: (Saliency map, No-Saliency map X Images with a high CNN score, Images with a medium CNN score). Through a combination of quantitative and qualitative methods, we can summarise the key findings of this study in light of the research question as follows:

1. Considering participants' ability to predict the CNN classification outcome of images as one measure of users' functional understanding, the presence of saliency maps did not result in user's higher prediction accuracy.
2. Across conditions, higher scores seem to be easier to predict than lower scores. Moreover, saliency maps do not seem to help in that regard.

However, an interesting finding emerged from the qualitative data, which showed that for correct answers, participants tended to mention Saliency-Features more often than when they were not. This finding prompts us to consider a different sampling strategy based on locating and presenting images that share similar features. This strategy should yield examples that are visually similar to the task image (rather than selecting images with the closest score), which is the subject of the next chapter.

Chapter 5

Evaluating the role of saliency maps with visually similar examples

The work reported in this chapter has been published at the following IUI conference paper:

> Ahmed Alqaraawi, Martin Schuessler, Philipp Weiß, Enrico Costanza, and Nadia Berthouze. 2020. Evaluating Saliency Map Explanations for Convolutional Neural Networks: A User Study. In Proceedings of the 25th International Conference on Intelligent User Interfaces (Cagliari, Italy) (IUI'20). Association for Computing Machinery, New York, NY, USA, 275–285.

Previous study (Chapter 4) showed that the presence of saliency maps did not result in higher accuracy. One thing that we have noticed by "visual inspection" of many images is that the saliency map sometimes has a tendency to highlight specific parts of the object (e.g., the legs of the horse). Therefore, we were curious as to why saliency maps did not help participants and whether they actually paid attention to and used saliency maps for reasoning.

For that reason, in this study, we aim to prime participants to think and reason about features by asking them to list the features they think the system is sensitive to for each task image. Moreover, results from the previous study also indicated that for correct answers, participants in the saliency map condition mentioned Saliency-Features more frequently. Does this imply that providing images with similar saliency-features would provide them with more opportunities to reason about features, potentially leading to more correct answers?. This is what we are mainly seeking to investigate in this chapter.

Previous research demonstrated that the CNN embeddings extracted from the penultimate layer of a trained CNN contain powerful descriptor information that may be employed in a variety of tasks, including finding instances that share similar patterns; a task usually named in the literature as instance retrieval (Sharif Razavian et al., 2014). In this chapter, we made use of this CNN property and conducted another study in which we sampled images that were visually similar to the task image. Given this new sampling strategy of examples, we aim to investigate the same main research question we posed in the previous chapter:

(R3) How do saliency maps help with building functional understanding?

In addition, we would like to understand more about the kind of features participants pay attention to in each group. Previously in Chapter 4, participants were asked to justify their answer by responding to this open-ended questions: ”why did you pick this choice”. In contrast, in this study, we focus on features by asking participants to list features they believe the system is sensitive to and features the system ignores. Given this new question that focuses primarily on features, we seek to confirm the previous findings that saliency maps seem to prime participants to think about Saliency-Features. In particular, we aim to address the following research question:

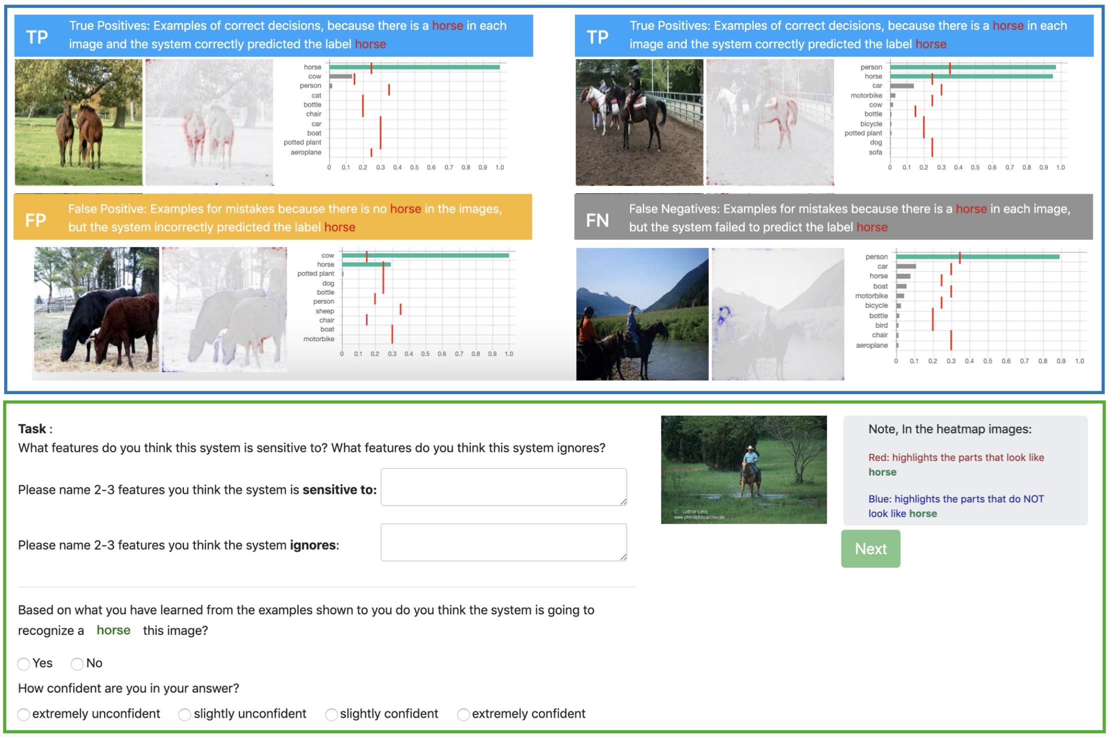
(R4): What features do lay users attend to in order to build a functional understanding of computer vision processes?

Finally, because classification scores produced by the CNN are the default sources of explanatory information at the instance level, we aimed to investigate whether visualising this additional numerical information would outperform, compliment or interact with the presence of saliency maps. Our study design in the previous chapter did not allow us to investigate this factor. Therefore, we designed this study to account for this factor. In the following sections, we begin by describing the study, then reporting the results, and finally discussing the key findings.

5.1 Study Design

We designed a between-group online study to evaluate whether saliency maps can help users understanding of a highly complex CNN used for multi-label image classification. The study included two independent variables that varied between groups, with a full factorial design. Both were related to the amount of information shown to participants: *presence of saliency maps* and *presence of classification scores.*

A screenshot of the experimental setup is shown in Figure 5.1. In the following sections, we lay out a more elaborate description of the study. At this point, it is essential to point out that, similar to the study reported in the previous chapter, we needed to strike a balance between the number of participants, the duration of the study and the variation of experimental factors.



TP True Positives: Examples of correct decisions, because there is a horse in each image and the system correctly predicted the label horse

TP True Positives: Examples of correct decisions, because there is a horse in each image and the system correctly predicted the label horse

FP False Positive: Examples for mistakes because there is no horse in the images, but the system incorrectly predicted the label horse

FN False Negatives: Examples for mistakes because there is a horse in each image, but the system failed to predict the label horse

Task :
What features do you think this system is sensitive to? What features do you think this system ignores?

Please name 2-3 features you think the system is **sensitive to:**

Please name 2-3 features you think the system **ignores**:

Based on what you have learned from the examples shown to you do you think the system is going to recognize a horse this image?

○ Yes ○ No

How confident are you in your answer?

○ extremely unconfident ○ slightly unconfident ○ slightly confident ○ extremely confident

Note, In the heatmap images:

Red: highlights the parts that look like horse

Blue: highlights the parts that do NOT look like horse

Figure 5.1: The interface: what participants were shown for one of the 14 different tasks. Examples are presented in the blue box at the top. The task is shown in the green box at the bottom. All participants worked on the same tasks and where shown the same examples. Conditions differed only in terms of the additional information that was presented alongside each example. Here, saliency maps and scores are shown.

5.1.1 Materials

The dataset, model, and the method for producing saliency maps are similar to those described in Section 4.1. In contrast to our previous studies, and because the CNN performs differently across classes, we revisited the model outcomes definition (i.e. TP, FN, and FP). In, particular, we calculated threshold values for each class (e.g. horse, cat) such that it maximises the F1-score for the class on the dataset. In Figure 5.1, the small vertical red lines represent these selected thresholds.

5.1.2 Tasks

We gave our participants the task to predict the classification outcome of the CNN described in Section 4.1 for a fixed set of 14 task images from the hold-out test set. More specifically, for each task image, we asked participants:

1. to predict whether the system will recognise an object of interest (‘cat’ or ‘horse’) in the given task image.
2. to rate their confidence in their forecast on a 4-point forced Likert item.
3. to list (2-3) features they believe the system is sensitive to and (2-3) features the system ignores.

Figure 5.1 depicts the interface for one task image (with a reduced number of example images). Half of the participants started with images of *horses*, while the other half, began with images of *cats*. Seven task images were concerned with the class *“cat”* and another seven with the class *“horse”*. For each task image, participants were shown 12 example images from the CNN training set to inform their judgement. All participants worked on the same task images and were shown the same example images.

Same as in previous study (Chapter 4), to increase participants engagement in the study, in addition to an £8 payment for their time, participants received an additional performance-based bonus of £0.5 for each correct answer as an incentive.

5.1.2.1 Selection of Example Images

The rational of displaying examples with different outcomes is the same as in the previous study (Chapter 4), however, in this study, example images for every task image were selected based on their cosine distance (Tolias et al., 2015) from the task image in the embeddings space generated from the penultimate layer of the network (Sharif Razavian et al., 2014) (with the guidance of the similarity retrieval tool, please see Section A.2). In particular, we sampled examples of different outcomes for each task image as follows:

- 6 examples of True Positives (TP), where a label had been correctly assigned;
- 3 examples of False Negatives (FN), where the CNN had failed to assign the label;
- 3 examples of False Positives (FP), where the CNN had incorrectly assigned the label.

Furthermore, we noticed in the previous study that the saliency maps of TP examples are more informative than FN and FP. Thus we decided to show more TP than FN or FP examples.

5.1.2.2 Selection of Task Images

In Chapter 4, we argued that given the short time frame of the study, the task complexity increases when more classes are included. To simplify the task further, in this study, we decided to limit our experiment to two classes but included three TP, two FN and two FP for each class.

We drew task images randomly from the hold-out test dataset, with the constraint of having a mid-range classification score. In our pilot studies we had found that images with a low classification score (close to the threshold) were almost unpredictable for participants, while images with a high score were easily predictable (as we observed in Chapter 4) when we sampled task images with a high classification score). Consequently, we chose to sample images with a medium CNN score, as we expect to see the most performance variation this way.

5.1.3 Conditions

The study included the following two independent variables:

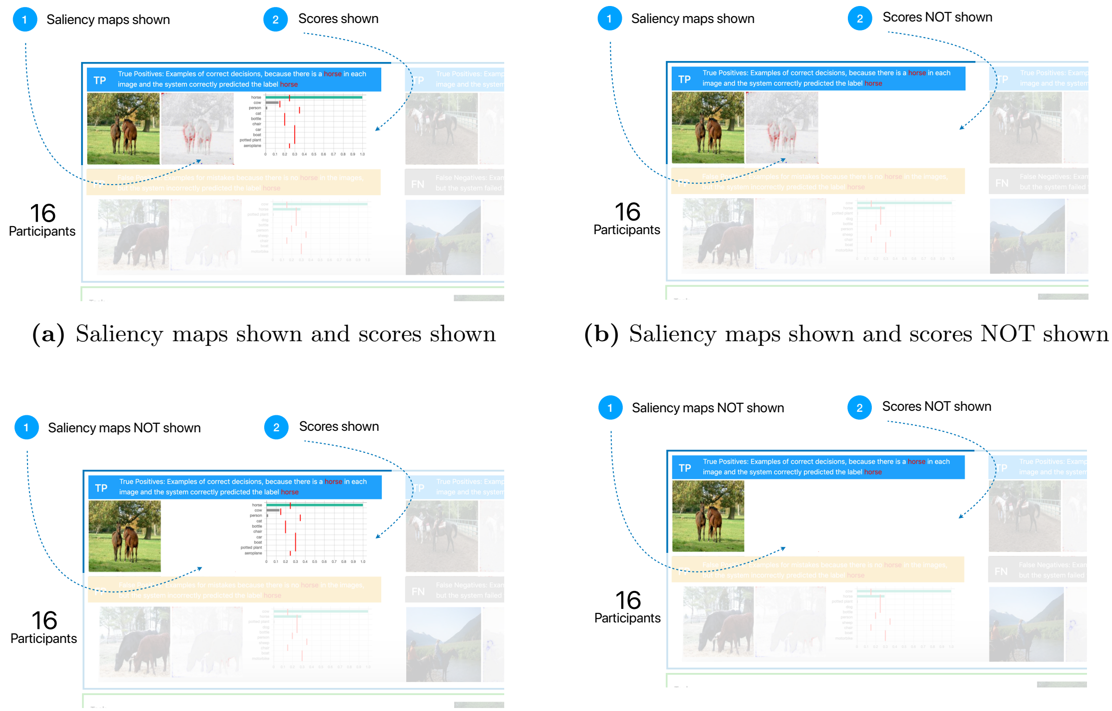
Presence of saliency maps This factor had two levels: shown or omitted. When shown, the saliency map for the relevant class was displayed next to each example image. It is important to note that saliency maps were not shown for the task image but only for the examples.

Presence of Classification Scores This factor also had two levels: shown or omitted. When shown, a bar chart of the top 10 classification scores was displayed next to each example image.

The two independent variables were combined in a full factorial design, resulting in the following four conditions:

- **Saliency maps shown** and **scores shown** (Figure 5.2a).
- **Saliency maps shown** and scores not shown (Figure 5.2b).
- Saliency maps not shown and **scores shown** (Figure 5.2c).
- Saliency maps not shown and scores not shown (Figure 5.2d).

Figure 5.1 illustrates the **saliency maps shown** and **scores shown** condition. In other conditions, the interface appeared the same, with the exception of not presenting the saliency maps, not presenting the scores, or not presenting both, as demonstrated in Figure 5.2.


1 Saliency maps shown
2 Scores shown
16 Participants
1 Saliency maps shown
2 Scores NOT shown
16 Participants
1 Saliency maps NOT shown
2 Scores shown
16 Participants
1 Saliency maps NOT shown
2 Scores NOT shown
16 Participants
TP
True Positives: Examples of correct decisions, because there is a horse in each image and the system correctly predicted the label horse
FN

(a) Saliency maps shown and scores shown

(b) Saliency maps shown and scores NOT shown

(c) Saliency maps NOT shown and scores shown

(d) Saliency maps NOT shown and scores NOT shown

Figure 5.2: Study conditions. The study recruited 64 participants (16 per condition)

5.1.4 Participants

We recruited 64 participants (16 per condition) through Prolific [1], with the same criteria we described in previous study (please see 4.3.3).

5.1.5 Procedure

After providing informed consent, each participant went through a short tutorial (Section B.3) providing the necessary background about the experiment as well as clear instructions for using the system. The tutorial included examples of how the model classified a specific image and clear definitions of TP, FN and FP. We presented participants who belonged to conditions that would show saliency maps with additional information and examples that described this explanation technique and how they can be interpreted. Similarly, participants assigned to a condition showing scores received additional advice on their interpretation.

Upon completion of the introduction, participants commenced completing their 14 tasks. At the end of the study, we gave them feedback for each task images and showed them their earned bonus.

[1] https://prolific.ac/

5.2 Results

5.2.1 Data analysis: choices and processes

The prediction data was analysed in terms of the percentage of correct predictions. Because in this study, conditions represent a full factorial between-subject design with two independent variables: the presence of saliency maps, and the presence of classification Scores, we chose to apply a two-way independent ANOVA test.

The confidence data were coded by numbers 1-4 and summed up per participant. A one-way independent Kruskal-Wallis test were used since we have one factor with 4 levels that represent the 4 conditions.

The open questions about what features the classifier is sensitive to and what features it ignored were analysed using a qualitative content analysis similar to the one we detailed in the previous study (Section 4.3.6.2), where we decided to assign each code to one of two code groups: **Saliency-Features** and **General-Attributes**. We counted the number of Saliency-Features codes and General-Attributes codes. We noticed that some participants wrote a lot in the qualitative response and therefore mentioned a lot of features, while others did not. To prevent this from skewing the results, we calculated a ratio. We obtained the **Saliency-Features ratio** for each participant by dividing the number of Saliency-Features codes by the total number of Saliency-Features and General-Attribute codes that we had assigned to their answers. Therefore a ratio of 0.6 means that 60% of the features that a participant mentioned were Saliency-Features. In the same fashion, we calculated ratios for all codes. Similar to (1), to analyse the data, we also applied a two-way independent measures ANOVA using white-corrected coefficient covariance matrix (White, 1980).

5.2.2 Outcome prediction accuracy

We summarized the data in Figure 5.3. A Shapiro-Wilk test revealed that the percentage of correct forecasts within groups were approximately normally distribute ($W = 0.957$, p=0.027). A Levene's Test showed performance variances between groups were similar ($F_{(3,60)} = 0.156$, $p = 0.925$).

A two-way independent ANOVA revealed a statistically significant main effect of the presence of saliency maps on the performance ($F_{(1,60)} = 4.191$, $p = 0.045$, $\eta^2 = 0.063$). In the presence of saliency maps participants were more accurate in predicting the outcome of the classifier ($\mu = 60.7\%$, $\sigma = 11.0\%$ vs. $\mu = 55.1\%$, $\sigma = 10.8\%$). There was no significant main effect of the presences of scores on performance ($F_{(1,60)} = 1.938$, $p = 0.169$, $\eta^2 = 0.029$). Furthermore, there was no interaction effect ($F_{(1,60)} = 0.060$, $p = 0.807$, $\eta^2 = 0.001$).

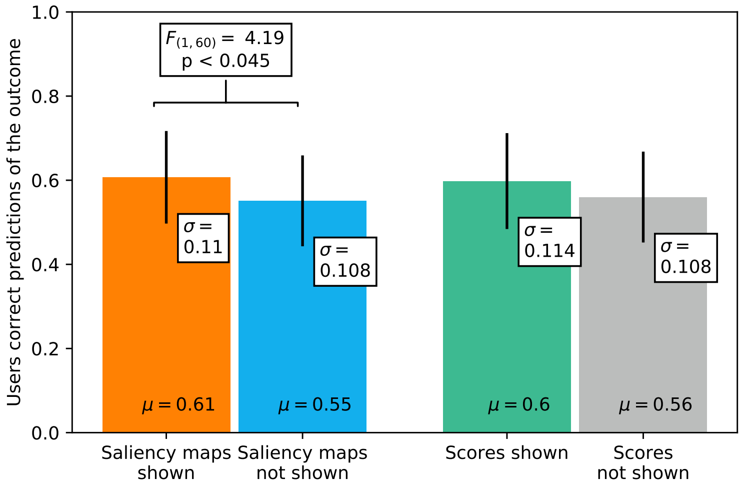


Figure 5.3: Left: When saliency maps were shown, participants were significantly more accurate in predicting the outcome of the classifier . Right: Scores did not significantly influence the participant's prediction performance. Success rates were relatively low across conditions, showing that tasks were very challenging.

We also consider participants' accuracy on the subsets of images corresponding to different outcomes (i.e. TP, FP, FN). Overall the accuracy was higher for TP images, on average 79.4%, it was lower for FP, on average 46.9%, and even lower for FN, on average 36.7%. An interpretation of this result is that participants are possibly inclined to over-estimate the performance of the systems on challenging cases. Such cases are represented by FP and FN images. In fact, in 67.3% of all cases, participants predicted that the system would be correct, whereas it was only correct in 42.9% of the cases (i.e. 6 out of 14 task images were TP, which represents 42.9%). Because we did not fully counterbalance the order of tasks and True Negatives (TN) were not part of the task set, unfortunately, we were not able to run statistical tests on the different outcomes.

5.2.3 Confidence

We also asked participants to rate their confidence in their forecast on a 4-point forced Likert item. Answers were coded by numbers 1-4 and summed up per participant. A one-way independent Kruskal-Wallis test showed that no statistically significant differences in confidence were found across conditions ($H(3) = 1.130$, $p = 0.770$). On average participants tended to be *"slightly confident"* in their answers (Median = 3.00).

5.2.4 Mentioned Saliency maps Features

Besides making a prediction, we asked participants what features they think the classifier is sensitive to and what features it ignored.

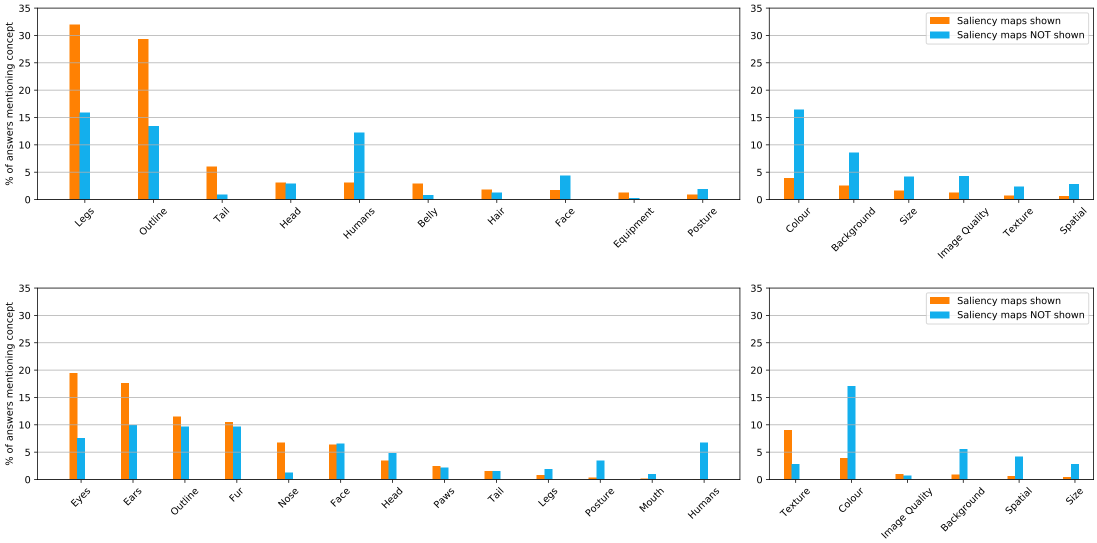


Figure 5.4: Frequencies of individual features mentioned by participants for images of horses (top) and cats (bottom). Left: Features belonging to the Saliency-Features. Right: Features belonging to the General-Attributes (frequencies were normalised for each participant).

5.2.4.1 Excluded data

An analysis of the qualitative data revealed that two participants misunderstood these tasks. Consequently, they were excluded from this analysis. It also became apparent that many of the remaining participants misinterpreted the question about the features the system *ignored*. Therefore, we focused only on replies participants gave regarding the sensitivity of the classifier to features.

5.2.4.2 Mixed-Method Analysis of Answers

We carried out a qualitative content analysis similar to the one we detailed in the previous study (Section 4.3.6.2), where we decided to assign each code to one of two code groups: **Saliency-Features** and **General-Attributes**.

The top of Figure 5.4 shows the ratios for the answers participants gave for images of cats, while the bottom of Figure 5.4 shows them for images of horses.

The Saliency-Features ratio was subjected to a statistical analysis. The data is summarised in Figure 5.5. A Shapiro-Wilk test revealed that the rate of Saliency-Features within groups were approximately normally distributed ($W = 0.900$, $p < 0.01$). A Levene's Test showed that the variances between groups were significantly different ($F_{(3,58)}$ $= 3.749$, $p = 0.016$). To account for heteroscedasticity we ran a two-way independent measures ANOVA using white- corrected coefficient covariance matrix (White, 1980). It revealed a statistically significant main effect of the presence of saliency maps on the

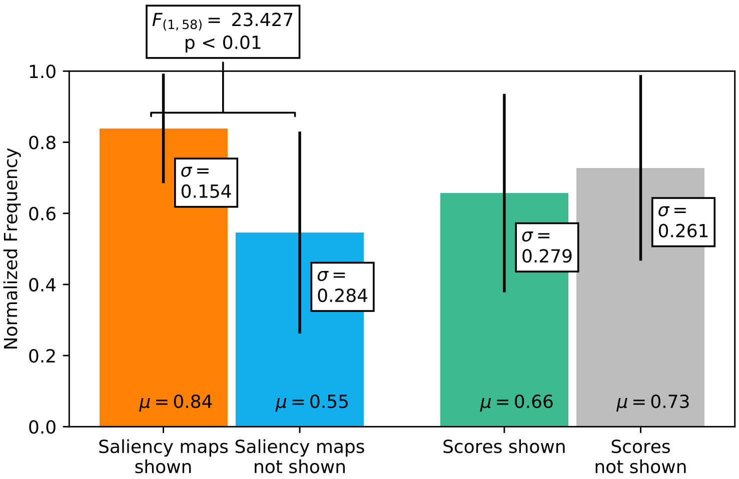


Figure 5.5: The ratio of mentioned Saliency-Features. It summaries the share of saliency-features participants mentioned per task. They mentioned significantly more such features when saliency maps were present (Left). Scores did not have an influence (Right).

rate of mentioned Saliency-Features ($F_{(1,58)} = 23.427$, $p < 0.01$, $\eta^2 = 0.295$). Participants mentioned a larger share of Saliency- Features (e.g. *legs, outline*) compared to General-Attributes (e.g. *colour, image quality*) when saliency maps were present ($M = 83.9\%$, $SD = 15.4\%$ vs. $M = 54.6\%$, $SD = 28.4\%$). There was no significant main effect for the presences of scores ($F_{(1,58)} = 1.384$, $p = 0.244$, $\eta^2 = 0.013$) and no interaction effect ($F_{(1,58)} = 0.004$, $p = 0.948$, $\eta^2 = 0.001$).

The effect of saliency maps can be explored in more detail in Figure 5.4. It shows that saliency maps seem to lead people to pay attention to specific parts of the object of interest. For example, Figure 5.4 depicts the share of mentioned features for images of horses. It is evident that some features such as *legs*, *outline*, *tail* and *belly* were mentioned much more frequently by participants exposed to saliency maps, while general-attributes such as *background* and *colour* are mentioned more often when the saliency maps are not shown.

5.3 Discussion

Through a combination of quantitative and qualitative analysis, the results of our study highlight the potential to use saliency maps as an explanatory tool for non-expert AI users, as well as their limitations. In the following subsections, with respect to the research questions posed at the start of this chapter, we discuss the key issues and their implications for design and future research.

5.3.1 The utility of saliency maps exists, but it is limited

our results show that when saliency maps were shown, participants predicted the outcome of the classifier significantly more accurately, however, success rates were still relatively low (i.e., 60.7%). Hence, the task of estimating the system's predictions on a new image remained challenging. This is also reflected by our participant's self-reported confidence in their answers, which was not affected by the presence of saliency maps or scores, and was on average still quite low. Moreover, Participants across conditions seemed to be better in predicting the system's outcome when it was correct (i.e., TPs). They were mainly struggling with the prediction of errors (i.e., FPs and FNs), performing worse than chance. One of the envisioned applications of explanations is aiding users in building appropriate trust into a system (Dzindolet et al., 2003; Bussone et al., 2015). Unexpected and unpredictable failures of a system (i.e. in our case, the ability to predict the FP and FN images) affect trust more negatively than those that can be understood and anticipated (Lee and See, 2004; Dzindolet et al., 2003). Therefore, it is important that users can understand when the system will fail. As detecting errors is a claimed utility of instance-level explanations (Ribeiro et al., 2016a; Lapuschkin et al., 2019), we suggest that **future work** should evaluate this empirically in more detail.

Reasoning on Examples In contrast to the last study (Chapter 4), we explained at the beginning of this chapter that we are exploring **R3** in a setting where sampled examples are based on images with similar embeddings to the task image. The rationale behind this choice was that people might learn more effectively from examples that are similar in appearance to the task image (Cai et al., 2019a,b). It might help them to reflect upon the *visually similar* images that the system had successfully classified (i.e. TPs) and images the system had classified incorrectly (i.e FN, FP). We hypothesised that such contrasting reasoning (Miller, 2019a) can help users to understand the system's causes of successes and failures. However, when considering the examples presented to participants, we noticed that the usefulness of FN saliency maps is negligible. They usually highlight very little evidence (e.g. see the FN example in Figure 5.1). For FN examples, the actual image and the other saliency maps (TP, FP) become the only source of information for understanding why an example has not been recognised by the system. This insight suggests that the utility of saliency maps varies according to the classification score. In other words, a saliency map may highlight what supports the prediction of some class, but it will fail to provide counter-factual evidence, namely, the absence of evidence. In the next study (Chapter 6), we designed a task that investigates this point.

