Data Model for Computer Vision Explainability, Fairness, and Robustness

Simran Karnani

Data Model for Computer Vision Explainability, Fairness, and Robustness

by

Simran Karnani

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Tuesday September 11, 2023 at 9:30 AM.

Student number:	4798341	
Project duration:	November 30th, 2022 – September 11th, 2023	
Thesis committee:	Dr. Avishek. Anand,	TU Delft, Supervisor
	Dr. Jie. Yang,	TU Delft, Daily supervisior
	Dr. Qing. Wang,	TU Delft

This thesis is confidential and cannot be made public until September 11th, 2023.

Preface

Two years ago, as I embarked on my master's program, I found myself wondering, ”How can I possibly complete a research project in just nine months?” On the surface, this timeframe seemed generous, yet in retrospect, time has swiftly flown by. What made this particular project so captivating was the different methodology used, which injected an element of excitement and novelty into my academic journey.

This project stood apart from those I had previously undertaken. Throughout my bachelor's and master's studies, I had not encountered an opportunity that required gathering insights directly from professionals in the field. The experience of employing a different methodology was a gratifying aspect of my master's journey, one for which I hold deep gratitude.

Much of the research I had encountered during my master's program had centered on the development of novel methods. Prior to this experience, I had not fully grasped the myriad challenges that researchers face in this domain. This eye-opening encounter not only broadened my understanding but also added an intriguing dimension to my academic exploration.

I would like to start by thanking my supervisor, Dr. Jie Yang or his unwavering support and encouragement during this scholarly endeavor. My initial interest in the WIS group was kindled through his lectures and the profound research he conducted. I also want to thank Agathe Balayn and Lorenzo Corti who provided invaluable guidance and support throughout the course of this project. Their willingness to share their knowledge and offer constructive criticism significantly contributed to the formulation and refinement of my research ideas. Their patience, continuous availability, and willingness to assist were instrumental. Their contributions rendered this project not only feasible but also intellectually enriching.

Subsequently, my heartfelt appreciation extends to my parents, brother, and grandmother, who consistently provided unwavering encouragement from afar, particularly during moments of demotivation. Their emotional support transcended physical distances and was a source of immense strength. In addition, I wish to extend my gratitude to my circle of friends who, at a moment's notice, were ever-ready for impromptu study sessions or simply to engage in conversations unrelated to university matters. Their camaraderie and availability for both academic and personal exchanges were invaluable and deeply cherished.

This journey was not the easiest but it definitely felt rewarding. I take immense satisfaction in the knowledge and skills I have acquired throughout the course of my master's program. I am happy to end my academic journey with this project and people. I am profoundly thankful for the entirety of the support network that has accompanied me on this academic journey. I hope this contribution can make a difference in the field of Computer Vision.

Simran Karnani
Delft, September 2023

Abstract

In recent years, there has been a growing interest among researchers in the explainability, fairness, and robustness of Computer Vision models. While studies have explored the usability of these models for end users, limited research has delved into the challenges and requirements faced by researchers investigating these requirements. This study addresses this gap through a mixed-method approach, involving 20 semi-structured interviews with researchers and a comprehensive literature analysis. Through this investigation, we have identified a practical need for a data model that encompasses the essential information researchers require to enhance explainability, fairness, and robustness in Computer Vision applications. We developed a data model that holds the potential to improve transparency and reproducibility within this field, speed up the research process, and facilitate comprehensive evaluations, whether quantitative or qualitative, of proposed methodologies. To refine and demonstrate the practicality of the data model, we have populated it with four existing datasets. Additionally, we have conducted two user studies to validate the model's usability. We found that participants were enthusiastic about using the data model. Some potential uses described by the participants were comparing models and datasets, accessing (niche) datasets and models, creating and exploring datasets, and having access to ground truth explanations. However, participants also had concerns about the data model, mainly with its usability being restricted to people with database knowledge and the richness of data in the database. Nonetheless, hope that this research constitutes the first step for data modelling for researchers in the field of Trustworthy AI.

Contents

1 Introduction

Computer Vision (CV) and Machine Learning (ML) have witnessed exponential growth and advancement in recent years, revolutionizing various domains and applications, like the finance, health, and political sectors [35], [80]. To accomplish high performance, developers are building bigger and bigger models, with millions of parameters, which overall becomes a black box for humans to understand [73]. While the model shows great performances, in terms of accuracy, it has raised further questions about the transparency, trust, and reproducibility of outputs from these systems. The lack of transparency can make it difficult for end users to interpret and understand the outcomes of such systems. We do not know how the model is making its predictions, where the model is biased, or how it will perform on real-world data. It also makes it difficult for users to form a meaningful trust relationship with such systems [4]. Furthermore, there has been a lack of reproducibility and knowledge of such systems. It is shown that with the rapid pace of development of new models, datasets and explanation methods, users tend to make choices using unprincipled reasons, such as popularity and familiarity, rather than understanding the strengths and weaknesses of the methods [19].

In an attempt to solve these issues, researchers shifted their focus to developing new models and metrics that look beyond the common evaluation metrics, such as accuracy. Their focus is on other requirements such as the model's *robustness* to distributional shifts in real-world settings [77] [50], *fairness* [108], [25], and *explainability*, in terms of interpretability [83]. Over the last couple of years, both the algorithmic and Human-Computer Interaction (HCI) research communities have shifted their focus to fulfil these requirements. The algorithmic community has been developing new methods and metrics to explain model inferences and ensure the models are fair and robust. On the other hand, the HCI community is studying the practical uses of these existing methods to see whether the proposed evaluations align with human expectations and are building tools to make these methods more accessible to developers.

With this in mind, researchers have also built tools [17], [14], [45] and resources [37], [75], [19] that can be used to encourage transparency and reproducibility in such systems about these new requirements. There is no single definition for these requirements, so for the remainder of this report, we will use the following definitions concerning Computer Vision.

- explainability: The reasons or justifications for a given output or decision in a human-readable manner [31].
- Fairness: The ethical principle of ensuring that these systems do not exhibit bias or discrimination towards individuals or groups based on protected characteristics such as race, gender, age, or other sensitive attributes [58].
- Robustness: The ability of these systems to maintain accurate and reliable performance under various challenging conditions, such as changes in lighting conditions, image noise, occlusions, variations in viewpoints or adversarial attacks [107].

While these contributions have been an important asset to the ML community, little work has been done to address the challenges researchers have been facing when working on such tasks. To our knowledge, the majority of the works focus on the challenges and needs of practitioners in this field to improve their experiences with AI [9], [8], [104], [51]. However, there are few to no works that focus on the needs of researchers in their daily tasks of implementing new CV methods or analyzing existing ones. The existing toolkits either target a single requirement (explainability [48], fairness [85]) or multiple requirements (explainability and fairness [14]). However, no tool allows users to address these three requirements combined.

There has also been a large focus on creating new algorithms and optimizing the existing ones, while very little attention is given to the data. Due to the lack of attention to the optimization of data and the way it is structured, researchers and practitioners spend more time organizing data when working on CV and ML tasks. They also spend quite some time trying to get access to datasets because datasets that are mentioned in many studies are not accessible. The lack of structure and accessibility makes it difficult for researchers to reproduce studies.

Research Question We believe that providing a tool that acts like a central hub containing structured and sufficient information about the needs of researchers for such tasks, can help users to search and have access to datasets and models, clear explanations, and quickly set up experiments. To achieve this, we have drawn inspiration from works situated at the intersection of machine learning and HCI that investigate how machine learning and related tools are used. With this as a starting point, this research aims to answer the following research question: *How to design a data model that can support researchers when performing Computer Vision classification tasks in terms of explainability, fairness, and robustness?*

Method We followed a mixed-method study to create this data model, more specifically the data model, which encompasses works towards solving these problems in a structured manner. We first conducted a rigorous literature study to identify the requirements for computer vision classification tasks in terms of explainability, fairness, and robustness. With this information, we created an initial data model. We fine-tuned the data model by fitting four existing datasets. We found that the literature study did not cover the researcher's challenges and desires they had to make their work easier and faster, thus we followed with semi-structured interviews, with 20 participants to understand the limitations and challenges they have encountered, along with verifying our data model design. These interviews were analyzed using a thematic analysis to complete the data model and examine the potential use cases of the model. Finally, we conducted two more interviews asking users to interact with the data model to validate its usability.

Results Through this, we were able to create a data model for CV classification tasks, which can be used as a support research tool for explainability, fairness, and robustness. Through the interviews, we found that the participants were enthusiastic about using such a tool, where they can easily access and explore data, compare models, and have clear definitions of explainability. We believe that, through the interviews, we have found sufficient results about the challenges and needs of researchers in this field. Having this tool can be a stepping stone to solving the time-consuming and expensive computations of annotating images, ground truth explanations, and ground truth labels. As more data is labelled and shared, researchers can evaluate their methods on a variety of data rather than only the most popular datasets. However, there were also some concerns in terms of storage and reliability of data and the restriction to SQL knowledge. Nevertheless, we hope that this research constitutes a new avenue for future works around data modelling for CV applications and research.

Users of the model This data model is intended to address the needs of research creators and research consumers. For the creators of a new dataset, model, or innovative method, this model aims to encourage them to carefully reflect on the process of creating, defining, distributing, and maintaining these innovations, while increasing transparency and usability. Including this information in the data model will not only give other researchers access to these new methods and datasets but also translate the required knowledge so that they are used as intended. The research consumers on the other hand are provided with all the information necessary to make informed decisions about the datasets, models, and explanations. This can mitigate unwanted biases in machine learning models, foresee and prepare for potential fails when the model is deployed in the real world, and help researchers and practitioners to choose the datasets, models and explanations for their research and problems [37].

Contributions The main contributions of this research are:

1. A (flexible) structured data model containing metadata of CV models, datasets and semantic concepts and explanations for Computer Vision classification tasks. A database is created to store this information for four datasets.

2. A comprehensive qualitative analysis of the existing challenges and limitations encountered by researchers in the domains of explainability, fairness, and robustness

3. A qualitative study evaluating the data model based on the design principles: completeness and usability

The structure of the report adheres to the following organization: Chapter 2 provides an overview of the background and tools specifically developed to address the challenges of explainability, fairness, and robustness in the context of the study. In Chapter 3, the methodology employed in the research is detailed, outlining the approach taken to collect and analyze data. The subsequent section, Chapter 4, presents a detailed explanation of the data model, specifically describing the way it is modelled and an in-depth description of the entities and attributes. Chapter 5 describes the findings obtained through the interviews, highlighting user suggestions to complete the data model and analyzing its usability. Finally, Chapter 6 concludes the research by discussing the strengths of our data model in comparison to other toolkits that are available in this field, users' concerns about the data model, and the limitations found in our study.

2 Related Works

The increasing adoption of AI systems has led to greater recognition of their limitations and challenges [91]. To address these issues, there has been a surge of research focused on developing new methods and conducting user studies in the domains of explainability, fairness, and robustness to increase understanding, reproducibility and transparency in these systems[12], [73]. In this section, we provide an overview of relevant prior works in the field of CV that have tackled these topics, aiming to advance the understanding and improvement of AI systems. In Section 2.1, we discuss prior works for explainability, fairness and robustness of CV applications and Section 2.2 discusses some literature and tools that were built to encourage transparency and reproducibility for the above-mentioned non-functional requirements in models and the research process. Lastly, Section 2.3 discusses data modelling.

2.1. Background

In the field of CV, the importance of explainability, fairness, and robustness has become increasingly evident. Researchers have devoted significant efforts to developing innovative methods that address these three essential non-functional requirements [89]. In this section, we provide a comprehensive exploration of prior studies that delve into these requirements, examining their intricacies and implications within the domain of CV. By delving into these works, we aim to gain a deeper understanding of the significance and impact of explainability, fairness, and robustness in the context of AI systems for the classification of visual tasks.

2.1.1. Explanations

The goal of explainability for CV tasks is to highlight the visual features that were used by the system to make a prediction. The explanations can be visual (e.g., a visual representation of the features, such as saliency methods) or textual (e.g., a semantic concept that represents the visual feature). These explainability types are further described in this section, along with their example methods.

Saliency Methods Saliency methods have emerged as valuable tools for establishing the relationship between a model's predictions and the inputs that significantly influence those predictions [60], [19]. The explanations provided by saliency methods find diverse applications [83], including the debugging of a model's prediction process, verification of the absence of spurious correlations in the model's learning [83], and scrutiny of the model for potential fairness-related concerns [86]. Many different saliency methods have been developed over the years, often guided by visual appeal on image data [1].

One approach to creating saliency maps is by using occlusion techniques or calculations with gradients to assign an importance score on each pixel to reflect its influence on the final classification. Smilkov et al. [93] proposed the method SmoothGrad based on this approach. Unlike previous methods which produced noisy saliency maps, by highlighting some randomly selected pixels in addition to the important pixels, SmoothGrad, tends to reduce the visual noise by first adding noise. SmoothGrad can help to visually sharpen gradient-based sensitivity maps by taking an image of interest, sampling similar images by adding noise to the image, and then taking the average of the resulting sensitivity maps for each of the sampled images. Figure 2.1 shows some examples of saliency maps using SmoothGrad on the ColorMNIST dataset. The highlighted pixels are the ones the model used to make its prediction.

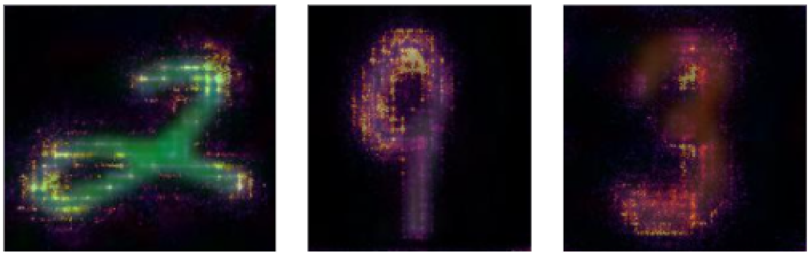


Figure 2.1: Examples of saliency maps using SmoothGrad

Another saliency method commonly used in practice is Local Interpretable Model-agnostic Explanations (LIME) [83]. It is built upon a vital concern with AI systems that if users do not trust the model or prediction, they will not use it. Trust can be broken down into trusting an individual prediction to task some action based on it and trusting a model to behave in a certain way when it is deployed. LIME, an algorithm that can explain predictions of any classifier or regressor in an interpretable and faithful manner, was designed as a solution to build trust in a prediction and a model. It does this by learning an interpretable model locally around the prediction. The explanations provided by LIME have four characteristics: interpretability (providing qualitative understanding between the input variables and the response), local fidelity (corresponds to how the model behaves in the vicinity of the instance being predicted), model-agnostic (it should be able to explain any model), and global perspective (gives users a reason to trust the model). Figure 2.2 shows some examples of saliency maps using LIME on the Imagenette dataset. The pixels enclosed within the yellow border are the ones the model used to make its predictions.

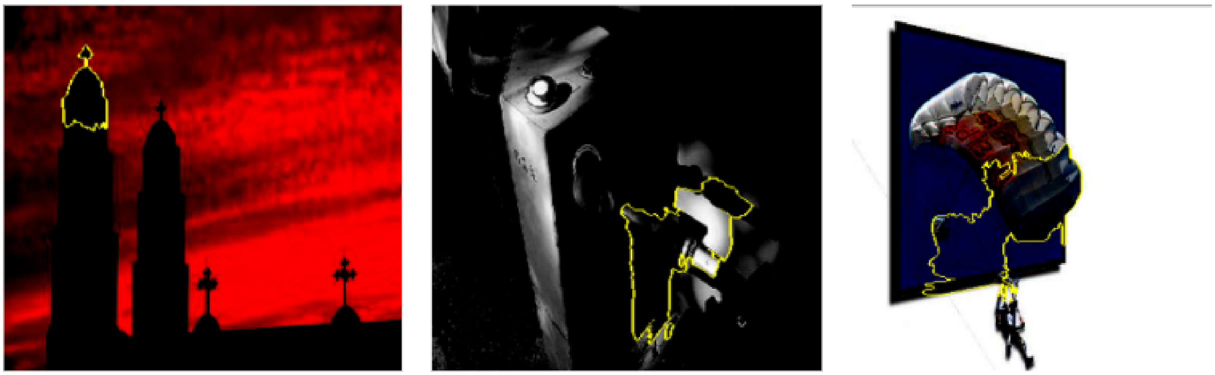


Figure 2.2: Examples of saliency maps using LIME

The last saliency method to be described is XRAI [60], a region-based saliency method based on Integrated Gradients [97]. XRAI is a saliency method that follows a two-step process. Firstly, it over-segments images by dividing them into smaller regions. Then, it iteratively assesses the importance of each region by assigning attribution scores. Smaller regions are progressively added to larger segments based on their attribution scores. This approach allows XRAI to highlight the significant regions in an image. The versatility of this saliency method lies in its compatibility with any deep neural network-based model, provided that there is a means to cluster input features using a similarity metric, such as color similarity in the case of images. Figure 2.3 shows some examples of saliency maps using XRAI. The top row shows the original image and the bottom row shows the parts of the images that the model used to make its predictions.

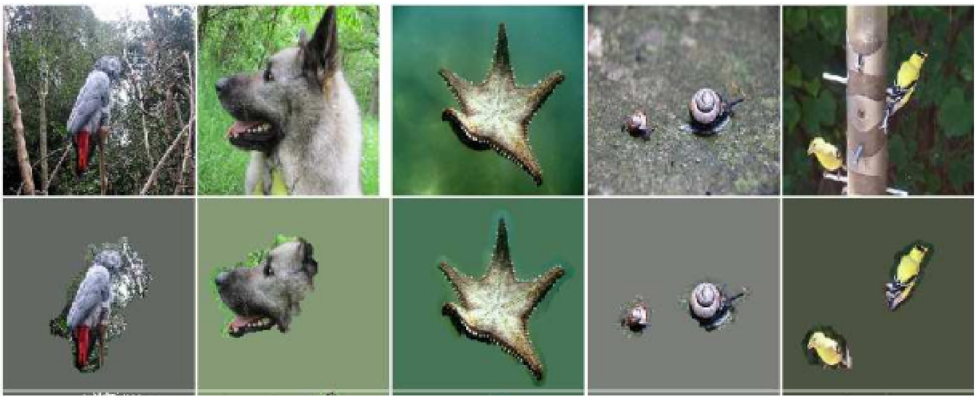


Figure 2.3: Examples of saliency maps using XRAI (Image taken from [60])

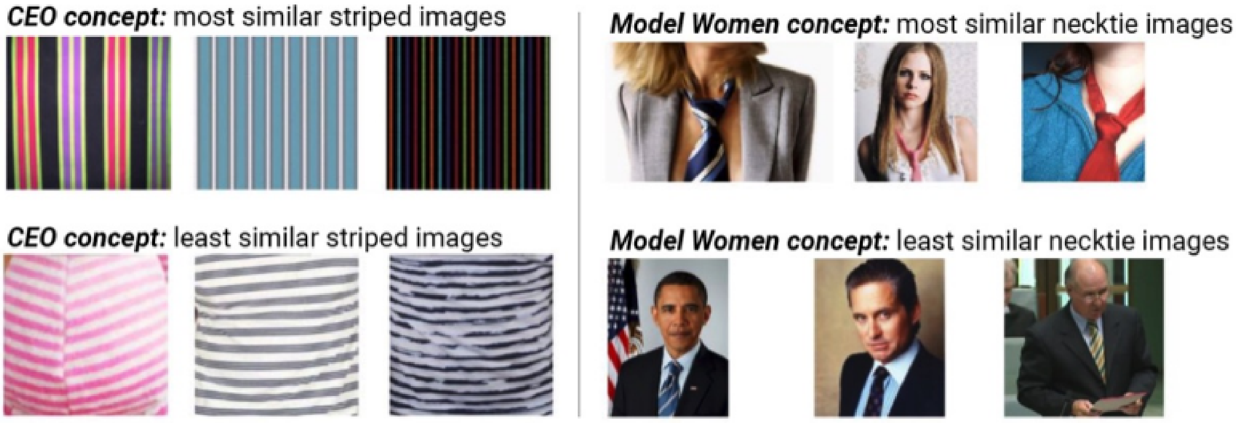

CEO concept: most similar striped images
Model Women concept: most similar necktie images
CEO concept: least similar striped images
Model Women concept: least similar necktie images

Figure 2.4: Global interpretability using TCAV (image taken from [61])

Semantic Concept-based Explanations Explanations play a vital role in enabling humans, including domain experts, researchers, and practitioners, to comprehend the reasoning behind model decisions. Traditional methods of highlighting important features in a sample tend to focus on the local behavior of individual data points, rather than providing a global understanding of how the model operates. Moreover, these methods may not be the most intuitive explanations for human comprehension, particularly when relying on low-level features such as raw pixel values. Human reasoning involves concept-based thinking, wherein similarities are extracted from numerous examples and semantically grouped based on their resemblances [114]. In this subsection, we discuss a few concept-based explainability methods.