We would like to emphasise that for a human, it is easy to spot and point to the absence of a feature concept, while it is not for a CNN. Humans can easily break down an image into meaningful regions (semantics) (Fei-Fei et al., 2007). In contrast, CNNs look for patterns in a sub-symbolic fashion that lead to an outcome (Bishop, 2006; Lipton, 2018). Because CNNs do not process data in a 'semantic' fashion, other patterns in an image (which may not belong to the concept) can contribute towards a classification outcome in unexpected ways (Lapuschkin et al., 2019). **An implication** for the design is that we need to develop explanation algorithms that bridge the gap between humans and machines by leading the user to understand that the system is not basing its classification decision on higher-level 'semantics' of the image. Furthermore, we would like to emphasise that choosing representative examples with their corresponding saliency maps, which summarise the behaviour of the system well, is an under-explored topic. New approaches for generating saliency maps and for applying them to various machine learning problems exist (see (Adadi and Berrada, 2018)). However, very little work exists that investigates for which instances users should examine salience maps. Researchers have acknowledged that users can only inspect a limited number of saliency maps (Ribeiro et al., 2016a), but to the best of our knowledge, only two works explore sampling strategies (Ribeiro et al., 2016a; Lapuschkin et al., 2019) - none of which where applicable for this work. An important implication, then, is that **further research** needs to characterise the effect of different sampling strategies of saliency map examples on users interpretation of the system operation.

5.3.2 Facilitating global model understanding by explaining local features

It is worth emphasising that even when users notice features, this does not necessarily imply that they will perform better in predicting the outcome of the CNN or reach a global understanding of the model. Saliency maps provide only a visualisation of the importance of pixels in a single image. Transferring knowledge about potential features to new images, where they are presented in different orientations, scales, forms and perspectives, is very challenging. Furthermore, it is hard to get a quantifiable measure of the importance of individual features in an image. Again complexity increases if one attempts to quantify the importance of a feature on new images. In other words, it is difficult to estimate how the classification score would change if a feature would be absent. Would the score go down by a factor of 0.1, 0.2 or 0.6? Moreover, does the presence of different features cause an interaction effect between the highlighted features? It is challenging for users to reason about this, especially when considering that CNNs process the input data in a non-linear fashion (Bishop, 2006).

An implication for the design of explanation systems, then, is that saliency maps should be complemented by a global measure that explains how sensitive the presence of a feature is to the prediction of some class. For example, how sensitive the presence of *nose* is to the prediction of *cat*? In that regard, complementing saliency maps with this additional information could be valuable for users to build quantifiable measures of saliency maps, and perhaps avoid biases that might arise from exploring an unrepresentative subset of the dataset. (Kim et al., 2017) proposed an algorithm in that direction, where a user can test how sensitive the model's predictions are to a global concept defined by the user. For example, how important the *strips* concept is to the ”zebra” class. Informed by the discussion above, in the next study (Chapter 6), we investigate whether the saliency map technique we are evaluating can help participants in quantifying the importance of individual parts in an image.

5.3.3 The importance of general attributes

Another reason why noticing Saliency-features does not necessarily facilitate a better understanding of a model is that general-attributes (e.g. colour, contrast) might influence the classification outcome. However, these general-attributes are usually not directly highlighted by saliency maps, because as a more general image property, they can not be localised to individual pixels. Moreover, saliency maps might even prime participants to primarily consider only highlighted features, and give less weight to other attributes that are not highlighted but important. In fact, our data indicates that when saliency maps are present, participant mentioned general-attributes less. This finding complements the previously stated limitation of the expressive capabilities of saliency maps (Schuessler and Weiß, 2019). In contrast, users preconceptions may cause them to focus on attributes such as the *brightness* of the image, even if it is not a major cause of failure. **An implication** for design is to develop explanations that convey the right expectation to users. We suggest that saliency maps should be complemented by more global representations of the image features. For example, saliency information could be related to global descriptors of the images, such as overall contrast or brightness measures.

5.4 Summary

This chapter reported on a between-group user study designed to evaluate the utility of “saliency maps”. A total of 64 participants were asked to estimate the CNN outcome based on task images, with 12 visually similar examples shown to help them make a decision. The study included two independent variables that varied between groups, with a full factorial design. Both variables were related to the amount of information shown to participants: the presence of saliency maps and the presence of classification scores.

Reflecting on the research questions we raised earlier, we can summarise the key findings of this study as follows:

1. In relation to R3, we can see that selecting examples with similar embeddings seems to enhance participants' ability to predict the outcome of the network for new images. However, even with saliency maps present, the CNN model remained largely unpredictable for participants (60.7% prediction accuracy). In addition, data indicates that scores had no influence on participants' ability to predict the outcome of the network.

2. In response to R4, when saliency maps were present, participants tended to mention Saliency-Features more often and give less weight to other attributes such as colour and contrast.

We may attribute the moderate performance in predicting the outcome of the network to saliency maps' limited utility. However, similar to the previous study, the complexity of the task, which involves asking participants to learn complex patterns from a few examples and then apply this knowledge to a new task, may have overshadowed what the saliency map truly offers. Although we found an improvement over the results reported in the prior study, the task's complexity may still exist. In the following chapter, we will attempt to reduce these elements in order to better understand the role of saliency maps.

Chapter 6

Evaluating the role of saliency maps through a simplified study design

In previous studies (Chapter 4 and Chapter 5), task images were presented *without* their saliency map explanations. Through this design we intended to simulate a realistic settings where a user would spend time browsing examples (with or without) explanations in order to understand a model's behaviour. To evaluate participants' understanding, we asked them to predict (or ”simulate”) the model outcome on new images. Results of the previous study (Chapter 5) indicate that when saliency maps were available, participants answered correctly more frequently than when they were absent (60.7% vs. 55.1%, p = 0.045). However, the overall performance was generally low even with the presence of saliency maps.

To better understand the limited results of the previous study, we hypothesised that learning how a model works from a few examples may be a task intrinsically too difficult, limiting participants' accuracy even when explanations were provided. Consequently, we decided to evaluate the explanation techniques through a task of lower complexity. Thus, for this study, we designed a task to try and assess what the various explanation techniques communicate to users by presenting them alongside the task image. Our reasoning is that if the information gained from an explanation is genuinely low, the user's performance in a complex task that does *not* show the corresponding explanation of the task image will likely to be worse.

In addition, previous results indicate that when saliency maps were present, participants mentioned Saliency-Features more often. Informed by this findings, we designed and evaluated **a new saliency map technique ”Semantic occlusion” (sem-occl)** to specifically focus on features that are meaningful to people. As a generalisation of this approach, we also proposed another occlusion technique ”multi-scale occlusion” (m-scale-occl), which does not rely on semantic annotation and instead uses rectangular occluding regions arranged on multi-scale grids. In both techniques, parts of the image get occluded and the image is fed through the CNN. A saliency map is generated based on how big the effect of the occluded pixels is on the CNN result. Examples produced by both techniques can be seen in figure 6.7.

Examining users' functional understanding through other measures: Given the new designed tasks (i.e. where saliency maps are presented alongside the task image), we aim to investigate the main research question we posed in previous chapters:

(R3): How do saliency maps help with building functional understanding, including the relation to varied system confidence?

where we measure this understanding by:

(1) Users' ability to predict the CNN classification outcome on task images. Which is the same measure we used in previous chapters, but this time, we present a saliency map alongside the task image.

In addition, in the previous study (Chapter 5), we discussed that one of the claimed utilities of saliency maps is detecting errors, and that one of the potential applications of explanations is to assist users in developing appropriate trust in a system. Despite Deep Neural Networks' greater performance, past research has shown that these networks are vulnerable to well-designed little perturbations of the input samples, which are commonly referred to as adversarial examples, which are a set of images that look almost identical to the original images to the human eye, but not to the network Yuan et al. (2019). Figure 6.1 shows an adversarial example produced by an algorithm developed by Chen et al. (2019). The significance of dealing with these adversarial samples is highlighted in safety-critical applications such as autonomous vehicles, where for instances traffic signs might be altered to deceive the smart system. Therefore, in this study, we are also interested in evaluating the role of saliency map techniques in informing users about this sort of error called adversarial images. In particular, we seek also to measure users understanding in **R3** by:

(2) their ability to predict the CNN classification outcome of adversarial images.

Moreover, in Section 5.3, we hypothesis that it is challenging for saliency map techniques to quantify the importance of individual parts in an image.

Quantifying the importance of a part across a large number of examples could be helpful in gaining a global understanding of a part's importance (in contrast to what saliency maps offer locally) and discovering the challenging cases that the system may struggle to identify. Moreover, such global perspective could be a first step towards providing a

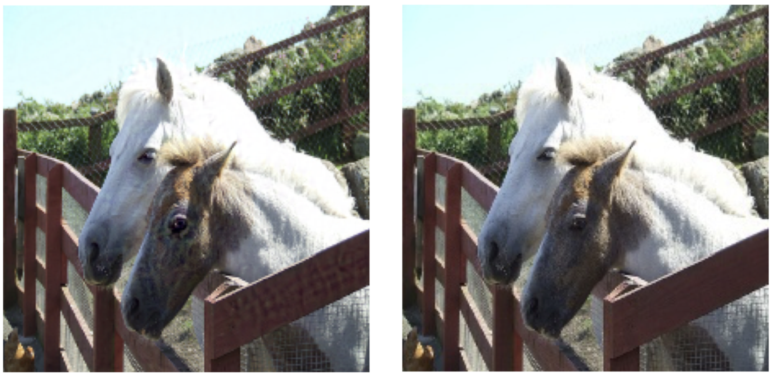


Figure 6.1: (Left: an adversarial example produced by Chen et al. (2019) algorithm. Right: the original image.

counter-factual explanations. For example, in addition to identifying the relevant parts, the explanation technique may inform users that the system had difficulty identifying the "cat's eyes" in an image because the "eyes" were not visible. Therefore, we also seek to measure users understanding in **R3** by:

(3) their ability to quantify the importance of individual parts of an image.

Comparing different saliency map techniques: Finally, in our previous studies, we demonstrated that the design space is large with many elements, therefore, one of the design choices (to constrain this large space) was to work with a single saliency map technique, namely the LRP. When compared to the previous study design, the task in this study is simpler, which should enable us to experiment with different saliency map methods. It is therefore feasible to address this additional research question:

(R5): How do different saliency map generation techniques perform to build functional understanding?

A total of 144 participants took part and were randomly assigned to one of six conditions, each with its own set of saliency maps generated using a different explanation technique. In all conditions, participants were asked to perform 6 tasks, each based on a different image. Each task includes 6 sub-tasks (i.e., they need to answer 36 questions in total). The tasks are detailed below in Section 6.2.

The work detailed in this chapter and Chapter 4 form the biases for the following paper in submission:

Ahmed Alqaraawi, Enrico Costanza, Nadia Berthouze and Emma Holliday. Evaluating and Improving Heatmap Explanations for CNNs through to online user studies. ACM Transactions on Computer-Human Interaction (TOCHI).

6.1 Background: Occlusion-Based Saliency maps

The findings of our previous studies prompted us to consider alternative strategies to generate saliency map explanations, we describe each in the following subsections.

6.1.1 Semantic Occlusion

Our previous studies revealed that the features of an object played an important role when our participants reasoned about how the CNN would classify an image. The importance of object features is also highlighted in the previous study (chapter 5). Arguably, it is natural for participants to refer to the parts of the object they are reasoning about, as that is how we as humans reason about visually recognising objects. Therefore, we decided to explore whether an explanation technique that explicitly refers to object features may better match users' expectations (their mental model), and hence be more informative and useful.

In particular, we decided to highlight the effect that different sub-regions of an object have on the classification score of the image. For the horse class, we considered the legs, head and body. For the cat class we considered, the head, eyes, nose, ears, legs and body. We leveraged the annotations available as part of the PASCAL-part dataset (Chen et al., 2014) to define these regions, merging them into higher level ones when needed (for example, the face, eyes, muzzle, and ears of the horse is merged into head).

Initially, we developed an interactive tool that allows users to check the contribution of the different parts of the object. Figure 6.2 shows a screen shot of the tool, in which in (1) we show the images, saliency map and the classification score of the presented image. In (2) The system highlights some of the object parts as the user hovers over an image. (3) Once a part is selected, it will be occluded using the DeepFill technique (Yu et al., 2018) to make it appear as if the selected part does not exist. Consequently, a new saliency map and a classification score are displayed for this modified image. The interactive element in this tool proved to be overwhelming for users because they had to recall the different saliency maps and scores that corresponded to the different occluded parts. To address this limitation, we developed a saliency map visualisation named: *sem-occl*, which attempts to provide almost the same information in a single static saliency map.

In particular, to estimate the effect of each sub-region on the classification score for an image I, we occluded the corresponding pixels obtaining a new image $I_{withOccludedRegion}$ and we fed the modified image through the CNN to compute the score: $score_{withOccludedRegion}$. We then used the difference between the original CNN score S and $S_{withOccludedRegion}$ to estimate $S_{region} = S - S_{withOccludedRegion}$. We then map S_{region} to a shade of red or blue to represent the importance of this region, where red indicates parts of the image that

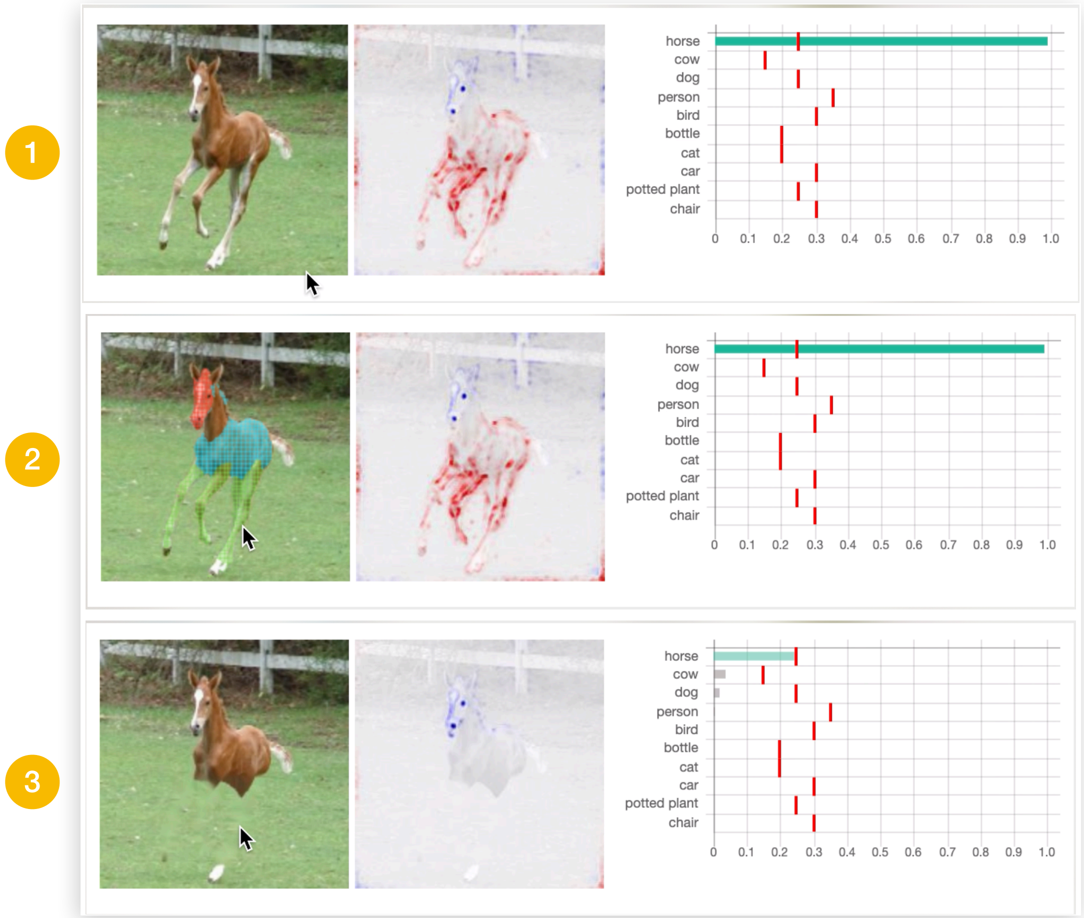


Figure 6.2: The click and hide tool

support the classification, while blue indicates parts that are against this classification. Example saliency maps generated through this method are shown on the fourth column ('sem-occl') of Figure 6.7.

The semantic occlusion method was designed primarily as a contrasting element for this study. Due to the method's reliance on the manual annotations of the images, in its current form it would not be easily applicable beyond a specific dataset like PASCAL-part. If user evaluation results show promise they would provide further motivation to try and address the technical challenge of automatic object segmentation of arbitrary images - a challenge already recognised by the AI community (Chen et al., 2014).

6.1.2 Multi-scale Occlusion

The limitations of the semantic occlusion method led to explore another occlusion-based method, which we refer to as "multi-scale occlusion"[1]. Unlike semantic occlusion, this

[1] While the initial development of this technique was developed by myself, further work needed to combine the data at multiple scales was done by Emma Holiday as part of her MSc thesis at UCLIC. Information about her work is included here for clarity, but labelled accordingly.

method occludes all sub-regions of an image using an iterative algorithm, so it is more general. By occluding at multiple scales, we believe this can roughly approximate different sized features of an object. We believe that the *conceptual simplicity* of occluding a region of the image to (coarsely) estimate its contribution to the CNN classification is attractive and promising because it can be easily communicated to non-expert users, without requiring much (if any) technical knowledge of how a CNN works. This is in stark contrast to method like LRP or GradCAM, which instead involve referring to the layers of the network and their operation. Indeed, our own experience when writing the participant instructions for previous studies, using LRP, was that we had to rely on a very generic description of the outcome of the explanation techniques, rather than mentioning how the explanation was generated.

We started by considering a grid of rectangular regions as a basis for the occlusion as a generalisation of regions representing parts of the objects. Each rectangle may contain a feature useful for the classification by the CNN, if so occluding it would affect the CNN classification score, and this could be visualised similar to the above for the semantic occlusion saliency maps. Inspired by methods like Wavelet Transforms (Grossmann and Morlet, 1984) and SIFT (Lowe, 2004) (ORB is a similar approach to SIFT that we have utilised in Chapter 3), we used a multi-scale approach, in recognition that features of interest might appear with different sizes within an image.

The image is initially divided into quarters (Figure 6.3, centre), and each quarter is used as an occlusion rectangle. Each quarter is then subdivided into quarters again, resulting in smaller occlusion rectangles as shown in Figure 6.3 (right). The process can be iterated to different maximum depths. The saliency maps used in this work were generated at maximum depth equal to 3. For each occluding rectangle a score can be estimated, following the same process as for semantic-occlusion saliency maps. However, for multi-scale saliency maps occluding rectangles of finer resolution overlap with those at coarser resolution. Given the hierarchical nature of the recursive subdivision of the image in smaller and smaller rectangles, the scores of all rectangles can be stored in

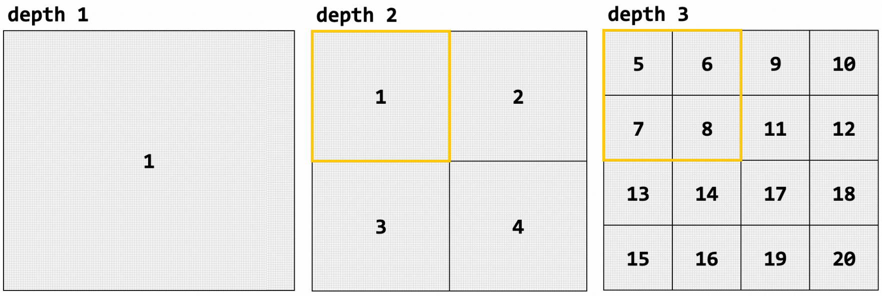


Figure 6.3: an example image with occlusion areas as per recursive programmatic occlusion. There are 20 occlusion areas, 4 large and 16 small. Note that the small areas are contained within the large areas, as highlighted by the yellow box. The order of occlusion starts from 1.

a tree data structure [2]. To synthesise a single multi-scale saliency map, the tree is pruned starting from the leaves as follows: child nodes are pruned if the sum of their scores is lower than the parent node (and the parent node is kept instead). Conversely, if the sum of the scores of the children nodes is higher than or equal to the score of the parent, the children are kept. Thus, once fully pruned, the tree contains the finest granularity regions possible without diminishing the score (importance). One limitation of the process as described so far is that features of interest might fall across adjacent rectangles, and hence their importance would be missed. To address this issue, we followed an approach that is well established in Computer Vision, i.e., to consider overlapping rectangles. Specifically, the entire multi-scale grid is offset by half of the size of the smallest rectangle, and repeated at this interval across the width of the largest occlusion size. As illustrated in Figure 6.4, for an image of W pixels in width and a saliency map with maximum depth N, the smallest occluding rectangle will have width $W_{small} = W/2^N$ and the horizontal offset will be $W_{off} = W_{small}/2 = W/2^{N+1}$. As shown in Figure 6.4, the offset is applied horizontally 2^{N-1} times, where N is the maximum depth of the saliency map, after which point the offset pattern would repeat

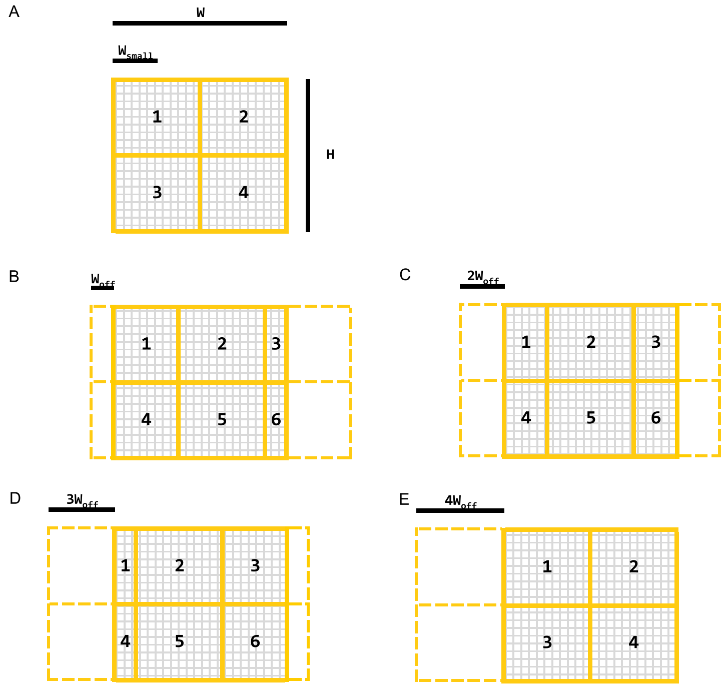


Figure 6.4: an example of the occlusion areas resulting from the offset pattern. Note how A and E are effectively the same. The dotted lines indicate the theoretical regions while the solid lines indicate the final occlusion regions, after cropping to the bounds of the image. In this Figure, W is the width of the original image, H is the height of the original image, W_{small} is the width of the smallest occluding rectangle (the rectangle is not shown), and $W_{off} = W_{small}/2$ is the width offset

[2] The development of this tree data structure and its processing was carried out by Emma Holiday as part of her Master's thesis at UCLIC, and it is included here for clarity.

(for the saliency maps in this paper $N = 3$ and the offset is repeated horizontally $2^2 = 4$ times). The offset is applied both vertically and horizontally, and for each application a new saliency map is generated, resulting in $2^{N-1} \times 2^{N-1} = 2^{2(N-1)}$ offset versions of the saliency map (for the ones in this chapter, it is equal to 16). This potentially large number of offset saliency maps is aggregated using a heuristic rule which favours the ones that have the highest score for the smallest area (i.e. where the score distribution over area has higher density). In particular the top 5 offset saliency maps by this heuristic are averaged to produce a single saliency map. Example saliency maps generated through this method are shown on the third column (m-scale-occl) of Figure 6.7.

While the naive implementation of this process is computationally intensive (because it requires feeding through the CNN a large number of occluded versions of the image), a large proportion of the calculations is repeated, offering a clear opportunity for optimisation and execution in parallel. Moreover, in contrast to the semantic occlusion, the multi-scale occlusion does not rely on manual annotations of the images.

6.2 Study Design

In this study, Participants were asked to complete 6 tasks, each based on a different image. Each task includes 6 sub-tasks, which are presented on separate pages. Figure 6.5 demonstrate how tasks and sub-tasks are structured and Figure 6.6 depicts the subtasks that participants have to complete for each task. These sub-tasks were designed to evaluate the amount of information gained by each technique, and they are defined as follows:

- **Q1-score-noH:** a baseline sub-task in which participants were asked to predict the CNN classification outcome of an image by assigning one of five levels of scores: [very low score, low score, medium score, high score, and very high score].
- **Q2-score-H** is the same as Q1-score-noH, but with the presence of a saliency map.
- **Q3-select-part:** Participants were given the image and the saliency map and asked to choose the part of the object that would reduce the score the most if obscured.
- **Q4-cover-part:** similar to Q3-select-part but with a slightly different question, in which participants were asked to predict whether the score would (decrease, increase, or remain nearly the same) if a certain part of the object was covered.
- **Q5-score-H** is the same as Q2-score-H, except that the image is the adversarial version of the original image (and the saliency map corresponds to this new version).

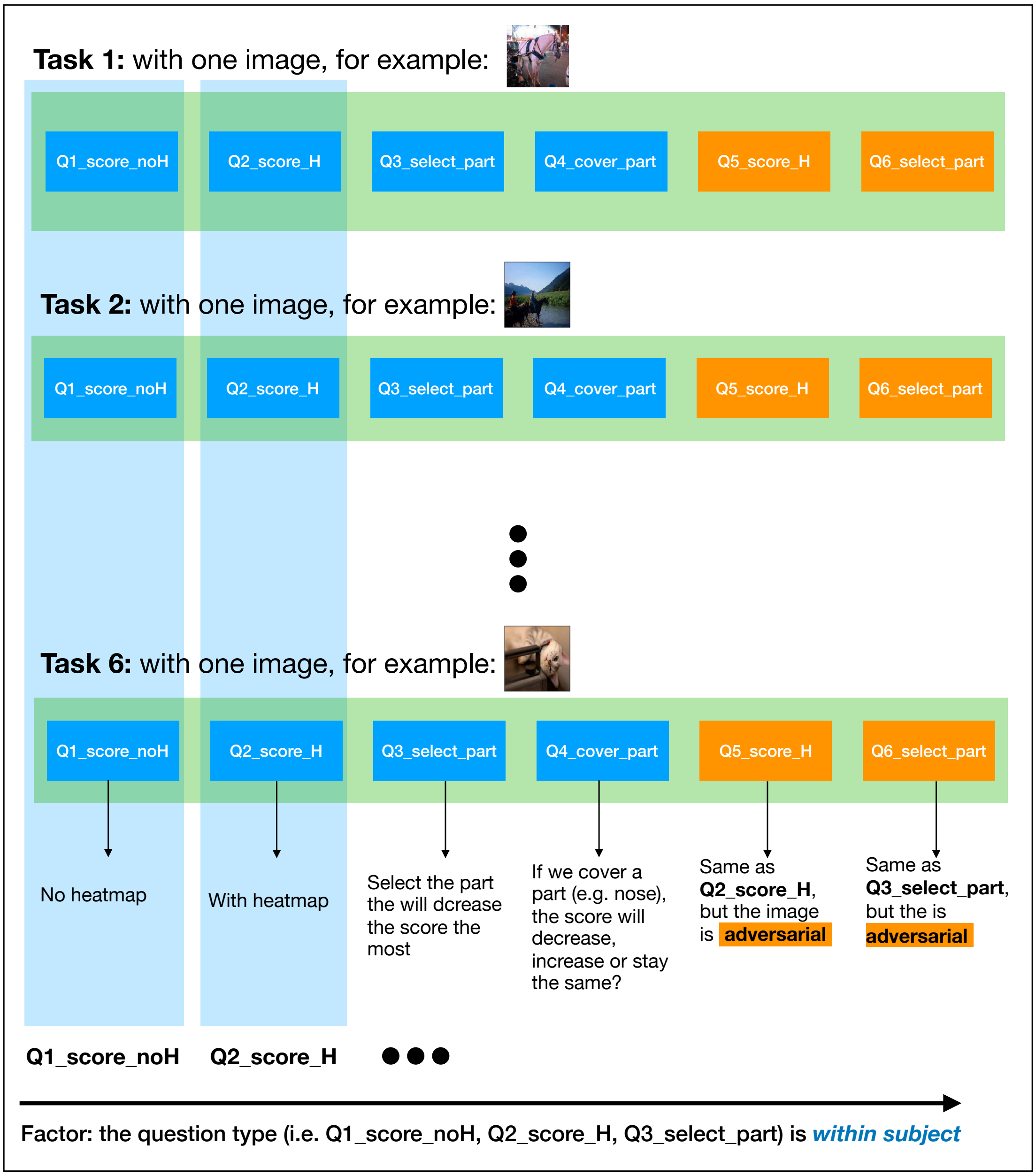

Task 1: with one image, for example:
Q1_score_noH
Q2_score_H
Q3_select_part
Q4_cover_part
Q5_score_H
Q6_select_part
Task 2: with one image, for example:
Q1_score_noH
Q2_score_H
Q3_select_part
Q4_cover_part
Q5_score_H
Q6_select_part
Task 6: with one image, for example:
Q1_score_noH
Q2_score_H
Q3_select_part
Q4_cover_part
Q5_score_H
Q6_select_part
No heatmap
With heatmap
Select the part the will dcrease the score the most
If we cover a part (e.g. nose), the score will decrease, increase or stay the same?
Same as Q2_score_H, but the image is adversarial
Same as Q3_select_part, but the is adversarial
Q1_score_noH
Q2_score_H
Factor: the question type (i.e. Q1_score_noH, Q2_score_H, Q3_select_part) is within subject

Figure 6.5: How tasks and sub-tasks are structured. All conditions were exposed to the same set of images. However, different conditions received different explanations.

- **Q6-select-part** is the same as Q3-select-part, except that the image is the adversarial version of the original image (and the saliency map corresponds to this new version).

For all these sub-tasks, we measure users' performance based on the number of correct selections.

Three images are from the ”cat” class, while the other three are from the ”horse” class. Half of the participants started with images of horses, while the other half with images of cats. To capture a variety of cases, we sampled task images from three levels of CNN

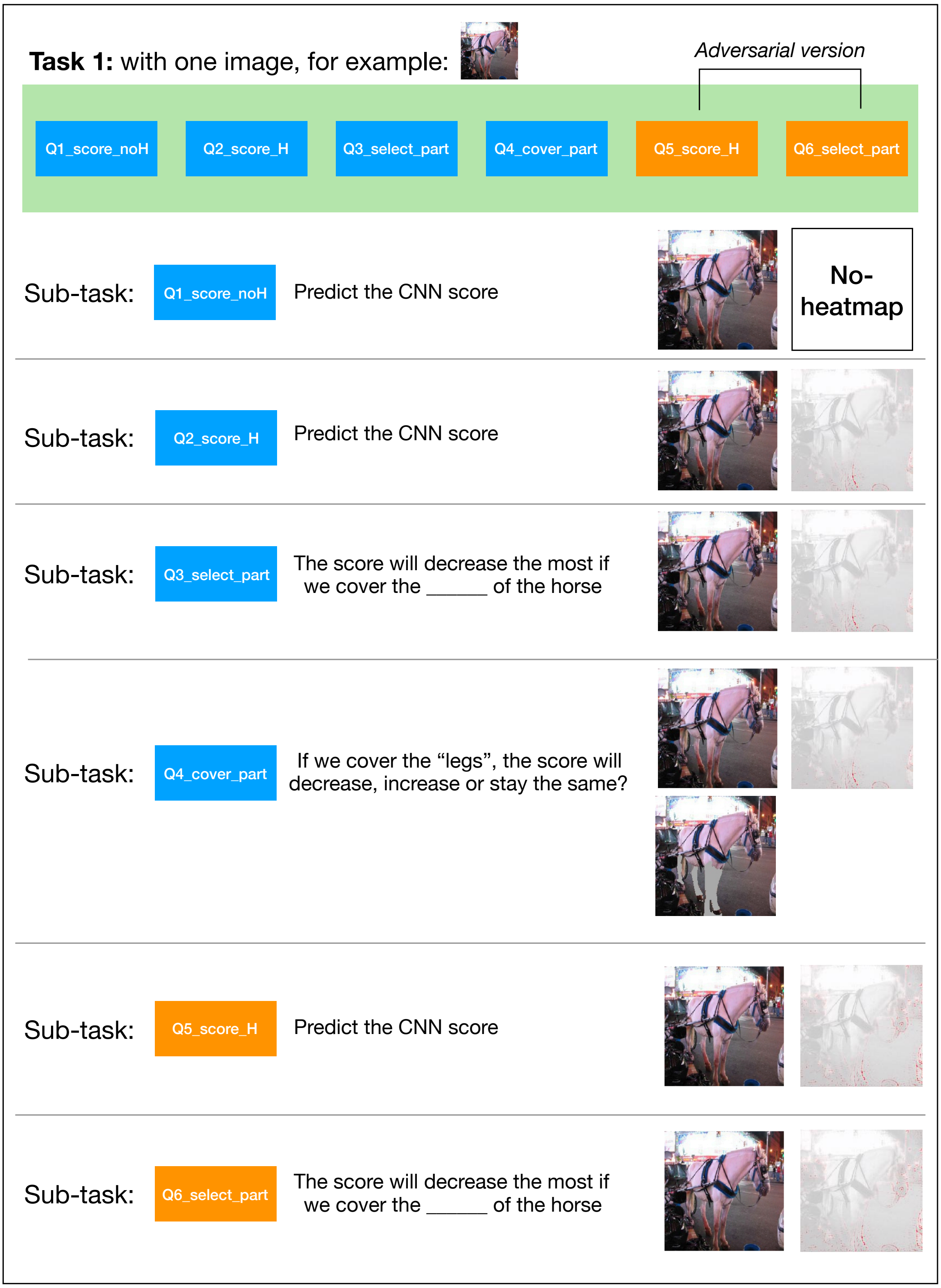


Figure 6.6: Details of the sub-tasks for Task 1 as an example. Note that the order of tasks (which is associated with images) are counter-balanced; meaning that other images might be presented for task1 instead of the one shown here.

scores: very low score (two images), medium score (two images) and very high score (two images). The images (tasks) were counter-balanced according to the Latin square scheme. In the tutorial session participants were shown five examples images on which the system achieved different levels of scores. Saliency map explanations were shown for all five examples. The work and implementation of (Chen et al., 2019) was used to produce the adversarial examples. All participants worked on the same task images and were shown the same example images. To explore the effect of covering different areas of the object on the classification score, we developed two interactive tools, the one we represented in Figure 6.2 and the one detailed in Section A.3.

The saliency map techniques involved in this study are LRP (Bach et al., 2015b), Grad-CAM (Selvaraju et al., 2017), Guided-backpropagation (Springenberg et al., 2015)) and the two saliency map techniques we introduced above (i.e. sem-occl and m-scale-occl). For contrast, we also included a simple edge detector as an additional condition, which is visually similar to other saliency map techniques (Adebayo et al., 2018), yet independent of the CNN model. Figure 6.7 shows the generated saliency maps for the images used in this study, while Figure 6.8 shows the generated saliency maps for the adversarial images version.

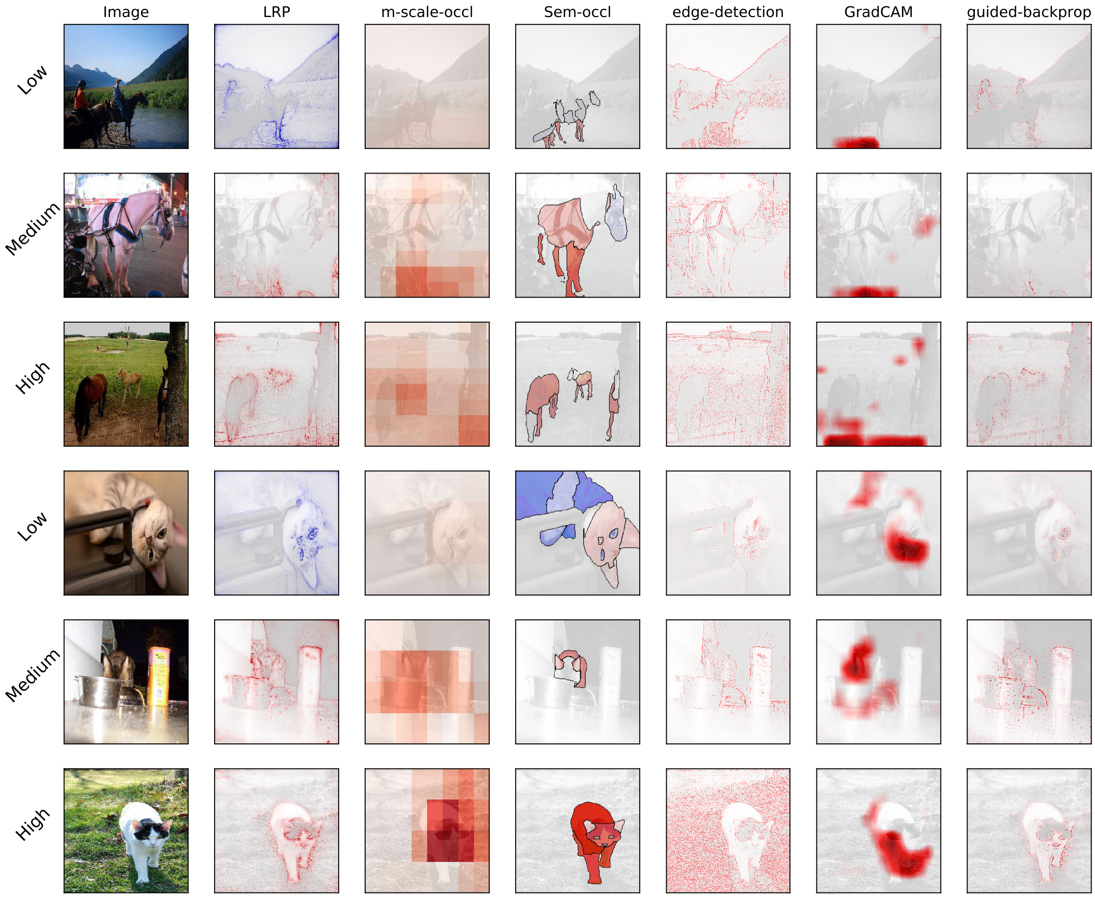


Figure 6.7: Generated saliency maps for the images used in the study (Q2-score-H, Q3-select-part and Q4-cover-part). The score is displayed on the left side: low, medium, or high

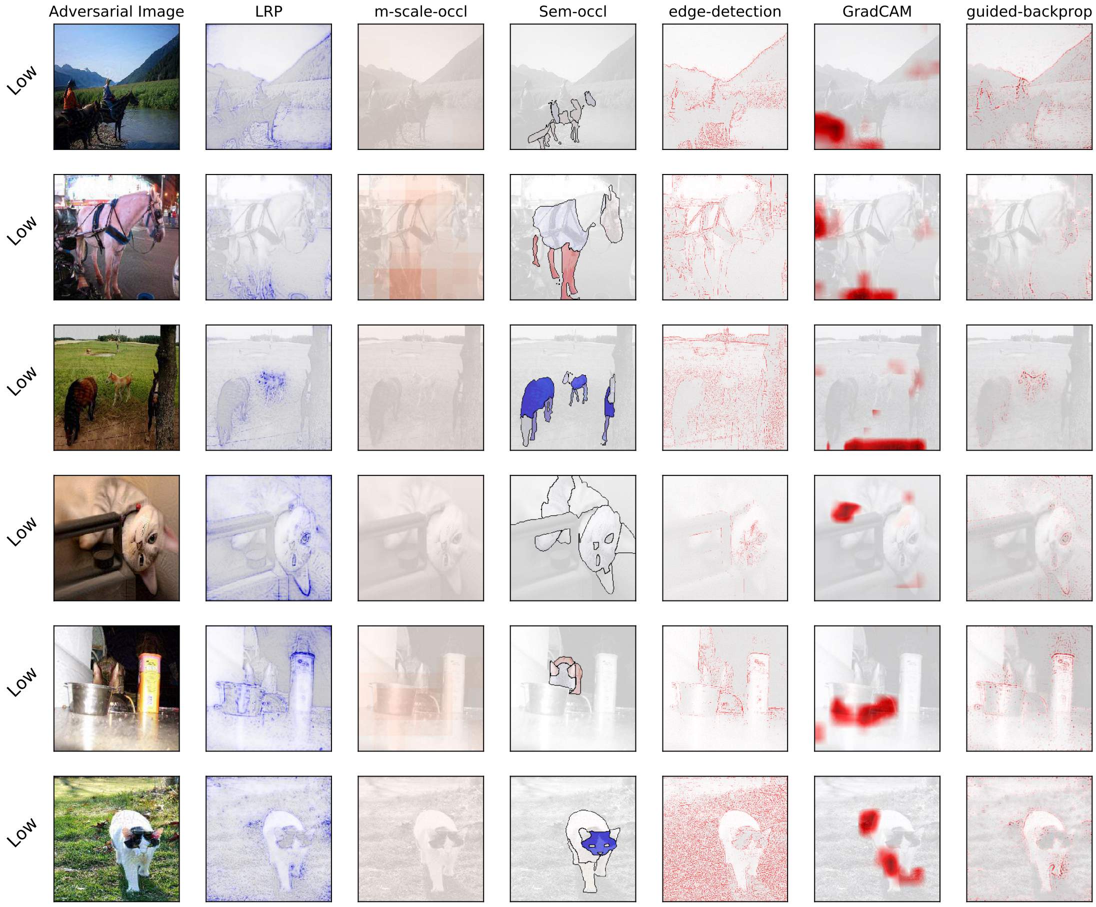


Figure 6.8: Generated saliency maps for the adversarial images used in the study (Q5-score-H and Q6-select-part). The score is displayed on the left side: low, medium, or high

6.2.1 Conditions

The study utilised a mixed factorial design with two independent variables (factors):

1. **The saliency map generation method** which is ***between subjects*** and has six levels which represents all techniques under assessments, which are: LRP, GradCAM, Guided-backpropagation, multi-scale occlusion (m-scale-occl), Semantic occlusion (sem-occl) and edge-detection.

2. **The sub-task type** (detailed in Section 6.2) which is ***within subjects*** and has six levels as well which represents all six sub-tasks: Q1-score-noH, Q2-score-H, Q3-select-part, Q4-cover-part, Q5-score-H and Q6-select-part.

6.2.2 Participants

We recruited 144 participants (24 per condition) through Prolific [3], with the same criteria mentioned in Chapter 4 (subsection 4.3.3).

6.2.3 Procedure

After providing informed consent, each participant went through a short tutorial (Section B.4) providing the necessary background about the experiment as well as clear instructions for using the system. The tutorial included examples of how the model classified a specific image. Further, we provided participants information about the explanation technique and how they can be interpreted. To avoid that one participant would spend lots of time going back and forth and another would not, examples were not presented during the actual task.

Upon completion of the introduction, participants were asked to complete 6 tasks (each with 6 sub-tasks). At the end of the study, they were debriefed: feedback for each task was presented together with the corresponding amount of bonus reward earned.

6.3 Results

6.3.1 Data analysis: choices and processes

We ran two types of statistical tests. The first is concerned with the main effects. The second examines the statistics in relation to a specific sub-task.

In Section 6.3.2 we report on the main effects (i.e. main effect of condition, main effect of sub-task) using the Aligned Rank Transform (ART) (Wobbrock et al., 2011). ART is based on aligned ranks and was proposed to analyse Non-parametric data from multi-factor, where common non-parametric tests such as Friedman test is inadequate (Wobbrock et al., 2011). In our study, we have a between subjects factor (i.e. The saliency map generation method) and a within-subject factor (i.e. The sub-task type). Therefore, we chose to apply ART to check for the main effects which can then be followed by post-hoc pairwise comparisons within each factor.

To check if there is a difference between conditions within a specific sub-task, we additionally run a Kruskal-Wallis test (which is also based on ranks) followed by a Mann-Whitney U test for the post-hoc pairwise comparisons. In particular, we report on the results of

[3] https://prolific.ac/

- Q1-score-noH and Q2-score-H: in which we examined participants' ability to predict the CNN score for images with and without the presence of saliency map (Section 6.3.3.1).
- Q5-score-H: in which we examined the effect of saliency maps on participants' ability to predict the CNN score for adversarial images (Section 6.3.3.2)
- Q3-select-part, Q4-cover-part and Q6-select-part: in which we examined the effect of saliency maps on participants' ability to understand how different regions of the object contribute to the CNN score (Section 6.3.3.3 and Section 6.3.3.4).

In addition, to further understand the impact of each explanation on participants' prediction accuracy, we ran a Wilcoxon signed rank test when the saliency maps were present and when they were not (i.e. between (Q1-score-noH, Q2-score-H) and between (Q1-score-noH and Q5-score-H)). Wilcoxon signed rank test is suitable to compare between two related samples (i.e. within-subject settings). Finally, in Section 6.3.4 we investigate how participants performed overall across the different sub-tasks (regardless of which condition they came from) as well as across different CNN outcomes (i.e. when task images have a low, medium or high CNN score).

6.3.2 Main effects

The Aligned Rank Transform (ART) test showed a statistically significant difference between conditions (F= 12.2, p <0.001, df=5). It also showed that participants' performance in the various sub-tasks differed significantly (F= 67.9, p <0.01, df=5).

Further to this, we report on the contrasts between conditions in Table 6.1. Additionally, because Q1-score-noH, Q2-score-H and Q5-score-H are related to participants' ability to predict the CNN score, we report on the contrasts between these sub-tasks in Table 6.2.

	guided-backprop	GradCAM	edge-detection	LRP	m-scale-occl
GradCAM	0.9029				
edge-detection	0.7312	0.1516			
LRP	0.3362	0.9241	**0.0107**		
m-scale-occl	**<.0001**	**0.0015**	**<.0001**	**0.0357**	
Sem-occl	**0.0006**	**0.0212**	**<.0001**	0.2385	0.9671

Table 6.1: Overall contrasts between condition (p-values)

6.3.3 Contrasts between conditions within a specific sub-task

In the following subsections, we report on several statistical tests to check for contrasts between conditions *within* a specific sub-task.

contrast	p-value
Q1-score-noH vs Q2-score-H	<.0001
Q1-score-noH vs Q5-score-H	<.0001
Q2-score-H vs Q5-score-H	0.4398

Table 6.2: Overall contrasts between sub-tasks (p-values)

6.3.3.1 Participants ability to predict the CNN score (Q1-score-noH and Q2-score-H)

In Q1-score-noH and Q2-score-H, the percentage of correct prediction per participant served as the basis for our evaluation (Figure 6.9 and Figure 6.10).

A Kruskal-Wallis test showed that the performance of participants across conditions were similar in Q1-score-noH, where saliency map explanation is *not* present (*H(5)*= 3.8, *p*= 0.58). The same test revealed a statistically significant difference between conditions in Q2-score-H (*H(5)*= 13.9, *p*= 0.016).

A Mann-Whitney U test for post-hoc pairwise comparisons, with Benjamini–Hochberg correction, revealed that (Figure 6.10) *m-scale-occl* and *LRP* performed statistically significantly better than *GradCAM* and *edge-detection*.

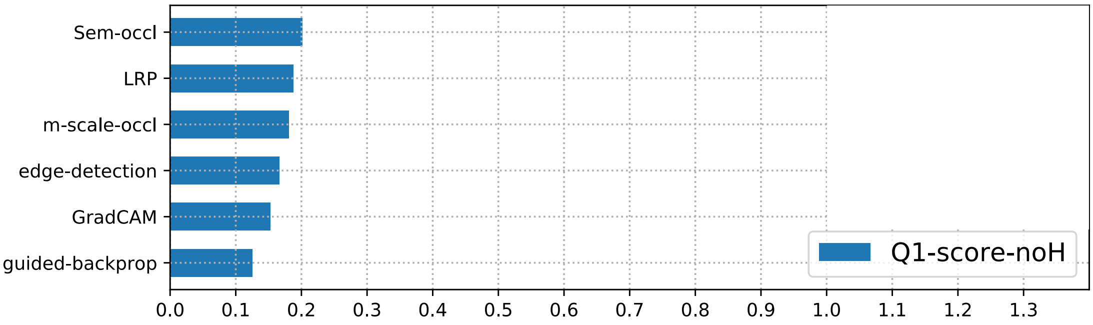


Figure 6.9: The percentage of correct answers across techniques. Lines denote that two techniques are (statistically) significantly different.

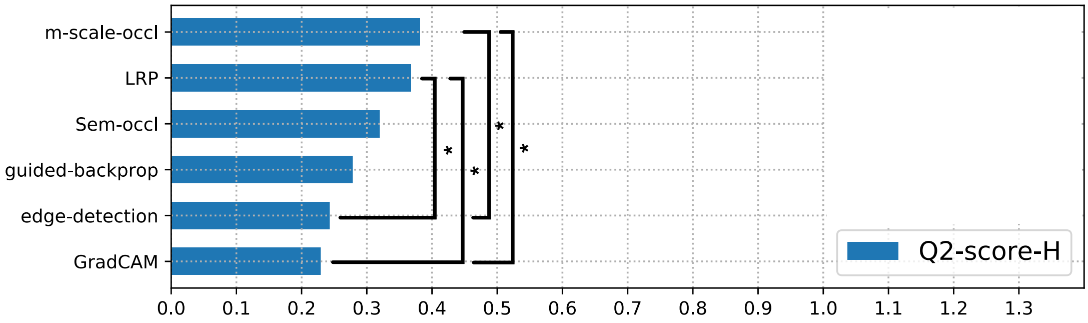


Figure 6.10: The percentage of correct answers across techniques. Lines denote that two techniques are (statistically) significantly different.

To further understand the impact of each explanation on participants' prediction accuracy, we ran a Wilcoxon signed rank test with Benjamini–Hochberg correction on the performance difference between Q1-score-noH and Q2-score-H (e.g. LRP Q1-score-noH vs LRP Q2-score-H). The test showed that the performance of participants has been improved significantly after showing the saliency map for all conditions (Table 6.3 - first column).

Condition	Q1-score-noH vs Q2-score-H (p-value)	Q1-score-noH vs Q5-score-H (p-value)
GradCAM	**0.023**	**0.0009**
guided-backprop	**0.003**	**0.0077**
LRP	**0.012**	**0.0077**
Sem-occl	**0.0216**	0.72
m-scale-occl	**0.0018**	**0.0002**
edge-detection	**0.0195**	0.72

Table 6.3: Wilcoxon signed rank test (corrected p-values) on the performance difference between (Q1-score-noH,Q2-score-H) and (Q1-score-noH,Q5-score-H)

6.3.3.2 Participants ability to predict the CNN score for adversarial images (Q5-score-H)

In Q5-score-H, we asked participants to predict the CNN classification outcomes of an adversarial version of the same images used in Q2-score-H (Figure 6.11). (saliency maps for those adversarial images are shown in Figure 6.8).

A Kruskal-Wallis test revealed a statistically significant difference between conditions on the performance (*H(5)*= 104.8, p = <0.001). A Mann-Whitney U test for post-hoc pairwise comparisons, adjusted with Benjamini–Hochberg procedure, showed that:

- *m-scale-occl* had a statistically significant better performance than the rest.
- *GradCAM* is significantly better than all except *m-scale-occl* and *LRP*.
- *LRP* is better than all except *m-scale-occl* and *GradCAM*.
- *Guided-backprob* is better than *edge-detection*.

In addition, we ran a Wilcoxon signed rank test with Benjamini–Hochberg correction on the performance difference between Q1-score-noH and Q5-score-H (e.g. LRP Q1-score-noH vs LRP Q5-score-H). The test showed that showing the saliency map in (Q5-score-H) helps in all conditions except *sem-occl* and *edge-detection* (Table 6.3 - last column).

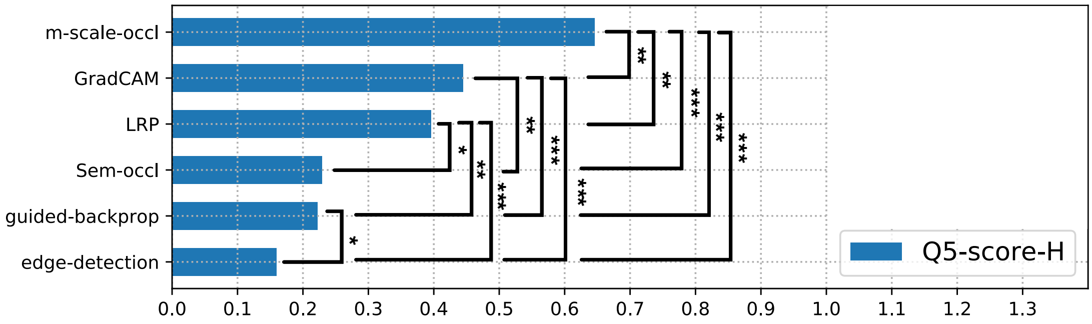


Figure 6.11: The percentage of correct answers across techniques. Lines denote that two techniques are (statistically) significantly different.

6.3.3.3 Participants ability to understand how different regions of the object contribute to the CNN score (Q3-select-part and Q4-cover-part)

In Q3-select-part and Q4-cover-part, we sought to see if saliency map methods might help participants understand how different parts of the object contribute to the CNN score. The summary of the results can be found in Figure 6.12 and Figure 6.13.

For both sub-tasks, a Kruskal-Wallis test revealed that participants' performance differed across conditions (**Q3-select-part:** *H(5)*= 32.12, *p*= <0.001, **Q4-cover-part:** *H(5)*= 19.4, *p*= 0.0016).

For Q3-select-part, a Mann-Whitney U test for post-hoc pairwise comparisons with Benjamini–Hochberg correction showed that participants presented with *Sem-occl* performed better than all other conditions (Figure 6.12). Similarly, for Q4-cover-part, the same test showed that *Sem-occl* condition performed better than all other ones except *m-scale-occl*.

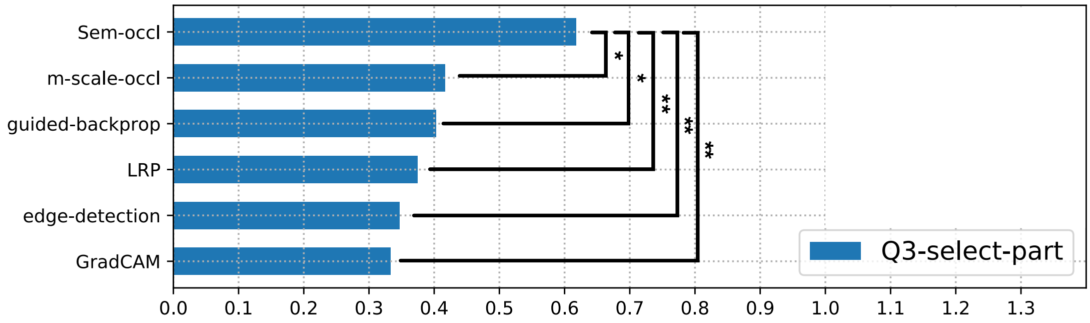


Figure 6.12: The percentage of correct answers across techniques. Lines denote that two techniques are (statistically) significantly different.

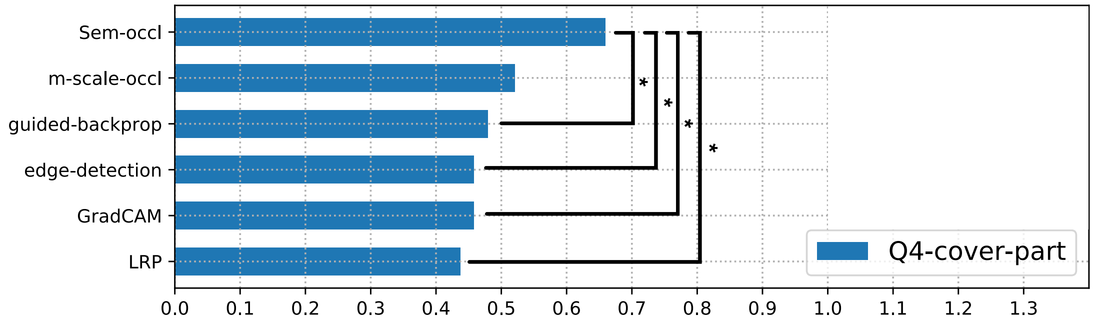


Figure 6.13: The percentage of correct answers across techniques. Lines denote that two techniques are (statistically) significantly different.

6.3.3.4 Participants ability to understand how different regions of the object contribute to the CNN score for adversarial images (Q6-select-part)

Similar to Q3-select-part, Q6-select-part asked participants to predict the performance of the system given an adversarial image and its corresponding saliency map. The summary of the results can be found in Figure 6.14.