Testing with Concept Activated Vectors (TCAVs) and Automatic Concept-based Explanations (ACE) are two methods that explain a model's behavior by identifying salient patches. TCAV is a method that leverages Concept Activated Vectors (CAVs) to analyze the sensitivity of a model's predictions to high-level concepts [61]. CAVs represent the direction of activations associated with a specific concept's set of examples. They are obtained by training a linear classifier between the concept's examples and counterexamples and then extracting the vector orthogonal to the decision boundary. TCAVs utilize directional derivatives to quantify how sensitive the model's predictions are to the underlying high-level concept learned by the CAV. Different types of CAVs were collected from popular image search engines, such color, texture, race, gender, and objects. Figure 2.4 shows examples of global interpretations learnt by a model. The image on the top left shows a vertical pinstripe pattern that was most commonly associated with a CEO (tie or suit). The image on the right shows sorted images of neckties on 'model women'. The images on the top right show women wearing neckties.

ACE is a systematic framework that automatically identifies higher-level concepts that are meaningful to humans and are important for the machine learning model. It does this by aggregating related local image segments across the data [39]. It takes a trained classifier and a set of images of a class as

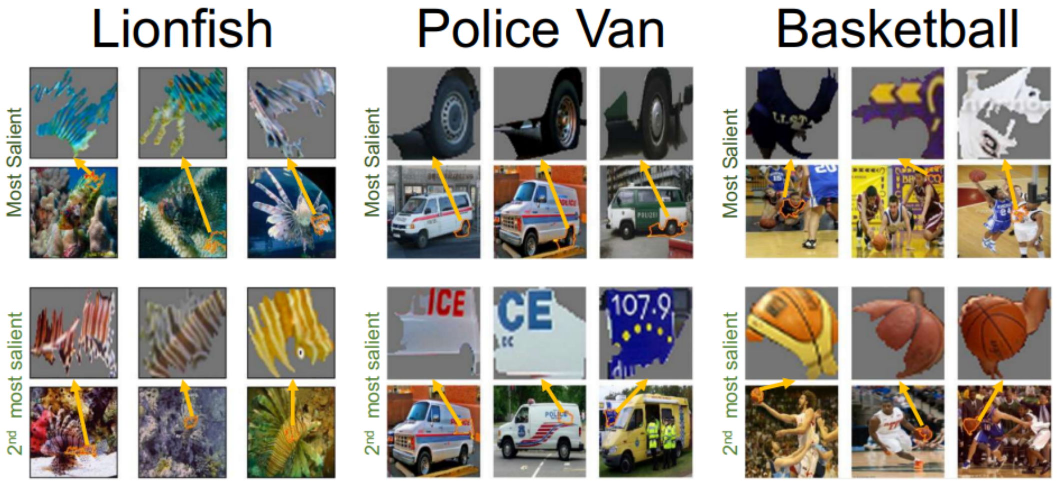
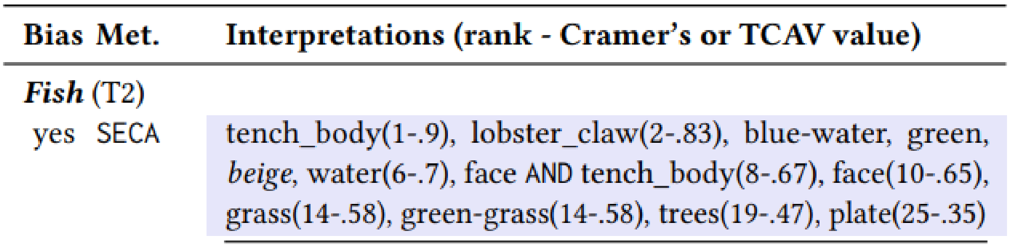


Figure 2.5: Global interpretability using ACE (image taken from [39])

input, extracts concepts present in that class and finally returns the importance of each concept. It also segments images with multiple resolutions. This helps to capture concepts from simple fine-grained granularity, such as textures and colors, to more complex and course-grained concepts such as parts and objects. Figure 2.5 shows examples of the most salient concepts for some Imagenet classes. It can be seen that the network classifies basketball using basketball jerseys and the basketball itself.

While TCAV and ACE focus on global patches, Balayn et al. [11] proposed SECA, a human-in-the-loop Semantic Concept extraction and Analysis framework that provides explanations at a global and local level using textual concepts. It generates explanations with a set of semantic concepts that are easily understood by humans. It combines local interpretability methods that identify image patches that are relevant to the prediction for an image with human computation to annotate the patches with semantic concepts (i.e., visual entities with types and attributes). Figure 2.6 shows the interpretations outputted by SECA using statistical testing.



Bias Met.	Interpretations (rank - Cramer's or TCAV value)
Fish (T2)	
yes SECA	tench_body(1-.9), lobster_claw(2-.83), blue-water, green, *beige*, water(6-.7), face AND tench_body(8-.67), face(10-.65), grass(14-.58), green-grass(14-.58), trees(19-.47), plate(25-.35)

Figure 2.6: Textual representations of global explanations for a fish (image taken from [11])

Explainability researchers are engrossed in developing methods and their evaluation procedures to increase the understanding, reliability, and usefulness of machine learning models. However, the lack of a tool where such methods can be appropriately compared makes it difficult for users to compare existing explainability methods. The data model we propose allows users to compare different explainability methods for different networks and datasets such that users can decide which explainability method they prefer for the given dataset and CV model.

2.1.2. ML Fairness and Robustness

Prior works regarding ML fairness focus on identifying bias in datasets and models. They work on mitigating bias to make the output of models fair concerning various protected attributes, such as skin color, gender, or age. It has been seen that algorithms that have been trained on biased data have resulted in algorithmic discrimination [21]. Furthermore, it has been noticed that some face recognition systems misidentify people of color, women, and young people at high rates [36]. Researchers also found that these models tend to underperform when given real-world data. This is because these samples contain various perturbations and corruptions that are usually not present in the training set. Thus researchers are looking for ways to create more robust CV models. In this subsection, we discuss some of the biases found in datasets and models along with the methods that are used to make models more fair and robust.

Identifying bias The study conducted by Buolamwini and Gebru [25] assessed the presence of bias in automated facial analysis algorithms and datasets, specifically focusing on phenotypic subgroups. Instead of solely evaluating models based on skin color or gender, the authors adopted an intersectional approach to examine subgroups. Their evaluation revealed that commercial face recognition systems exhibited significant misclassification rates for dark-skinned females. The authors also discovered that the widely used facial analysis benchmarks, namely Adience and IJB-A, predominantly consist of individuals with light skin tones. They also curated a new dataset with balanced data for each intersectional subgroup. In this paper, they found that bias was present in these phenotypic subgroups rather than simply skin color or gender separately. Identification of these subgroups helps developers understand where the model is biased and the subgroup in which data is lacking.

Protected attributes encompass not only gender, skin color, or age, but also attributes indirectly associated with these factors. When CV models perform tasks, they extract relevant statistics from the training data. These statistics can range from low-level features like color or composition (e.g., dolphins are typically grey, cars have wheels) to contextual or societal cues (e.g., basketball players wearing jerseys, the association of programmers with males) [92]. The models learn these discriminatory cues while training on unrelated and general tasks, often amplifying their influence. Similar associations were found when researchers [108] analyzed the CelebA dataset. They discovered correlations between attribute presence and the gender of individuals depicted in the images. For instance, they found a correlation between smiles and females in the dataset. These findings shed light on the intricate connections between attributes, discriminative cues, and their relationships to protected characteristics.

Bias does not only occur from objects or identification attributes (gender, skin color) present in an image. Wang et al. [108] conducted a study that revealed how data containing more information, such as colored images, can result in reduced control over the discriminatory cues present in the data [22]. They found that the lack of information in black and white images results in the network picking up on less bias.

Mitigating bias After identifying the multiple ways in which bias can be present in a dataset, Ramaswamy, Kim, and Russakovsky [82] proposed a strategy to mitigate bias that stems from correlations. This was done by using GANs to generate realistic images and make perturbations to the images in the latent space to generate training data that is balanced for each protected attribute. Experimenting with the CelebA dataset, they found attributes that were gender-dependent and gender independent. For example, attributes such as 'ArchedBrows', 'Attractive', 'BushyBrows', 'PointyNose' and 'RecedingHair'. Many annotations were also gender independent, meaning that they did not depend on gender expression. Examples are 'Bangs', 'BlondHair', 'Chubby', 'Earrings', 'Glasses', 'MouthOpen', and 'Smiling'. De-correlating attributes from gender, where possible, results in more fair classifiers as shown in the study.

Robustness It was found that models often lack robustness to small translations in the input data [6], small adversarial perturbations [98], [41], and commonly occurring image corruptions (e.g. brightness, fog, blurring) [81]. Designing models to be robust to these distributional shifts is essential for deploying models in complex, real-world settings where the test data is not a perfect reflection of the training data [77].

Hendrycks and Dietterich [50] introduce two benchmarks designed to assess the robustness of machine learning models to common corruptions and perturbations. The first benchmark, called ImageNet-C, consists of a collection of common visual corruptions applied to the ImageNet dataset. These corruptions encompass various types such as noise, blur, weather effects, and digital artefacts. The second benchmark, ImageNet-P, comprises perturbed or subtly modified versions of the original ImageNet images. These perturbations include changes in blur, brightness, zoom, rotation, scale, and more. Figure 2.7 shows some examples of corruptions and perturbations applied to a bird image. While humans generally do not struggle with small changes or the presence of corruptions in images, machine learning models can easily be misled and produce incorrect predictions when faced with such challenges. This

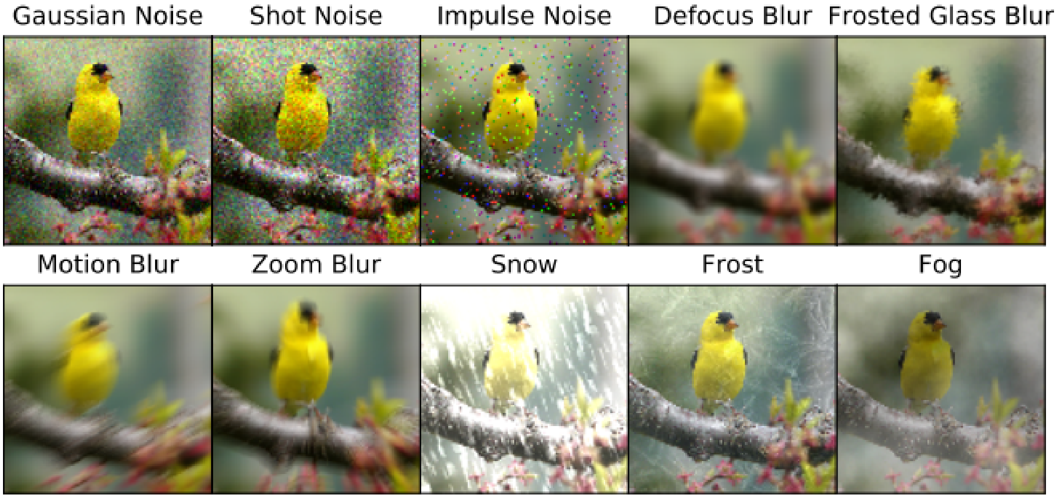


Figure 2.7: Visual corruptions and perturbations applied to images

discrepancy between human and machine perception is particularly evident when models are trained on simulated datasets and then perform poorly on real-world data. By incorporating these corruptions and perturbations into the testing distribution, researchers can gain insights into the robustness and generalization capabilities of models, helping to bridge the gap between simulated and real-world performance. They can also understand how the model will work and ways in which it may fail when given real-world data.

These studies take into consideration various constraints and offer approaches to address fairness and robustness concerns in machine learning models. They propose transformations of the training data, feature embeddings, training procedures, or adjustments in the post-processing of inferred label likelihoods. Another line of research explores procedural fairness or robustness, aiming to ensure that models do not rely on certain ”unfair” features or features that do not directly correlate to the class [79], [42]. In this context, explainability plays a vital role in identifying the features learned by the model, enabling a deeper understanding of the decision-making process and potential biases. To our knowledge, there are no tools that incorporate all three non-functional requirements: explainability, fairness, and robustness. These three requirements rely on many common types of information, therefore having a tool that addresses all three requirements is a possibility. By incorporating explainability techniques in our data model, researchers can assess the fairness of models and identify any discriminatory factors that may have been learned. Additionally, having a tool that contains information about the dataset and model can help users make informed decisions.

2.2. Reproducibility and Transparency

Machine learning models are used to automate tasks and even go beyond human performance. However, the lack of transparency and reproducibility in these models hinders their scientific value [44]. To address these issues, researchers have proposed several approaches to enhance transparency and reproducibility in machine learning. This section provides an overview of the works that have been proposed to tackle these challenges, along with the artefacts developed to facilitate transparency and reproducibility in the field.

2.2.1. Works Encouraging Reproducibility and Transparency

Over the last few years, authors have shifted their focus to work to make data, machine learning models, and explainers transparent and reproducible. Gebru et al. [37] proposed Datasheets for Datasets, which are additional documentation for datasets. Data plays a crucial role in a machine learning model’s performance. Datasheets play a crucial role in providing comprehensive information about datasets, including their motivation, composition, collection process, and recommended uses. By documenting such details, datasheets enable developers to gain insights into potential biases in the system and anticipate the model's performance post-deployment. They serve as a means to enhance transparency and accountability within the machine learning community, mitigate unintended social biases in machine learning models, promote reproducibility of results, and aid researchers and practitioners in selecting suitable datasets for their specific tasks. The availability of datasheets contributes to a more informed and responsible use of machine learning models.

In addition to Datasheets for Datasets [37], Mitchell et al. [75] proposed Model Cards, which are short documents that accompany trained machine learning algorithms. They provide benchmarked evaluations for several conditions, such as across different cultural, demographic, or phenotypic groups and intersectional groups. They also contain the context in which models are intended to be used and details of the performance evaluation procedures. Model cards can provide users with information about what machine learning systems can and cannot do, the types of errors they make, and additional steps that could create more fair and inclusive outcomes with the technology.

For the explainers, Boggust et al. [19] proposed, Saliency Cards, a framework to characterize and compare saliency methods. Saliency cards describes how saliency methods operate and their performance across a large number of evaluation metrics. They document a saliency method's methodology, sensitivity, and perceptibility, such that users can easily access and compare the implications of different methods.

While these lay the groundwork and motivate reproducibility and transparency of the data and models, they do not contain the relationships between these entities. These relationships are important to know how the model performs on different data, the type of explainer required for the dataset and model, and the type of data required to train the model. They also do not store as much information as stored in the data model, such as the samples, with labels and ground truth explanations. These works contained information required by users to replicate an experiment and thus we rely on these works for our data model. Furthermore, the data model also gives users access to the datasets and models, which is usually a major constraint for many experiments.

2.2.2. Artefacts

With the high surge of new ML methods, tools were developed to support researchers and practitioners in using relevant metrics and methods or visualising relevant information for non-functional requirements.

Explainability For the explainability requirements, tools such as XAITK [53], Quantus [48], and AI Explainability 360 [5] have been proposed. XAITK (eXplainable AI ToolKit) [53] offers a collection of algorithms and methods that enable users to generate explanations for predictions made by ML models. It included techniques such as saliency mapping, feature importance analysis, and concept activation analysis. This toolkit also provides tools for model interpretability, model debugging, and fairness assessment, allowing users to evaluate and improve the trustworthiness and fairness of their AI systems.

Quantus [48], another explainable AI toolkit, focuses on providing tools and methods to assess the quality and reliability of explanations generated by neural networks. The toolkit offers tools and techniques to assess the quality of explanations and examines their impact on human decision-making, facilitating the development of more trustworthy and user-friendly AI systems. The evaluation techniques can be used to assess the faithfulness, stability, and sensitivity of the explanations, allowing users to determine how well the explanations align with the model's internal workings and how consistent they are across different inputs.

The AI Explainability 360 Toolkit [5] is an open-source software toolkit featuring a collection of diverse state-of-the-art explainability methods, evaluation metrics, and an extensible software architecture that organizes these methods according to their use in the AI modelling pipeline. The algorithm can be applied to various machine learning models to allow users to gain insights into the inner workings of the models. It also provides a user-friendly interface and visualization tools to help users understand and interpret the explanations generated by the algorithms. This allows users to explore the feature importance, decision boundaries, and other relevant information related to model predictions. Furthermore, the toolkit also includes evaluation metrics and techniques to assess the quality and fairness of the explanations, enabling users to gauge the reliability and unbiasedness of the generated explanations.

Fairness In terms of fairness, tools such as Aequitas [85], AI Fairness 360 [14], and FairLearn [17] were proposed. Aequitas [85] is a bias and fairness audit toolkit that was designed to evaluate and assess bias and fairness in predictive models. It enables users to quantitively measure and analyze different dimensions of bias, compare model performances, and explore fairness interventions. It also offers statistical methods and visualizations to identify and analyze potential biases and disparities in model predictions.

AI Fairness 360 (AIF360) [14] is an open-source toolkit that facilitates the assessment and mitigation of bias and unfairness in ML models. It provides a comprehensive set of algorithms, metrics, and tutorials to help developers and researchers analyze and address bias-related issues in their systems. The algorithms help in adjusting training data to reduce unfairness or modifying the predictions for ML fairness without significantly sacrificing overall accuracy. AIF360 also includes tools for data exploration, visualization, and fairness-aware model selection.

Fairlearn [17] is an open-source toolkit that equips practitioners and researchers working on fairness in AI with the necessary tools and methods to assess, analyze, and improve fairness in machine learning models. It provides a set of algorithms, metrics, and visualization tools to measure and mitigate various forms of unfairness in machine learning models. The algorithms help to reduce unfairness and achieve better balance among different groups and the metrics enable users to quantify and evaluate disparities and biases present in models.

Robustness Lastly, for robustness, the Amazon SageMaker Clarify [45] was proposed. Amazon SageMaker Clarify is a cloud-based service that offers a comprehensive set of tools and features for machine learning bias detection and explainability. The service offers pre-built bias detection algorithms that enable users to identify potential biases across various protected attributes, such as race or gender, in their training data and model predictions. These algorithms utilize statistical techniques to measure and quantify bias, providing actionable insights into the fairness of the models. It also provides visualizations and reports that help users visualize and communicate the detected biases and explanations effectively.

Platforms storing datasets and models Some resources contain information about datasets and models. For example, Papers with code [1], Kaggle [2], and Huggingface [56]. Papers with Code is a free online resource that aims to bridge the gap between research papers and code to promote transparency and reproducibility. It does this for both models and datasets. Kaggle serves as a hub for sharing datasets, code, and kernels. Users can explore and download publicly available datasets, access and contribute to open-source code repositories, and share their projects and analyses. Lastly, Huggingface is an open-source community that hosts several fine-tuned models and datasets.

The toolkits and resources mentioned above do not follow a common, structured data model, thus making it difficult to compare the requirements in terms of models and datasets. To our knowledge, Huggingface is the only dataset library that encourages a structured description of data. While some tools might account for the requirements, they do not directly enable researchers to structure datasets. Moreover, they do not allow researchers to compare models for datasets and provide them with human-readable explanations. Therefore, we propose a data model that contains information required to research explainability, fairness and robustness, along with access to structure, search, and reuse datasets in a unique format. It also provides a more in-depth description of what the samples of a dataset contain before having to download the dataset.

Works at the intersection of HCI and AI There have been many studies focusing on practitioners' needs and challenges revolving around explainability, fairness, and robustness for CV models. Many tools have been designed to address these three requirements, however, these tools must be built with real-world needs in mind. Therefore, researchers are investing more time into understanding these needs from practitioners in the field. Studies have focused on understanding the goals and needs of practitioners, their workflows, the artefacts they use and the challenges and limitations they face during this process. In the works of CV and explainability, Balayn et al. [8] conducted interviews with

[1] https://paperswithcode.com/

[2] https://www.kaggle.com/

practitioners to understand how CV model failures are being handled. Balayn et al. [9] also interviewed ML practitioners to understand how explainability methods can be used to identify bugs in CV models. Verma et al. [104] conducted a study where they interviewed medical professionals to understand what they want AI to be in oncology. In the works of fairness, Holstein et al. [51] conducted interviews and surveys with ML practitioners to identify the challenges and needs to create fairer systems. However, to our knowledge, there are no works that focus on the challenges and needs of researchers in this field. Taking inspiration from prior works at the intersection of HCI and AI, we used similar methods and interviews to understand the challenges and needs of researchers in the field of CV to make these models more explainable, fair, and robust.

2.3. Data Modelling

A data model comprises a set of rigorously defined mathematical concepts that facilitate the exploration and representation of both the static and dynamic characteristics within data-intensive applications [24]. PostgreSQL, an example of a relational model, possesses mechanisms capable of emulating numerous semantic and object-oriented data modeling structures. These encompass generalization, complex objects with shared components, and attributes that establish references to tuples within other relations [84]. These data models which are based on Entity-Relation (ER) models can be visually represented using ER diagrams. According to the study performed by Song, Evans, and Park [94], ER diagrams can have different features and notations. Some of these differences are whether they allow attributes in a relationship, how they represent cardinality, whether they show total/partial specialization, and whether they model the foreign key at the ER diagram level.

Using this information, we designed our data model to allow attributes in a relationship and show total/partial specialization. We also did not include foreign keys at the ER diagram level. The cardinality representation is further described in Section 4.1.

3
Method

To address the current problems with transparency, usability, and feasibility of data, explanations fairness, and robustness, a data model was built that can structure the data such that data is available to all researchers in a coherent structure. To do this the data model was created using a 3 stage procedure. The model was initially designed based on literature, followed by an iterative updating procedure and fitting different datasets to it, and then completing the model based on interview results using a thematic procedure. Another round of interviews was held to analyze the usage of the data model. Figure 3.1 shows a visual representation of this procedure. The subsequent sections provide a comprehensive elaboration of the aforementioned steps.