A Kruskal-Wallis test revealed a statistically significant difference between conditions on the performance (*H(5)*= 48.3, *p* = <0.001). A Mann-Whitney U test for post-hoc pairwise comparisons showed that:

- *m-scale-occl* and *sem-occl* had a better performance than *guided-backprob* and *edge-detection*
- *LRP* and *GradCAM* performed better than *edge-detection*

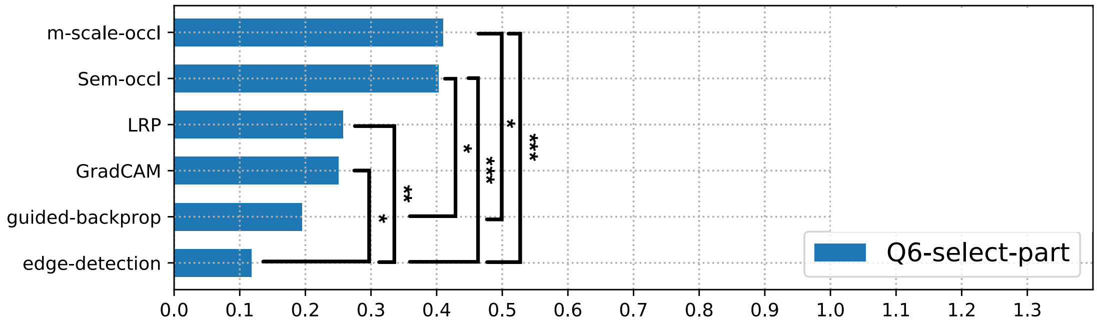


Figure 6.14: The percentage of correct answers across techniques. Lines denote that two techniques are (statistically) significantly different.

6.3.4 Participants overall scores, and across different CNN outcomes

As described above, the dataset of images used in this study included two images for which CNN score was high, two for which it was medium, and two for which it was low.

Figure 6.15 reports the average participant accuracy plotted against the CNN score (as well as overall). The figure shows the average score for all participants regardless of which condition they came from.

To reflect on the overall performance of the various saliency maps, we find it useful to compare the results to the chance level. Q1-score-noH, Q2-score-H and Q5-score-H are multiple choice questions with 5 levels (very low, low, medium, high, and very high), thus, the chance level is 1 out of 5 or 0.2. In relation to *users' ability to predict the CNN classification outcome on task images*, we found that overall the success rate in Q1-score-noH (no saliency map) is slightly lower than chance (average score=0.17) (figure 6.15), suggesting that the task is not easy. In contrast, when saliency maps were shown in Q2-score-H, participants across conditions predicted the classifier's outcome significantly more accurately than in Q1-score-noH (Table 6.2), reaching about 0.3. While this is above chance level, it remains low overall, suggesting that saliency maps do not provide a major gain when it comes to estimating the system's predictions.

It should be noted that for the two sub-tasks involving adversarial images (Q5-score-H and Q6-select-part) the CNN score was always *very low* (5 images) or *low* (1 image), in virtue of the definition of such images. Therefore data for these two sub-tasks is not included in this analysis. However, the overall average accuracy for Q5-score-H was 0.35 and for Q6-select-part was 0.27.

To better understand participants' performance in Q2-score-H, we broke down the results based on the different CNN scores on the study images (i.e. when the CNN produced low, medium and high scores), as illustrated in Figure 6.15. Participants across conditions did better at predicting the system's outcome when the CNN score was high (participants' average accuracy: 0.46). They were mostly struggling to predict the CNN outcome when it was low (participants' average accuracy: 0.23) or medium (participants' average accuracy: 0.22). In these cases, the results were similar to chance. The same effect does not hold for Q5-score-H, where the images were adversarial. For this sub-task, the CNN score was always *low*, in virtue of the definition of such images (see Figure 6.8). In this case, participants' overall performance was a bit higher (around 0.35). However, participants' average accuracy on images that *used to be high* in the original version is 0.26 (Q2-score-H with high CNN scores was 0.46), which may also confirm the observation above about overestimating the performance of the system.

6.4 Discussion

The results of the study show that different saliency map explanation techniques offer different information gains, highlighting the key strengths and limitations of each **(R5)**. At the beginning of this chapter, we raised a couple of research questions. In the

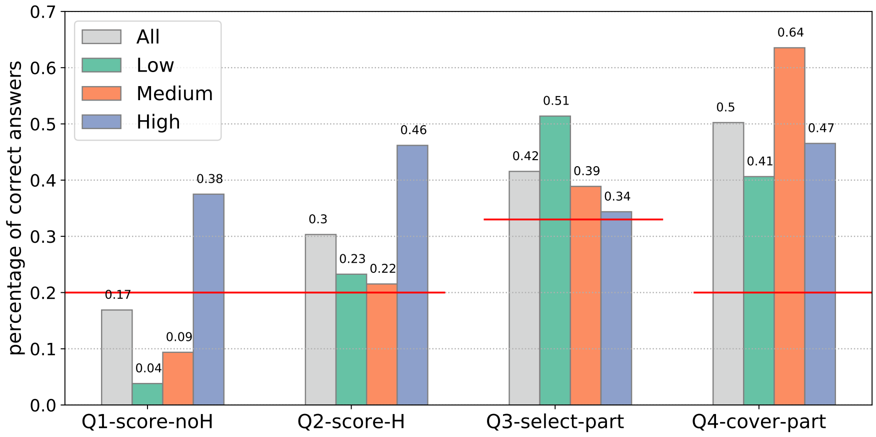


Figure 6.15: Participants' overall scores (All), and across different CNN outcomes (Low, Medium, High). The red horizontal line indicates the chance level

following subsections, we reflect on those questions and highlight implications for design and further research.

6.4.1 Participants' ability to predict the CNN score

All evaluated saliency maps techniques proved to help in the task of predicting the CNN classification outcome of an image. However, even after simplifying the task by presenting the corresponding saliency maps of the task images, overall participants' performance remained low (average score was around 0.3). It's worth noting that even edge detection seemed to assist participants in Q2-score-H. One possible explanation is that, while not being linked to the CNN model, it may draw attention to characteristics that are known to be selected from CNNs.

With respect to users' ability to predict the CNN classification score of adversarial images, we found that overall, saliency maps seem to help for most conditions. Adversarial images are defined as being almost identical to the original versions to the human eye, but not to the network. So that's where the explanations are most needed and helpful. However, we can observe that not all explanations are equal in their effectiveness. In particular, (1) edge-detection stops being helpful because the algorithm only detects the edges and does not take into account all of the CNN parameters. For that reason, the saliency map generated for the original image in Q2-score-H and the saliency map generated for the adversarial version in Q5-score-H appear nearly identical (for comparisons, please refer to Figure 6.7 and Figure 6.8 - the fifth column). (2) When it comes to sem-occl, its saliency map contains red and blue colours to represent the parts that look like the object of interest and those that do not (this is in contrast to m-scale-occl which only shows the red colour). We hypothesise that analysing both colours requires more cognitive load and can be hard to map to a score. In addition, sem-occl is a simplified but incomplete explanation in that only the parts of the object get highlighted, whereas CNN can classify an object using other patterns that are not necessarily related to the object (e.g. the grass around a horse). (3) Finally, for guided-backprob, although statistics show that it is helpful in Q5-score-H, this result is likely due to the low score of this condition in Q1-score-noH. If the score in Q1-score-noH was the same as other techniques (i.e. around 0.2), its significance would vanish. In (Adebayo et al., 2018), it is argued that guided-backprop is untrustworthy according to their evaluation test, which showed that such a method is invariant under model randomisation, which is compatible with our findings.

When we broke down the results for Q2-score-H based on the different CNN scores on the study images (i.e. when the CNN produced low, medium and high scores), we noticed that participants seem prone to overestimating the performance of the system: they expect the CNN score to be high even when it is actually low. One possible interpretation of this result is that it offers further evidence to the idea that we presented in Section 4.3.7.1: participants might be ”replacing” the difficult question ”does the CNN recognise a horse/cat in this image?” with the easier question ”do I recognise a horse/cat in this image?”.

6.4.2 Some saliency maps directly reflect the CNN score

Our results in Q2-score-H show that m-scale-occl and LRP helped the most when it comes to the sub-task of predicting the classification outcome (Figure 6.10). These results could be explained by considering the correlation between the presentation of a saliency map and the score value. LRP is a gradient-based technique that relies on a conservation principle (Bach et al., 2015b) which ensures that the back propagated score is preserved in each layer until it reaches the input space and is visualised as a saliency map. In contrast, any method that does a sort of normalisation (such as GradCAM) or loses part of the evidence while back-propagating through the network (such as guided-backprob) performed at the same level as the edge-detection condition (which does not take the model into account). For example, when examining the saliency map of the guided-backprob in Figure 6.7, we can notice that the high and low score saliency maps are visually similar. In fact, we investigated the participants' performance further in the guided-backprob condition, and found that they performed well in Q2-score-H for images with high CNN scores. In contrast, they performed poorly in the adversarial versions of these same images, i.e., Q5-score-H, which are in fact adversarial images with low scores.

M-scale-occl is generated according to the difference in value between the original score and the score after occluding a selective patch (or selective part of the object when it comes to the semantics technique). Thus, its saliency maps provide a clear indication of the score (see Figure 6.7), and it appeared to perform better in general. Basic forms of these occlusion techniques, on the other hand, are known to be computationally inefficient. Furthermore, the sem-occl technique, in particular, is based on human annotation of object parts, which represents a limitation. An implication for design then is the need for further improvements to such techniques, especially given recent developments in CNN-based segmentation algorithms (e.g. (Tao et al., 2020)) which can potentially automate the generation of sem-occl saliency maps.

When it comes to analysing the results of Q5-score-H, we would expect a truthfulness explanation (saliency map) not to be fooled by an adversarial image. M-scale-occl performed statistically significantly better than all conditions (Figure 6.11). The second best were LRP and GradCAM. For LRP, because all scores for adversarial images are low, associated LRP saliency maps were always in the shades of blue (see Figure 6.8), which makes the user's forecasting relatively easy. For GradCAM, one can notice that its saliency maps are not localised to pixels in the proximity of the object of interest for adversarial images compared with the saliency maps generated for the original images, which may have helped participants to predict that the score of the system is unlikely to be high for the given adversarial images.

Given these differences, the intended task should guide the saliency map approach selection; for example, if the goal is to detect errors or find a unique pattern in the data, a method that applies some sort of normalisation should suffice (e.g. GradCAM). An application that requires some form of mapping between the saliency map and the score (e.g. predicting the CNN outcome) would, on the other hand, require a technique that meets these criteria.

6.4.3 Highlighted regions in some saliency map techniques could be misleading

For **Q3-select-part**, the sem-occl performed significantly better than the rest, as shown in Figure 6.12. It's interesting to note that no condition performed better than edge-detection, apart from sem-occl. This is fairly surprising, since we expected other approaches to highlight features (parts) other than the general edges to which the model responds. For **Q4-cover-part**, because we have 3 multiple choices (increase, decrease, or stay almost the same), the chance level is 0.33. Similar to Q3-select-part, all conditions except sem-occl performed in a similar level to the baseline condition: edge-detection. The situation for **Q6-select-part** appears to be more complicated; sem-occl has stopped to be the best, possibly because, when looking at its associated saliency maps (Figure 6.8), and given that the CNN score for all the provided adversarial images was low, there is no clear indication of which part will reduce the score the most if covered (with the exception of image 2 and image 5). It's also worth noting that guided-backprob did not perform well in this task, which may support previous research (Adebayo et al., 2018) that suggests that guided-backprop is not truthful.

Since the design of sem-occl matches these tasks (i.e., highlighting the effect of occluding different parts of the object), its superior performance in Q3-select-part and Q4-cover-part was to be expected. It's worth remembering how Q3-select-part and Q4-cover-part were defined to better understand the result. In Q3-select-part, participants were asked to choose the part of the animal that would cause the score to drop the most. A saliency map technique has to provide a fine-grained indication of the importance of the different parts to help users achieve this task. The formulation in Q4 is slightly different, which asks whether covering part of the animal will shift the score in either direction or hold it almost the same. This sub-task is more challenging since we discovered that removing a part of an image does not guarantee that the score will change; other patterns may exist that are sufficient for the CNN to recognise an object. This point reminds us that the learning process of CNN is complex and can be non-linear.

As a summary, and in response to *users' ability to quantify the importance of individual parts of an image*, the saliency map techniques we examined do not appear to help in quantifying the significance of individual parts (semantics) in an image. The inclusion of the sem-occl condition was mainly for the purpose of contrast. However, since its performance in Q3-select-part and Q4-cover-part was better than other techniques, this may suggest that developing a technique in that direction could be promising. Furthermore, as we mentioned earlier, quantifying a part's relevance across a large number of examples may help in gaining a global perspective about how relevant this part is to the classification process. All saliency maps considered in this study are designed to convey what is essential for the classification score, but not the contrasting question: what if this part is occluded or does not exist? This, perhaps, could be a first step towards the overarching aim of providing an explanation that relies on a cause-and-effect scheme rather than a simple correlation.

6.5 Summary

This chapter reported on a between-group user study designed to evaluate the utility of "saliency maps". In contrast to our previous studies reported in Chapter 4 and Chapter 5, in this study, we assess the utility of saliency map techniques through a task of lower complexity, in which saliency maps are presented alongside the task image. Reflecting on our research questions we raised earlier, we can summarise the key findings of this study as follows:

- All evaluated saliency map techniques proved to help in the task of predicting the CNN classification outcome of an image. However, even after simplifying the task by presenting the corresponding saliency maps of the task images, overall participants' performance remained low (average score was around 0.3).

- For the task of predicting the CNN classification outcome of adversarial images, all techniques helped as well except sem-occl.
- The saliency map techniques we examined do not appear to help in quantifying the significance of individual parts (semantics) in an image nor to estimate how the classification score would change if a part would not be present in the image.
- Results of the study show that different saliency map explanation techniques offer different information gains (R5).

Results also suggest that techniques based on some form of occlusion could be promising, as such techniques have the characteristics of being model-agnostic and more truthful than gradient-based techniques. They do, however, have some limitations. For example, although sem-occl was designed to highlight features that are meaningful to people, such visualisation represents a simplified and incomplete explanation because other patterns that are not necessarily connected to the object (e.g. the context) may also contribute to image classification. Finally, because the results highlight that different saliency map generation techniques provide varied levels of value, selecting a saliency map technique should be in light of the intended task.

Chapter 7

General Discussion and Conclusions

This chapter provides a summary of the thesis, and a general discussion of the reported studies and how they are related, as well as future research directions.

7.1 Summary and Key Findings

In Chapter 1 and Chapter 2, we discussed that while ”data-driven” ML systems have grown common and successful in a wide range of disciplines, many of these algorithms are currently opaque boxes. We pointed out how previous research has suggested that explanation techniques can help detect unexpected behaviour and build appropriate trust in the system. Despite the fact that various explanation techniques have been developed and analytically examined, we highlight that only a limited number of user studies have been carried out to determine their utility to users. Based on this premise, we mentioned that the main objective of this thesis is to investigate the role of explanations in informing end-users. We demonstrated that evaluating complex systems is challenging, where there is a need to strike a balance between multiple factors, which include the number of participants, the duration of the study, and the variation of experimental factors. As a result, we decided to specify a defined scope in which we would try to focus on specific types of users, datasets, models, explanation techniques, and the evaluation measure choice. We came to the conclusion that analysing explanation techniques in these contexts would provide useful design guidelines for explanation techniques.

In Chapter 3, we examined the role of visual feedback in informing user understanding in the context of pattern recognition systems. Through the application of creating a stop-motion animation, we designed and conducted a between-groups study with four conditions with the aim of seeing if visual feedback obtained from various stages of the processing pipeline can assist a user in a task of automatic alignment of frames that results in a stop-motion animation. Our findings show that Keypoint markers can help users in building better functional understanding, so long as the meaning being conveyed is inline with user expectations **(R1)**. In particular, participants who received Keypoint markers derived from later stages (higher levels) of the processing pipeline demonstrated an improved understanding of the system operation compared to explanations derived from an early stage (lower level) of the pipeline **(R2)**. A main implication from the study was that when designing an explanation for users who are not experts in AI, we should seek an explanation that provides a ”functional understanding” of the system and how it will be utilised rather than an explanation that focuses on the system's inner workings.

In Chapter 4, we discussed how CNNs became the preferred algorithm for computer vision applications, then explained their main limitations and how explanation techniques are required. We also explained why we chose saliency maps to investigate. We then begin by conducting an online study investigating the role of saliency maps in informing technical users. In particular, the following research question was defined:

(R3): How do saliency maps help with building functional understanding, including the relation to varied system confidence?.

To examine users' functional understanding, participants were asked to estimate the CNN outcome on task images, with 12 examples (each with a similar score) shown to help them make a decision. Our data highlighted a number of key findings which includes: (1) There was no significant difference between conditions in terms of correct guessing of whether images would be correctly or incorrectly classified. (2) Across conditions, higher scores seem to be easier to predict than lower scores, and saliency maps do not appear to aid in this regard. (3) When images with low CNN scores were sampled, features were mentioned a lot less frequently by our participants suggesting that the utility of saliency maps varies according to the classification score. (4) Participants applied different strategies to reason about the provided examples trying to find patterns about the system's behaviour, highlighting the limitation of saliency maps being defined on individual points. (5) we report on some instances in which despite having access to the saliency maps some participants expected the system to understand human high-level concepts, where in reality, CNN learns patterns in a bottom-top hierarchy fashion in which meaningful patterns that *look like* what we humans refer to as ”semantics” may emerge in the deep layers of the network, but that is not guaranteed. (6) Finally, a main theme that emerged from the study was the mentioning of features across all conditions. Findings showed that for correct answers, saliency map participants often rely on features that can be highlighted by the saliency map (i.e. Saliency-Features), while participants in the no-saliency map condition do not.

Building on the findings of the previous study, Chapter 5 focuses on features by displaying examples and task images that share similar patterns, utilising the power of CNN-embeddings. We specifically seek to understand the kind of features users attend to by asking them to explicitly list the features they think the system is sensitive to. In particular, the following research question was defined:

(R4): What features do lay users attend to in order to build a functional understanding of computer vision processes?.

Results showed that presenting saliency maps seem to prime our participants to attend to features that can be highlighted by saliency maps and give less attention to other attributes (e.g. colour, contrast) that may also influence the CNN outcome but cannot be directly highlighted by the saliency map. Generally, our data indicates that lay users have a tendency to interpret explanations as an outcome rather than a progress of an intermediary stage. Moreover, in this study, participants were significantly more accurate in predicting the outcome of the classifier when saliency maps were shown, nonetheless, the total success rate remained low (R3). In an attempt to explain this moderate outcome, we discussed a number of factors that may have contributed to it, which includes the tasks' complexity, which required participants to learn complex patterns from a few samples and then apply this knowledge to a new task image, had overshadowed what the saliency map truly offered. Although we discovered an improvement over the previous study's results, the task's complexity may still remain.

Therefore, to limit possible confounds that may emerge as a consequence of the complexity of the study design which involves a variety of factors, in Chapter 6, we devised a task that, unlike earlier studies, assesses what the different explanation techniques communicate to users by presenting them alongside the task image. Moreover, informed by the results of our previous studies which indicate that when saliency maps were present, participants mentioned Saliency-Features more often. we designed and evaluated two new saliency map techniques that focus on features that are meaningful to people. Given the new task, we raised the following research question:

(R5): How do different saliency map generation techniques perform to build functional understanding?.

Results showed that (1) all evaluated saliency map techniques proved to help in the task of predicting the CNN classification outcome of an image. However, even after simplifying the task by presenting the corresponding saliency maps of the task images, overall participants' performance remained low. (2) For the task of predicting the CNN classification outcome of adversarial images, all techniques helped as well except sem-occl. (3) The saliency map techniques we examined do not appear to help in quantifying the significance of individual parts (semantics) in an image nor to estimate how the classification score would change if a part would not be present in the image. (4) findings indicate that techniques appear to perform differently across tasks. As a result, the saliency map technique should be guided by the intended task. (5) techniques based on occlusion performed better overall, especially when images were adversarial. Basic forms of these occlusion techniques, on the other hand, are known to be computationally inefficient, providing a clear possibility for optimisations, especially given recent developments in CNN-based segmentation algorithms.

Table 7.1 summarises the contributions made by this thesis.

7.2 Design Implications

In this section, we will summarise several design implications that have been highlighted throughout this thesis.

Chapter 3 Design Implications

- When designing an explanation for users who are not experts in AI, we should seek an explanation that provides a ”functional understanding” of the system and how it will be utilised rather than an explanation that focuses on the system’s inner workings.
- Choosing the stage of processing from which feedback is derived is important in users’ ability to construct coherent understandings of a system’s operation.
- Misunderstanding a feedback could result in a worse functional performance than received no feedback at all.

Chapter 4 and 5 Design Implications

- Saliency maps should be complemented by global descriptors such as overall contrast, brightness, and histograms to constrain the locality effect.
- Human expectation can give emphasis on some factors that are not the main cause of a model’s failure, which highlight the importance of developing explanations that convey the right expectation to users.
- Further research needs to be conducted to better characterise the effect of different sampling strategies on users interpretation of the system operation.
- Further studies could be conducted to measure whether the inclusion of saliency maps in the UI changes users behaviour.

Contribution	Chapter
Conducted an investigation of how feedback from different stages of the processing pipeline helps users build a functional understanding of the computer vision processes. Our findings indicate that participants who received keypoint markers derived from later stages (higher levels) of the processing pipeline demonstrated an improved understanding of the system operation compared to explanations derived from an early stage (lower level) of the pipeline.	Chapter 3
Examined the role of saliency maps in informing end users through three user studies. The thesis provided a number of key findings. For example, saliency maps could help participants predict the outcome of the model, but overall, the success rates were relatively low.	Chapter 4, 5, 6
Conducted a study to investigate the features lay users attend to in order to build a functional understanding of computer vision processes. Our findings indicate that saliency maps appear to prime participants to primarily focus on what saliency maps highlight (which we called Saliency-Features), but potentially distracting them from other attributes such as colour and contrast, which saliency maps cannot highlight.	Chapter 5
Compared between different saliency map generation techniques in terms of their impact on building functional understanding. The utility of different saliency map approaches appears to vary depending on the task at hand.	Chapter 6
Presented methodological contribution in the form of how to design user studies for evaluating explanation techniques.	all studies chapters
Highlighted a number of implications for the design of explanation techniques and further research in that area.	all studies chapters
Investigated several methods for selecting and displaying example images to users, one of which was influenced by the results of our first saliency map study.	Chapter 4, 5
Developed, implemented and evaluated two novel occlusion-based saliency map techniques: ”semantic occlusion” (sem-occl) and ”multi-scale occlusion” (m-scale-occl).	Chapter 6
Developed a number of helpful tools to aid in testing and understanding on how to design our studies.	Chapter 6, Appendix A

Table 7.1: A summary of the contributions made by this thesis

Chapter 5 Design Implications

- Choosing representative examples with their corresponding saliency maps, which summarise the behaviour of the system well, is an under-explored topic.
- Need to develop explanation algorithms that bridge the gap between humans and machines by leading the user to understand that the system is not basing its classification decision on higher- level 'semantics' of the image.
- The utility of saliency maps for low score images is negligible suggesting that new techniques which provides counter-factual evidence is needed.
- Transferring knowledge about potential features to new images, where they are presented in different orientations, scales, forms and perspectives, is very challenging. Therefore, saliency maps should be complemented by a global measure that explains how sensitive the presence of a feature is to the prediction of some class.

Chapter 6 Design Implications

- Because different saliency map explanation techniques offer different information gains, selecting a saliency map technique should be in light of the intended task.
- Occlusion-based techniques performed better, especially when images are adversarial. Design implications include the need for further improvements to such techniques, especially given recent developments in DNN-based segmentation algorithms which can automate the generation of sem-occl saliency maps.
- Highlighted regions in some saliency maps techniques could be misleading as they might give the wrong impression that covering these regions would result in a lower prediction score.

7.3 Discussion

7.3.1 Seeking a functional understanding of the system

Findings from all the studies in the thesis indicate the importance of deriving an explanation that is inline with the user's expectations. In the following paragraphs, we shall reflect on some aspects that are related to this notion.

The first is related to **how explanations are generated**. When we reflect on some of the AI algorithms, we may notice that some of their aspects can be easily mapped to visual representations. For example, keypoint markers (Chapter 3) are often naturally represented as dot visualisation over-laying an image. Similarly, a saliency map is a simple and natural map of the network's neuron activations to a visual representation that overlays the image (Chapter 4, 5, 6). The simplicity and readiness of these representations provoke a tendency to use them for explanation. However, we must ensure that such a representation is truly in line with what a user would expect.

"Key point markers", for example, may provide an accurate representation of one stage of the processing pipeline, but our results indicate that participants related them to the outcome of the algorithm (i.e. indicated regions where the stabilisation process had identified matches), while in reality they are obtained from an early stage of the processing pipeline. Therefore, we argue that when designing an explanation that targets end users, we should distance the user from understanding the internal states of the algorithm, and instead consider the end result of the system and how it will be used. This finding is also inline with Kulesza et al. (2013) work, in which they showed that completeness (i.e., to what extent the feedback describes all of the underlying processes of the system) is more significant than soundness (the accuracy with which the feedback accurately reflects the underlying process of the system). In our context, "key point markers" may provide a sound but incomplete representation, as it explains only one stage of the processing pipeline.

The situation for saliency maps is different. Saliency maps tend to highlight all patterns that contribute to the final outcome (because they are designed to summarise the contribution of all neurons in the network), but they are still incomplete because they are defined on individual instances.So, a complete *functional understanding* is hard to achieve without observing many representative examples, a topic that will be discussed in more detail in Section 7.3.3 below.

When it comes to assessing the soundness of saliency map techniques, we should remind ourselves that saliency maps represent a separate (though related) process from how CNN analyses data, and this separate process is not guaranteed to be accurate. In Chapter 6, we showed that some saliency map techniques are not truthful (e.g. guided-backprob), in the sense that the generated saliency map for an image and its adversarial version look almost identical, even though the CNN outcome is very different between the two. Results of the study showed that such similar visualisation of the two versions led participants to a worse performance in that sub-task. In addition, the locality aspect of saliency maps adds another degree of complexity and potentially confusion, where locality in this context refers to the fact that changing a few pixels in the input image can result in a significantly different saliency map. (Lipton, 2018; Ghorbani et al., 2019) In addition to the soundness and completeness factors, another dimension to consider is understanding what patterns are required by a model to classify an image. For example, the saliency map visualisation highlights what supports the prediction of a class (e.g. cat). This may include, for example, the eyes and the nose of the cat, however, this visualisation does not tell us whether the presence of all (i.e. eyes and nose) or just some of the patterns is required by the model. In Chapter 6, we studied this point and discovered that, for some task images, covering some of the highlighted regions (e.g. eyes of the cat) had no effect on the classification outcome. Participants who were exposed to various saliency map generation techniques, however, predicted that the score would drop if this particular ”highlighted” part was covered.

7.3.2 Do explanations contribute to an overestimation of the system's capabilities?

Across the different studies, we reported instances in which explanation may have contributed to participants over-estimating what the system is actually doing. For example, in Chapter 3, participants in the Keypoints condition tended to overestimate the meaning of the key feedback and associate it with higher level concepts such as the separation of background and foreground objects.

With heatmeaps, although this technique represents an approximate representation of the output of the algorithm as it summarises the neuron activations across the whole network, there were also instances of over-estimation. For example, in CNN, the learning process has an iterative nature, in which the system slowly updates its parameters to reach the optimum model. This learning process has a bottom-top hierarchy in which several works showed (through saliency maps) that a meaningful patterns that *look like* what we humans refer to as ”semantics” may emerge in the network's deep layers. We reported in Chapter 4 cases in which some participants expected the system to have a global label of an object (e.g. animal), and saliency maps may have contributed to this understanding. We conclude that proposing techniques that attempt to bridge the gap between concepts defined by users and how these concepts are represented by the model is important.

In Chapter 6, we introduced sem-occl as an attempt to achieve this goal. However, as we discussed before, this representation has some limitations. First, it may contribute more towards this inaccurate understanding that the system has a global label for an object. Second, it may allude to the understanding that CNN processes parts or features independently, where in fact the learning process of CNN is more complex and can be non-linear (e.g. there might be an interaction effect between multiple features). Finally, this technique does not highlight other patterns that are not part of the object, which can also contribute to image classification.

7.3.3 Limitations of instance-level explanations

Some explanation techniques are defined on individual data points rather than on a global scale. This fact suggests that such explanations are incomplete and may lead users to an inaccurate understanding of the model if the displayed saliency maps are not representative instances. In fact, building a coherent understanding of the underlying model by examining these individual instances can exceed users' cognitive load. As a result, previous work argued that it is critical to select and display representative data points or to find ways to summarise these individual instances. However, when it comes to dealing with images, we found the situation to be more complex; creating a suitable aggregate representation of data types such as images is not obviously clear.

To mitigate the impact of this limitation, previous work (Miller, 2019a) suggested to apply the contrasting strategy in which two examples with different outcome displayed. Our results complement the suggested strategy in most cases, but also report the contrary in others. In particular, in Chapter 3: we report on instances in which the user's understanding has been improved by looking at contrasting cases. However, we also witnessed cases where participants failed to correct their misunderstanding despite witnessing evidence to the contrary; a behaviour pattern previously reported in work on intelligent system Tullio et al. (2007).

Qualitative data reported in Chapter 4 revealed that participants use examples and employ various types of reasoning to explain why a specific image would be recognised or not recognised by the model. Examples of such reasoning include the reference to a general similarity to a specific example, or group of examples. or even to some general aspect such as "image brightness". In addition, multiple comments justify their choice by referencing similar TPs or FNs or specific features of those outcomes.

In Chapter 5, we attempt to improve the way of selecting examples to show images that are visually similar to the "task image" (i.e. using CNN embeddings). Previous work showed that embeddings may include a high-level representation of what people refer to as semantics. However, even with these attempts, the task seems to exceed the user's capacity to build a global understanding of a model that has learnt complex patterns in which objects are often displayed in different orientations, scales, forms and perspectives. Finally, for instance-level explanations, we would also like to emphasise the importance of studying how many images to show so that participants can learn from multiple instances. In our studies that examine saliency maps, one challenge was to display an adequate number of examples with the aim of being informative but not overwhelming. Although previous works (please see Section 2.5) have investigated multiple aspects in that area, we believe there is still room for contribution.

7.3.4 Explaining challenging cases

One strategy we used in our studies to inform participants about the system's behaviour was to display instances where the system is likely to succeed and other instances where it is not. The role of explanation in these distinctive cases is to highlight the reasons for possible successes and failures. Recall for keypoint-markers, tasks were designed to expose participants to “feature rich” backgrounds, where the algorithm is likely to work, and “feature poor” backgrounds, where the algorithm is likely to fail in the stabilisation process. Keypoint-markers appear to be useful in some cases, particularly when they are obtained from a later stage of the processing pipeline (i.e. Matching-Keypoints condition) as opposed to earlier stages (i.e. Keypoints condition). We applied a similar strategy with saliency maps displaying contrasting examples (i.e. TPs, FNs and FPs). However, for both explanations (keypoints and saliency maps), the absence of evidence was the only indicator that a system had failed to function with an instance (e.g. no enough keypoints in the image, or no highlighted pixels for saliency maps). A more helpful explanation would be to explain why the evidence is missing. For example, to inform a user about the behaviour of a keypoint matching algorithm, a more helpful explanation should indicate that distinctive shapes and features are required for the algorithm to perform the task.

7.4 Future Work

The investigations described in this thesis identified a number of potential areas for future research. In this section, we highlight some limitations and potential possibilities for further research. Because the design space for the studies we presented is vast, we made some design decisions that represent both a limitation and potential avenues for future work.

The first limitation is the small number of image classes we considered. We decided for this compromise considering the limited time for each session, and the limited knowledge participants would have been able to obtain about class-specific behaviour. Future work should run a long-term evaluation (i.e. lasting several days or weeks) to allow participants to explore a large dataset with multiple classes in more depth. In addition, it would be valuable to run field studies with participants who can apply these models in real world applications. As mentioned previously, lab studies are faster, cheaper, and more regulated than field studies. In field studies, on the other hand, participants can be exposed to saliency maps in more realistic settings. However, evaluating saliency maps in field studies requires controlled and well-motivated experimental tasks.

Another limitation of our design is the usage of one specific network architecture (VGG16 (Simonyan and Zisserman, 2014)) and one specific technique to generate saliency maps for the first two studies (LRP (Bach et al., 2015b)). Through a series of pilot studies, we have tried to explore other techniques that provide saliency maps that participants found to be informative. However, this also means that results might change with a different combination of techniques. Moreover, our participants were required to have a technical background, whereas in our studies, we did not control for ML expertise. We see a potential opportunity to repeat our studies with different participant populations, such as ML-experts or lay users.

For the first two saliency map evaluation studies, we considered the user's ability to predict the outcome of a ML classifier as a measure to assess how transparent or explainable a system is. This measure has been proposed and utilised in other studies (Lipton, 2018; Muramatsu and Pratt, 2001; Poursabzi-Sangdeh et al., 2018). However, there are potential avenues to discover the value of saliency maps from varied angles. Figure 2.8 shows a number of alternative measures that could be used to evaluate saliency maps. For example, as we mentioned in our studies, generally, when saliency maps were present, participants referenced Saliency-Features more often. A possible research question would be to investigate whether participants who receive the saliency maps are better at providing specific suggestions (informed by the highlighted features) on ways to improve the model accuracy. This implication aligns with the argument that Lipton (2018) put forward by mentioning informativeness as one of the desiderata of interpretability research. He suggested that although the outcome of the model is the obvious way of conveying information to people, other intermediate aspects might also convey useful information.

Training ML models on specific datasets may not necessarily result in good generalisation. When the system is deployed in the field, new captured instances could come from a different probability distribution. For example, Nguyen et al. (2015), showed how to easily deceive a state-of-the-art CNN by feeding it images with certain patterns that a model responds to. These images can be easily verified "visually" by people, and yet the model incorrectly classifies the image with very high confidence. In less obvious cases, the problem becomes more challenging. For example, our results in Chapter 6 showed that different generation techniques provide different utility in helping participants predict the CNN classification outcome of adversarial images. However, adversarial attacks and biases can take different forms, highlighting an opportunity for future work in that area.

With current advancements in computational capabilities, there are opportunities to introduce an interactive element to such systems. By so doing, this may perhaps help in situations which require a substantial cognitive load and in other situations where the user would have the ability to ask questions such as "what if" instead of just observing the output of a static model. This argument is also supported by (Abdul et al., 2018b), where they point out that explanation techniques reported in the explainable AI literature are mainly static. They argue that interaction would be a promising direction to explore and derive insights from complex models. In Appendix A, a number of interactive tools have been developed to explore and understand image datasets.

Saliency maps generated from images with low scores do not appear to be useful. Our findings show that saliency maps can emphasise what supports the prediction of a class, but they cannot explain what is missing in the image. For example, if the ”eyes” are the only component of the cat that contributes to the prediction, saliency maps will typically highlight the ”eyes” as evidence. Saliency maps, on the other hand, will highlight nothing if the ”eyes” are covered. In other words, saliency maps do not show which patterns are missing in the image so that the model can recognise them. CNN is well-known for its ability to acquire complicated and abstract visual concepts in its deep layers (Chollet, 2017). Inspecting several channels in a deep layer reveals that the network learns concepts such as ”eyes” and ”legs.” where these concepts are typically learnt through multiple channels. One possible direction for future research is to investigate the possibility of utilising this special property of CNN by identifying the existence of a specific concept (e.g., nose), then feeding this information back to the user as an explanation.

Finally, future work should assess the effect of saliency maps in terms of trust, or measure whether the inclusion of saliency maps in the UI changes users' behaviour, along the lines of what (Verame et al., 2016) evaluated for confidence information.

7.5 Conclusion

This thesis reports on a series of user studies which evaluate the role of explanation in informing end-user understanding of complex system decisions. Building on prior work, the thesis highlights the importance of explaining complex system decisions and underlines the need to evaluate the proposed explanation techniques through user studies.

Two explanation techniques were evaluated. The first is keypoint markers, which are often derived from a keypoint matching algorithm. Our results indicate that they can be more informative to users when obtained from the later stages of the processing pipeline, as this is more inline with user expectations.

The second is saliency maps produced to explain the decisions of Convolutional Neural Networks (CNNs). Our results indicate that the presence of saliency maps helps participants predict the outcome of the model, but overall, the success rate is relatively low. Through a combination of quantitative and qualitative methods, our data highlights a number of key findings that may explain this moderate outcome. Our data shows that:

- saliency maps might prime users to consider features that can be highlighted by the saliency map (Saliency-Features) and give less weight to other attributes such as colour and contrast.
- when images with low CNN scores were sampled, features were mentioned a lot less frequently than when images were sampled with a high CNN score, suggesting that the utility of saliency maps varies according to the classification score.

- participants applied different strategies to reason about the provided examples and build a pattern about the system's behaviour, highlighting the limitation of saliency maps being defined on individual points.
- some saliency map generation techniques do not seem to help in predicting the CNN score of adversarial images.
- saliency maps can be misleading in that the highlighted regions do not always indicate that covering these regions would result in a lower prediction score.

Overall, we argue that reaching a solid understanding of how the CNN model classifies images is not possible with the sole use of instance-level based explanations (of which saliency maps are an example). Even with very informative examples, saliency maps can only highlight the importance of features that are localisable to pixel-regions. We suggest using saliency maps in conjunction with other more global explanation methods. Furthermore, we view saliency map sampling strategies (i.e. what instances to display to users) as a promising direction for future research. Finally, while developing or deploying an explanation for non-expert users, we would want to emphasise the importance of looking for an explanation that offers a "functional understanding" of the system and how it will be used rather than one that concentrates on the underlying algorithm. We hope that the work presented in this thesis will promote discussions and future research on interaction with complex systems, as well as draw attention to the importance of creating and evaluating new explanation techniques centred around human needs.

Appendix A

Tools

We have developed a number of tools to aid in testing and understanding on how to design our studies. In this appendix chapter, we introduce each tool briefly and show a screenshot of how it looks and how it may be used.

A.1 The browser

This is a helpful tool to navigate a dataset based on multiple aspects such as the category and the classification outcome. Figure A.1 shows a screenshot of the interface. In (1), the user can filter images by category (e.g. horse, dog). In (2), a subset of images can be displayed based on whether the image is considered as TP, FN or FP. The user can

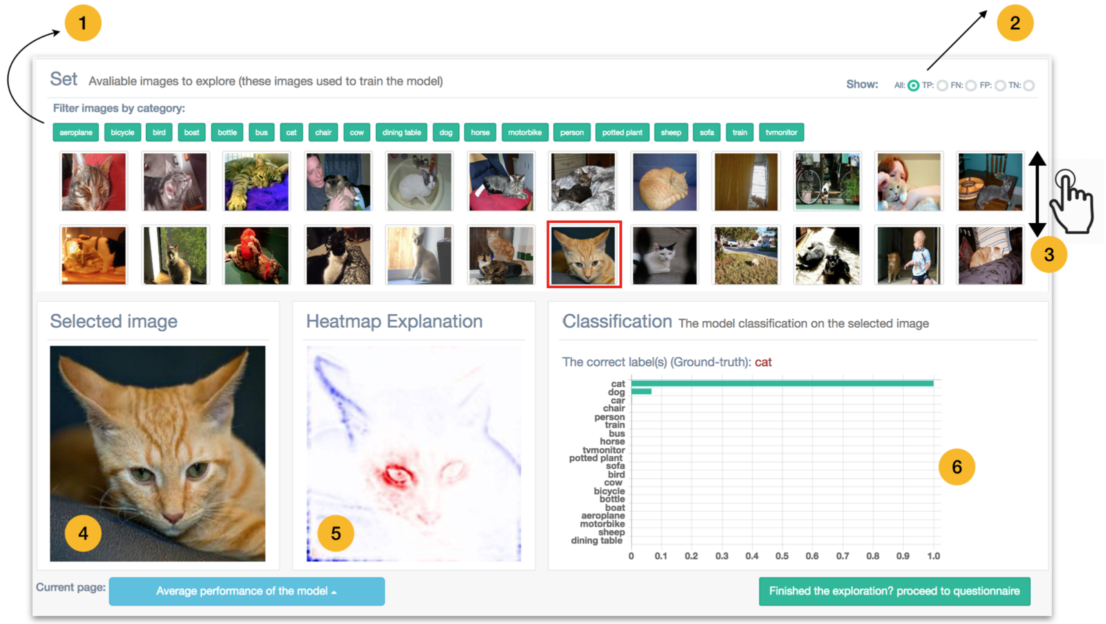
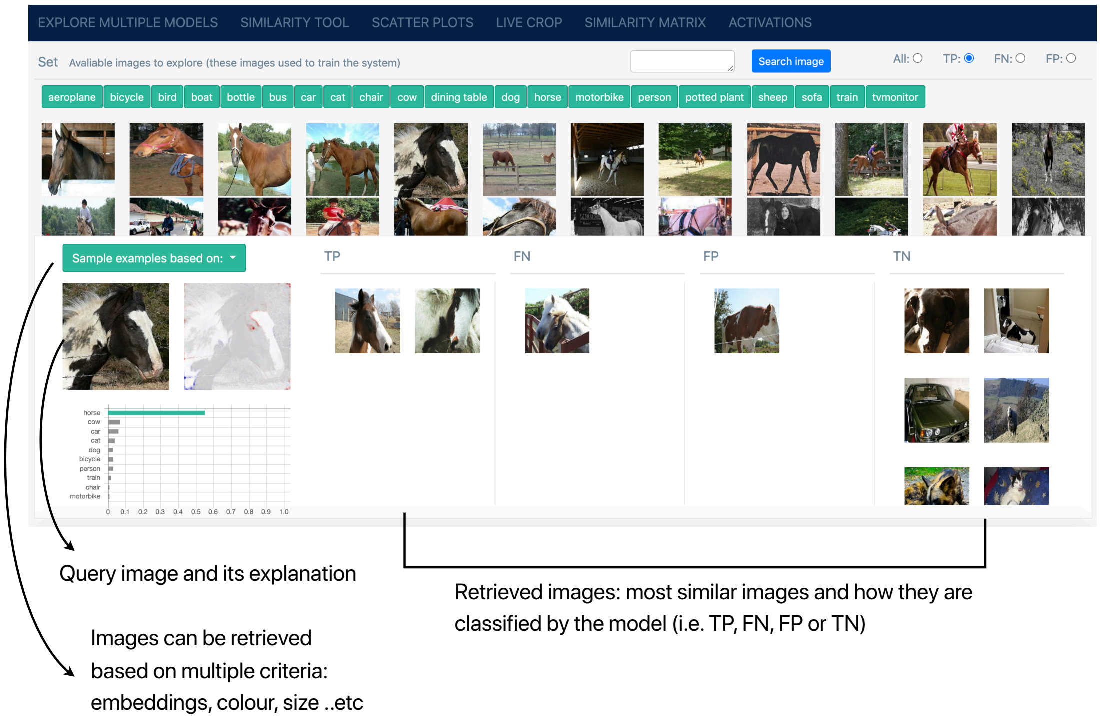

1
2
3
4
5
6
Set Avaliable images to explore (these images used to train the model)
Show: All: ● TP: ○ FN: ○ FP: ○ TN: ○
Filter images by category:
aeroplane
bicycle
bird
boat
bottle
bus
cat
chair
cow
dining table
dog
horse
motorbike
person
potted plant
sheep
sofa
train
tvmonitor
Selected image
Heatmap Explanation
Classification The model classification on the selected image
The correct label(s) (Ground-truth): cat
cat
dog
car
chair
person
train
bus
horse
tvmonitor
potted plant
sofa
bird
cow
bicycle
bottle
boat
aeroplane
motorbike
sheep
dining table
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
Current page:
Average performance of the model ▴
Finished the exploration? proceed to questionnaire

Figure A.1: The browser: a tool to navigate the dataset



Query image and its explanation

Images can be retrieved based on multiple criteria: embeddings, colour, size ..etc

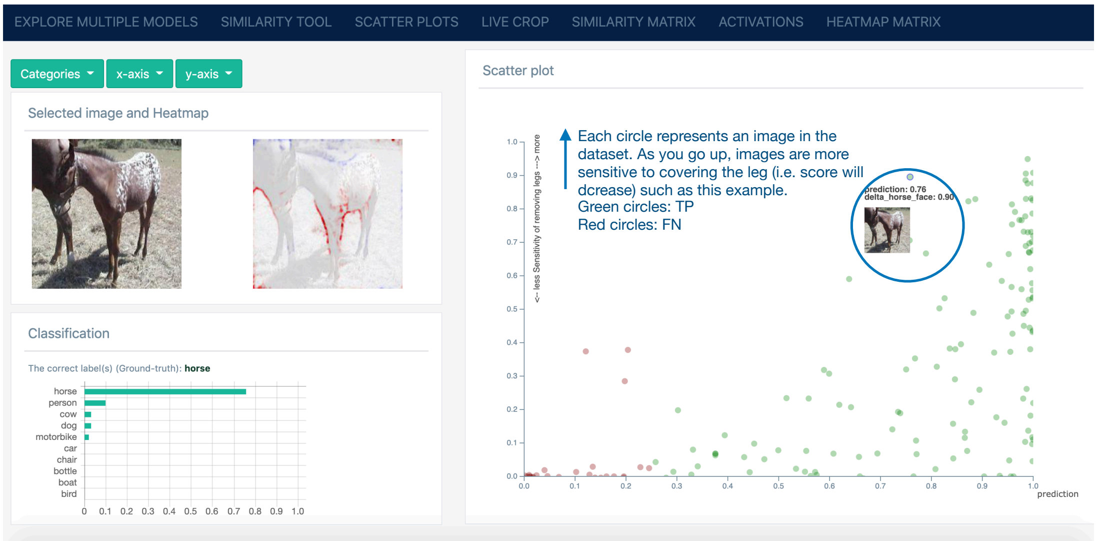
Retrieved images: most similar images and how they are classified by the model (i.e. TP, FN, FP or TN)

Figure A.2: Similarity tool: to retrieve similar images based on multiple criteria

scroll up and down to select an image (3). Once an image is selected, it will be displayed in (4). At the same time, a heatmap explanation will be shown in (5) and a bar chart representing the probability scores for each category will be presented in (6) along with the ground truth of this particular image. The bar chart is interactive; by clicking on any of the bars in the chart, users can display the heatmap for the corresponding class. In addition to these functions, the tool allow the user to explore how different models classify certain image, along with the corresponding heatmap explanation.

A.2 Similarity tool

In Chapter 4 and Chapter 5, we displayed few examples that are related to the task image based on some criterion. To help us inspect and explore the different selection options, we developed this tool which allow a user to retrieve instances based on multiple options, which includes different embeddings extracted from different layers of the network. We also include the option to retrieve examples based on other aspects such as the dominant colour and size of the main object. Figure A.2 shows a screenshot of the tool.



The correct label(s) (Ground-truth): **horse**

Each circle represents an image in the dataset. As you go up, images are more sensitive to covering the leg (i.e. score will dcrease) such as this example.
Green circles: TP
Red circles: FN

Figure A.3: The interactive scatter plots tool

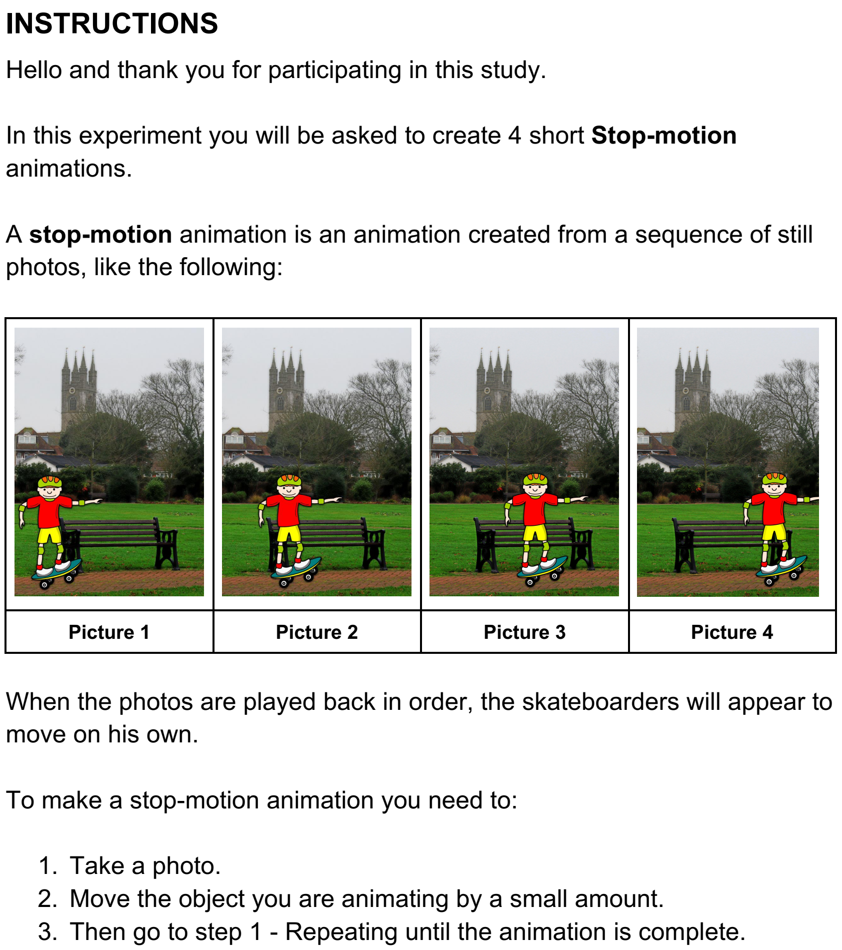
A.3 Scatter plots tool

In Chapter 6, we had to find a convenient way to explore the effect of covering different part of the object on the classification score. For that reason, we created an interactive scatter plot (Figure A.3 in which a user can explore the relationships between different attributes. For example, what is the relationship between the classification score and the size of the object. The tool offer multiple options to explore including the size, colour, image brightness and the sensitivity index to covering a part of the object (e.g. the eyes of a cat).

Appendix B

Study Tutorials

B.1 Chapter 3 tutorial



INSTRUCTIONS

Hello and thank you for participating in this study.

In this experiment you will be asked to create 4 short **Stop-motion** animations.

A **stop-motion** animation is an animation created from a sequence of still photos, like the following:

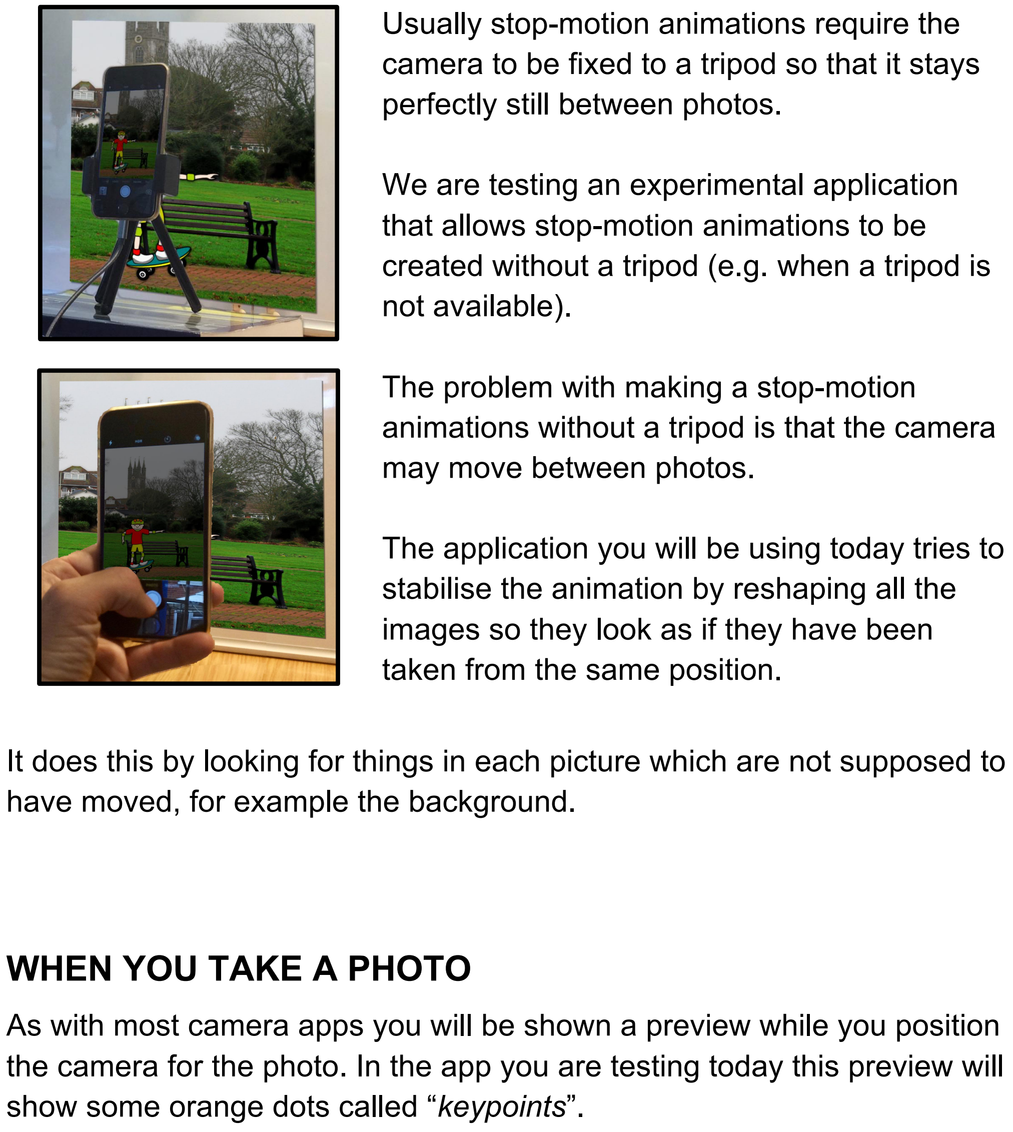
Picture 1	**Picture 2**	**Picture 3**	**Picture 4**

When the photos are played back in order, the skateboarders will appear to move on his own.

To make a stop-motion animation you need to:

1. Take a photo.
2. Move the object you are animating by a small amount.
3. Then go to step 1 - Repeating until the animation is complete.

Figure B.1: Chapter 3 instructions - page 1



Usually stop-motion animations require the camera to be fixed to a tripod so that it stays perfectly still between photos.

We are testing an experimental application that allows stop-motion animations to be created without a tripod (e.g. when a tripod is not available).

The problem with making a stop-motion animations without a tripod is that the camera may move between photos.

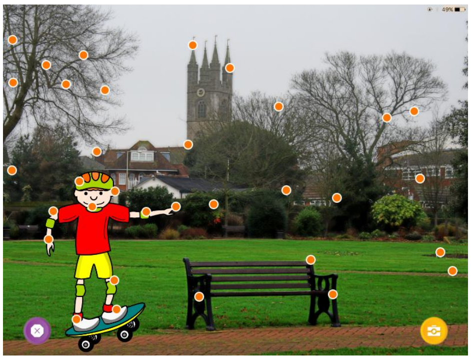
The application you will be using today tries to stabilise the animation by reshaping all the images so they look as if they have been taken from the same position.

It does this by looking for things in each picture which are not supposed to have moved, for example the background.

WHEN YOU TAKE A PHOTO

As with most camera apps you will be shown a preview while you position the camera for the photo. In the app you are testing today this preview will show some orange dots called “*keypoints*”.

Figure B.2: Chapter 3 instructions - page 2



These *keypoints* represent pixels which the app finds to be distinctive - please note that what is distinctive to the app may be different from what is distinctive to the human eye.

For the app to work well:

- keypoints should appear on the background, and
- they should be spread evenly, rather than concentrated on a small area

Note: Keypoints on a moving object (e.g. the character) don't help.

Note

The application we are testing is not yet perfect, so you may need to "work around it" to get good results. If a photo looks wrong after the app has processed it, then please delete it and try again. If this happens repeatedly or there is a technical issue, then please let the investigator know. We are testing the app and not you.

Any Questions?

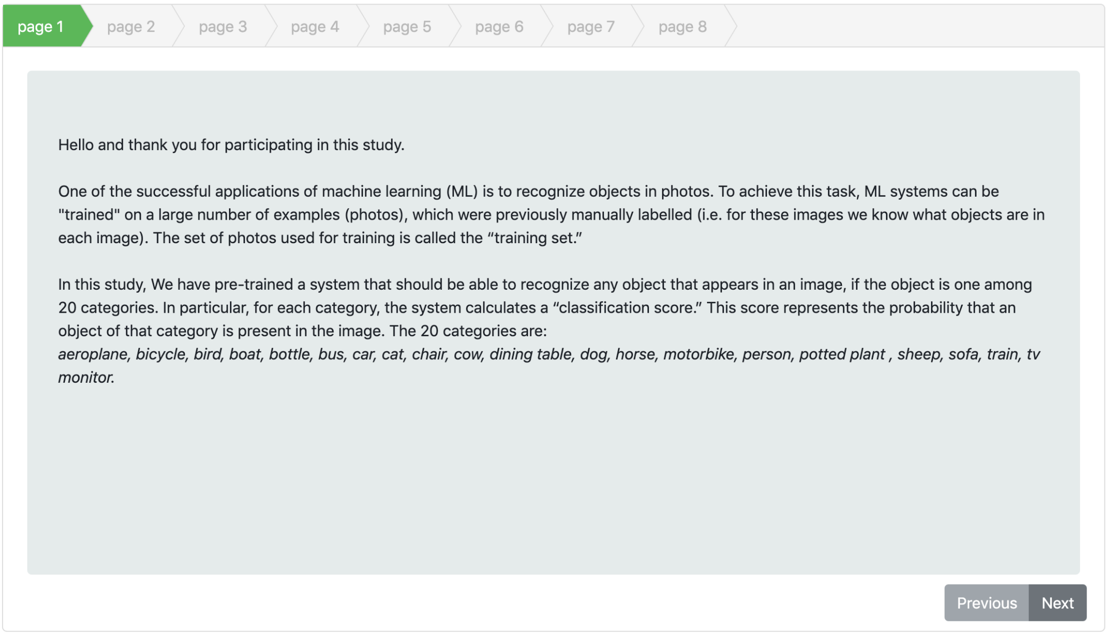
Please ask the investigator now.