3.1. Model from Literature

To design the initial model, a literature analysis was conducted to find relevant information based on prior knowledge. The papers collected varied over a range of topics such as CV explanations, concept-based explanations, and ML fairness and robustness. The papers were subdivided into the following sections: propositions of new CV methods or explanations for CV methods, evaluations of existing methods and models, and studies on how researchers may use or develop existing methods, and the needs and additional requirements related to them. Table 3.1 shows an overview of the papers used for the literature analysis and the total count for each category. The papers were gathered through a combination of keywords: *XAI, Deep Learning, CV, Explanation, ML Fairness, Robustness and Transparency*. Along with a keyword search, papers were also collected from scientific venues: NeurIPS, FAccT, and CHI, and snowball sampling. A total of 63 papers were collected and analyzed.

After collecting the literature, information was extracted based on what the authors focused on, metadata, the details they provided when performing experiments and results (eg. the model used, the performance) and the strengths and limitations of the method. For new methods, we looked at the contributions brought by the author and for the evaluations of these methods and user studies, we

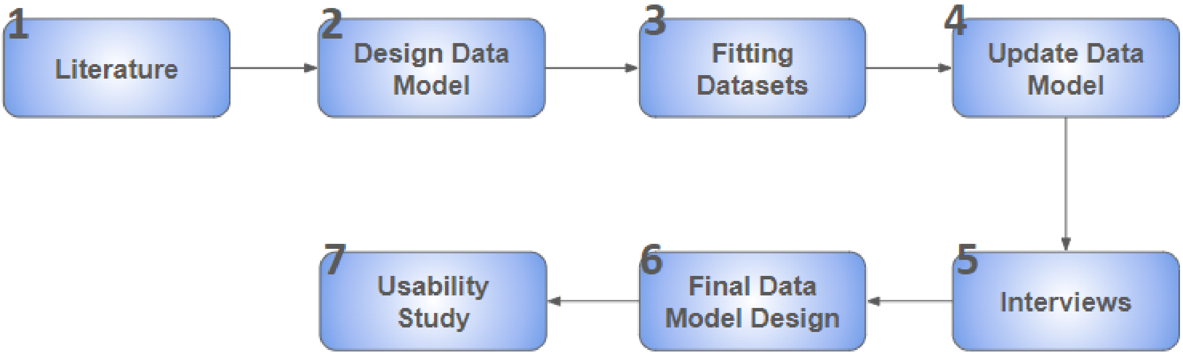


Figure 3.1: Steps taken to build the data model

Contribution category	Explainability	T	Fairness	T	Robustness	T
New Method	[117], [90], [83], [28], [61]. [120], [39], [11], [18], [10], [20], [87], [88], [60], [86], [114]	16	[108], [111], [58], [33], [40], [46], [54], [57], [59], [63], [103], [109], [110], [116]	14	[27], [38], [62], [55], [118], [68], [66], [2], [113], [65], [47], [3], [119]	13
Evaluation	[1], [52], [60]	3	[25], [112], [15], [72], [13]	5	[108], [111], [115], [77], [32], [78], [30], [16], [106]	9
Human-centred Study	[74], [37], [75], [19]	4	[4], [37], [75], [19], [26]	5	[8]	1
Total		23		24		23

Table 3.1: Bibliography for the literature analysis

focused on the strengths and limitations of the method and information provided in the literature. An example of each of these is displayed in Table 3.2.

	Method	Paper
Strength	Injected bias can be used to debug a model to see whether it is able to pick up on this bias [18]	Describes additional quality metrics (e.g. amount of time users took, user retention) for human in the loop methods [76]
Weakness	Human in the loop concept explanations can be tedious and require a lot of manual labour [7]	Lacks information to reproduce the method (e.g. model metadata, dataset metadata) [29]

Table 3.2: Examples of information that was collected from literature

After collecting information from the literature, the information was transitioned into a data model. Metadata about datasets, models, and explanations were identified and translated into a model with additional entities and relationships. The model was iteratively constructed using a thematic approach where themes such as explanations, datasets, and models were identified and backed by codes within each. Figure 3.2 shows some example themes (blue), subthemes (green), and codes (yellow) that were identified in the literature. For example, a dataset can be constructed using a human-in-the-loop process, where humans are asked to label the dataset. In this case, *dataset* is the theme, *human in the loop* is the subtheme and *labelling* is the code. Another example is that a model is evaluated on its performance and the performance can either be an evaluation metric (e.g. f1-score, confusion matrix, accuracy) or something more of human understanding (e.g. whether this was the expected outcome, do the outputs make sense, are they happy with the model and is it doing what it was intended to do). In this case, the theme is *model*, *performance* is the subtheme and the two codes within this subtheme are *metrics* and *human understanding*. It was also noticed that some entities (themes) had relationships between them. For example, some explanation methods (e.g., GradCam) were built to work with models such as Convolution Neural Networks [86] or some models perform better with certain dataset domains. The initial data model was designed to encompass the codes and relationships identified throughout the literature. To enhance readability and accommodate the numerous entities and attributes involved, we opted to represent the model using a conceptual ER diagram. This approach also facilitates understanding for users with varying levels of experience with databases.

3.2. Model Refinement using Real Datasets

After designing the initial data model based on the literature, we fitted four datasets to refine the model. The datasets we used were Birds [9], Imagenette [1], Adience [34] and Colored MNIST [69]. We chose datasets that contain information that covers a vast variety of literature, such as injected bias, protected attributes, and perturbations. These datasets also cover the requirements of concept-based explanations, ML fairness and robustness. We chose these four datasets as they cover all the required literature and requirements, which will be further described in the following subsections. We modelled the conceptual ER diagram into a Relational Database. The datasets were qualitatively analyzed to find entities and attributes that were missing from the initial data model. After fitting the datasets, we

[1] https://github.com/fastai/imagenette

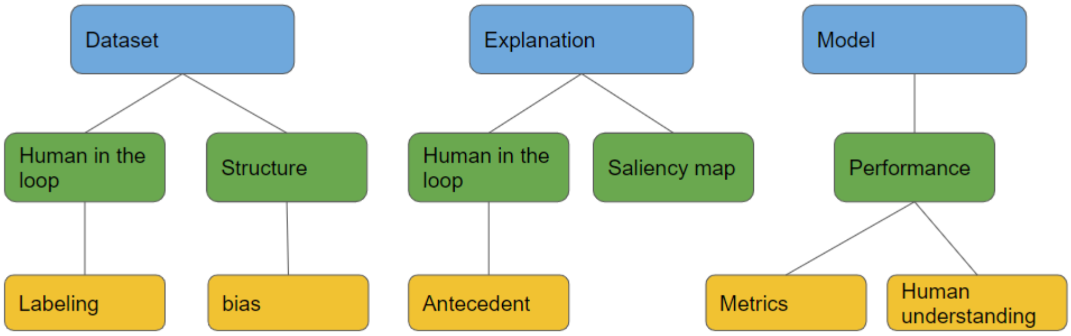


Figure 3.2: Themes and codes in literature identified using a thematic approach

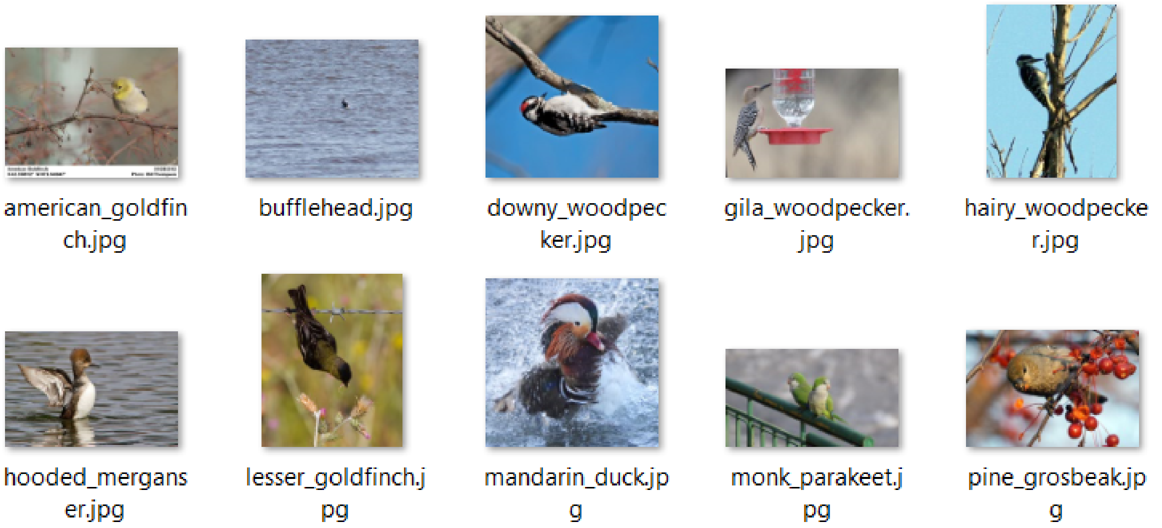


Figure 3.3: Sample images from the Birds dataset

found and included additional attributes such as color, pattern, clarity, and instances partially shown or not. We also realized issues with some attributes and relationships that were in the initial schema that had to be changed after we tried fitting these datasets to the model. Sections 3.2.1, 3.2.2, 3.2.3, 3.2.4 describe these datasets, why they were chosen and how information was extracted from them in detail.

3.2.1. Birds

Birds [9] is an artificially biased dataset of 10 bird classes: American Goldfinch, Lesser Goldfinch, Hairy Woodpecker, Hooded Merganser, Pine Grosbeak, Bufflehead, Downy Woodpecker, Monk Parakeet, Gila Woodpecker, and Mandarin Duck. Figure 3.3 shows a sample of each type of bird. This dataset was chosen because the images contain several concepts that cover the attributes required for explanations. Furthermore, it also covers the part of the literature about injected bias. The dataset was manually created by Balayn et al. [9] by sampling images with the injected bias. The dataset was utilized to fine-tune the Tensorflow model, InceptionV3 [99], which was subsequently employed to predict outcomes for 494 samples. The predictions were explained using a saliency explanation method, SmoothGrad [93]. SmoothGrad was chosen due to its sensitivity to model parameters while also minimizing noisy results [11]. The expected mechanisms were manually found by Balayn et al. [11] and instances and mechanisms were manually annotated by Balayn et al. [9], using the Amazon Mechanical Turk platform. These important entities (expected mechanisms, mechanisms and instances) that were identified in the literature are described further in Section 4.2.

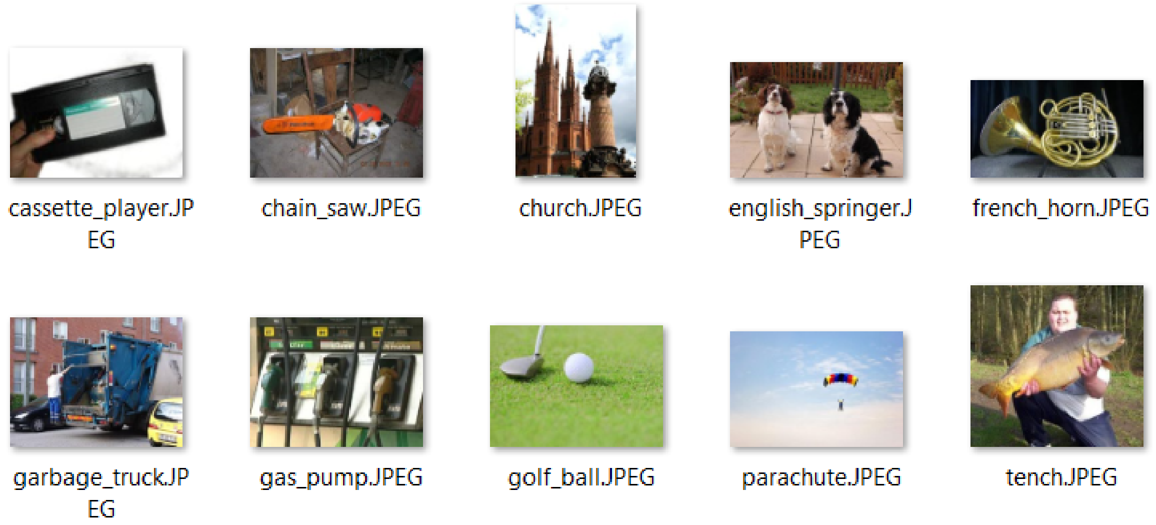


Figure 3.4: Sample images from the Imagenette dataset

3.2.2. Imagenette

Imagenette [2], a subset of ImageNet, is a widely used dataset. This dataset was used because of its generalizability and the images contain several concepts that can be used for explanability. Imagenette also consists of 10 classes: Parachute, Golf Ball, Chain Saw, Cassette Player, English Springer, Tench, Church, French Horn, Garbage Truck, and Gas Pump. Figure 3.4 shows a sample of each of the classes in the dataset. The dataset was used to finetune the InceptionV3 model [99] for a small number of epochs. The data was split into train, validation and test sets with approximately 9500, 3200, and 1200 samples respectively. LIME [83] was the chosen explainability method for this dataset and model combination. LIME was chosen due to its popularity within the explanation methods (13443 citations) and local fidelity. We manually found expected mechanisms (entity further explained in Section 4.2) for each of the classes (global expected mechanism) and a few of the images within each class (local expected mechanism). We manually annotated the mechanisms and used the Single-shot Multibox Detector (SDD) [70] to annotate the instances (entity further explained in Section 4.2). SSD was chosen since it was trained on a larger range of labels compared to the other object detection methods available. However, many of the images were incorrectly annotated, not annotated, or the annotations were very broad as shown in in Figure 3.5 respectively. For this reason, we manually annotated a couple of images per class to get a richer set of annotations.

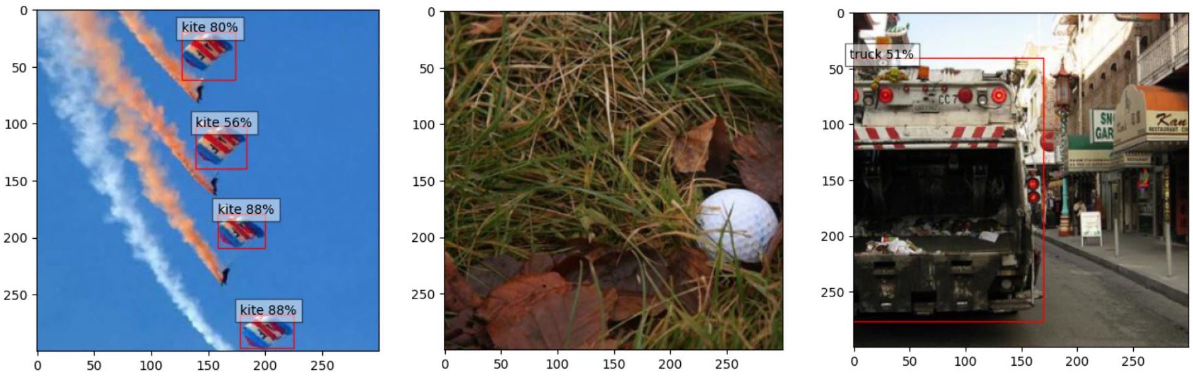


Figure 3.5: Annotations generated by Single-shot Multibox Detector

[2] https://github.com/fastai/imagenette

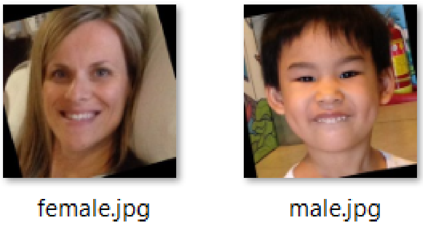


Figure 3.6: Sample images from the Adience dataset

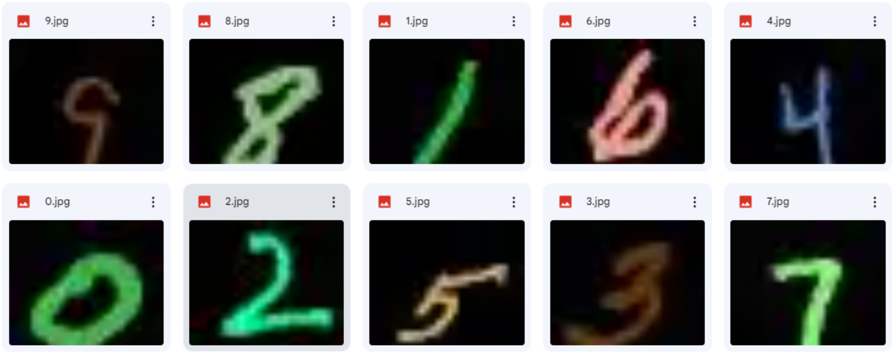


Figure 3.7: Sample images from the Colored MNIST dataset

3.2.3. Adience

Adience [34] is a multi-label dataset that contains samples of subjects' faces that can be grouped by age and gender. Since our model currently focuses on single-label classifications, we chose to simply use gender (male and female) as the target category. Figure 3.6 shows a sample of each of the classes in the dataset. Adience is a dataset that is typically used in works around ML fairness and also contains protected attributes (gender). The fundamental objective of this dataset revolves around capturing images under conditions closely resembling real-world scenarios, encompassing variations in appearance, pose, lighting conditions, and image quality, among others. Based on prior work [25], the dataset was used to finetune InceptionV3 [99]. The data was split into train, validation and test sets with about 1900, 500, and 700 samples respectively. LIME [83] was used to generate saliency maps for this dataset and model. This was the chosen explainability method due to its popularity (13443 citations). The global expected mechanisms for each class and local expected mechanisms for a few images per class were manually found and the local mechanisms and instances were manually annotated. The descriptions of these important entities can be found in Section 4.2.

3.2.4. Colored MNIST

Colored MNIST [69] is a modification of the MNIST dataset, where each number is colored in a specific color. Figure 3.7 shows a sample of each of the numbers in the dataset. This dataset was chosen for its natural perturbations that can be used to fulfil the requirement of robustness. Colored MNIST consists of 10 labels, the numbers 0 to 9. This dataset contains a comprehensive set of 15 corruptions that are applied to the MNIST dataset. The dataset was used to fine-tune the VGG19 model with 4000 training samples and 2000 validation samples. The model was then used to make predictions on 10,000 samples. Guided Backpropagation [95] was used to generate saliency maps for each of the predictions. This saliency detection approach was chosen due to its easy interpretability on numbers. The global expected mechanisms for each class and local expected mechanisms for a few images per class were manually found and the local mechanisms and instances were manually annotated. The descriptions of these important entities can be found in Section 4.2.

While fitting the datasets to the model, some gaps were found in the model as well as some lack of clarity. In addition to the attributes that were found after fitting the datasets, there were also some differences in the relationships and understanding of attributes. For example, when initially designing the dataset, a sample had one prediction, however, after fitting the datasets, it was found that it is more useful to change this relationship to a one-to-many, indicating that a sample has multiple predictions, indicated in Figure 3.8. A prediction of each of the labels in the dataset with the respective confidence so that the user can decide on how he/she would want to use the information, whether they prefer the highest predicted label, or in the case that it was incorrectly predicted, what was the confidence of the correct label, etc. Another example was the lack of understanding of the difference between protected attributes and injected bias, as the protected attributes in the dataset used by Joo and Kärkkäinen [58] were sunglasses and hats on all males. Based on this, we define protected attributes as attributes that are related to humans and injected bias as biases applied to non-human concepts.

After refining the model with these changes, the model was shown to other practitioners and researchers in the field of CV, Explainability, Fairness, and Robustness to further refine the model and make it complete. This process is described in the following section.

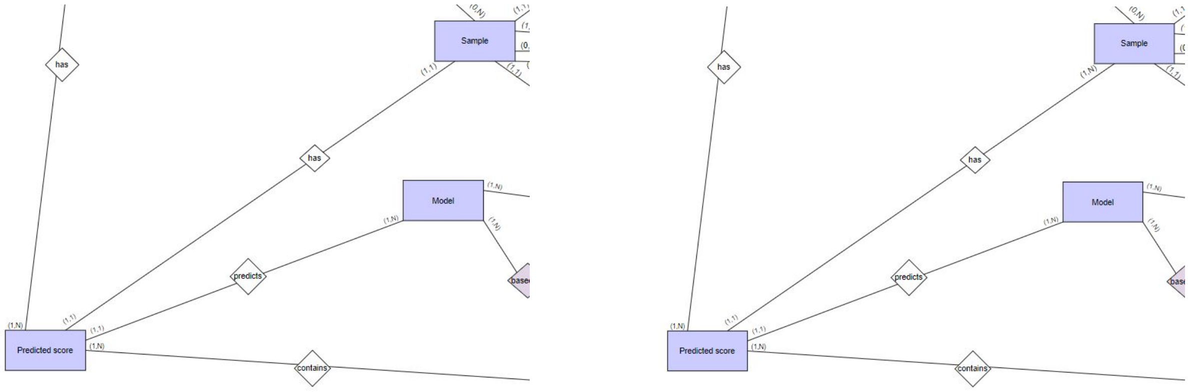


Figure 3.8: Cardinality between Sample and Prediction Before (left) and After (right) Fitting Datasets

3.3. Data model Evaluation

After designing the data model through literature and datasets, qualitative studies were held to evaluate the model for the two design principles: completeness and usability. Along with validating our model, the study also looked into researchers' workflows and discussed the advantages and limitations of the model. To our knowledge, there is not a lot of research focusing on the three non-functional requirements that address users' needs and challenges. Thus, this part of the research focuses on users and solving some of their problems with our data model.