Ready?

Please tell the investigator.

Figure B.3: Chapter 3 instructions - page 3 for the keypoints condition. Similar instructions are provided for other conditions

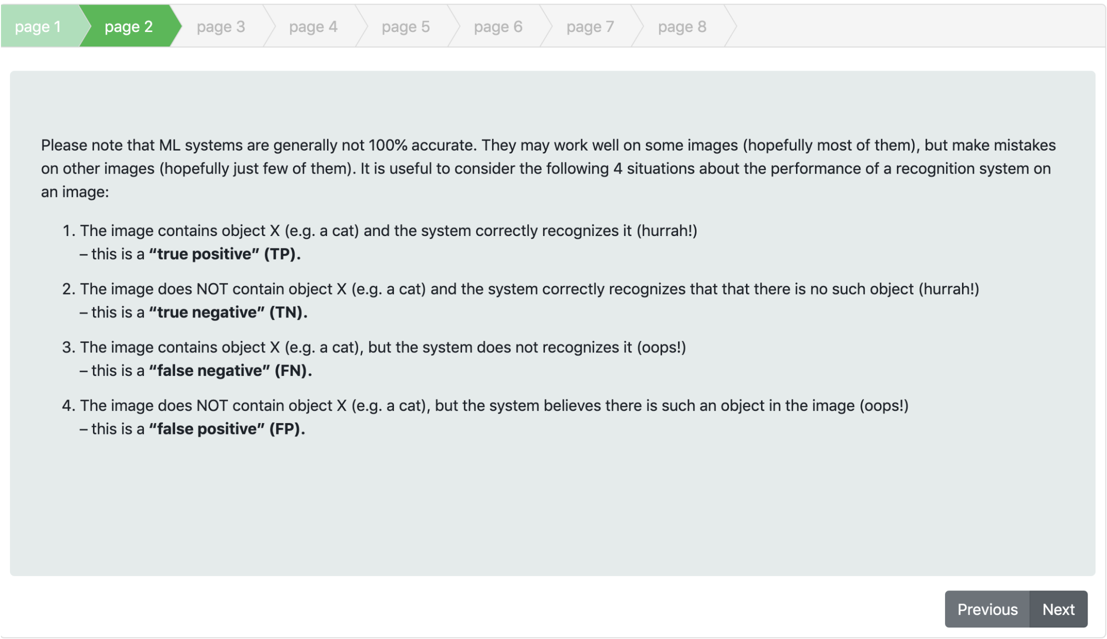
B.2 Chapter 4 tutorial



Hello and thank you for participating in this study.

One of the successful applications of machine learning (ML) is to recognize objects in photos. To achieve this task, ML systems can be "trained" on a large number of examples (photos), which were previously manually labelled (i.e. for these images we know what objects are in each image). The set of photos used for training is called the "training set."

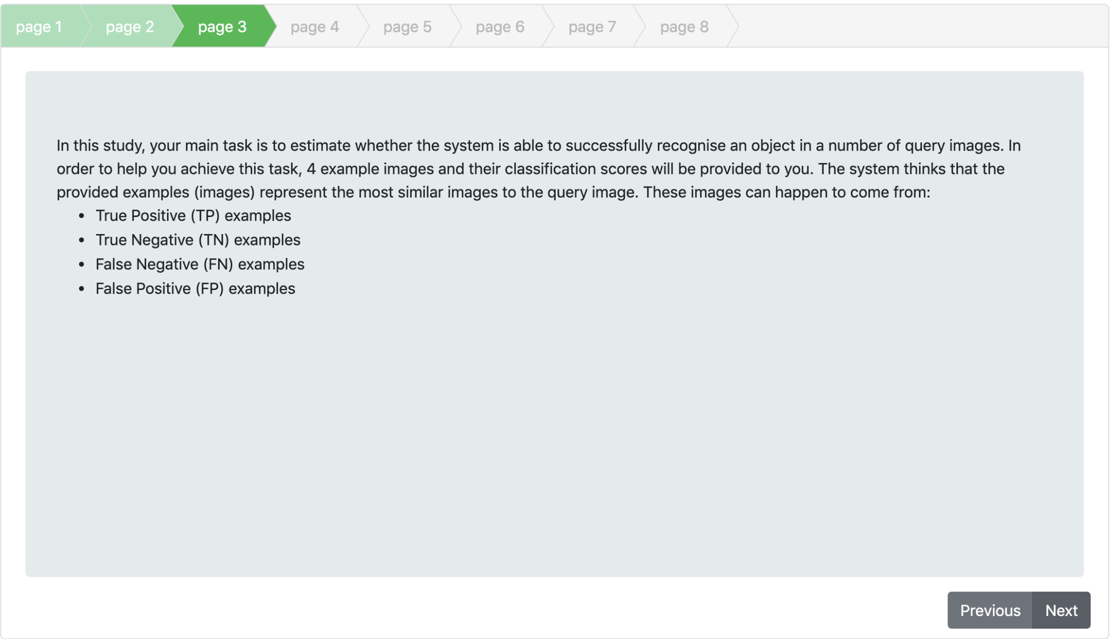
In this study, We have pre-trained a system that should be able to recognize any object that appears in an image, if the object is one among 20 categories. In particular, for each category, the system calculates a "classification score." This score represents the probability that an object of that category is present in the image. The 20 categories are:
aeroplane, bicycle, bird, boat, bottle, bus, car, cat, chair, cow, dining table, dog, horse, motorbike, person, potted plant , sheep, sofa, train, tv monitor.



Please note that ML systems are generally not 100% accurate. They may work well on some images (hopefully most of them), but make mistakes on other images (hopefully just few of them). It is useful to consider the following 4 situations about the performance of a recognition system on an image:

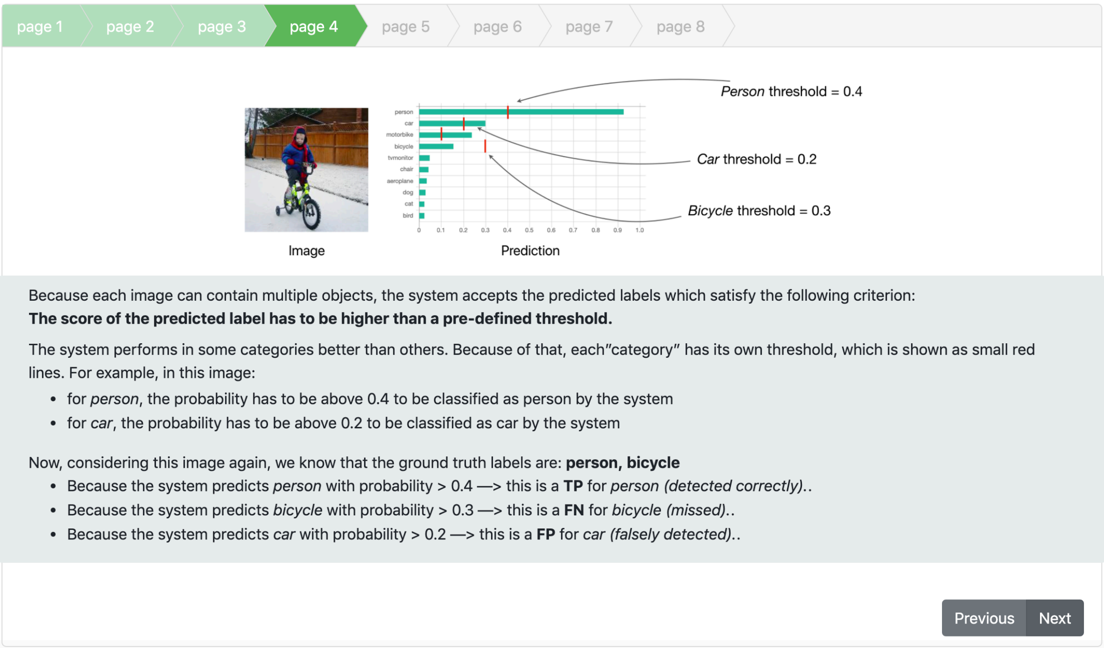
1. The image contains object X (e.g. a cat) and the system correctly recognizes it (hurrah!)
   – this is a **"true positive" (TP).**
2. The image does NOT contain object X (e.g. a cat) and the system correctly recognizes that that there is no such object (hurrah!)
   – this is a **"true negative" (TN).**
3. The image contains object X (e.g. a cat), but the system does not recognizes it (oops!)
   – this is a **"false negative" (FN).**
4. The image does NOT contain object X (e.g. a cat), but the system believes there is such an object in the image (oops!)
   – this is a **"false positive" (FP).**

Figure B.4: Chapter 4 tutorial - page 1,2



In this study, your main task is to estimate whether the system is able to successfully recognise an object in a number of query images. In order to help you achieve this task, 4 example images and their classification scores will be provided to you. The system thinks that the provided examples (images) represent the most similar images to the query image. These images can happen to come from:

- True Positive (TP) examples
- True Negative (TN) examples
- False Negative (FN) examples
- False Positive (FP) examples



Because each image can contain multiple objects, the system accepts the predicted labels which satisfy the following criterion:
The score of the predicted label has to be higher than a pre-defined threshold.

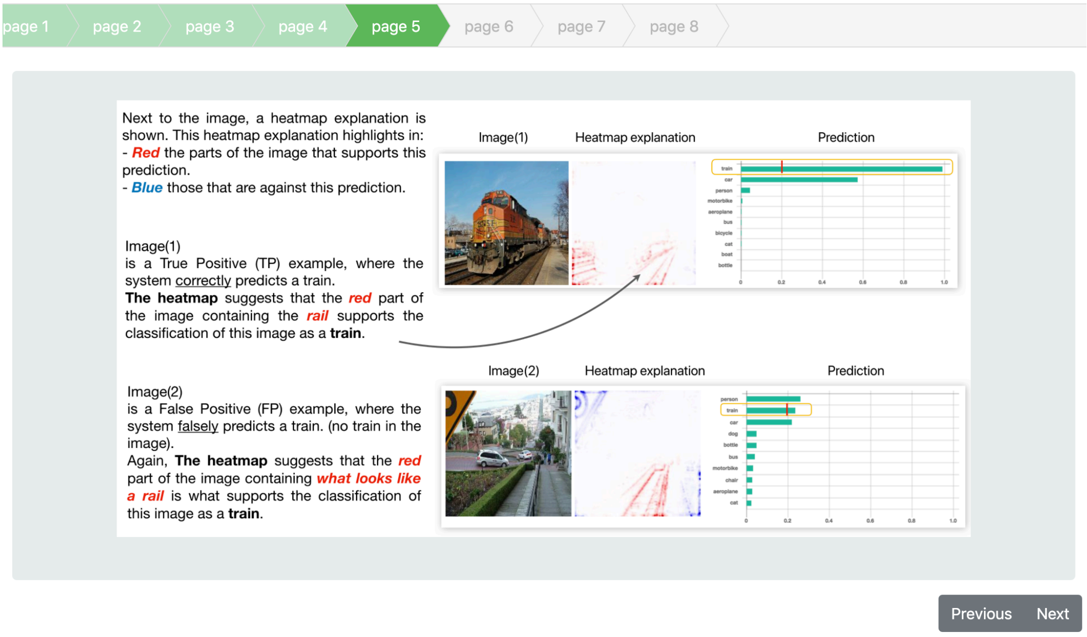
The system performs in some categories better than others. Because of that, each"category" has its own threshold, which is shown as small red lines. For example, in this image:

- for *person*, the probability has to be above 0.4 to be classified as person by the system
- for *car*, the probability has to be above 0.2 to be classified as car by the system

Now, considering this image again, we know that the ground truth labels are: **person, bicycle**

- Because the system predicts *person* with probability > 0.4 —> this is a **TP** for *person (detected correctly)*..
- Because the system predicts *bicycle* with probability > 0.3 —> this is a **FN** for *bicycle (missed)*..
- Because the system predicts *car* with probability > 0.2 —> this is a **FP** for *car (falsely detected)*..

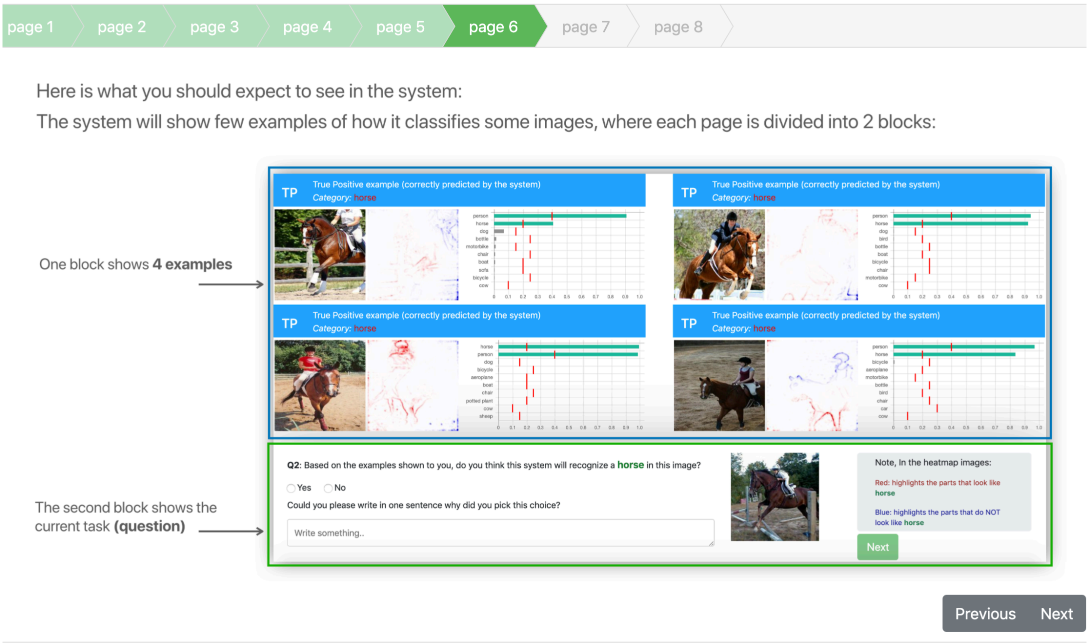
Figure B.5: Chapter 4 tutorial - page 3,4



Next to the image, a heatmap explanation is shown. This heatmap explanation highlights in:
- *Red* the parts of the image that supports this prediction.
- *Blue* those that are against this prediction.

Image(1)
is a True Positive (TP) example, where the system correctly predicts a train.
The heatmap suggests that the *red* part of the image containing the *rail* supports the classification of this image as a **train**.

Image(2)
is a False Positive (FP) example, where the system falsely predicts a train. (no train in the image).
Again, **The heatmap** suggests that the *red* part of the image containing *what looks like a rail* is what supports the classification of this image as a **train**.



Here is what you should expect to see in the system:
The system will show few examples of how it classifies some images, where each page is divided into 2 blocks:

One block shows **4 examples**

TP True Positive example (correctly predicted by the system)
Category: horse

TP True Positive example (correctly predicted by the system)
Category: horse

TP True Positive example (correctly predicted by the system)
Category: horse

TP True Positive example (correctly predicted by the system)
Category: horse

The second block shows the current task **(question)**

Q2: Based on the examples shown to you, do you think this system will recognize a **horse** in this image?

○ Yes ○ No

Could you please write in one sentence why did you pick this choice?

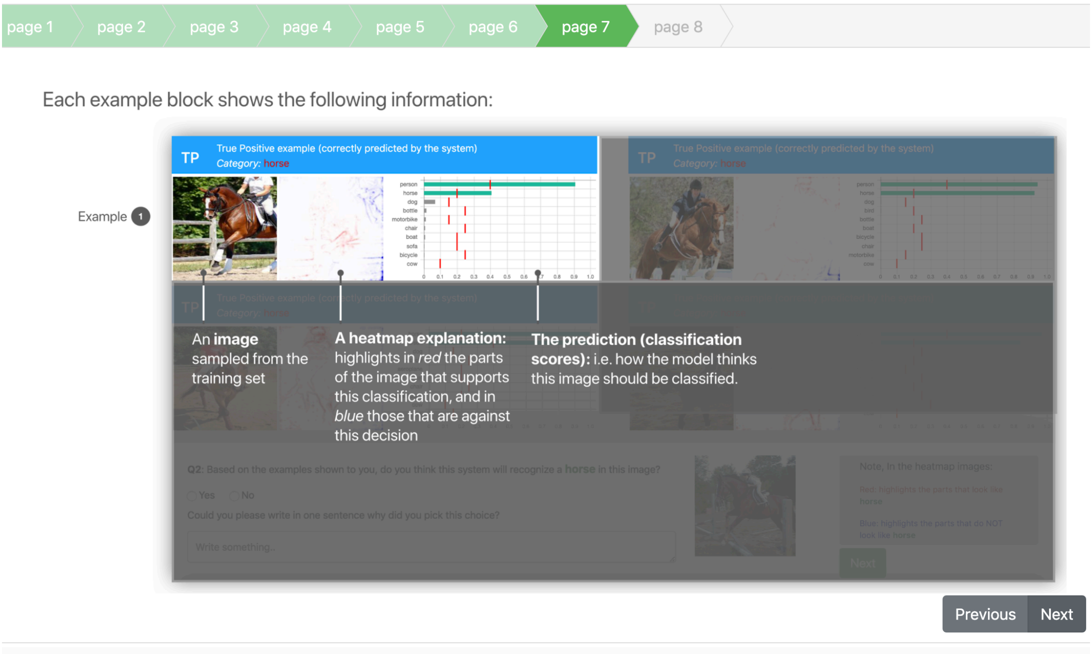
Write something..

Note, In the heatmap images:

Red: highlights the parts that look like **horse**

Blue: highlights the parts that do NOT look like **horse**

Figure B.6: Chapter 4 tutorial - page 5,6

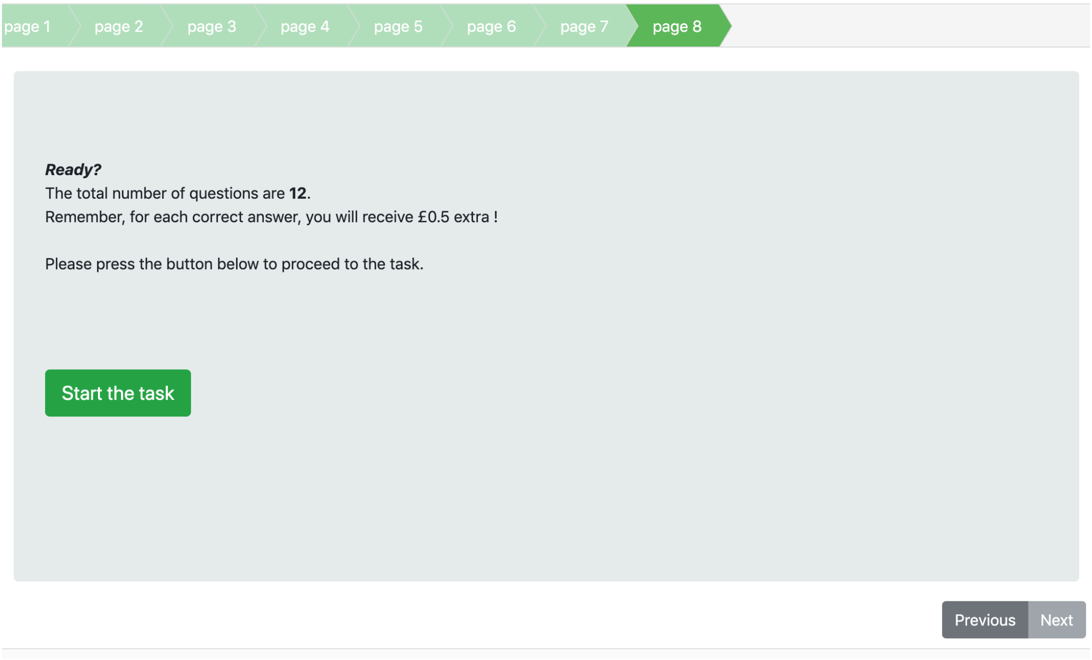

page 1
page 2
page 3
page 4
page 5
page 6
page 7
page 8
Example 1
TP
True Positive example (correctly predicted by the system)
Category: horse
person
horse
dog
bottle
motorbike
chair
boat
sofa
bicycle
cow
Q2: Based on the examples shown to you, do you think this system will recognize a horse in this image?
○ Yes ○ No
Could you please write in one sentence why did you pick this choice?
Write something...
Note, in the heatmap images:
Red: highlights the parts that look like horse
Blue: highlights the parts that do NOT look like horse
Next
Previous
Next

Each example block shows the following information:

An **image** sampled from the training set

A heatmap explanation: highlights in *red* the parts of the image that supports this classification, and in *blue* those that are against this decision

The prediction (classification scores): i.e. how the model thinks this image should be classified.



Ready?

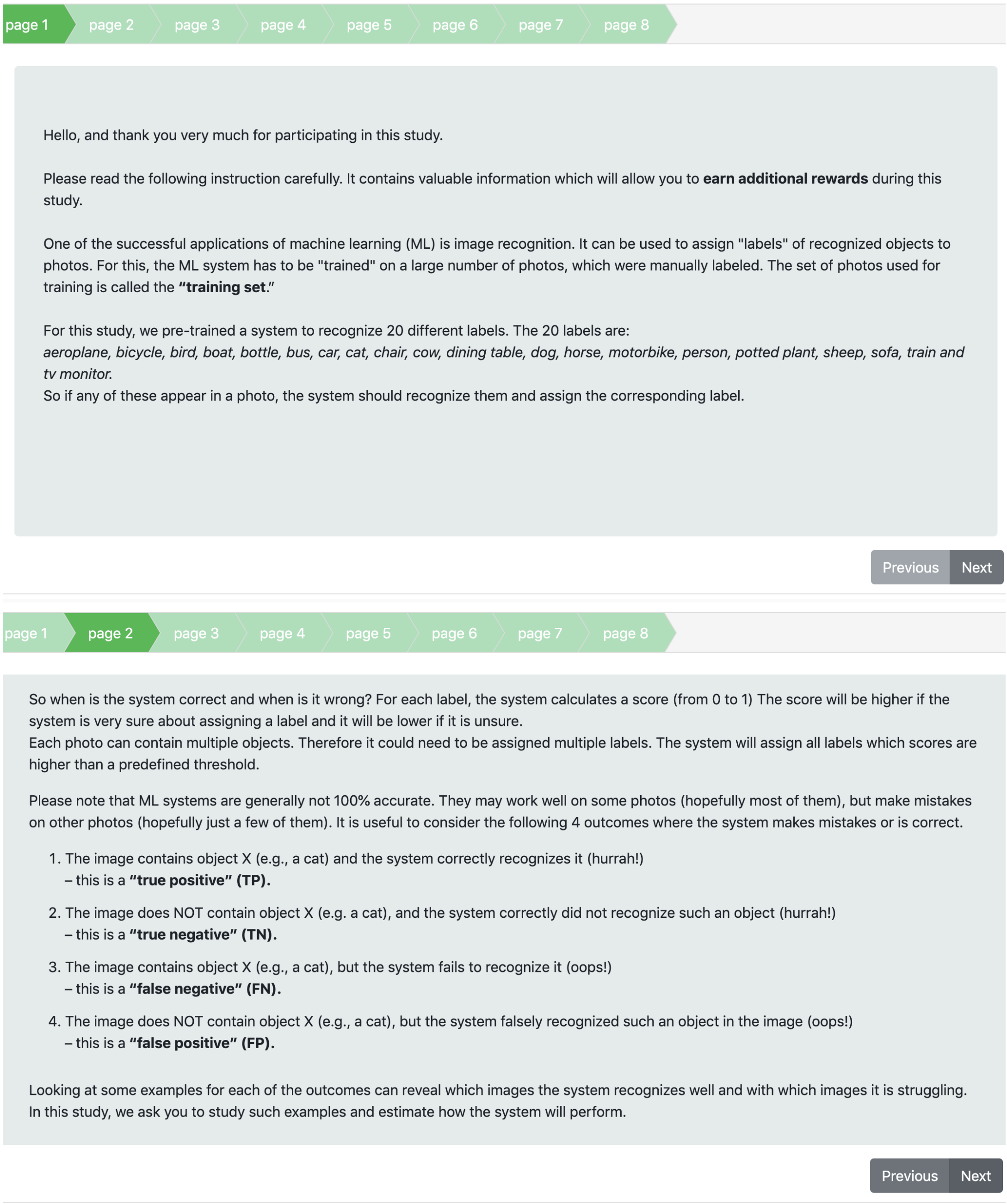
The total number of questions are **12**.

Remember, for each correct answer, you will receive £0.5 extra !

Please press the button below to proceed to the task.

Figure B.7: Chapter 4 tutorial - page 7,8

B.3 Chapter 5 tutorial



Hello, and thank you very much for participating in this study.

Please read the following instruction carefully. It contains valuable information which will allow you to **earn additional rewards** during this study.

One of the successful applications of machine learning (ML) is image recognition. It can be used to assign "labels" of recognized objects to photos. For this, the ML system has to be "trained" on a large number of photos, which were manually labeled. The set of photos used for training is called the **"training set**."

For this study, we pre-trained a system to recognize 20 different labels. The 20 labels are:
aeroplane, bicycle, bird, boat, bottle, bus, car, cat, chair, cow, dining table, dog, horse, motorbike, person, potted plant, sheep, sofa, train and tv monitor.
So if any of these appear in a photo, the system should recognize them and assign the corresponding label.

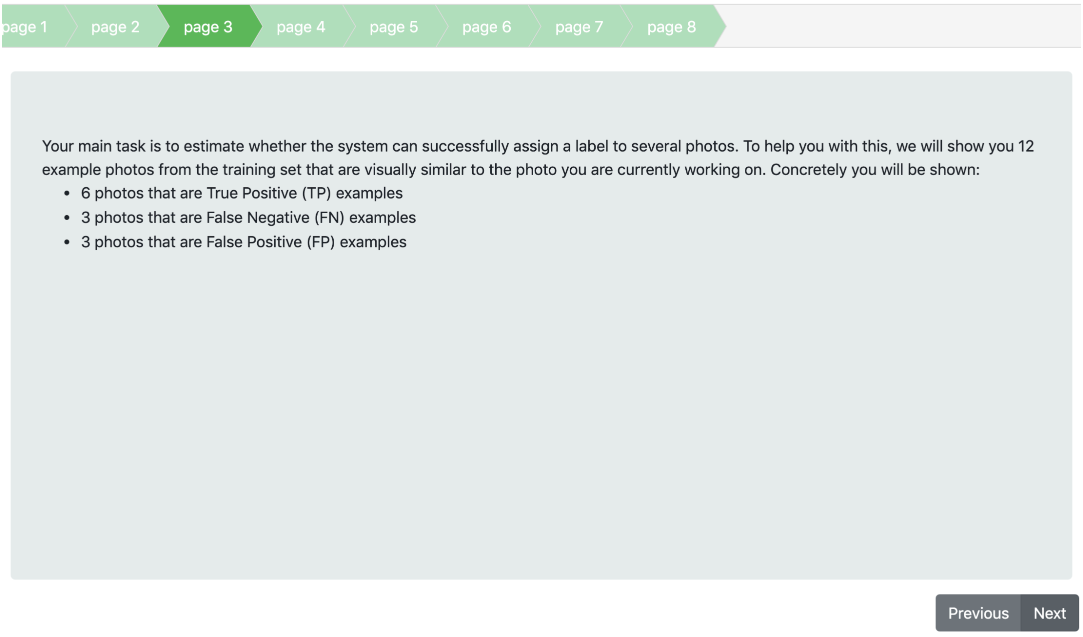
So when is the system correct and when is it wrong? For each label, the system calculates a score (from 0 to 1) The score will be higher if the system is very sure about assigning a label and it will be lower if it is unsure.
Each photo can contain multiple objects. Therefore it could need to be assigned multiple labels. The system will assign all labels which scores are higher than a predefined threshold.

Please note that ML systems are generally not 100% accurate. They may work well on some photos (hopefully most of them), but make mistakes on other photos (hopefully just a few of them). It is useful to consider the following 4 outcomes where the system makes mistakes or is correct.

1. The image contains object X (e.g., a cat) and the system correctly recognizes it (hurrah!)
   – this is a **"true positive" (TP).**
2. The image does NOT contain object X (e.g. a cat), and the system correctly did not recognize such an object (hurrah!)
   – this is a **"true negative" (TN).**
3. The image contains object X (e.g., a cat), but the system fails to recognize it (oops!)
   – this is a **"false negative" (FN).**
4. The image does NOT contain object X (e.g., a cat), but the system falsely recognized such an object in the image (oops!)
   – this is a **"false positive" (FP).**

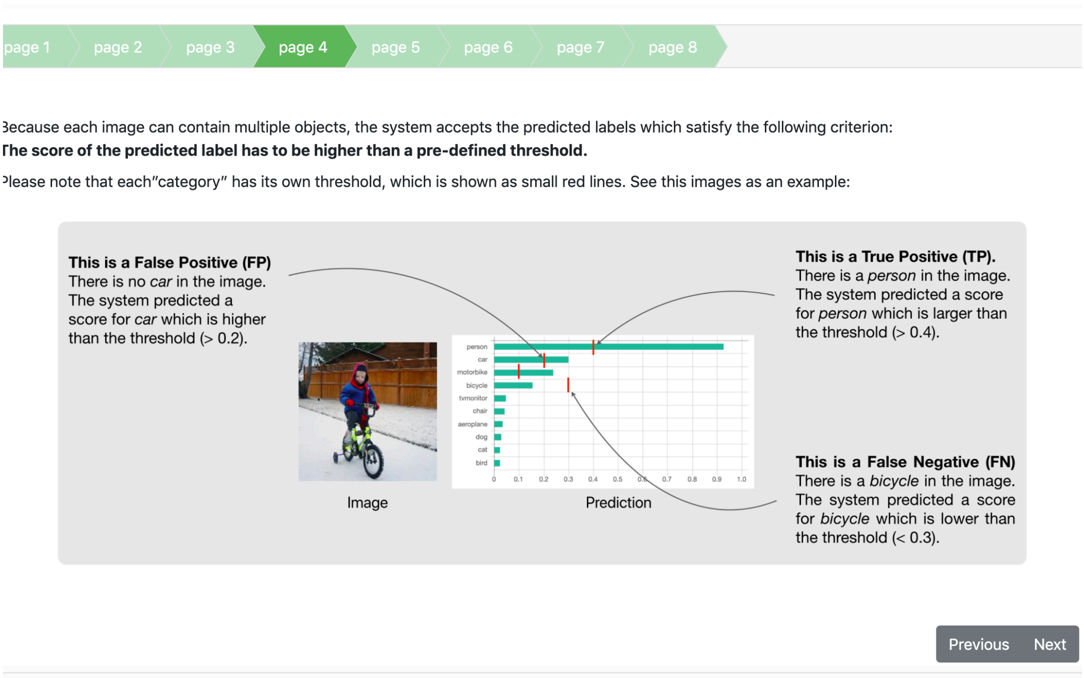
Looking at some examples for each of the outcomes can reveal which images the system recognizes well and with which images it is struggling. In this study, we ask you to study such examples and estimate how the system will perform.

Figure B.8: Chapter 5 tutorial - page 1,2



Your main task is to estimate whether the system can successfully assign a label to several photos. To help you with this, we will show you 12 example photos from the training set that are visually similar to the photo you are currently working on. Concretely you will be shown:

- 6 photos that are True Positive (TP) examples
- 3 photos that are False Negative (FN) examples
- 3 photos that are False Positive (FP) examples



Because each image can contain multiple objects, the system accepts the predicted labels which satisfy the following criterion:
The score of the predicted label has to be higher than a pre-defined threshold.

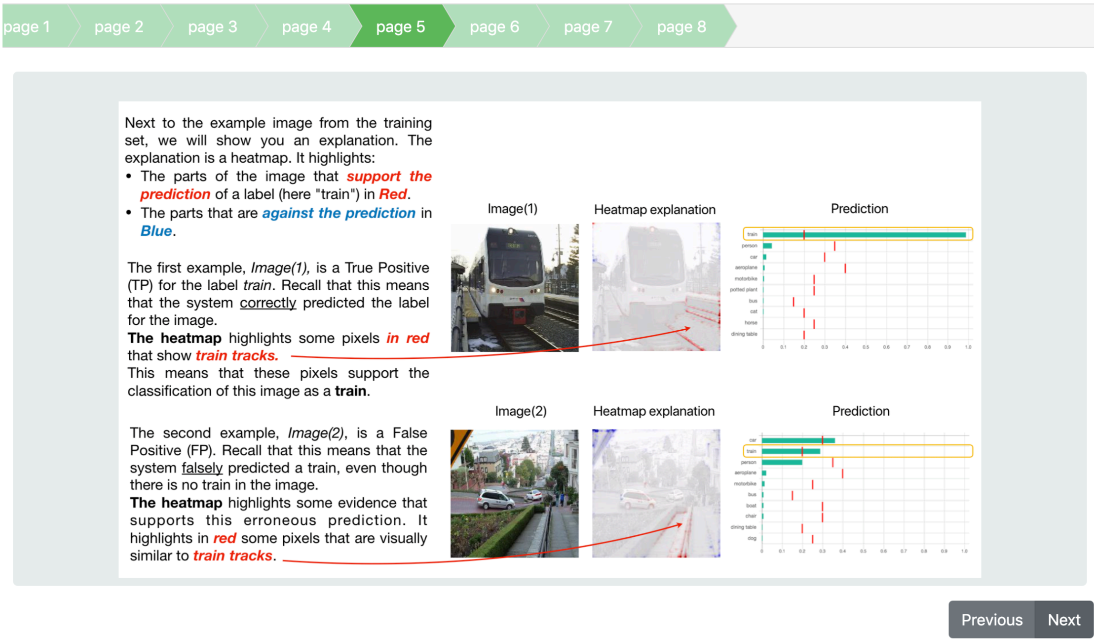
Please note that each"category" has its own threshold, which is shown as small red lines. See this images as an example:

This is a False Positive (FP)
There is no *car* in the image. The system predicted a score for *car* which is higher than the threshold (> 0.2).

This is a True Positive (TP).
There is a *person* in the image. The system predicted a score for *person* which is larger than the threshold (> 0.4).

This is a False Negative (FN)
There is a *bicycle* in the image. The system predicted a score for *bicycle* which is lower than the threshold (< 0.3).

Figure B.9: Chapter 5 tutorial - page 3,4

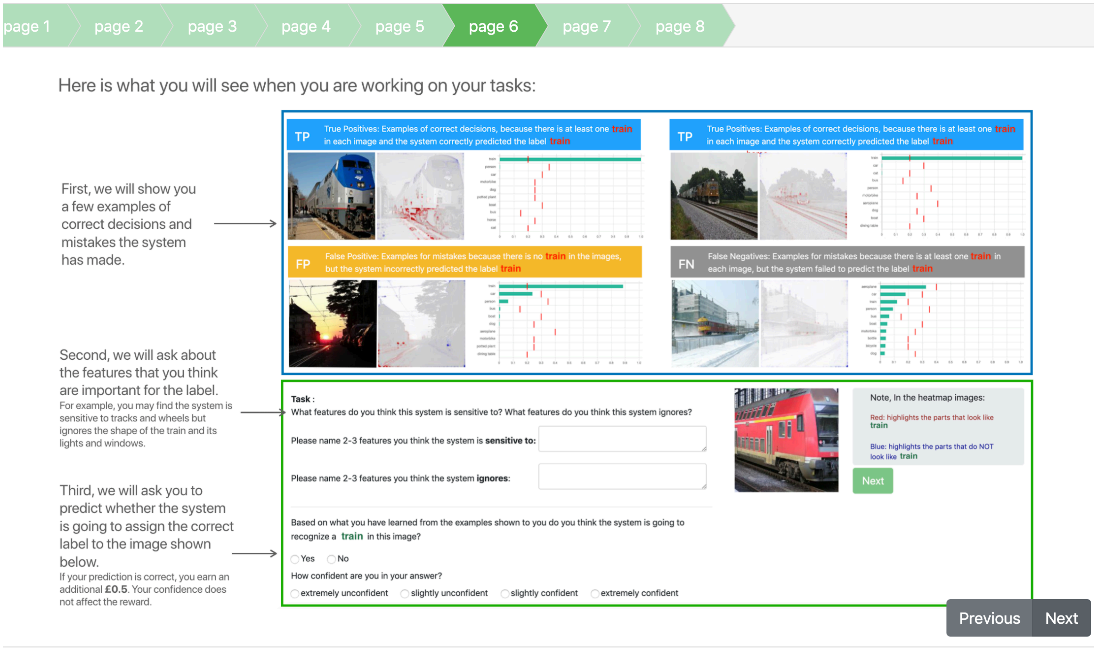


Next to the example image from the training set, we will show you an explanation. The explanation is a heatmap. It highlights:

- The parts of the image that ***support the prediction*** of a label (here "train") in ***Red***.
- The parts that are ***against the prediction*** in ***Blue***.

The first example, *Image(1),* is a True Positive (TP) for the label *train*. Recall that this means that the system correctly predicted the label for the image. **The heatmap** highlights some pixels ***in red*** that show ***train tracks.*** This means that these pixels support the classification of this image as a **train**.

The second example, *Image(2)*, is a False Positive (FP). Recall that this means that the system falsely predicted a train, even though there is no train in the image. **The heatmap** highlights some evidence that supports this erroneous prediction. It highlights in ***red*** some pixels that are visually similar to ***train tracks***.



Here is what you will see when you are working on your tasks:

First, we will show you a few examples of correct decisions and mistakes the system has made.

TP True Positives: Examples of correct decisions, because there is at least one train in each image and the system correctly predicted the label train

TP True Positives: Examples of correct decisions, because there is at least one train in each image and the system correctly predicted the label train

FP False Positive: Examples for mistakes because there is no train in the images, but the system incorrectly predicted the label train

FN False Negatives: Examples for mistakes because there is at least one train in each image, but the system failed to predict the label train

Second, we will ask about the features that you think are important for the label. For example, you may find the system is sensitive to tracks and wheels but ignores the shape of the train and its lights and windows.

Task :
What features do you think this system is sensitive to? What features do you think this system ignores?

Please name 2-3 features you think the system is **sensitive to:**

Please name 2-3 features you think the system **ignores**:

Note, In the heatmap images:

Red: highlights the parts that look like train

Blue: highlights the parts that do NOT look like train

Third, we will ask you to predict whether the system is going to assign the correct label to the image shown below. If your prediction is correct, you earn an additional **£0.5**. Your confidence does not affect the reward.

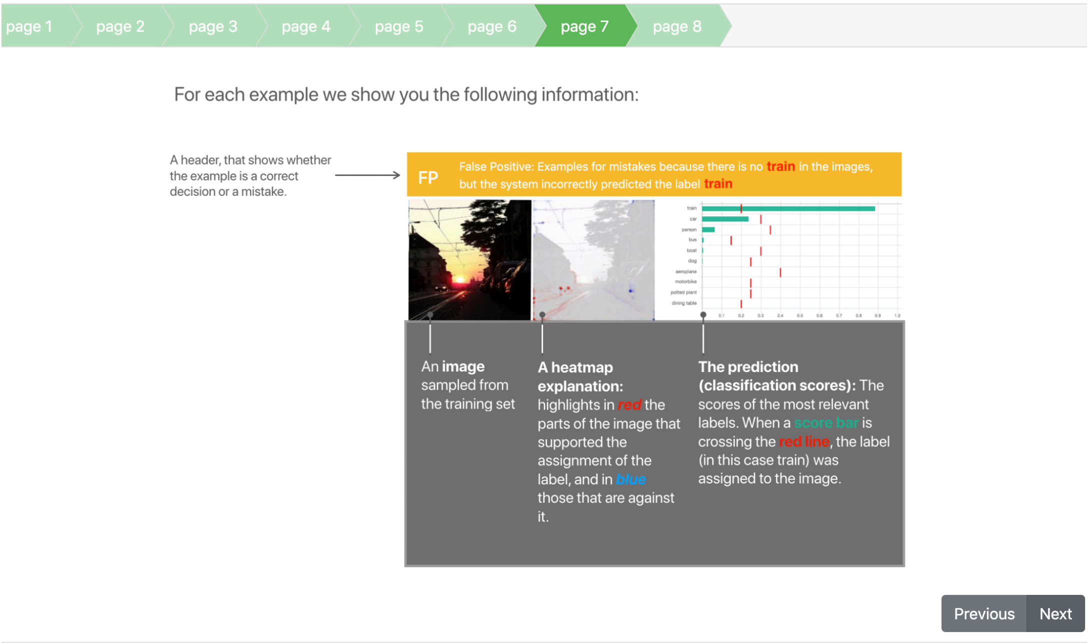
Based on what you have learned from the examples shown to you do you think the system is going to recognize a train in this image?

○ Yes ○ No

How confident are you in your answer?

○ extremely unconfident ○ slightly unconfident ○ slightly confident ○ extremely confident

Figure B.10: Chapter 5 tutorial - page 5,6



For each example we show you the following information:

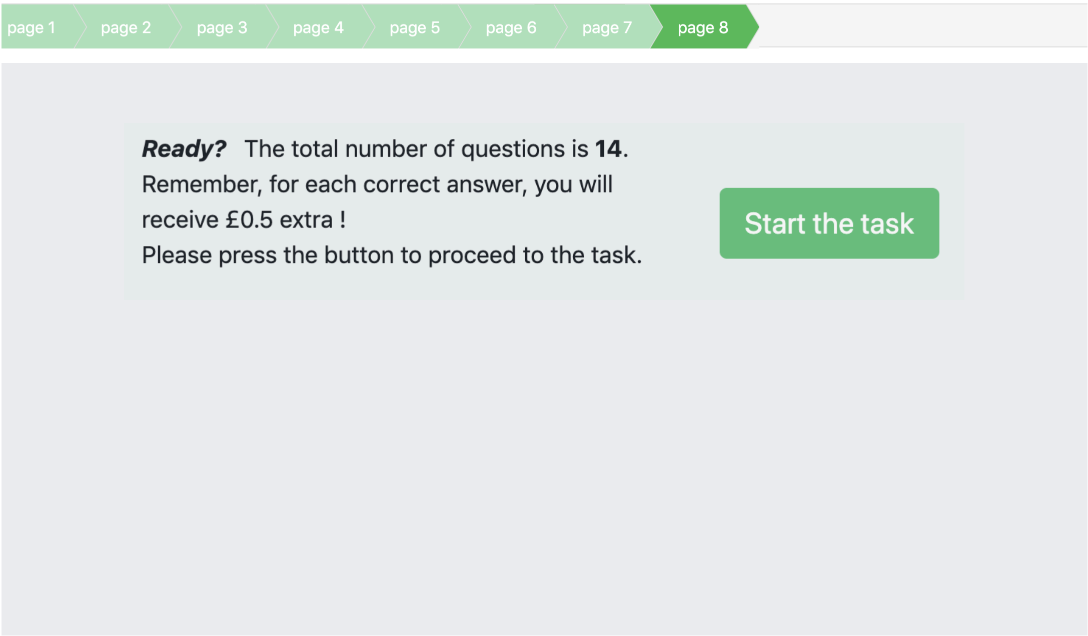
A header, that shows whether the example is a correct decision or a mistake.

FP False Positive: Examples for mistakes because there is no **train** in the images, but the system incorrectly predicted the label **train**

An **image** sampled from the training set

A heatmap explanation: highlights in ***red*** the parts of the image that supported the assignment of the label, and in *blue* those that are against it.

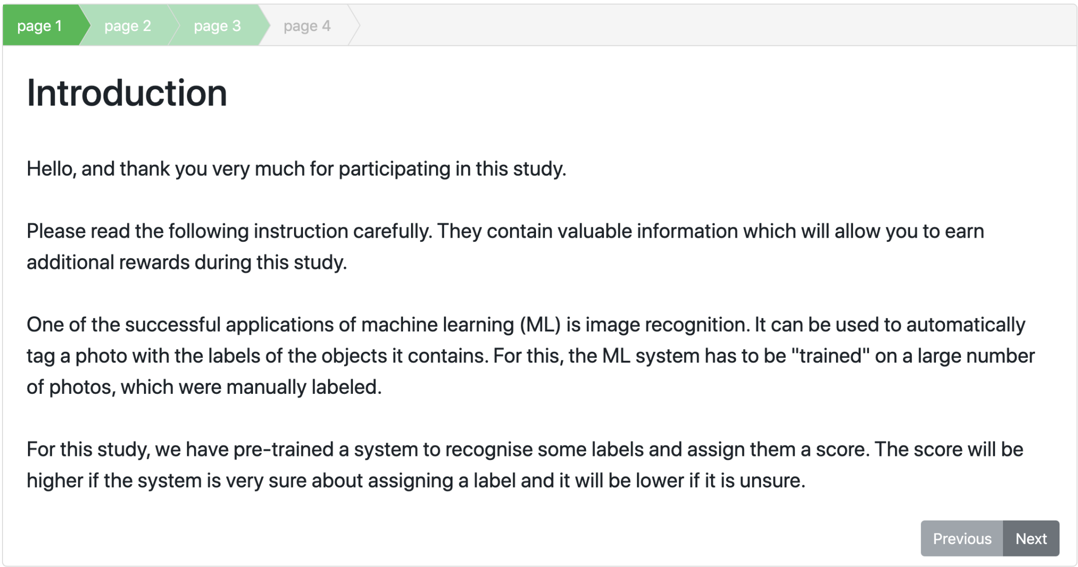
The prediction (classification scores): The scores of the most relevant labels. When a **score bar** is crossing the **red line**, the label (in this case train) was assigned to the image.



Ready? The total number of questions is **14**.
Remember, for each correct answer, you will receive £0.5 extra !
Please press the button to proceed to the task.

Figure B.11: Chapter 5 tutorial - page 7,8

B.4 Chapter 6 tutorial



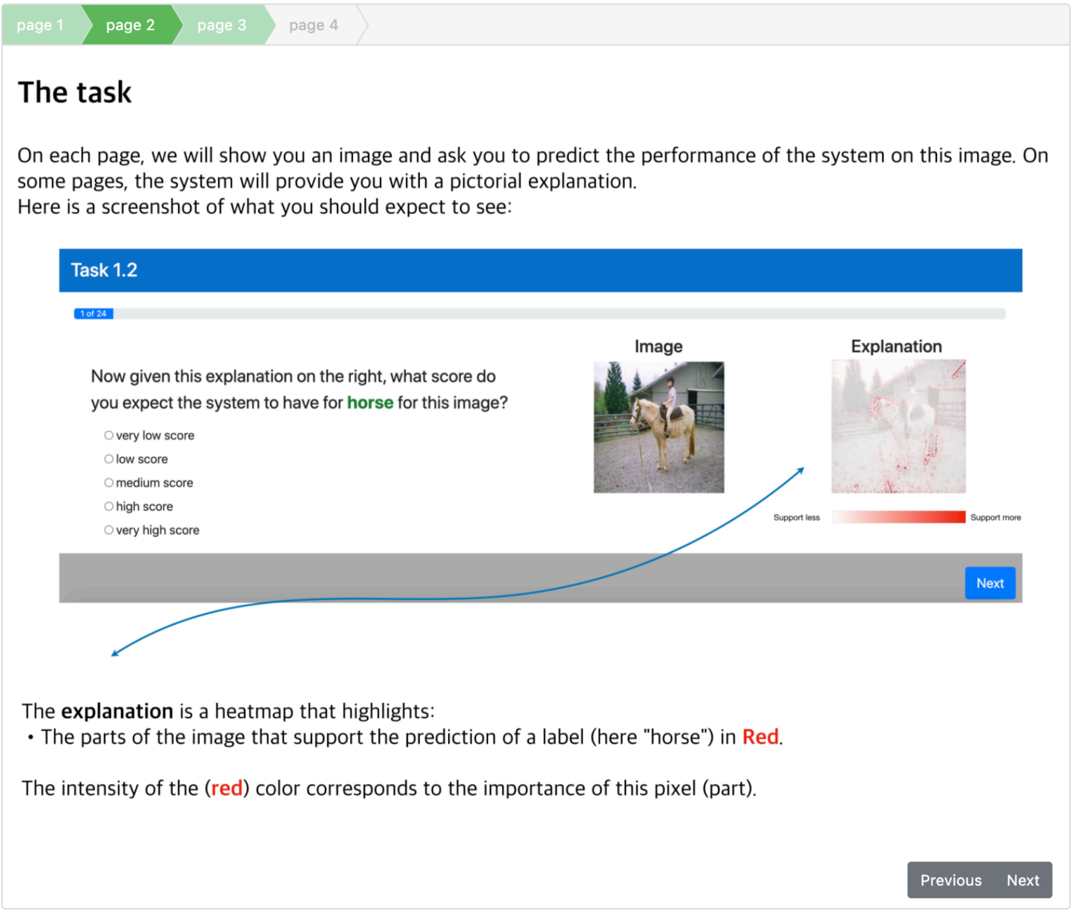
Introduction

Hello, and thank you very much for participating in this study.

Please read the following instruction carefully. They contain valuable information which will allow you to earn additional rewards during this study.

One of the successful applications of machine learning (ML) is image recognition. It can be used to automatically tag a photo with the labels of the objects it contains. For this, the ML system has to be "trained" on a large number of photos, which were manually labeled.

For this study, we have pre-trained a system to recognise some labels and assign them a score. The score will be higher if the system is very sure about assigning a label and it will be lower if it is unsure.



The task

On each page, we will show you an image and ask you to predict the performance of the system on this image. On some pages, the system will provide you with a pictorial explanation.
Here is a screenshot of what you should expect to see:

Task 1.2

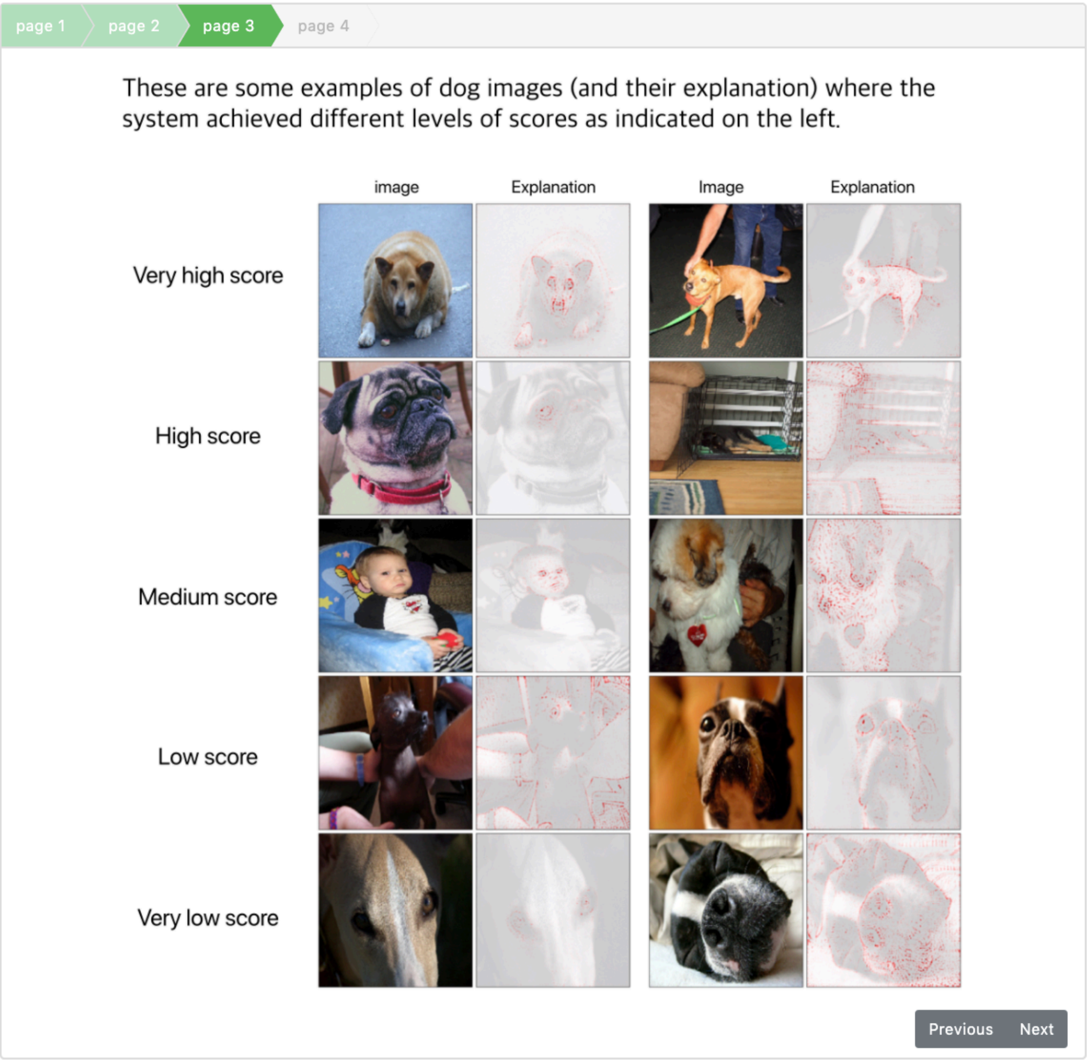
Now given this explanation on the right, what score do you expect the system to have for **horse** for this image?

○ very low score
○ low score
○ medium score
○ high score
○ very high score

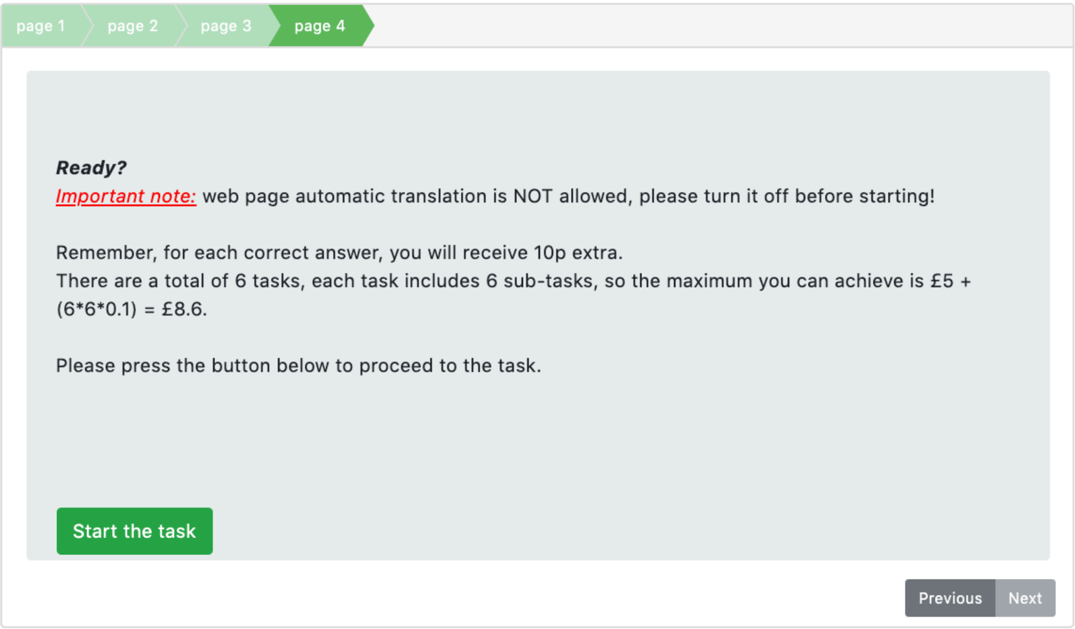
The **explanation** is a heatmap that highlights:
• The parts of the image that support the prediction of a label (here "horse") in Red.

The intensity of the (red) color corresponds to the importance of this pixel (part).

Figure B.12: Chapter 6 tutorial - page 1,2



These are some examples of dog images (and their explanation) where the system achieved different levels of scores as indicated on the left.



Ready?
Important note: web page automatic translation is NOT allowed, please turn it off before starting!