3.3.1. Interview structure

To evaluate our design principles and complete the data model, we performed two rounds of semi-structured interviews. Before doing the interviews, we held pilot studies to refine the interview.

Interview 1 Through the first round of interviews, we hoped to discover participant's needs, obstacles, and challenges related to CV tasks, explanations, and datasets. We also discussed how they envision using the model and where can we make improvements, along with additional information to refine the model. Lastly, they stated the advantages and limitations of such a model. The interview structure with some example questions is shown below.

- **Getting to know the participant** This section focuses on the participant's work background, the domain that they specialize in, and where they have acquired the knowledge for CV and explainability and/or fairness. Example questions include:
  - *Where do you work, what is your job title, what does your job briefly consist of?*
  - *What do you do with XAI/ CV fairness?*
  - *Do you work on tasks of a specific domain? If so where have you acquired the required knowledge?*

- **Challenges faced when working on projects**
  This section focuses on asking the users about challenges they face at any step of their project. Challenges can lie anywhere from understanding the research to evaluating their methods. Some questions that were asked at this stage were:

  – *What are the main steps when working on these tasks? What are the objectives of each of these steps? What are some challenges faced or are still facing in any of these steps?*

  – *Are there any potential challenges in terms of concerns about transparency, reproducibility, comparability across different publications, usability, and feasibility?*

- **Challenges concerning datasets** This part of the interview focuses on the uses of datasets and some concerns facing the uses of datasets. This section consisted of questions such as:

  – *How do you use datasets in your tasks?*

  – *How do you evaluate the methods used? Are there any challenges faced during the evaluation? Are there any specific aspects that you look for when doing an evaluation?*

  – *What specific characteristics do you look for in a dataset?*

  – *Are there any constraints when choosing a dataset? Are there any limitations with the current datasets that you are using?*

- **Validating the model** The last section is about validating the model and evaluating the usability and completeness of the model. To aid the understanding of the model to the participants, they were first presented with a model that just showed the entities and relationships (Figure 3.9) and then the ER diagram that was designed based on literature (Figure 3.10). The questions that were asked in this stage were:

  – *What would you want to see in such a model? (before showing the model)*

  – *After seeing the model, is there anything else you would want to add to it?*

  – *Can you see yourself using this model for any of your use cases? In the case that you can see yourself using it, can you give an example of a task you would use it for? In the case that they say no, why would you not use it?*

Interview 2 The second round of interviews focused on the model usability - how the model can be used and the level of difficulty in using the model. The participants were asked to either populate the database using a dataset of their choice or query through the database containing the above-mentioned datasets. In the case that the participant chose to populate the database with their dataset, they were given information about the file structures that we used to populate the database, along with some example datasets. The participants were asked to collect the necessary information shown in the data model and store it according to the file structures provided, or any other format preferred. They were also given access to the code [3] so that they could populate the database with their dataset.

Participants who opted to extract information from the database were asked to familiarize themselves with the data model and information they would be interested in extracting based on their work. During the interview, they were asked to write queries to extract the necessary information.

Due to the amount of information required to populate the dataset, all participants chose to extract information from the database - *I looked into the information required for population, and the metadata on the datasets and model was doable, however, the mechanisms seemed like a lot of work and I think that's the selling point of this data model*. Throughout the interview, the participants were observed based on their level of difficulty in querying information and the type of information they extracted from the database. The participants were also asked questions such as how they expect to incorporate this data model in their everyday workflows and which of the current processes that they perform can be replaced by this model to make their work easier and less time-consuming.

[3] https://github.com/delftcrowd/CV_datamodel

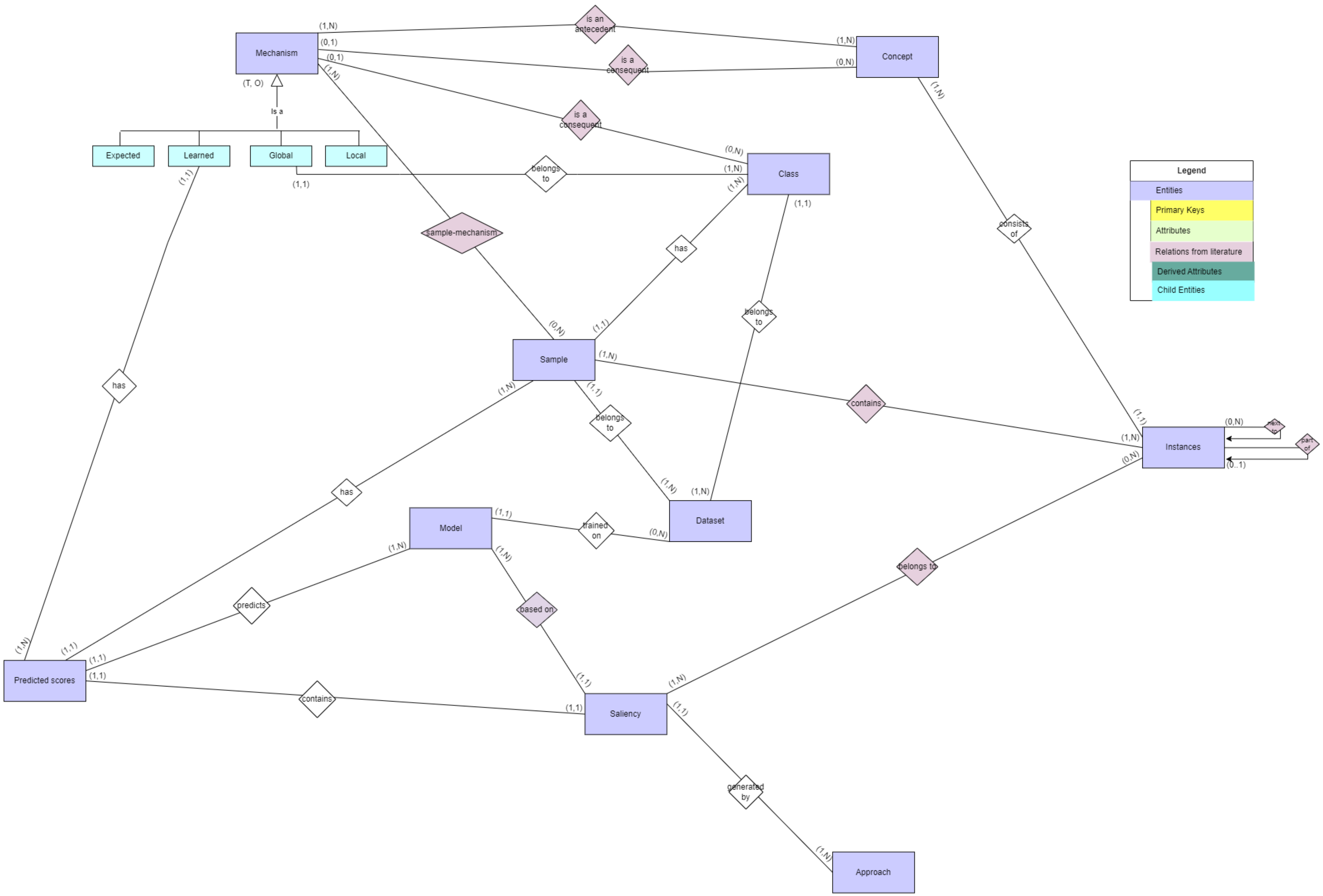


Figure 3.9: ER Diagram without attributes highlighting the entities and relationships

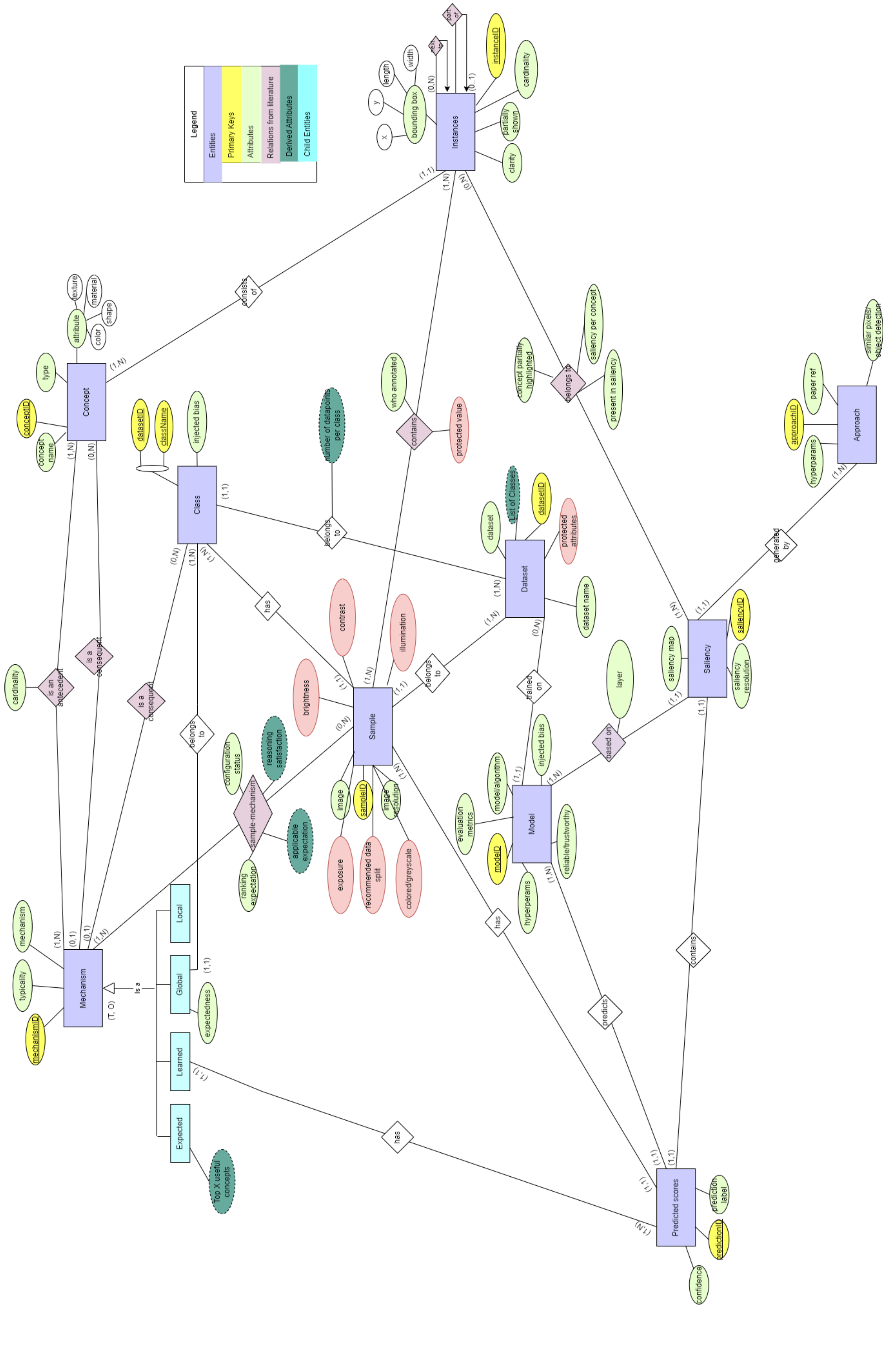


Figure 3.10: ER diagram presented designed based on literature

3.3.2. Participants and their recruitment

A total of 20 participants were interviewed in the first round of interviews, which we found sufficient to reach saturation. The participants were recruited via snowball and targeted sampling. Since this version of the model was designed for two stakeholder groups: researchers and people working in industry, we interviewed participants from both groups. Participants who have experience with CV, Explainability for CV tasks, and/or Fairness were selected for this study. The participants for this study were recruited using a combination of targeted sampling and snowballing techniques. Table 3.3 describes the participants. The first three participants took part in the pilot interviews to refine the interview structure and questions, while the others took part in the interviews that are mentioned above.

Two participants participated in the second round of interviews as these interviews required the participants to do some preparation. We interviewed one researcher and one practitioner. Participants were required to have some level of experience with databases and SQL. The participants were recruited using targeted sampling. Table 3.4 describes these participants and their proficiency with databases.

Prior to the interviews, we received approval from the ethics committee of TU Delft and all participants signed an informed consent form. The interviews lasted on average one hour.

ID	Job Title	Focus of Research	Experience	Domain of Expertise
P1	PhD Researcher focused on human-centered explainable AI and human-AI decision making	CV and XAI	3 years	Domain agnostic
P2	Student researching the adaptation of AI in the medical domain	CV and XAI	about 1 year	Healthcare domain: pulmonology department
P3	Machine Learning Consultant - building the bridge between academia and industry	CV and XAI	-	Healthcare domain: histopathology department
P4	AI research scientist - applying deep learning-based CV research onto multiple natural sciences and real-life application scenarios	CV and XAI	over 3 years	Domain agnostic
P5	Senior Machine Learning Engineer - finding gender bias in occupation images	CV and XAI	year	Domain agnostic
P6	PhD Researcher - Machine Learning and Databases	CV	1 year	Domain agnostic
P7	Assistant Professor - Theoretical topics of Machine Learning	CV and XAI	4 years	Domain agnostic
P8	PdH Researcher - Human-AI Collaboration in Video-based Design for Social Good	CV	-	Domain agnostic

ID	Job Title	Focus of Research	Experience	Domain of Expertise
P9	Assistant Professor and Researcher - Making machine learning models and neural networks trustworthy when applied to high-stakes settings in which human agents are involved	CV and XAI	5 years	Domain Agnostic
P10	Senior CV Scientist - Development of AI models	CV	10 years	Healthcare domain: medical imaging; Defense and Security; Aviation
P11	PhD Researcher - Contestability of algorithmic decision making processes	CV and XAI	1 year	Domain agnostic
P12	Assistant Professor - Human-centered computing approaches for trustworthy machine learning	CV, XAI, CV Fairness	XAI - 3 years; Fairness - 1-2 years	Domain agnostic
P13	Junior Software Engineer - R&D to develop and implement algorithms for digital signal processing	CV and XAI	1 year	Domain agnostic
P14	PhD Researcher - Explanability for NLP and CV tasks	CV and XAI	3 years	Domain agnostic
P15	Assistant Professor - Data Management and Human-Computer Interaction	CV, XAI, and CV fairness	1 project	Medical Domain
P16	Machine Learning Engineer - Train and deploy ML models into production	CV and XAI	1 project	Domain agnostic
P17	PhD Researcher - Explainability of AI applications	CV for Humans, XAI, CV Fairness, and Robustness	3 years	Finance Domain: anti-money laundering and life insurance plans
P18	PhD Researcher - Developing deep learning models for CV tasks	CV	3 years	Domain agnostic
P19	AI Engineer - Generate Synthetic Cancer Data	CV	3 years	Medical Domain - Cancer
P20	Research Scientist	CV and XAI	5-6 years	domain agnostic

Table 3.3: Overview on participants' daily work and background from the first round of interviews

ID	Job Title	Focus of Research	Proficiency with Databases
21	Machine Learning Engineer - Train and deploy ML models into production	CV and XAI	Intermediate
22	PhD Researcher - Security and Privacy	CV and Robustness	High

Table 3.4: Overview on participants' background of database proficiency from the second round of interviews

3.3.3. Transcription Analysis

Each of the anonymous interviews was transcribed and analysed using inductive and deductive coding. The transcripts were coded iteratively, arranging them into themes and codes. A total of 9 themes and 113 codes were found. Table 3.5 shows the themes and some of the codes within the themes that were obtained through the interviews. Besides coding the interviews, we also highlighted the information participants desired to see in such a data model. Based on these interviews, users' problems and suggestions for refining the schema were discussed based on how much they added to the model, while also keeping the model on the smaller side. These are summarized in Table 5.1.

Theme	Code
Annotation	confidence, user dependent, trust, expensive
Challenge	lack of knowledge, lack of resources, portray explainability, model complexity
Dataset	data structure, dataset size, quality, dataset access
Evaluation	biased evaluation, stakeholder dependent, fairness, inadequate evaluation
Experiment	collaboration
Explanation	clarity/presentation, tailored explanation, trust, definition
Model	combining models, model use, bias, debug
Problem	reproducibility, understanding AI, time-consuming, useful explanations
Schema	robust, easy, own datasets, preset structure

Table 3.5: Themes and codes obtained through the thematic analysis of the interviews.

4 Results: Data Model

In this section, we describe the data model obtained through our mixed-method approach. We discuss how the database was modelled and how it can be reproduced (Section 4.1), followed by a comprehensive description of the entities, attributes and relationships of the data model in Section 4.2. Finally, the content of the interviews is further analyzed in the User Study (Section 5).

4.1. Proposed Data Model

After conducting interviews with the participants, the data schema was once again iterated and modified to adapt to the participants' suggestions and needs (described in section 5). Figure 4.1 shows the final schema obtained from our mixed-method approach, which was described in Section 3. A description of the entities, attributes, and relationships can be found in Section 4.2. In this section, we explain how the database was modelled along with the necessary information to reproduce and populate the database with other datasets. We also explain some of the choices we made when designing the data model.

4.1.1. Implementation

The database was modelled as a Relational Database using Postgresql (Version 15). This was done using Python 3.8 and SqlAlchemy. Relational Database was chosen due to its tabular data structure which makes it easier to manage and understand relationships between entities. Separating entities into their own tables makes the design more modular, clearer to read, and adds flexibility to modifying the data schema. Furthermore, a relational database allows to retrieve complex data by using joins to combine multiple tables, which is useful in our data model, since users want to combine and compare models based on the datasets and the objects present in the samples [43]. Gyorödi, Gyorödi, and Sotoc [43] also found that the *select* operation is faster and more efficient in relational databases compared to non-relational databases. Since many of the potential uses suggested by the participants were about extracting information from the database, we decided to use a relational database.

When designing the data model, we tried to separate all entities into their own respective tables. Participants suggested this to add modularity and prevent too many empty columns. *Because some dataset owners or companies may not want to share everything about the dataset so there may be columns in there that are empty. (P1)* Figures 5.6 and 5.7 show parts of the ER-diagram before and after this suggestion (further explained in Section 5.1). We also chose to incorporate previous works in our data model, such as Datasheets [37], Model Cards [75], and Saliency Cards [19], rather than adding each characteristic separately to our data model. We chose to do this such that the database contains all the necessary information regarding datasets and models about explainability, fairness, and robustness, without overloading it. *You want to make a trade-off between what information you want to store in the database because there's a lot that can be stored. (P1)*

Tables 4.2 and 4.3 show the data type and source for each attribute belonging to the entities and relationships shown in the data model (Figure 4.1). The relationships table also includes the cardinalities of the relationships. For example, the relationship consists of (Instance ↔ Concept) with a cardinality of $(1,1) \leftrightarrow (1,N)$ reads as an instance is a representation of one and only one concept and a concept

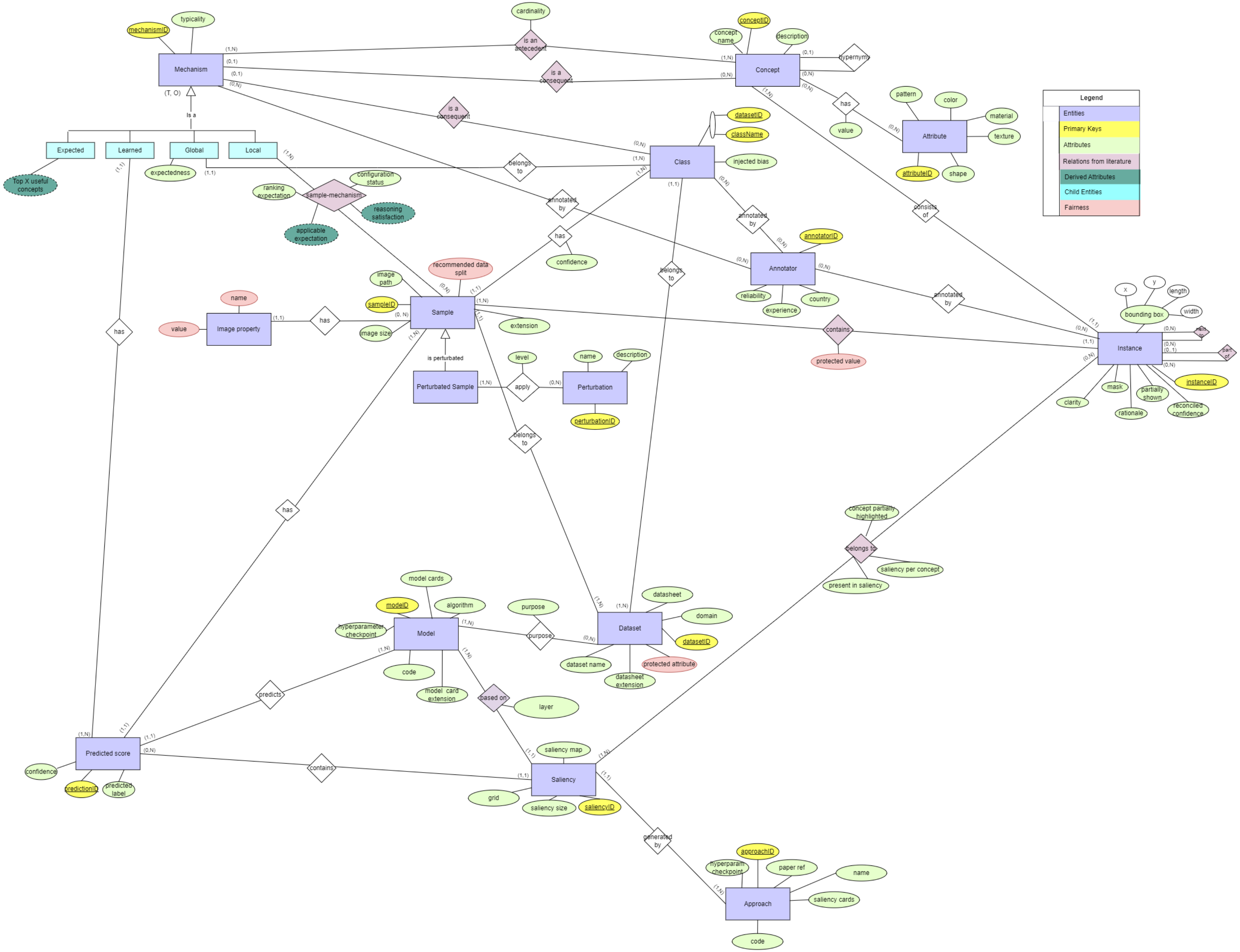


Figure 4.1: Final and Modified Data Schema obtained through the mixed-method approach

Dataset	Speed	Storage
Adience	20 mins	1.46MB
Birds	40 mins	12.6 MB
Colored MNIST	2h	8.42MB
Imagenette	1h	6.89MB

Table 4.1: Amount of time and space occupied by each of the datasets in the database

consists of at least one instance. Figure 4.2 shows the physical data model. The entities and respective attributes are in the orange tables, while the green tables display the relationships (if it’s a many-to-many relationship or the relationship contains attributes). The source code for the implementation can be found at Github [1].

The database can be created by running the Python script ‘initialize.py‘. This creates an empty relational schema containing all the tables and relationships described in tables 4.2 and 4.3. Examples of the required file structures can be found within the directories ‘labeled data_dataset name‘. The repository also includes dedicated Python scripts to populate the database with the four datasets. The datasets chosen cover different aspects of the data model, for example, injected bias, protected attributes, and more. The script ‘population_template.py‘ contains the core code to load the datasets into the database and the scripts ‘populate_dataset name.py‘ are used to load each dataset into the database.

The code has been designed with modularity in mind, enabling its utilization with various datasets, provided that the file structures containing the data are adhered to. The repository includes scripts (found in the directory preprocessing) that were used to structure the data into these respective file structures. These can be adapted depending on how the information is currently stored. However, if users want to use their own file structures, that is also possible by adapting the ‘population_template.py‘ script to accommodate the new file structures.

Furthermore, the repository incorporates scripts utilized for fine-tuning CV models on the aforementioned datasets. These can be found in the ‘models‘ directory.

The database was run with 16GB RAM. To optimize the storage and speed of queries, the paths to the images, saliency maps, and models were stored in the database rather than the images and models themselves. Furthermore, it was noticed that populating the predictions was relatively slow compared to the other entities in the model. Table 4.1 illustrates the data size of each of the datasets within the database along with the amount of time taken to load information for each of the datasets into the database.

Entity	Attribute	Data Type	Source
Dataset	Dataset_id	UUID	Authors
	Datasheet	String	[37]
	Datasheet Extension	String	Authors
	Domain	String	P6, P8, P10, P14
	Protected Attributes	String	[25]
Concept	Concept_id	UUID	Authors
	Concept name	String	[39]
	Description	String	[74]
Attribute	Attribute_id	UUID	Authors
	Type	Enum	[120]
Sample	Sample_id	UUID	Authors
	Image path	String	Authors
	Image size	String	[39]
	Recommended split	String	[25]
	Extension	String	P3
Image property	Image_property_id	UUID	Authors
	Property	String	[25], [108]
	Value	String	[25], [108]

[1] https://github.com/delftcrowd/CV_datamodel

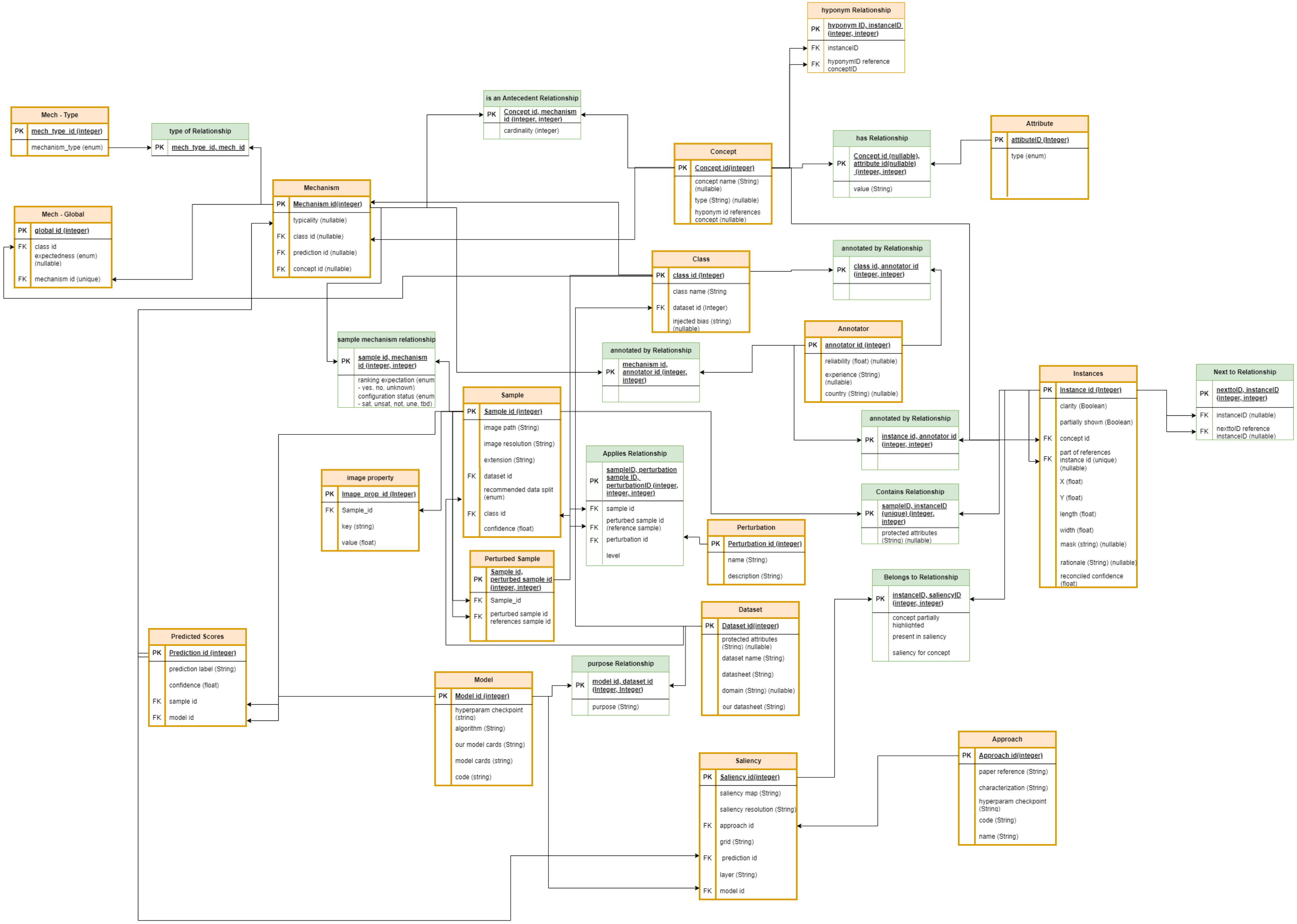
Entity	Attribute	Data Type	Source
Perturbation	Perturbation_id	UUID	Authors
	Name	String	P10, P15
	Description	String	P10, P15
Instance	Instance_id	UUID	Authors
	Mask	String	P18, P19
	bbX	Float	Authors
	bbY	Float	Authors
	bbWidth	Float	Authors
	bbLength	Float	Authors
	Clarity	Boolean	Existing dataset inspection
	Partially Shown	Boolean	Existing dataset inspection
	Rationale	String	P10
	Reconciled confidence	Float	P3, P15
Class	Class_id	UUID	Authors
	Class name	String	Authors
	Dataset_id	UUID	Authors
	Injected bias	String	[11], [88]
Annotator	Annotator_id	UUID	Authors
	Reliability	Float	P11, P15
	Experience	String	P11, P15
	Country	String	P16, P11, P15
Saliency	Saliency_id	UUID	Authors
	Saliency map	String	Authors
	Saliency Size	String	Existing dataset inspection
	Grid	String	P18
Model	Model_id	UUID	Authors
	Algorithm	String	[83]
	Hyperparameter checkpoint	String	Authors
	Model cards	String	[75]
	Model cards extension	String	Authors
	Code	String	P18, P19
Approach	Approach_id	UUID	Authors
	Name	String	Authors
	Hyperparameter checkpoint	String	Authors
	Paper reference	String	Authors
	Saliency cards	String	P12, P17
	Code	String	P19
Predicted Score	Prediction_id	UUID	Authors
	Prediction label	String	Authors
	Confidence	Float	Authors
Mechanism	Mechanism_id	UUID	Authors
	Typicality	Float	[88]
	Type	Enum	[7]
Global	Expectedness	Enum	[7]

Table 4.2: Data dictionary of the entities and attributes in the data schema

Relationship	Attribute	Data Type	Cardinality	Source
next to (Instance ↔Instance)	-	-	(0,N) ↔(0,N)	[88]

Relationship	Attribute	Data Type	Cardinality	Source
part of (Instance ↔Instance)	-	-	(0,1) ↔(0,N)	[88]
is Consequent (Mechanism ↔Concept)	-	-	(0,1) ↔(0,N)	[7]
is Consequent (Mechanism ↔Class)	-	-	(0,1) ↔(0,N)	[7]
sample mechanism (Mechanism ↔Sample)	ranking expectation	Enum	(1,N) ↔(0,N)	[7]
	configuration status	Enum		[7]
is Antecedent (Mechanism ↔Concept)	cardinality	Integer	(1,N) ↔(1,N)	[7]
annotated by (Mechanism ↔Annotator)	-	-	(0,N) ↔(0,N)	Authors
has (Learnt Mechanism ↔Predicted Score)	-	-	(1,1) ↔(1,N)	Authors
belongs to (Global Mechanism ↔Class)	-	-	(1,1) ↔(1,N)	[7]
contains (Sample ↔Instance)	protected value	Boolean	(1,N) ↔(1,1)	[25]
annotated by (Annotator ↔Instance)	-	-	(0,N) ↔(0,N)	Authors
belongs to (Saliency ↔Instance)	present in saliency	Boolean	(1,N) ↔(0,N)	Authors
	concept partially highlighted	Boolean		[61]
	saliency per concept	String		[120]
consists of (Instance ↔Concept)	-	-	(1,1) ↔(1,N)	Authors
generated by (Saliency ↔Approach)	-	-	(1,1) ↔(1,N)	Authors
annotated by (Annotator ↔Class)	-	-	(0,N) ↔(0,N)	Authors
has (Sample ↔Predicted Score)	-	-	(1,N) ↔(1,1)	Authors
predicts (Model ↔Predicted Score)	-	-	(1,N) ↔(1,1)	Authors
contains (Predicted Score ↔Saliency)	-	-	(0,N) ↔(1,1)	Authors
based on (Saliency ↔Model)	layer	String	(1,1) ↔(1,N)	[117]
purpose (Model ↔Dataset)	purpose	String	(1,N) ↔(0,N)	P6, P13
belongs to (Sample ↔Dataset)	-	-	(1,1) ↔(1,N)	Authors
has (Sample ↔Image Property)	-	-	(0,N) ↔(1,1)	Authors
belongs to (Class ↔Dataset)	-	-	(1,1) ↔(1,N)	Authors
apply (Perturbated Sample ↔Perturbation)	level	String	(1,N) ↔(0,N)	P14
has (Sample ↔Class)	confidence	Float	(1,1) ↔(1,N)	P9, P13
has (Attribute ↔Concept)	value	String	(0,N) ↔(0,N)	Authors
hyponym (Concept ↔Concept)	-	-	(1,N) ↔(0,N)	P12, P14

Table 4.3: Data dictionary of the relationships and their attributes (if any)



hyponym Relationship
PK hyponym ID, instanceID (integer, integer)
FK instanceID
FK hyponymID reference conceptID
Mech - Type
PK mech_type_id (integer)
mechanism_type (enum)
type of Relationship
PK mech_type_id, mech_id
is an Antecedent Relationship
PK Concept id, mechanism id (integer, integer)
cardinality (integer)
Concept
PK Concept id(integer)
concept name (String) (nullable)
type (String) (nullable)
hyponym id references concept (nullable)
has Relationship
PK Concept id (nullable), attribute id(nullable) (integer, integer)
value (String)
Attribute
PK attibuteID (Integer)
type (enum)
Mech - Global
PK global id (integer)
FK class id
expectedness (enum) (nullable)
FK mechanism id (unique)
Mechanism
PK Mechanism id(integer)
typicality (nullable)
FK class id (nullable)
FK prediction id (nullable)
FK concept id (nullable)
Class
PK class id (Integer)
class name (String
FK dataset id (Integer)
injected bias (string) (nullable)
annotated by Relationship
PK class id, annotator id (integer, integer)
sample mechanism relationship
PK sample id, mechanism id (integer, integer)
ranking expectation (enum - yes, no, unknown)
configuration status (enum - sat, unsat, not, une, tbd)
annotated by Relationship
PK mechanism id, annotator id (integer, integer)
Annotator
PK annotator id (integer)
reliability (float) (nullable)
experience (String) (nullable)
country (String) (nullable)
Instances
PK Instance id (Integer)
clarity (Boolean)
partially shown (Boolean)
FK concept id
FK part of references instance id (unique) (nullable)
X (float)
Y (float)
length (float)
width (float)
mask (string) (nullable)
rationale (String) (nullable)
reconciled confidence (float)
Next to Relationship
PK nexttoID, instanceID (integer, integer)
FK instanceID (nullable)
FK nexttoID reference instanceID (nullable)
Sample
PK Sample id (integer)
image path (String)
image resolution (String)
extension (String)
FK dataset id
recommended data split (enum)
FK class id
confidence (float)
annotated by Relationship
PK instance id, annotator id (integer, integer)
image property
PK Image_prop_id (Integer)
FK Sample_id
key (string)
value (float)
Applies Relationship
PK sampleID, perturbation sample ID, perturbationID (integer, integer, integer)
FK sample id
FK perturbed sample id (reference sample)
FK perturbation id
level
Contains Relationship
PK sampleID, instanceID (unique) (integer, integer)
protected attributes (String) (nullable)
Perturbation
PK Perturbation id (integer)
name (String)
description (String)
Perturbed Sample
PK Sample id, perturbed sample id (integer, integer)
FK Sample_id
FK perturbed sample id references sample id
Belongs to Relationship
PK instanceID, saliencyID (integer, integer)
concept partially highlighted
present in saliency
saliency for concept
Dataset
PK Dataset id(integer)
protected attributes (String) (nullable)
dataset name (String)
datasheet (String)
domain (String) (nullable)
our datasheet (String)
Predicted Scores
PK Prediction id (integer)
prediction label (String)
confidence (float)
FK sample id
FK model id
Model
PK Model id (integer)
hyperparam checkpoint (string)
algorithm (String)
our model cards (String)
model cards (string)
code (string)
purpose Relationship
PK model id, dataset id (Integer, Integer)
purpose (String)
Approach
PK Approach id(integer)
paper reference (String)
characterization (String)
hyperparam checkpoint (String)
code (String)
name (String)
Saliency
PK Saliency id(integer)
saliency map (String)
saliency resolution (String)
FK approach id
grid (String)
FK prediction id
layer (String)
FK model id

Figure 4.2: Logical Data Model

4.2. ER Descriptions

This section provides an in-depth explanation of the data model. Section 4.2.1 presents an overview of the entities and attributes within the schema. Subsequently, Section 4.2.2 delves into examples of attributes that can be derived from the model, followed by Section 4.2.3 which discusses the relationships present in the schema. Furthermore, Section 4.2.4 highlights the integration of datasheets, model cards, and saliency cards with the schema, enhancing transparency and facilitating comprehensive analysis.

4.2.1. Entities and Attributes

This section provides a comprehensive overview of the entities and attributes present in the schema, offering a detailed description of each. We also include quotes from participants for entities and attributes they thought were necessary for such a schema or would complete it. Table 5.1 summarizes these suggestions - users' suggestions for elements that they thought would be necessary for the data model and were already a part of it, and users' suggestions for elements that would complete the schema.

- **Dataset**: the quality and underlying characteristics of training datasets are fundamental to the system's performance. If there are any biases included, the system will reflect on these patterns. If the dataset is not representative of the target population, then certain groups are held at a disadvantage. Industry practitioners are aware of the importance of training datasets, which is why they often revisit datasets when they notice problems with the system. Therefore, understanding the data and knowing what it contains is necessary to create more fair and robust models [4].

  – *Dataset name*: The name of the dataset that is being described.

  – *Datasheet*: Reference (URI) to the datasheet. The datasheet serves as a valuable resource for obtaining further information about the dataset. Datasheets are further described in Section 4.2.4 - Datasheets and [37]. While our data model already contained some information about the dataset, many participants (P2, P3, P6, P9, P10, P11, P14, P15, and P19) found that adding additional information about the dataset (e.g. when and how the data was collected, the authors of the dataset, license, how the dataset was labelled) will add more transparency and will allow users to make more knowledgeable decisions based on this additional information. It also gives users enough information to debug a system so that they know where the network could fail. *I would put a link to the GitHub page or something where I can look for information on the actual dataset and also the author so I know at least who and if there's any paper describing it. (P14)*

  – *Datasheet extension*: Reference (URI) to the extended datasheet. The extended datasheet contains additional information that was requested by participants that was not initially proposed in the datasheets described by [37]. The questions contained in the extended version are described in Section 4.2.4 - Extended Datasheet.

  – *Domain*: This contains the domain of the dataset, such as Animals or Histopathology. This attribute enables users to easily locate datasets that align with their specific domain of interest. This addition was made based on the feedback from participants P14 and P10, who believed that it could improve the data model's efficiency, particularly in terms of dataset search and finding similar datasets within the database. *It could be very useful that you can quickly based on certain entity search terms find your data set and what would be convenient if there is a database like this to quite quickly find similar data sets with the search terms. So maybe a list of tags that describe the dataset. (P10)*

  – *Protected attributes*: This describes the sensitive data that may be present in a dataset. Including this data allows users to make more informed choices about fairness. In the study conducted by [25] on the IJB-A and Adience datasets, it was observed that the performance of models was notably higher for males and individuals with lighter skin tones. Upon deeper analysis, it was discovered that the representation of darker-skinned females in the IJB-A dataset was limited (4.4%), while darker-skinned males were underrepresented in the Adience dataset (6.4%). Thus bringing attention to the protected attributes can be the stepping stone to building fair, transparent, and accountable models.

- **Concept**: Entities, also known as semantic concepts, that can be used to identify a class and make up the expected mechanisms [39], [61]. Mechanisms are further explained in this section.
  - *Concept Name*: This entity contains the name of a semantic concept. However, the concept name is not just restrained to the name of the concept, as some concepts may be difficult to describe in words, such as patches or patterns [39].
  - *Description*: This gives a more fine-grained label to a concept, for example, a table can be categorized as a dining table, a computer table, or a dressing table [74].
- **Attribute**: These are the finer-grained entities that describe the concept. Attributes can either be used to further describe a concept or in some cases, a concept is simply the attributes. The model may not have learnt the entire concept, but rather just an attribute. For example, in the Birds dataset, the model kept learning the colors blue and green rather than concepts belonging to a monk parakeet bird.
  - *Type*: Attributes that describe concepts can be texture [38], material [120], shape, color [120], or pattern [39]. One widely accepted intuition on how CNNs can reach such impressive levels of performance is that they combine low-level features (e.g. colors, patterns, shapes) to increasingly complex shapes, such as wheels, car windows until the object (e.g. car) can be classified [71], [64]. Initially, each of these attributes had its own column, however, after multiple participants pointed out that there can be many more attributes that describe concepts, we decided to make ‘attribute‘ a separate entity where users can add the type of attribute and its value.
- **Sample**: This entity contains information about a single sample image from a dataset.
  - *Image path*: This points to the sample image in the dataset. Participants P3 and P10 said that knowing the image path can reduce one of the current limitations as having images in different folders, with no descriptions of where these images reside makes the task very cumbersome and time-consuming. *The data that I got was really messy. It was just that the folder structure was all broken up, the images were from different labs so you had different folders for the different labs all containing the images. (P3)*
  - *Image size*: This contains the resolution of the image. Sometimes the saliency maps are not the same size as the images; therefore having the resolution of the image makes it easier to map the highlighted area in the saliency map to the image. Additionally, [39] used varying resolutions to capture concepts from simple fine-grained ones such as textures and colors to more complex and coarse-grained concepts such as parts and objects. The image resolution can determine the level of intricacy of the annotations and explanations can be. This was also an important factor to participants P10, P19, and P14, as the image resolution can serve as a constraint for using the image, or users prefer all images in the dataset to be of the same or similar resolution. Some CV models also have a fixed input size, thus knowing the image resolution can help users pick the images that are suitable for the model.
  - *Recommended Split*: This describes which data split the sample belongs to (train, validation, or test set). This allows users to further analyze their model and check class balances [25]. This was also confirmed by participants P5, P17 and P20. Knowing the data split allows users to compare the training data to the testing data to see whether the model can correctly predict data that is not the exact representation of the training data. Users can also find the groups of data that are under-represented in the training set therefore identifying bias that may be present in the model.
  - *Extension*: The extension of the sample image. In some domains, images do not have common extensions and additional software is necessary to access these images. Thus including this attribute allows users to determine whether they can use this dataset based on the software needed before downloading the entire dataset. *Every lab could have its own type of like image extensions, maybe you need special programs to open different images. (P3)*
- **Image Property**: This describes the properties of the images. Researchers found that knowing these values for a sample can lead to more fair and robust CV models [108]. For example, a model may classify images based on the brightness or colors of the image, rather than concepts within the image

[108]. Participant P3 also found this bias when working on a project. *This image contains cancer and this other image also contains cancer. But maybe just the varying degrees of brightness make those two images slightly different. The model is biased towards one and not towards the other, so the model would say that this one image does not have cancer and this one image does have cancer, but the human eye can catch this very easily because we are not led astray by these noise factors. (P3)* Thus knowing these properties helps users determine why the model was biased and can thus create more fair and robust systems.