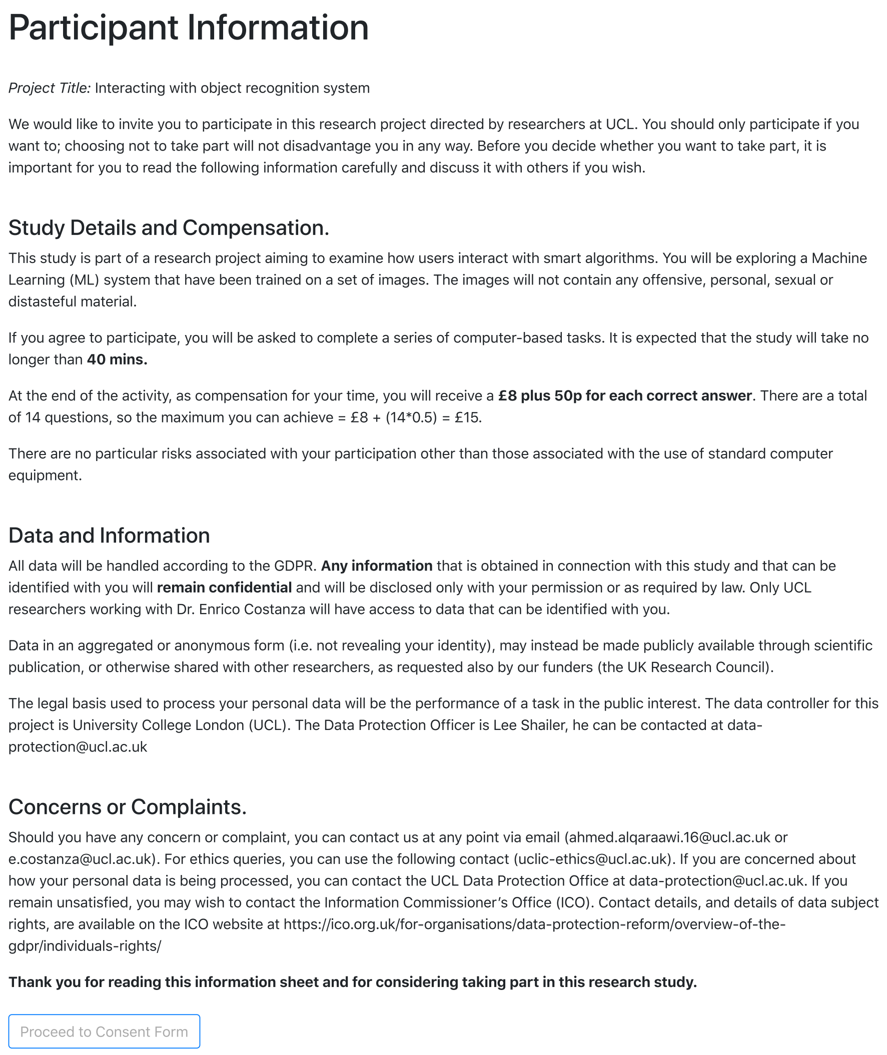
Remember, for each correct answer, you will receive 10p extra.
There are a total of 6 tasks, each task includes 6 sub-tasks, so the maximum you can achieve is £5 + (6*6*0.1) = £8.6.

Please press the button below to proceed to the task.

Figure B.13: Chapter 6 tutorial - page 3,4

Appendix C

Forms



Participant Information

Project Title: Interacting with object recognition system

We would like to invite you to participate in this research project directed by researchers at UCL. You should only participate if you want to; choosing not to take part will not disadvantage you in any way. Before you decide whether you want to take part, it is important for you to read the following information carefully and discuss it with others if you wish.

Study Details and Compensation.

This study is part of a research project aiming to examine how users interact with smart algorithms. You will be exploring a Machine Learning (ML) system that have been trained on a set of images. The images will not contain any offensive, personal, sexual or distasteful material.

If you agree to participate, you will be asked to complete a series of computer-based tasks. It is expected that the study will take no longer than **40 mins.**

At the end of the activity, as compensation for your time, you will receive a **£8 plus 50p for each correct answer**. There are a total of 14 questions, so the maximum you can achieve = £8 + (14*0.5) = £15.

There are no particular risks associated with your participation other than those associated with the use of standard computer equipment.

Data and Information

All data will be handled according to the GDPR. **Any information** that is obtained in connection with this study and that can be identified with you will **remain confidential** and will be disclosed only with your permission or as required by law. Only UCL researchers working with Dr. Enrico Costanza will have access to data that can be identified with you.

Data in an aggregated or anonymous form (i.e. not revealing your identity), may instead be made publicly available through scientific publication, or otherwise shared with other researchers, as requested also by our funders (the UK Research Council).

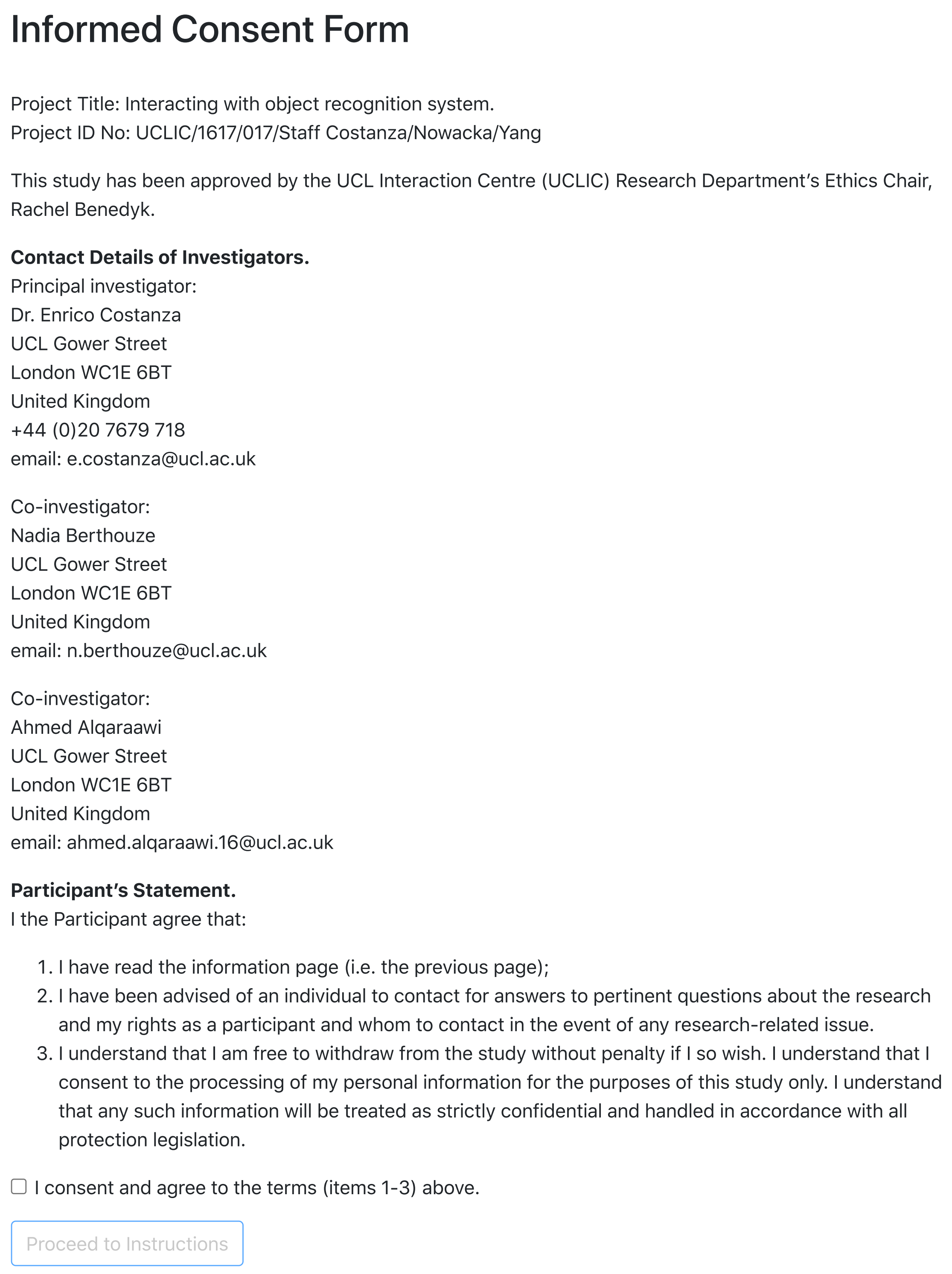
The legal basis used to process your personal data will be the performance of a task in the public interest. The data controller for this project is University College London (UCL). The Data Protection Officer is Lee Shailer, he can be contacted at data-protection@ucl.ac.uk

Concerns or Complaints.

Should you have any concern or complaint, you can contact us at any point via email (ahmed.alqaraawi.16@ucl.ac.uk or e.costanza@ucl.ac.uk). For ethics queries, you can use the following contact (uclic-ethics@ucl.ac.uk). If you are concerned about how your personal data is being processed, you can contact the UCL Data Protection Office at data-protection@ucl.ac.uk. If you remain unsatisfied, you may wish to contact the Information Commissioner's Office (ICO). Contact details, and details of data subject rights, are available on the ICO website at https://ico.org.uk/for-organisations/data-protection-reform/overview-of-the-gdpr/individuals-rights/

Thank you for reading this information sheet and for considering taking part in this research study.

Figure C.1: Participant Information sheet



Informed Consent Form

Project Title: Interacting with object recognition system.
Project ID No: UCLIC/1617/017/Staff Costanza/Nowacka/Yang

This study has been approved by the UCL Interaction Centre (UCLIC) Research Department's Ethics Chair, Rachel Benedyk.

Contact Details of Investigators.
Principal investigator:
Dr. Enrico Costanza
UCL Gower Street
London WC1E 6BT
United Kingdom
+44 (0)20 7679 718
email: e.costanza@ucl.ac.uk

Co-investigator:
Nadia Berthouze
UCL Gower Street
London WC1E 6BT
United Kingdom
email: n.berthouze@ucl.ac.uk

Co-investigator:
Ahmed Alqaraawi
UCL Gower Street
London WC1E 6BT
United Kingdom
email: ahmed.alqaraawi.16@ucl.ac.uk

Participant's Statement.
I the Participant agree that:

1. I have read the information page (i.e. the previous page);
2. I have been advised of an individual to contact for answers to pertinent questions about the research and my rights as a participant and whom to contact in the event of any research-related issue.
3. I understand that I am free to withdraw from the study without penalty if I so wish. I understand that I consent to the processing of my personal information for the purposes of this study only. I understand that any such information will be treated as strictly confidential and handled in accordance with all protection legislation.

□ I consent and agree to the terms (items 1-3) above.

Figure C.2: Informed Consent Form

References

A. Abdul, J. Vermeulen, D. Wang, B. Y. Lim, and M. Kankanhalli. Trends and Trajectories for Explainable, Accountable and Intelligible Systems: An HCI Research Agenda. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, pages 582:1–582:18. ACM, 2018a.

A. Abdul, J. Vermeulen, D. Wang, B. Y. Lim, and M. Kankanhalli. Trends and trajectories for explainable, accountable and intelligible systems: An hci research agenda. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, pages 582:1–582:18, New York, NY, USA, 2018b. ACM.

A. Adadi and M. Berrada. Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI). *IEEE Access*, 6:52138–52160, 2018.

J. Adebayo, J. Gilmer, M. Muelly, I. Goodfellow, M. Hardt, and B. Kim. Sanity checks for saliency maps. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 9525–9536, Red Hook, NY, USA, 2018. Curran Associates Inc.

J. Adebayo, M. Muelly, I. Liccardi, and B. Kim. Debugging tests for model explanations. *arXiv preprint arXiv:2011.05429*, 2020.

A. T. Alan, E. Costanza, S. D. Ramchurn, J. Fischer, T. Rodden, and N. R. Jennings. Tariff Agent: Interacting with a Future Smart Energy System at Home. *ACM Trans. Comput.-Hum. Interact.*, 23(4):25:1–25:28, August 2016.

D. Ardila, A. P. Kiraly, S. Bharadwaj, B. Choi, J. J. Reicher, L. Peng, D. Tse, M. Etemadi, W. Ye, G. Corrado, D. P. Naidich, and S. Shetty. End-to-end lung cancer screening with three-dimensional deep learning on low-dose chest computed tomography. *Nature Medicine*, 25(6):954, June 2019.

S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140, 2015a.

S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek. On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation. *PLOS ONE*, 10(7):e0130140, July 2015b.

D. Baehrens, T. Schroeter, S. Harmeling, M. Kawanabe, K. Hansen, and K.-R. Müller. How to explain individual classification decisions. *The Journal of Machine Learning Research*, 11:1803–1831, 2010.

C. M. Bishop. *Pattern recognition and machine learning.* Information science and statistics. Springer, New York, NY, 2006.

V. Braun and V. Clarke. Using thematic analysis in psychology. *Qualitative research in psychology*, 3(2):77–101, 2006.

L. Breiman et al. Statistical modeling: The two cultures (with comments and a rejoinder by the author). *Statistical science*, 16(3):199–231, 2001.

A. Bussone, S. Stumpf, and D. O'Sullivan. The role of explanations on trust and reliance in clinical decision support systems. In *2015 International Conference on Healthcare Informatics*, pages 160–169, Oct 2015.

C. J. Cai, J. Jongejan, and J. Holbrook. The Effects of Example-based Explanations in a Machine Learning Interface. In *Proceedings of the 24th International Conference on Intelligent User Interfaces*, IUI '19, pages 258–262, New York, NY, USA, 2019a. ACM.

C. J. Cai, E. Reif, N. Hegde, J. Hipp, B. Kim, D. Smilkov, M. Wattenberg, F. Viegas, G. S. Corrado, M. C. Stumpe, and M. Terry. Human-Centered Tools for Coping with Imperfect Algorithms During Medical Decision-Making. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, pages 4:1–4:14, New York, NY, USA, 2019b. ACM.

A. Campolo, M. Sanfilippo, M. Whittaker, and K. Crawford. AI Now 2017 Report. *Microsoft Research*, February 2018.

R. Caruana, Y. Lou, J. Gehrke, P. Koch, M. Sturm, and N. Elhadad. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1721–1730. ACM, 2015.

C.-H. Chang, E. Creager, A. Goldenberg, and D. Duvenaud. Explaining image classifiers by counterfactual generation. *arXiv preprint arXiv:1807.08024*, 2018.

S. Chen, X. Huang, Z. He, and C. Sun. Damagenet: A universal adversarial dataset. *arXiv preprint arXiv:1912.07160*, 2019.

X. Chen, R. Mottaghi, X. Liu, S. Fidler, R. Urtasun, and A. Yuille. Detect what you can: Detecting and representing objects using holistic models and body parts. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1971–1978, 2014.

F. Chollet. *Deep learning with Python*. Simon and Schuster, 2017.

M. Chromik and M. Schuessler. A taxonomy for human subject evaluation of black-box explanations in xai. In *ExSS-ATEC@ IUI*, 2020.

J. Deng, W. Dong, R. Socher, L.-J. Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, Miami, FL, June 2009. IEEE.

A. Dix. *Human Issues in the Use of Pattern Recognition Techniques*, page 429–451. Ellis Horwood, USA, 1992.

F. Doshi-Velez and B. Kim. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*, 2017a.

F. Doshi-Velez and B. Kim. Towards a rigorous science of interpretable machine learning. 2017b.

M. T. Dzindolet, S. A. Peterson, R. A. Pomranky, L. G. Pierce, and H. P. Beck. The role of trust in automation reliance. *Int. J. Hum.-Comput. Stud.*, 58(6):697–718, June 2003.

M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007). 2007.

M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012). 2012.

J. Fails, D. Olsen, a. a, and b. b. A design tool for camera-based interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '03, pages 449–456, New York, NY, USA, 2003. ACM.

L. Fei-Fei, A. Iyer, C. Koch, and P. Perona. What do we perceive in a glance of a real-world scene? *Journal of Vision*, 7(1):10–10, January 2007.

A. Ghorbani, A. Abid, and J. Zou. Interpretation of neural networks is fragile. 33: 3681–3688, Jul. 2019.

A. Grossmann and J. Morlet. Decomposition of hardy functions into square integrable wavelets of constant shape. *SIAM journal on mathematical analysis*, 15(4):723–736, 1984.

R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi. A Survey of Methods for Explaining Black Box Models. *ACM Comput. Surv.*, 51(5): 93:1–93:42, August 2018.

M. Hamidi-Haines, Z. Qi, A. Fern, F. Li, and P. Tadepalli. Interactive naming for explaining deep neural networks: A formative study. *arXiv preprint arXiv:1812.07150*, 2018.

S.-H. Han, M.-S. Kwon, and H.-J. Choi. Explainable ai (xai) approach to image captioning. *The Journal of Engineering*, 2020(13):589–594, 2020.

R. R. Hoffman, S. T. Mueller, G. Klein, and J. Litman. Metrics for explainable ai: Challenges and prospects. *arXiv preprint arXiv:1812.04608*, 2018.

D. Kahneman and S. Frederick. Representativeness revisited: Attribute substitution in intuitive judgment. *Heuristics and biases: The psychology of intuitive judgment*, 49: 81, 2002.

M. Kahng, P. Y. Andrews, A. Kalro, and D. H. P. Chau. A cti v is: Visual exploration of industry-scale deep neural network models. *IEEE transactions on visualization and computer graphics*, 24(1):88–97, 2018.

J. Kato, S. McDirmid, and X. Cao. Dejavu: Integrated support for developing interactive camera-based programs. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology*, UIST '12, pages 189–196, New York, NY, USA, 2012. ACM.

B. Kim, R. Khanna, and O. O. Koyejo. Examples are not enough, learn to criticize! criticism for interpretability. *Advances in neural information processing systems*, 29, 2016.

B. Kim, C. Rudin, and J. A. Shah. The bayesian case model: A generative approach for case-based reasoning and prototype classification. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 1952–1960. Curran Associates, Inc., 2014.

B. Kim, M. Wattenberg, J. Gilmer, C. Cai, J. Wexler, F. Viegas, and R. Sayres. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). 2017.

P.-J. Kindermans, S. Hooker, J. Adebayo, M. Alber, K. T. Schütt, S. Dähne, D. Erhan, and B. Kim. The (un) reliability of saliency methods. In *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, pages 267–280. Springer, 2019.

J. Kleinberg, H. Lakkaraju, J. Leskovec, J. Ludwig, and S. Mullainathan. Human Decisions and Machine Predictions. *The Quarterly Journal of Economics*, 133(1): 237–293, February 2018.

J. Koenemann and N. J. Belkin. A Case for Interaction: A Study of Interactive Information Retrieval Behavior and Effectiveness. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '96, pages 205–212, New York, NY, USA, 1996. ACM.

P. W. Koh and P. Liang. Understanding black-box predictions via influence functions. In *International Conference on Machine Learning*, pages 1885–1894. PMLR, 2017.

J. Krause, A. Perer, and E. Bertini. A User Study on the Effect of Aggregating Explanations for Interpreting Machine Learning Models. page 10, 2018a.

J. Krause, A. Perer, and E. Bertini. A user study on the effect of aggregating explanations for interpreting machine learning models. *arXiv*, 2018b.

J. Krause, A. Perer, and K. Ng. Interacting with predictions: Visual inspection of black-box machine learning models. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 5686–5697. ACM, 2016.

T. Kulesza, M. Burnett, W.-K. Wong, and S. Stumpf. Principles of explanatory debugging to personalize interactive machine learning. In *Proceedings of the 20th International Conference on Intelligent User Interfaces*, pages 126–137. ACM, 2015.

T. Kulesza, S. Stumpf, M. Burnett, and I. Kwan. Tell Me More?: The Effects of Mental Model Soundness on Personalizing an Intelligent Agent. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 1–10, New York, NY, USA, 2012. ACM.

T. Kulesza, S. Stumpf, M. Burnett, S. Yang, I. Kwan, and W.-K. Wong. Too much, too little, or just right? ways explanations impact end users' mental models. In *Visual Languages and Human-Centric Computing (VL/HCC), 2013 IEEE Symposium on*, pages 3–10. IEEE, 2013.

V. Lai and C. Tan. On Human Predictions with Explanations and Predictions of Machine Learning Models: A Case Study on Deception Detection. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, FAT* '19, pages 29–38, New York, NY, USA, 2019. ACM.

S. Lapuschkin, A. Binder, G. Montavon, K.-R. Muller, and W. Samek. Analyzing classifiers: Fisher vectors and deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2912–2920, 2016.

S. Lapuschkin, S. Wäldchen, A. Binder, G. Montavon, W. Samek, and K.-R. Müller. Unmasking Clever Hans predictors and assessing what machines really learn. *Nature Communications*, 10(1):1096, March 2019.

L. Lascau, S. J. J. Gould, A. L. Cox, E. Karmannaya, and D. P. Brumby. *Monotasking or Multitasking: Designing for Crowdworkers' Preferences*, page 1–14. Association for Computing Machinery, New York, NY, USA, 2019.

J. D. Lee and K. A. See. Trust in Automation: Designing for Appropriate Reliance. *Human Factors*, 46(1):50–80, March 2004.

B. Y. Lim, A. K. Dey, and D. Avrahami. Why and Why Not Explanations Improve the Intelligibility of Context-aware Intelligent Systems. In *Proceedings of the SIGCHI*

Conference on Human Factors in Computing Systems, CHI ’09, pages 2119–2128, New York, NY, USA, 2009. ACM.

P. Linardatos, V. Papastefanopoulos, and S. Kotsiantis. Explainable ai: A review of machine learning interpretability methods. *Entropy*, 23(1):18, 2021.

Z. C. Lipton. The Mythos of Model Interpretability. *Commun. ACM*, 61(10):36–43, September 2018.

D. Lowe. Distinctive image features from scale-invariant keypoints. 60(2):91–110, November 2004.

D. Maynes-Aminzade, T. Winograd, and T. Igarashi. Eyepatch: Prototyping camera-based interaction through examples. In *Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology*, UIST ’07, pages 33–42, New York, NY, USA, 2007. ACM.

T. Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, 267:1–38, 2019a.

T. Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, 267:1 – 38, 2019b.

G. O. Mohler, M. B. Short, S. Malinowski, M. Johnson, G. E. Tita, A. L. Bertozzi, and P. J. Brantingham. Randomized Controlled Field Trials of Predictive Policing. *Journal of the American Statistical Association*, 110(512):1399–1411, October 2015.

C. Molnar. *Interpretable machine learning*. Lulu. com, 2020.

G. Montavon, W. Samek, and K.-R. Müller. Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*, 2017.

J. Muramatsu and W. Pratt. Transparent Queries: Investigation users’ mental models of search engines. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 217–224. ACM, January 2001.

M. Narayanan, E. Chen, J. He, B. Kim, S. Gershman, and F. Doshi-Velez. How do humans understand explanations from machine learning systems? an evaluation of the human-interpretability of explanation, 2018a.

M. Narayanan, E. Chen, J. He, B. Kim, S. Gershman, and F. Doshi-Velez. How do Humans Understand Explanations from Machine Learning Systems? An Evaluation of the Human-Interpretability of Explanation. February 2018b.

A. Nguyen, J. Yosinski, and J. Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 427–436, 2015.

K. Patel, N. Bancroft, S. M. Drucker, J. Fogarty, A. J. Ko, and J. Landay. Gestalt: Integrated support for implementation and analysis in machine learning. In *Proceedings of the 23Nd Annual ACM Symposium on User Interface Software and Technology*, UIST '10, pages 37–46, New York, NY, USA, 2010. ACM.

K. Patel, J. Fogarty, J. A. Landay, and B. Harrison. Investigating statistical machine learning as a tool for software development. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, pages 667–676, New York, NY, USA, 2008. ACM.

V. Petsiuk, A. Das, and K. Saenko. RISE: randomized input sampling for explanation of black-box models. *CoRR*, abs/1806.07421, 2018.

F. Poursabzi-Sangdeh, D. G. Goldstein, J. M. Hofman, J. W. Vaughan, and H. Wallach. Manipulating and measuring model interpretability, 2018.

S. Pouyanfar, S. Sadiq, Y. Yan, H. Tian, Y. Tao, M. P. Reyes, M.-L. Shyu, S.-C. Chen, and S. S. Iyengar. A Survey on Deep Learning: Algorithms, Techniques, and Applications. *ACM Comput. Surv.*, 51(5):92:1–92:36, September 2018.

M. T. Ribeiro, S. Singh, and C. Guestrin. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 1135–1144, New York, NY, USA, 2016a. ACM.

M. T. Ribeiro, S. Singh, and C. Guestrin. Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144. ACM, 2016b.

Y. Rogers. *INTERACTION DESIGN: beyond human-computer interaction, 3rd Edition / Rogers, Yvonne.* 1st edition edition, 2011.

E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. Orb: An efficient alternative to sift or surf. In *Proceedings of the 2011 International Conference on Computer Vision*, ICCV '11, pages 2564–2571, Washington, DC, USA, 2011. IEEE Computer Society.

W. Samek, A. Binder, G. Montavon, S. Lapuschkin, and K.-R. Müller. Evaluating the visualization of what a deep neural network has learned. *IEEE transactions on neural networks and learning systems*, 2017.

A. L. Samuel. Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, 3(3):210–229, July 1959.

M. Schuessler and P. Weiß. Minimalistic explanations: Capturing the essence of decisions. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI EA '19, pages LBW2810:1–LBW2810:6, New York, NY, USA, 2019. ACM.

R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 806–813, 2014.

H. Sharp. *Interaction design*. John Wiley & Sons, 2003.

G. Shmueli et al. To explain or to predict? *Statistical science*, 25(3):289–310, 2010.

B. Shneiderman. Opinion: The dangers of faulty, biased, or malicious algorithms requires independent oversight. *Proceedings of the National Academy of Sciences*, 113(48): 13538–13540, November 2016.

K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps, 2013.

K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *In Workshop at International Conference on Learning Representations*. Citeseer, 2014.

K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition, 2014.

L. Sixt, M. Granz, and T. Landgraf. When explanations lie: Why many modified BP attributions fail. In H. D. III and A. Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 9046–9057. PMLR, 13–18 Jul 2020.

J. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller. Striving for simplicity: The all convolutional net. In *ICLR (workshop track)*, 2015.

A. Springer and S. Whittaker. Progressive Disclosure: Empirically Motivated Approaches to Designing Effective Transparency. In *Proceedings of the 24th International Conference on Intelligent User Interfaces*, IUI '19, pages 107–120, New York, NY, USA, 2019. ACM.

P. Stone, R. Brooks, E. Brynjolfsson, R. Calo, O. Etzioni, G. Hager, J. Hirschberg, S. Kalyanakrishnan, E. Kamar, K. Leyton-Brown, D. C. Parkes, W. Press, A. Saxenian, J. Shah, M. Tambe, and A. Teller. "artificial intelligence and life in 2030." one hundred year study on artificial intelligence: Report of the 2015-2016 study panel, 2016.

H. Strobelt, S. Gehrmann, B. Huber, H. Pfister, and A. M. Rush. Visual analysis of hidden state dynamics in recurrent neural networks. *arXiv preprint arXiv:1606.07461*, 2016.

L. H. Sullivan. The tall office building artistically considered. *Lippincott's Magazine*, 57 (3):406, 1896.

A. Tao, K. Sapra, and B. Catanzaro. Hierarchical multi-scale attention for semantic segmentation. *arXiv preprint arXiv:2005.10821*, 2020.

The IEEE Global Initiative on Ethics of Autonomous and Intelligent Systems. Ethically Aligned Design: A Vision for Prioritizing Human Well-being with Autonomous and Intelligent Systems, version 2, 2017.

G. Tolias, R. Sicre, and H. Jégou. Particular object retrieval with integral max-pooling of cnn activations. *arXiv preprint arXiv:1511.05879*, 2015.

J. Tullio, A. K. Dey, J. Chalecki, and J. Fogarty. How it works: A field study of non-technical users interacting with an intelligent system. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '07, pages 31–40, New York, NY, USA, 2007. ACM.

J. K. M. Verame, E. Costanza, and S. D. Ramchurn. The effect of displaying system confidence information on the usage of autonomous systems for non-specialist applications: A lab study. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pages 4908–4920, New York, NY, USA, 2016. ACM.

H. White. A heteroskedasticity-consistent covariance matrix estimator and a direct test for heteroskedasticity. *Econometrica*, 48(4):817–838, 1980.

J. O. Wobbrock, L. Findlater, D. Gergle, and J. J. Higgins. The aligned rank transform for nonparametric factorial analyses using only anova procedures. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 143–146, 2011.

K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057. PMLR, 2015.

R. Yang and M. W. Newman. Learning from a learning thermostat: lessons for intelligent systems for the home. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, pages 93–102. ACM, 2013a.

R. Yang and M. W. Newman. Learning from a learning thermostat: Lessons for intelligent systems for the home. In *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 93–102. ACM, August 2013b.

M. Yin, J. W. Vaughan, and H. Wallach. Understanding the effect of accuracy on trust in machine learning models. 2019a.

M. Yin, J. Wortman Vaughan, and H. Wallach. Understanding the Effect of Accuracy on Trust in Machine Learning Models. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, pages 279:1–279:12, New York, NY, USA, 2019b. ACM.

J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang. Generative image inpainting with contextual attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5505–5514, 2018.

X. Yuan, P. He, Q. Zhu, and X. Li. Adversarial examples: Attacks and defenses for deep learning. *IEEE transactions on neural networks and learning systems*, 30(9): 2805–2824, 2019.

M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.

A. Zhang, Z. C. Lipton, M. Li, and A. J. Smola. Dive into deep learning. 2020. *URL https://d2l. ai*, 2020.

Y. Zhao, S. Szpiro, J. Knighten, and S. Azenkot. Cuesee: Exploring visual cues for people with low vision to facilitate a visual search task. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '16, pages 73–84, New York, NY, USA, 2016. ACM.