  - *Property*: This describes the properties of an image, such as the image brightness, contrast, illumination, color space, exposure, etc. Exploring and leveraging variations in these properties contributes to the development of more robust systems [108], [25], [78].
  - *Value*: This stores the value of the image property.

- **Perturbation**: The evaluation of models using corrupted images contributes to the development of robust systems [50]. Many times, the model is evaluated on test data that is drawn from the same distribution as the training data and performs poorly on out-of-distribution data. The models lack robustness to small changes in the data, such as input translations, adversarial perturbations, and image corruptions. Including perturbated images in the dataset is an essential step for deploying models in complex, real-world settings where the test data does not perfectly match the training distribution [77]. This suggestion was made by participant P14 as it gives users clarity on what the model really understands. *Even if the picture is quite weird, it would allow you still to understand whether the network really understood what a cat is, even if there would be more multiple features. (P14)* Thus it will allow users to understand what the model really learnt.

  - *Name*: The name of the perturbation (e.g. rotation, blur, brightness, gaussian noise)
  - *Description*: A description of the perturbation added to a given image.

- **Instance**: This entity represents the instances of concepts that are contained in a sample image.

  - *Mask*: A mask around an instance. This can be useful, especially for objects that are not rectangular in shape and users want to get the exact outline of the instance instead. The incorporation of a mask proved to be a significant attribute for participant P19, as it facilitates the curation of a dataset through the utilization of synthetic object generation. *Because you wouldn't be interested in the background if you wanted to generate like full-bodied people, you would just be interested in the outline of the person and then trying to make a square box around them, remove them and then pass it with like a neutral background is difficult. (P19)*
  - *Bounding Box*: The coordinates of the instance's bounding box are stored here. Bounding boxes play a crucial role in enabling users to identify instances within an image and annotate them efficiently.
    - ◇ bbX: the starting 'x' coordinate for the bounding box around a given instance. The reference system is centered in the top-left corner of the image.
    - ◇ bbY: the starting 'y' coordinate for the bounding box around a given instance. The reference system is centered in the top-left corner of the image.
    - ◇ bbWidth: the width of the bounding box around a given instance.
    - ◇ bbHeight: the height of the bounding box around a given instance.
  - *Clarity*: Sometimes instances are present in an image but may be unclear which makes it difficult for a model to recognize. This is important because it shows how much a model can recognize an object, and what level of granularity it requires the concepts to be to recognize them. Figure 4.4 shows an example of a shark that is not clearly visible.
  - *Partially Shown*: Sometimes the entire concept is not shown in an image. This may make it difficult for a model to identify the partially shown concept. Figure 4.5 shows part of the shark's body. However, without any knowledge of what the dataset is about, neither humans nor models may recognize that it is a shark.

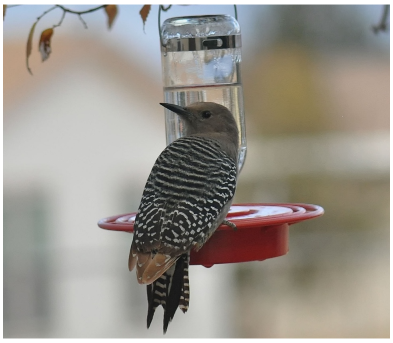


Figure 4.3: Rationale - Is the perch partially obstructed?

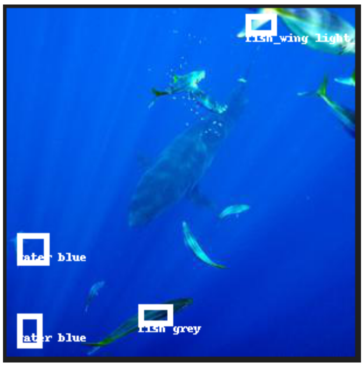


Figure 4.4: Clarity of concepts in images

Figure 4.5: Partially shown concepts in images

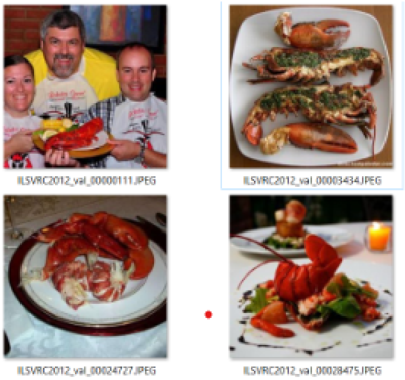

ILSVRC2012_val_00000111.JPEG
ILSVRC2012_val_00003434.JPEG
ILSVRC2012_val_00024727.JPEG
ILSVRC2012_val_00028475.JPEG

Figure 4.6: Bias within a dataset

  - *Rationale*: A rationale behind how an annotation was made. For example, some annotators may consider one obstructed pixel to be a partially shown object while another may consider the object to be fully shown. Figure 4.3 shows a bird sitting on a perch. Some may argue that the perch is partially obstructed, while others may argue that one can still see that it is a perch and thus will say that it is fully visible. The threshold at which an object is considered to be partially shown was a concern raised by Participant P10.
  - *Reconciled confidence*: The confidence around an instance's annotation. This depends on the annotator's reliability and experience. It is assumed that the reconciled confidence is calculated using majority voting, but users can choose to use other aggregation methods as well. Annotations are very subjective and many objects can have multiple names (synonyms or words of different granularities). Thus multiple annotations for one object are preferred for better quality annotations (quality control). This was suggested by participants P3 and P15 as having multiple crowd workers or experts annotate a sample can lead to interesting inter-annotator agreements.

- **Class**: The ground truth label of a given sample.
  - *Class name*: This stores the name of the ground truth label.
  - *Dataset ID*: This shows which dataset the class belongs to.
  - *Injected Bias*: This includes a description of artificially injected bias, aiming to examine how models respond to such biases. For instance, in the Fish dataset, a majority of the images featuring lobsters (Figure 4.6) also include a plate. The presence of bias in a dataset allows for assessing whether a model can effectively learn and account for such biases [11], [88].

- **Annotator**: This stores information about the annotators of an instance, class, or mechanism. This information gives users further insight into how much they can rely on the annotations. Participants P11 and P15 found this to be an important factor because annotations can be quite open and subjective. *So for instance, in a western country, the table that you would have in a dining room has certain characteristics which might not be reproducible or are not the same in Asian countries. (P11)*

  - *Reliability*: The reliability of a human annotator. This can be provided by, for example, a crowdsourcing platform. The reliability can be based on how well the annotator performed previous tasks.
  - *Experience*: The amount of experience the human annotator has in a particular domain.
  - *Country*: The country of residence of the human annotators provides insights into their cultural background, which can be significant in understanding the subjective nature of annotation.

- **Saliency**: This stores information about the saliency map of an image.
  - *Saliency map*: Reference (URI) to where the saliency is stored. The saliency map highlights the parts of the image that the model used to make its predictions.
  - *Saliency size*: As previously mentioned, the resolution of the image and saliency may differ. Thus including the resolution makes it easier to map the highlighted concepts from the saliency map to the image.
  - *Grid*: The saliency map is displayed using a specific color scheme, with a grid overlay that visually represents the level of importance for each pixel. This aids users in visualizing the salient regions of the image [101]. *Something you need to provide the color so we know which pixels have more importance. (P18)*

- **Model**: This stores information about the CV model used.
  - *Algorithm*: This shows which model and algorithm were used to classify the images. Different algorithms use different concepts to identify images, thus knowing which model was used is an important factor [83].
  - *Hyperparameter checkpoint*: A reference (URI) to where the checkpoint is stored. Every model has several hyperparameters. The hyperparameter values influence the model's behaviour, making this a useful attribute for replication [83].
  - *Model cards*: Reference (URI) to the model card. The model card serves as a valuable resource for obtaining more information about the model and its purposes. Model cards are further described in Section 4.2.4 - Model Cards and [75]. While our data model already contained some information about the CV model used, some participants (P1, P6, P18) suggested adding some additional information such as the version of the model, and use cases, for more transparency and reproducibility.
  - *Model cards extension*: Reference (URI) to the extended model card. The extended model card contains additional information that was not initially proposed in the model cards described by [75]. The questions contained in the extended version are described in Section 4.2.4 - Extended Model Cards.
  - *Code*: A reference (URI) is provided to access the code. This allows users to conveniently access the code used for the model, which can save time and facilitate replication of the study. This was suggested by participants P18 and P19 because some models have complex code which makes it difficult and time-consuming to reproduce. *I would also add the GitHub there if there is good code. If there is a Keras or pytorch implementation I would also put it there because sometimes it is difficult to understand how the model was generated. (P19)*

- **Approach**: This refers to the explanation approach that was used to explain the model.
  - *Name*: The name of the explainability approach used.
  - *Hyperparameter checkpoint*: A reference (URI) to where the hyperparameter is stored. This allows users to easily replicate the approach that was used to explain a model.
  - *Paper reference*: Reference (URI) to the scientific publication that introduced the given explanability method.
  - *Saliency cards*: A reference (URI) to the saliency card. Saliency cards serve as a valuable resource for obtaining more information about the saliency approach used. Saliency cards are further explained in Section 4.2.4 - Saliency cards and [19]. This was suggested by participants

P12 and P17 because saliency methods consist of several characteristics and having this information allows users to pick a saliency method in a more informed manner rather than simply choosing the widely known ones. *I think on the characterization of saliency maps, there is probably a lot more to say like on the evaluation and the comparisons of how well they do. You could also compare saliency maps depending on how well they do for different tasks. (P17)*

  - *Code*: A reference (URI) is provided to access the code. This allows the user to easily access the code for this explanability method, which can save them time when replicating the study.

- **Predicted Score**: This shows the predictions a model made for a given sample.

  - *Prediction label*: The class that the model could have assigned for a given image.
  - *Confidence*: The confidence of a CV model in predicting the class for a given image. This can be the confidence for the final predicted label or other potential labels.

- **Mechanism**: A set of general rules used to describe a class. Mechanisms are used to understand the relationship between concepts to identify a consequent (class or concept). For example, if a garden is associated with plants, the model should not classify everything that contains plants as a garden. It should be able to look at a combination of concepts before making a prediction [7]. Throughout the interviews, several participants (P5, P9, P12, P14) mentioned that they would like to know what the model has learnt and what they expect the model to learn in human-understandable terms. *We really needed to have better ways of well, first of all, explaining what exactly the model has learned in human concepts. (P12)*; *And while sometimes you do have tags, for instance, associated with images, you don't have annotations about what parts of the image make the annotator think that that that's actually an image of Bob Kennedy. (P9)*

  - *Typicality*: When a global expected mechanism is mapped to a sample, there is uncertainty as the entire mechanism may not be relevant to that sample. For example, if an image of a bedroom does not contain a bed, then we cannot map the mechanism <bed AND nightstand -> bedroom>. Therefore, the typicality score tells one the certainty of each mechanism across samples [88].
  - *Type*: Mechanisms can be at a global level, which can be a set of concepts present at a class level, or they can be at a local level, which is a set of concepts that are present in a certain sample. We also store mechanisms that we expect the model to learn and mechanisms that the model has learnt. Thus the type can be a combination of these, which leads to four types of mechanisms: global learnt mechanisms, global expected mechanisms, local learnt mechanisms, and local expected mechanisms.
    - ⋄ *Local*: A local mechanism consists of the set of concepts (rules) present in a particular sample.
    - ⋄ *Global*: A global mechanism consists of a set of concepts (rules) that are generally present in a class. Global expected mechanisms are directly associated with the consequent in general rather than the consequent for that dataset, meaning that injected bias is not included in the global mechanisms.
      - · Expectedness: This determines whether the mechanism is expected, unexpected, or whether there may be a possibility of seeing this mechanism across all samples.
    - ⋄ *Expected*: Expected mechanisms are a set of concepts (rules) that are expected to be present in a sample image or consequent. These can be used as ground truth explanations for a dataset (P1).
    - ⋄ *Learned*: The learned mechanisms are a set of concepts (rules) that a model is able to learn from a sample image or consequent.

4.2.2. Derived Attributes

The following subsection provides insights into some examples of derived attributes that can be obtained from the existing attributes and relationships within the data model. It also presents some example queries demonstrating the utilization of these attributes.

- **List of Classes**: The list of classes belonging to a dataset can be derived using queries. This is especially helpful for users to see the level of granularity within the dataset. For example, a dataset containing different types of birds is more fine-grained than a dataset containing different types of animals. Users can then decide the level of granularity they want concepts to use learnt [39].
  Example query: SELECT class_name FROM class WHERE dataset_id = <your_dataset_id>;

- **Number of Datapoints per Class**: Developers have an overview of the dataset they are using to train models, but the end users of the system are not always aware of the attributes of the dataset. One important attribute is the number of data points per class. Many times, datasets have minority classes that only make up a small percentage of the entire dataset. Users can then determine for which samples the model outcomes are reliable. Moreover, it also lays out in which contexts and for which classes its use is not recommended. In an example found by [100], models that were trained on light-skin-toned individuals were less reliable in detecting melanoma in darker-skin-toned people. Knowing the distribution of data within a dataset was an important factor for participants P5, P17, and P20. *So in case that you have a dataset that is not balanced then you have bias before even starting to train a model. (P5)*. Participant P17 also wanted to use this data model to change the number of samples per data split because, in a previous project, he/she had to do it manually which took up a lot of his/her time. Thus using this data model, users can derive X number of images per data split.

- **Top X Useful Concepts**: Based on the typicality, the top number of concepts can be derived from the database for a certain class. This tells the user which concepts are highly expected to be present in a sample image of a class. Users can then determine whether the concepts are generalizable to the real world or not [20].
  *Example query: SELECT * FROM mechanism WHERE class_id = <your_class_id> AND (type = 'GEM' OR type = 'LEM') ORDER BY typicality DESC LIMIT 10;*

- **Applicable Expectation**: Mechanisms can either be expected for a consequent globally or they can be unexpected. The expected mechanisms are the ones that are most likely always present in sample images, while the unexpected ones are those that are discovered by the model in the learning process but are not relevant to humans. This can however be useful to indicate mechanisms that are discovered during the diagnosis session, though they were not expected [7].

- **Reasoning Satisfaction**: At a local level, comparing the expected mechanisms for a sample to the learnt mechanisms for a sample can lead to three categories of bugs when there is no overlap between the expected and learnt sets: incomplete, irrelevant, or incorrect. [7]

  - incomplete concept(s): a concept is considered missing when the concept in the expected set is not present in the set of learnt concepts.
  - irrelevant concept(s)t: an irrelevant concept is one that is in the learnt mechanism set but not a part of the expected mechanisms.
  - incorrect inferred class: this happens when the class the model associates with the sample is different from the ground truth.

  Example query: SELECT s.image_path, c.class_label, ps.predicted_label, ps.confidence FROM sample s JOIN class c ON s.sample_id = c.sample_id JOIN predicted_scores ps ON s.sample_id = ps.sample_id WHERE ps.confidence = (SELECT MAX(confidence) FROM predicted_scores WHERE sample_id = s.sample_id)

- **Error rate per Class/ per protected group**: The error rate per group helps users further analyze a model to understand whether there was some bias towards a certain group. For example, in a study conducted by [25], they noticed that dark-skinned females are the most misclassified group with an error rate of 34.7% while light-skinned males have the lowest error rate of 0.8%. Participant P6 also expressed that getting the performance per class helps him/her to debug CV models and determine which model to use. *We also like to measure the detailed performance per class so that we can evaluate and search the models based on this metadata. (P6)*

4.2.3. Relationship

This section delves into the exploration and discussion of the relationships between entities within the schema. The focus is on relevant relationships that have been identified through literature, suggestions from participants, or those that contain specific attributes. These relationships are concisely explained for better understanding. Table 4.3 shows an overview of all relationships present in the data model.

- **Part of**: Within an instance, multiple other instances may exist. For example, a tree is an instance and it contains branches and leaves which are also instances. Thus, a branch is a part of a tree. Thus a recursive relationship is added to indicate the presence of an instance within another instance[88].
- **Next to**: A sink is placed in a bathroom as well as a kitchen. To differentiate both scenes, it can be checked if a mirror is placed next to the sink or a fridge. If it is a mirror, the scene is a bathroom, and if a fridge, it is a kitchen. During the interview with Participant P12, it was brought to our notice that the next to relationship should encompass all spatial relationships, such as on top of, below, to the right of, or to the left of. The type of spatial relationship can be calculated with the bounding box positions. This relationship is portrayed as a recursive relationship 'next to' is added to the instance entity [88].
- **is consequent**: As humans, we can classify images by simply looking at them and interpreting the semantic concepts. However, machine learning models work differently. They cannot simply identify concepts, but rather look into more physical finer-grained entities such as color, shape, texture, etc). For example, a model can identify a tree by looking at the leaves and branches. The branches themselves are identified by their wooden texture, long narrow rectangular shape, brown color, etc. Therefore, it can be seen that a mechanism is not just an association of concepts to classes but also a hierarchical set of associations between concepts [7].
- **Sample Mechanism**: Mechanisms can be globally expected or learnt but they can also have a local life, where the mechanism is associated with a sample. In other words, this relationship denotes whether the mechanism can be reasonably expected for a specific sample.
  - *Ranking Expectation*: Humans may expect a priority for one mechanism over another, thus a ranking of expectation for each consequent association is required [7].
  - *Configuration Status*: A mechanism can be satisfied by a model if it uses it to predict a sample. It can also be unsatisfied when the model does not use the mechanism at all, it is not entirely learnt, or if additional irrelevant concepts have been learnt [7].
- **is Antecedent**: Each mechanism is composed of an antecedent which is a set of concepts that define a class or a concept of higher granularity [7].
  - *Cardinality*: one concept might be relevant multiple times for a particular consequent. An example of an antecedent with cardinalies would be (flat countertop AND 4 legs -> table) [7].
- **Contains**: This relationship shows that a sample can contain one or multiple instances.
  - *Protected value*: When a dataset contains protected attributes, this attribute indicates whether a certain instance is a protected value.
- **Belongs to**: This relationship maps objects in a saliency map to their respective object.
  - *Present in Saliency*: Sometimes instances present in an image are not learnt by the model to make its prediction and therefore are not present in the saliency map. This attribute shows which instances are used by the model.
  - *Concept Partially Highlighted*: Sometimes the model only uses part of a concept to identify an image. For example, instead of highlighting the entire zebra, it may only highlight part of the zebra's body which shows the black and white stripes or for a car, it may only highlight the car wheels to predict traffic. This shows that the model does not learn the entire concept of a car, but rather parts of the car [61].

  - *Saliency per Concept*: [120] generate saliency maps for each concept so one can see what elements of the concept the model is looking at and which aspects of the elements are most important to the final prediction. Therefore, it can be seen whether the entire concept or only part of the concept is highlighted. A similar approach is followed by [23], where they generate saliency maps for concepts to be able to determine the overall class the image belongs to. Having a saliency map per concept was also some data Participant P5 wanted to clearly see which pixels within a concept were most important to the model.

- **Based on**: The pixels that were used by a model to predict images are highlighted in a saliency map. The based on relationship shows the saliency map that was generated for that model.

  - *Layer*: Layers in a model have different purposes. The lower levels are more focused on identifying edges while the higher layers focus on concepts that may be more relevant and easily understood by humans. Thus, some methods extract saliency maps at different layers of the model architecture [117]. Understanding the layer at which the saliency map was generated was also a crucial attribute for Participant P14, as each layer possesses varying levels of comprehensibility for humans. *The first layers are not capable of discerning features very clearly like the shapes or general info. So this is also a thing to understand, at which point of the network, the explanations are good enough for the human to understand them. (P14)*

- **Purpose**: The purpose of a dataset for a given model. A dataset can be used to train, validate, fine-tune, or test a model.

  - *Purpose*: This describes the purpose of the dataset for a given model. This was suggested by participants P6 and P13 to know whether the model was trained on a given dataset or simply fine-tuned on it. *I'm not sure what the relation is between models and datasets. There are some cases for example a model is trained from scratch on this set and sometimes the model is fine-tuned on another dataset. (P6)*

- **Applies**: This describes the type of perturbation applied to a sample with regard to another sample. This relationship was inspired by the interview with Participant P14 to reflect the relationship between the original sample and the perturbed sample.

  - *Level*: This describes the level of corruption applied to a sample. For example, in the research performed by [50], each type of perturbation had 5 levels of intensity.

- **Has (Sample ↔Class)**: This relationship shows that ground truth class a sample belongs to.

  - *Confidence*: Sometimes, ground truth labels for a dataset are obtained through crowd workers or model inference [25]. Thus having the confidence for a label is important, to add more transparency and trust for a dataset. Participants P9 and P13 suggested including the confidence of ground truth labels for this reason.

- **Has (Concept ↔Attribute)**: This relationship shows attributes that can be used to describe a concept.

  - *Value*: This describes the value of the attribute for a given concept. This could be the name of a color or a pattern description.

- **Hyponym**: Concepts can be general or specific. This relationship describes the superclass a concept belongs to. For example, guitar is a hyponym of musical instrument. When concepts are being annotated, guitar and musical instrument can be classified as the same annotation and the user can simply pick whether they want a more specific or general label. It can also be used to see the hierarchical relationship between concepts, as Participant P12 suggested.

4.2.4. Additional documentation

As there are existing resources that were designed to encourage transparency for datasets, models, and explainers, we decided to include those resources in our schema rather than adding each of the attributes separately and duplicating already existing work. These resources include datasheets [37], model cards [75] and saliency cards [19]. In this subsection, we highlight what each of these works is about, along with some additional information that was requested by the participants.

Datasheets

The primary objective of datasheets for datasets [37] is to enhance transparency and accountability within the machine learning community. By providing comprehensive information about datasets, datasheets aim to address and mitigate potential societal biases that can be present in machine learning models. Furthermore, they promote the reproducibility of machine learning results by enabling researchers and practitioners to access and evaluate the datasets used in various studies. Ultimately, datasheets assist in the informed selection of appropriate datasets for specific tasks, fostering responsible and ethical machine learning practices.

During the interviews, participants mentioned a few attributes that were already covered in the datasheets proposed by Gebru et al. [37]. The datasheet consists of seven sections: Motivation, Composition, Collection process, Preprocessing/cleaning/labelling, Uses, Distribution, and Maintenance. Below is a description of each of these sections along with a few example questions that can be possibly answered as an addition to our schema. It further includes a subset of example questions, which may be deemed unnecessary for addressing given that they are either already encapsulated within our data schema or do not hold significance for our designated use cases (questions in *italics*). Questions that were mentioned by the participants end with an asterisk (*). There were additional questions that were suggested by participants that are not covered in the datasheets proposed by Gebru et al. [37] and are therefore covered in our extended datasheet which is described in the following section.

- **Motivation**: This section encourages data creators to explain their motivations for creating the dataset.
  - For what purpose was the dataset created? *
  - Who created the dataset (for example, which team, research group) and on behalf of which entity (for example, company, institution, organization)? *
  - *Who funded the creation of the dataset?*
- **Composition**: The questions in this section are to provide the data consumers with the information they may need to make informed decisions about using the dataset for their tasks.
  - Does the dataset contain all possible instances or is it a sample (not necessarily random) of instances from a larger set?
  - Does the dataset identify any subpopulations (for example, by age, or gender)?
  - *How many instances are there in total (of each type, if appropriate)? **
  - *Are there any errors, sources of noise, or redundancies in the dataset?**
- **Collection process**: This section contains questions that may help other researchers and practitioners to create datasets with similar characteristics.
  - How was the data associated with each instance acquired? Was the data directly observable (for example, raw text, movie ratings), reported by subjects (for example, survey responses), or indirectly inferred/derived from other data (for example, part-of-speech tags, model-based guesses for age or language)?*
  - Who was involved in the data collection process (for example, students, crowd workers, contractors) and how were they compensated (for example, how much were crowd workers paid)?*
- **Preprocessing/cleaning/labelling**: This section provides data consumers with information regarding the preprocessing of the data and whether it is compatible with their tasks.

  - Was any preprocessing/labelling of the data done (for example, discretization or bucketing, tokenization, part-of-speech tagging, SIFT feature extraction, removal of instances, processing of missing values)?*
  - Was the “raw” data saved in addition to the preprocessed/cleaned/labelled data (for example, to support unanticipated future uses)?

- **Uses**: The questions in this section are to help dataset consumers reflect on which tasks the dataset should be used for and which ones it should not. It is intended to help these creators to make informed decisions without any potential risks or harms.
  - What (other) tasks could the dataset be used for?*
  - Are there tasks for which the dataset should not be used?
  - *Has the dataset been used for any tasks already?**
  - *Is there a repository that links to any or all papers or systems that use the dataset?**
- **Distribution**: This section contains questions related to the distribution of the dataset which can either be internally within the entity on behalf of which the dataset is created or externally to third parties.
  - How will the dataset be distributed (for example, tarball on website, API, GitHub)?
  - Have any third parties imposed IP-based or other restrictions on the data associated with the instances? *
  - *Will the dataset be distributed to third parties outside of the entity (for example, company, institution, organization) on behalf of which the dataset was created?*
- **Maintenance**: This section is to encourage dataset creators to plan for dataset maintenance and inform the dataset consumers to plan as well.
  - Will older versions of the dataset continue to be supported/hosted/maintained?
  - Who will be supporting/hosting/maintaining the dataset?
  - *If others want to extend/augment/build on/contribute to the dataset, is there a mechanism for them to do so?*
  - *. Is there an erratum?*

Extended Datasheet
The extended datasheets consists of questions for information that was pointed out by participants during the interviews and are not covered by the original Datasheets proposed by [37]. Following is a compilation of questions encompassed within the extended datasheets, which aim to foster enhanced transparency and knowledge regarding the dataset in use.

1. What methodology was employed for labelling the dataset, providing insights into the process and techniques utilized to assign labels to the data instances? Add a description of how the data was annotated. If multiple annotations were provided for one instance, what aggregation method was used to make a consensus? (P6, P9, P11, P15)
2. Provide a concise overview of the dataset, including its domain and a description of the types of pictures encompassed within the dataset. (P8, P10, P14, P18)
3. Can you specify the geographical locations where the pictures in the dataset were captured? (P3)
4. How can the dataset be used? Provide some example use cases. (P15)
5. Does this dataset represent real-world data or is it a toy dataset? Is this dataset real or synthetically generated? (P9)
6. How much storage is necessary for this dataset? (P6, P8)

7. Are there any additional details about the dataset? (e.g. Camera may have moved so the image may be taken from +- X position) (P18)
8. Are there duplicated images within the dataset? If yes, do the duplicates add value to the dataset or can they be removed? Are there minor differences between the original sample and the duplicate (eg. markings)? What is the significance of these minor differences? (P3)
9. What software is required to view/preprocess the image? (P3, P15, P18)

Model Cards
The intended purpose of model cards is to encourage transparent model reporting. Model cards are short documents that provide benchmarked evaluations in a variety of conditions, such as across cultural, demographic, or phenotypic groups and intersectional groups that are relevant to the intended application domains. They also clarify the intended purposes of the model and the tasks they are not suited for. Furthermore, they provide details of the performance evaluation procedures and other relevant information.

During the interviews, participants suggested adding some model-specific attributes to give users additional information about the models. Some of these questions were already covered by the model cards proposed by Mitchell et al. [75]. The model cards contains a set of sections that a model card should have. These sections are intended to provide users with relevant details to consider, however, they are not intended to be complete or exhaustive. They may be tailored according to the model, context, and stakeholders. Below are the descriptions of what these sections contain along with a few example questions that should be answered within each section and a few samples of information that are already provided by the data model and thus do not need to be part of the model cards (these are provided in *italics*). The questions that were also asked by the participants end with an asterisk (*). There were additional questions that were suggested by participants that are not covered in model cards proposed by Mitchell et al. [75] and are therefore covered in our extended model cards which are described in the next section.

- **Model details**: This section contains basic questions about the model.
  - What person or organization developed the model? *
  - Which version of the model is it, and how does it differ from previous versions? *
  - *What type of model is it?*
  - *Where can resources for more information be found?*
- **Intended use**: This section provides users with what the model should and should not be used for, and why it was created.
  - Primary intended uses *
  - Out-of-scope uses
- **Factors**: This section describes the model's performance across a variety of relevant factors such as groups, instrumentation, and environments.
  - What are foreseeable salient factors for which model performance may vary, and how were these determined?
  - Which factors are being reported, and why were these chosen?
- **Metrics**: This section is about the metrics that can be used to describe the model.
  - What measures of model performance are being reported, and why were they selected over other measures of model performance? *
  - How are the measurements and estimations of these metrics calculated? *
  - *If decision thresholds are used, what are they, and why were those decision thresholds chosen?*
- **Evaluation data**: This section provides information about how the model was evaluated.

  - How were these datasets chosen?
  - How was the data preprocessed for evaluation (e.g., tokenization of sentences, cropping of images, any filtering such as dropping images without faces)?
  - *What datasets were used to evaluate the model?*

- **Training data**: Questions about the training data are similar to those of the evaluation data. *
- **Quantitaive analyses**: This section provides the results of evaluating the model according to the chosen metrics.
  - How did the model perform with respect to each factor? *
  - Intersectional results: How did the model perform with respect to the intersection of evaluated factors?
- **Ethical considerations**: This section describes the ethical considerations that went into the model development, surfacing ethical challenges and solutions to stakeholders.
  - What risk mitigation strategies were used during model development?
  - What risks may be present in model usage?
  - *Does the model use any sensitive data (e.g., protected classes)*

Extended Model Cards
In addition to the model cards proposed by [75], an additional set of questions that can potentially help with the transparency and reproducibility of these methods are also provided in a separate model card. These questions were suggested by the participants during the interviews. The list below offers a comprehensive compilation of the questions that arose during these interviews.

1. What hardware was utilized for model training/fine-tuning, and what were the observed performance characteristics in terms of speed, storage utilization, and other relevant metrics? (P6, P8, P15)
2. Which loss function was employed to measure the performance of the model during training and optimization? (P9)
3. What specific training strategy or approach was utilized by the model during the training process? (P9, P18)
4. What optimizer and scheduler were used for this model? (P8)
5. Paper or resources for information about the original version of this model: where can resources for more information be found? (P13, P19)
6. If the model is made for a specific stakeholder group, provide a quantitative/qualitative analysis on how they were able to use the model (e.g. did the model meet their needs, were they able to use it easily) (P10)
7. How was the training procedure carried out? (P9, P13, P18)

Saliency Cards
With the rapid pace of development in explanability, users find it difficult to stay informed about the strengths and weaknesses of different explanability methods and therefore choose to continuously work with the few popular and familiar methods. They also tend to blindly use these methods based on the information and results shown in their papers. However, it is seen that some explainability methods underperform and do not accurately display what the CV model has really learnt [105]. *That so if you learn a machine learning model and then extract an explanation, it is entirely possible that the algorithm you use for extracting the explanation is just outputting garbage right, so you're not really evaluating the model you're evaluating the explanation technique which and there are very often not that many guarantees that they perform as intended. (P9)* To overcome this, [19] introduced saliency

cards. Saliency cards are structured documentation of how saliency methods work and their performance across several evaluative metrics. This includes information about the developers, design goals, input, model and user assumptions, dependencies, usage considerations, benefits and limitations, and performance across various evaluations. During the interviews, participants suggested adding additional information about the saliency methods (P12, P17), such as their characteristics, therefore, we decided to include saliency cards [19] which provide an extensive description of the saliency method in use.

Saliency cards proposed by [19] contains 10 attributes that are grouped into three categories: methodology, sensitivity, and perceptibility. Below is a brief description of each of these categories along with the attributes within each category.

- **Methodology**: provides information to help users understand if a saliency method applies to their task.
  - *Determinism*: measures whether a saliency method will always produce the same output (saliency map) given a particular input, label, and method. Understanding the determinism of the method can determine if and how users will apply the method.
  - *Hyperparameter dependence*: measures a saliency method's sensitivity to user-specified hyperparameters. This informs users about the parameters and how to use them effectively.
  - *Model agnosticism*: measures how much access a saliency method needs to a model. This helps users identify whether a particular method might be incompatible with their use case.
  - *Computational efficiency*: measure how computationally intensive it is to produce the saliency map. This allows users to decide whether running a particular saliency method is feasible in their setting.
  - *Semantic directness*: Saliency methods look at various aspects of model behavior and sematic directness represents the complexity of the abstraction. Semantically direct saliency methods do not require users to understand complex algorithmic mechanisms such as surrogate models or accumulated gradients, therefore making their results more intuitive to users without formal ML expertise. Semantic directness can help users determine which saliency method to use depending on the backgrounds of the people who will be interpreting the saliency maps.
- **Sensitivity Testing**: focuses on whether a saliency method changes in proportion to meaningful changes in the model and data.
  - *Input sensitivity*: measures whether the saliency method accurately reflects the model's sensitivity to transformations in the input space. This is essential for tasks that use explanability methods to understand the impact of input changes.
  - *Label sensitivity*: measures the saliency method's response to changes to the target label. This is important for tasks that evaluate model behavior based on changes in the target label.
  - *Model sensitivity*: measures whether the output of a saliency method is sensitive to meaningful changes to the model parameters. This can be used for tasks that compare models.
- **Perceptibility Testing**: This describes attributes that are related to the human perception of the saliency map.
  - *Minimality*: measures how many unnecessary features are given a significant value in the saliency map. This informs users about the amount of noise they can expect in the saliency map.
  - *Perceptual correspondence*: measures if the perceived signal in the saliency map accurately reflects the feature importance. This is important, especially for high-risk tasks, where misleading signals may provide an unwarranted justification for decisions or lead users to make uninformed decisions.

5 Results and Discussion: User Study

During the interviews, participants were asked questions about their daily tasks revolving around CV applications, their current needs and struggles, and how they envision using such a data model. Through these interviews, we gathered information to validate the three design principles: completeness and usability. Thus we asked participants for suggestions to create a complete data model, whether they would use the model and for what tasks, and whether the information provided through this model can solve some of their current struggles. This chapter provides information obtained through the interviews along with a thorough discussion. Section 5.1 contains the suggestions from the participants to make a more complete data model and Section 5.2 contains information about the usability of the model across different stakeholder groups.

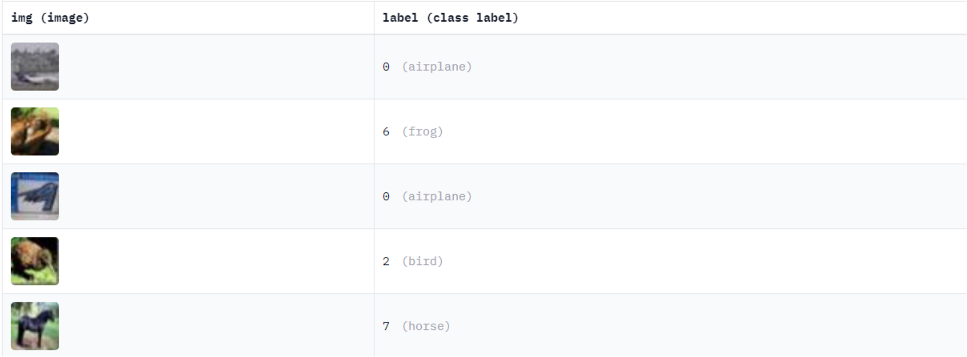
5.1. User Suggestions and Needs

In the first round of interviews, participants were asked about their work background, daily tasks regarding CV tasks, and their current needs and struggles with datasets and CV models. With this information and the participants' suggestions, we were able to complete our data model to store information about the datasets used, the models, explanation methods, ML fairness and robustness. In the last part of the interview, we told participants that we were creating a data model to add structure to datasets and increase transparency for models and datasets being used. We asked them about some of their requirements on entities, attributes, and relationships they would want to see in such a model before showing them the model we designed through literature (Figure 3.10). After hearing their requirements, we showed them the model we designed and asked for further suggestions to complete the model. Some suggestions that were made were already included in our data model, while the others were either included or excluded after our internal discussion. Although we aim to make a model that is as complete as possible, storing an overboard amount of information can slow the querying procedure. Therefore, we had to find a trade-off between completeness and usability. Table 5.1 summarizes the suggestions made by participants and whether the suggestion was already included in the initial schema (if mentioned before viewing it), whether it was subsequently incorporated into the final schema or can be derived with queries, or if it was ultimately not included. Suggestions that were already a part of the data model and those that were included in the model are described in Section Section 4.2.

5.1.1. Incorporated Recommendations

From the interviews, we gathered some interesting additions that can be incorporated into the data model. In this section, we describe a few of the suggestions, how they enhance the data model, and their importance in creating more trustworthy and reliable datasets and models.

Metadata Two of the most common suggestions were to include metadata of the datasets and models. While the data model that we designed through literature already contained some of this data, such as the name of the dataset, the labels within the dataset, the model architecture, and hyperparameters



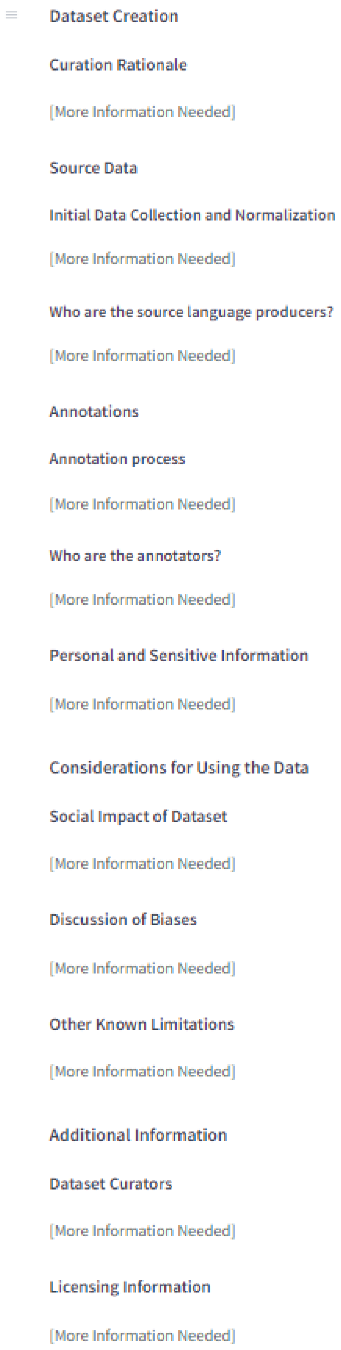
img (image)	label (class label)
	0 (airplane)
	6 (frog)
	0 (airplane)
	2 (bird)
	7 (horse)

Figure 5.1: Example of a dataset found on Kaggle without ground truth explanations

used for the model, there was some information that was missed out on. Some examples of missing metadata regarding the dataset included the dataset domain, license, data collection method, and authors of the dataset. For the model, examples of missing metadata included the version of the algorithm, the training process, model loss, scheduler, and optimizer. After gathering this information and continuing our research, we came across Datasheets for Datasets [37] (described in Section 4.2.4) and Modelcards [75] (described in Section 4.2.4) which included a majority of the information participants requested for. Instead of adding this information separately to our data model, we decided to include the already existing resources to enhance our data model. Additional metadata that was suggested by participants and was not found in Datasheets for Datasets [37] and Modelcards [75] were included in a separate document called Extended Datasheets (Section 4.2.4 - Extended Datasheets) and Extended Model Cards (Section 4.2.4 - Extended Model Cards) respectively. Diving further into research, we also found resources for saliency methods, namely Saliency Cards [19] (described in Section 4.2.4), and decided to include it in our data model to give users additional information about the characteristics of the saliency detection method. Including this information adds more transparency and allows users to use the information more reliably. Knowing the data that is present in a dataset can help users find biases in the dataset and therefore foresee biases in a model. Furthermore, including the metadata for datasets, models, and explainers allows users to reproduce studies.

Ground truth explanations While datasets contain ground truth labels, to our knowledge, there are few to no datasets that contain ground truth explanations (an example of such datasets is shown in 5.1) (*It's hard to say to what extent they have been solved because I know maybe there are some datasets which do contain explanations, but these datasets whether it fits your select research interest (P1)*). When using explanation methods, participants want to know what they should expect the model to look at when making its predictions - *I hope they have some ground truths with the evaluation or like even one or two sentences from experts like how they think about it and then we can refer to it. (P1)*. This could be in the form of only saliency maps or saliency maps with bounding boxes and textual descriptions. Including these textual descriptions provides more semantically diverse, informative, and relevant explanations [7], [11]. In the data model designed, ground truth explanations are encapsulated within expected mechanisms. Local expected mechanisms can show users what the model is expected to learn in a given sample while global expected mechanisms can be used to give the user an overview of the mechanisms that can be expected for images of that class. Including ground truth, explanations help users evaluate a model's interpretability at a global and local level.

Data collection procedure Furthermore, many of the widely accessible datasets do not contain information on the data collection procedure of the datasets. This includes how the data was collected, how were the ground truth labels annotated, the confidence of these labels, and more. Figure 5.2 shows a snippet of the dataset card for CIFAR-10 at Hugging Face. As can be seen, information about the data source, annotation, creation, ethical considerations, licensing, and more are left unanswered. A similar example was found for another dataset at Kaggle (Figure 5.3) Having this information allows developers to measure how generalizable the machine learning model is and whether there are any biases in the model - *'Nowadays we have more machine learning models that are training on data*



Dataset Creation

Curation Rationale

[More Information Needed]

Source Data

Initial Data Collection and Normalization

[More Information Needed]

Who are the source language producers?

[More Information Needed]

Annotations

Annotation process

[More Information Needed]

Who are the annotators?

[More Information Needed]

Personal and Sensitive Information

[More Information Needed]

Considerations for Using the Data

Social Impact of Dataset

[More Information Needed]

Discussion of Biases

[More Information Needed]

Other Known Limitations

[More Information Needed]

Additional Information

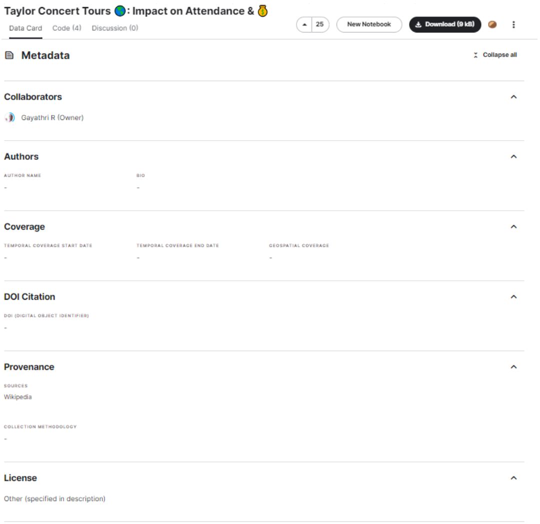
Dataset Curators

[More Information Needed]

Licensing Information

[More Information Needed]

Figure 5.2: Data collection and annotation data for CIFAR-10 at Hugging Face



Taylor Concert Tours 🌎: Impact on Attendance & 💰

Metadata

Collaborators

Gayathri R (Owner)

Authors

AUTHOR NAME	BIO
-	-

Coverage

TEMPORAL COVERAGE START DATE	TEMPORAL COVERAGE END DATE	GEOSPATIAL COVERAGE
-	-	-

DOI Citation

DOI (DIGITAL OBJECT IDENTIFIER)

-

Provenance

SOURCES

Wikipedia

COLLECTION METHODOLOGY

-

License

Other (specified in description)

Figure 5.3: Metadata for dataset at Kaggle

which is collected in the global north and annotated in the global south to be used again in the global north. And I guess this will, I mean this raised some you know some issue on everything you know on how generalizable is a machine learning model' (P15). Having this information allows users to identify where and why the CV model may fail and the groups of data the model may work better for. This is especially useful when the model is deployed to industry and the model has only been trained on some subgroups of people [100].

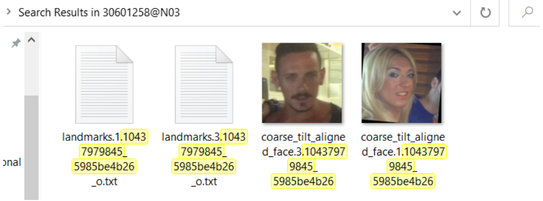
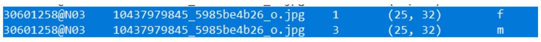

Search Results in 30601258@N03
onal
landmarks.1.1043 7979845_ 5985be4b26 _o.txt
landmarks.3.1043 7979845_ 5985be4b26 _o.txt
coarse_tilt_aligne d_face.3.1043797 9845_ 5985be4b26
coarse_tilt_aligne d_face.1.1043797 9845_ 5985be4b26

Figure 5.4: Images of the Adience dataset in their respective folder



```
30601258@N03    10437979845_5985be4b26_o.jpg    1       (25, 32)        f
30601258@N03    10437979845_5985be4b26_o.jpg    3       (25, 32)        m
```

Figure 5.5: Text file storing information about the images in the Adience dataset

Folder structure Participants also suggested adding folder structures where images can be found to reduce the amount of time taken to set up the data. For example, participant P3 spent months organizing the data and understanding the structure before being able to use the data for the task that was given to him/her - *That costs like months really just to fix that. And just talking to other people trying to know where they put the data, why they have put the data in a certain way, why there are duplicates or why there are duplicates with just a few things different marked differently. (P3)*. Including the folder structure to find images along with the Datasheet for Datasets and Extended Datasheets can be an initial solution to solving a problem like this and therefore cost participants less time to set up and understand the data. A similar problem was personally experienced during the engagement with the Adience dataset. Since this data is multilabeled, the images were categorized based on the user ID. The dataset also contained a text file that contained information about each image such as the user ID, image name, face ID, age and gender. Figures 5.4 and 5.5 show snippets of the images in their folder and the data in the text file respectively. As can be seen, the image names in the folders and those in the text file differ. While some may think it is easily understood that the image names are a combination of the image name and face ID in the text folder, it did take some time to figure this out. Thus having the file structure or a description of how images are stored can save quite some time.

Filtering images When introducing the tool, participants wished for a way to filter for images with certain concepts or attributes within a dataset or changing the number of images in a data split and more. These are not directly stored within the data model, but they can be derived with a few queries. Being able to filter images based on concepts or attributes within the image allows users to curate their own datasets, perhaps datasets with artificially injected bias. It can also allow developers to debug models or find biases in the model before sending it into production [4]. Participants also mentioned having a tool that can easily change the number of images per data split - *You could select like either want 20 images, 100 images, 50 images because I did that process manually for the purpose. But it could be also nice too if you if you if you're doing a user interface to like easily what you want to change is like the number of examples that you want to generate in for like. (P17)* While we did not design a user interface for using the data model (due to the time constraint), users can easily change the number of images per split using queries. Being able to control the number of images used each time can be used to find a threshold for the number of images necessary for training and how the performance differs as the amount of data is varied.

Varied attributes and image properties When showing the participants the data model, we had a set number of attributes and image properties that we had found in the literature. However, after speaking with several participants, we realized that many other image properties can be used by a model to make classifications, and thus find where and how the model is being biased and create robust systems. For attributes, we realized that material and texture are rarely filled and therefore lead to many empty cells in the database and a wider table, which is known to be a poor design choice [96]. Thus instead of having fixed image properties and multiple empty cells for attributes, as shown in Figure 5.6, we decided to make each of them a separate entity (Figure 5.7). With this design, users can add the name of the image properties with their respective values and pick one or more attributes and store its value in the relationship. With this design, the attribute table will only have five columns, and no empty cells, following the normalization practice of relational databases[67].

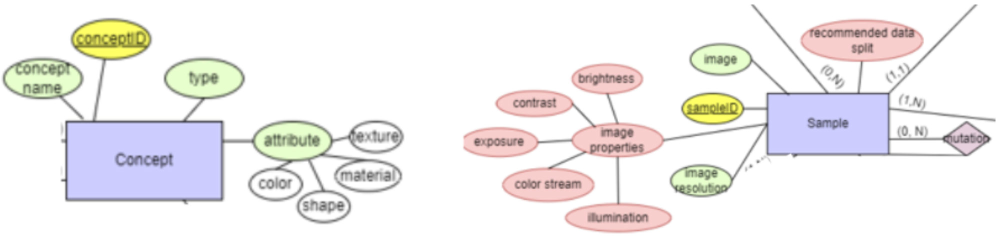


Figure 5.6: Fixed attributes and image properties (before interviews)

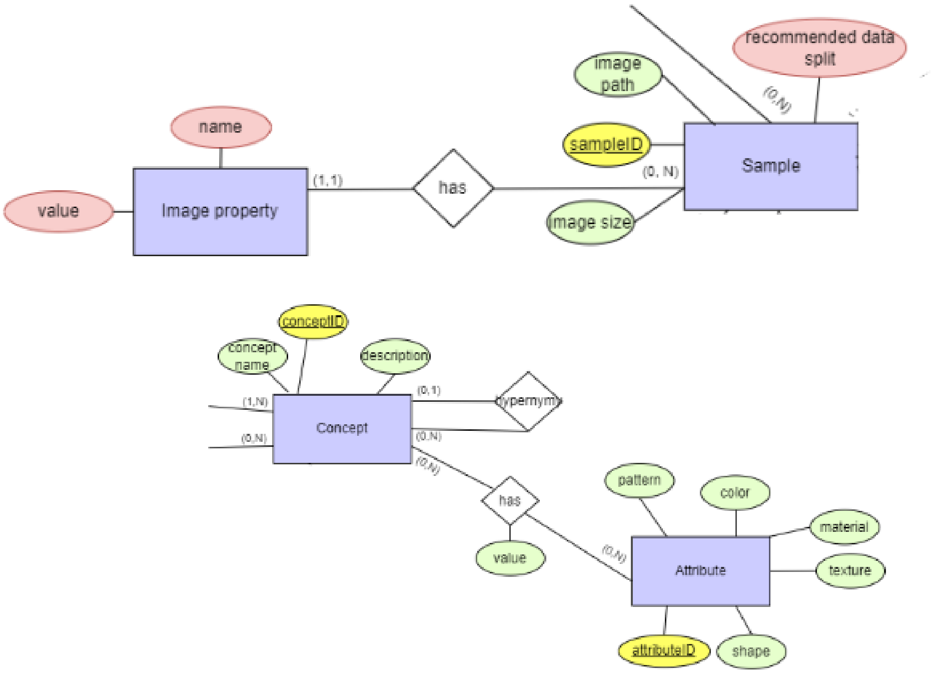


Figure 5.7: Images and attributes as separate entities (after interviews)

5.1.2. Omitted Recommendations

While all the suggestions made by the participants were quite insightful and valuable, a few were omitted as they did not fit the scope of our research or we had to make a trade-off between the completeness of the data model and its usability (as explained previously).

Multilabel Firstly, some participants did not want to constrict this data model to only single labelled datasets and thus suggested expanding the model to also include multilabel datasets. *If we have one image and we're only going after for cancer or not, the thing is that going from health tissue to cancer is a spectrum so it doesn't happen overnight. You are going to progress the cancer and those precursor lesions, they could also be present in your histopathology image. So that's why usually they also work with multiple labels, not just one label, but like let's say four or five. (P3)*. This is a possible extension for the future, by changing cardinalities. However, for the initial data model, we wanted to keep it simple to ensure that everything works as intended to. Moreover, many of the popular datasets for CV are also single labelled (ImageNet [1], CIFAR-10 [2], MNIST [3]), which made it easier to populate our data model and test the usability.

Multiple modalities Participant P3 also mentioned that in one of his/her recent projects, they were trying to classify histopathological scans using multiple modalities, both images and texts. Given this, Participant P3 claimed that for several domains, classification tasks are done using multiple modalities, and therefore suggested modifying the schema to support classification for multimodal representations. As mentioned previously, we wanted to keep the data model simple and ensure that everything was as intended before expanding the model to incorporate other modalities. Furthermore, our focus was mainly on CV applications.

Dataset and model reviews Another suggestion made by the participants was dataset reviews and model reviews. Participants requested these reviews so that they could read which datasets and models worked well together and which ones did not, the tasks that they worked well for, and more. *Could be convenient that if people can share some experiences or almost rate a certain model. (P10)* Including model reviews and dataset reviews was debated quite a bit because including user experiences, their problems and the solutions to those problems can be extremely helpful. However, including that in a database was not feasible. Moreover, Datasheets for Datasets (Section 4.2.4) [37] and Model Cards (Section 4.2.4) [75] included questions such as the tasks the model or dataset was made for and what they should not be used for. These questions already give users an insight into how to use the dataset and model. Furthermore, a link to the dataset and code for the model is also provided. These are usually on Github, thus users can create issues to add comments about the model or dataset in use.

Concept environment and characteristics A further recommendation made by Participant P3 was to include the environment of the concept. *Yeah, as in you can have birds from the Sahara, but also birds in Antarctica. (P3)* Including the environment can add more specification to where the image was taken or the type of bird in this case. While we did not specifically include the environment in our data model, we included the attribute 'description' and the relationship 'hyponym' (Figure 5.8). Here users can specify the environment of the concept if necessary, or even reference the specified concept (e.g. Antarctic bird) to the general concept (e.g. Bird). Furthermore, in the Extended Datasheets (Section 4.2.4) users can also include details about the provenances of the data. Another suggestion for the entity 'concept' was to include the characteristics of the concept. *What you can do is also look at the relationships of those objects like you see a car and it's driving on the road that you know that's driving or it's parked. (P10)* This was not included in the data model because we were solely focusing on classification tasks and the datasets and literature that we looked into did not require these characteristics to classify images.

[1] https://www.image-net.org/

[2] https://www.cs.toronto.edu/ kriz/cifar.html

[3] https://www.tensorflow.org/datasets/catalog/mnist

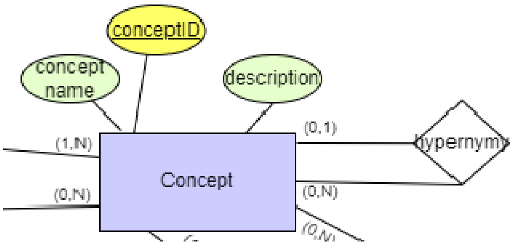


Figure 5.8: Description and Hyponym to add further specifications to a concept

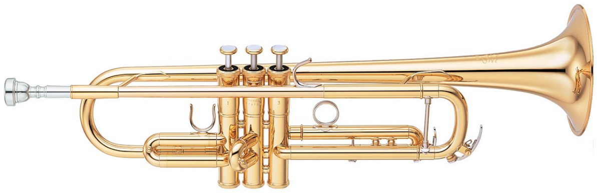


Figure 5.9: This image can be annotated as a 'trumpet' or 'musical instrument'

User interface Participants also suggested designing a user interface (UI) that can be put on top of the data model for easy extraction of data. One use of the UI was to download images in certain folders as finding the right images and then downloading them into separate folders can be quite time-consuming (P16, P17). Given the time constraint, we were not able to design a UI. However, users can simply write a script to extract images of a certain kind and download them into the respective folders.

Synonyms Participants were also concerned about the annotation given for a concept. Using the example of a use case encountered by participant P14, he/she said that for the image in Figure 5.9, he/she found that some participants labelled it as a 'trumpet' while others labelled it as a 'musical instrument'. For his task, both of these meant the same thing and he wanted a way to classify them as synonyms and keep the overall label as trumpet. In our data model, we see this as a hyponym and hyponymy relationship. We do not include an automated way to identify synonyms of annotations but there are resources such as WordNet [4] that can be used in collaboration with our data model to automate this process.

Relationships Participants also suggested some modifications in the relationships. One modification was to add a relationship between instance and attribute rather than concept and attribute. The suggested modification is shown in Figures 5.10 and 5.11. Participant P9 suggested this to make it easier to query for images with certain attributes. However, incorporating this change would force every concept to be filled. From the literature [39] [9] we read and the datasets we worked with, we found that many times the model does not learn a concept, but rather just the attribute of that concept. An example of this is shown in Figure 5.12, where a majority of the concepts learnt are random patches of the zebra pattern (Image obtained from [39]). Thus moving this relationship to instances would force users to fill in the concept name at all times. The other suggestion made was to move the relationships 'next to' and 'part of' (which are currently self-relationships with instance) to the concept. An example that was given by Participant P14 was that nostrils are always a part of the nose. However, the nostrils are not always shown in the image and keeping this relationship at an instance level will not show this relationship between nose and nostril. However, moving this relationship to concept will not make it generalizable. Nostrils are always a part of a nose but a fingerprint scanner is not always a part of a

[4] https://wordnet.princeton.edu/

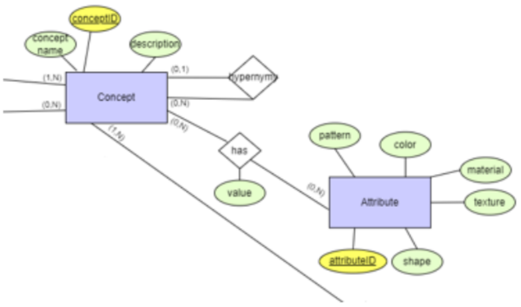


Figure 5.10: Relationship between concept and attribute

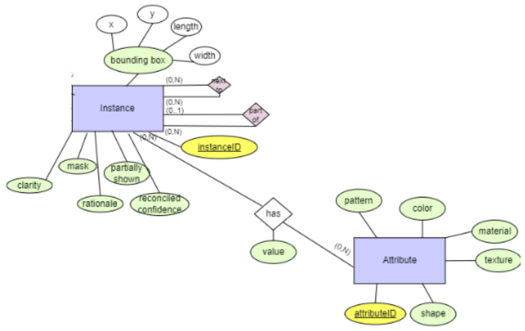


Figure 5.11: Relationship between instance and attribute

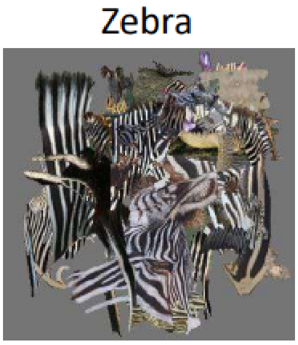


Figure 5.12: Patterns the model learnt (Snippet from Ghorbani et al. [39]

smartphone. The new smartphones do not include this feature, thus adding the ‘part of‘ relationship at a concept level does not make this generalizable for all concepts.

Requirement/Suggestion	Participant ID	Existing	Included	Excluded
Multilabel	P3, P9			✓
Multimodal	P3			✓
Folder structure (image path)	P3, P10	✓		
Image extension	P3		✓	
Instance annotation confidence	P3		✓	
Image brightness	P3	✓		
Environment of a concept	P3			✓
Model reviews	P3, P6, P10, P13			✓
Model metadata	P1, P6, P18	✓	✓	
Tasks the model was used for	P6		✓	
Statistical information about the data	P6, P20	✓		
Metadata of dataset	P2, P3, P6, P9, P10, P11, P14, P15 P19	✓	✓	
Amount of storage required for the model and datasets	P6, P8		✓	
Amount of time required to run the model (speed)	P6		✓	
How were the ground truth labeled annotated	P6, P9, P11, P15		✓	
Data distribution	P5, P17, P20	✓		

Requirement/Suggestion	Participant ID	Existing	Included	Excluded
Amount of data in a dataset	P20	✓		
Dataset reviews	P10, P20			✓
Annotations/concepts within a sample	P9, P12, P1	✓		
Saliency map	P9	✓		
Is the dataset labeled	P9		✓	
Training procedure	P9, P13, P18		✓	
Saliency per concept	P5, P9	✓		
Filters to get samples with certain characteristics/bias	P8, P17	✓		
Description of training data	P8, P18		✓	
Code	P18, P19		✓	
Instance mask	P18, P19		✓	
Color grid for saliency maps	P18		✓	
Download images into folders	P17			✓
Changing the number of images in each of the splits	P16, P17	✓		
Extract saliency maps for multiple samples from a class	P17	✓		
Synonyms for concepts	P14, P17			✓
Saliency map characteristics	P12, P17		✓	
Object overlapping/rationale	P10, P13		✓	
Model purpose for that dataset	P13		✓	
Global mechanisms of a class	P12	✓		
Concepts at an abstract level	P12		✓	
Annotation source	P11, P15		✓	
Data preprocessing	P11		✓	
Image resolution	P10, P14, P19	✓		
Characteristics of a concept	P10			✓
Dataset updates	P10		✓	
Data augmentation/mutation	P10, P14		✓	
Link to original model	P19		✓	
Link to dataset	P14		✓	
Example usage of the dataset	P15		✓	
Hardware model was trained on	P6, P8, P15		✓	
Ground truth confidence	P15		✓	
Ground truth explanation	P1	✓		

Table 5.1: Participants' requirements and suggestions extracted from the interviews.

5.2. Uses of the Data Model

During the first round of interviews, participants were inquired about prevailing challenges encountered with their daily tasks concerning CV and datasets. Additionally, they were prompted to mention their desires to solve these problems or enhance the efficiency and expediency of these tasks. Participants were further queried regarding their intended utilization of the data model. This section discusses the findings of the interviews. Section 5.2.1 discusses the motivation gained from the interviews and how they envisioned incorporating the model into their daily CV tasks. In the second round of interviews, we asked participants to interact and use the model to further examine the usability of the model. Section 5.2.2 examines the usability of the data model through participant interactions.

5.2.1. Potential Use Cases of the Data Model

When speaking with the participants during the first round of interviews, they mentioned some desires they had to make explainability, fairness, and robustness for CV tasks easier, faster, and more understandable. While speaking with them, we gained more confidence in our data model, that it can be used to solve many of their current problems. After showing them the data model, we also asked them about potential uses of the data model and how they envision themselves using it in the future. In this section, we discuss the participants' challenges and desires, ways in which the data model can be used to solve these problems and how they see themselves benefiting from the use of the data model. In this section, we discuss the potential use cases of the data model. We have grouped our findings according to the following categories: CV model, Dataset and Samples, Explanation, and Structure, Documentation, Definitions.

CV Model

- **Debug and test model** CV models are essentially black boxes. We do not know how and why these models make predictions. *I think there's a lack of interpretability in deep learning tasks like for instance when you want to train a model. We don't know the reason why it's producing these results sometimes. (P16)* This was a concern for many participants (P3, P4, P6, P9, P16, P19). They want to be able to debug the model and know more about the data the model was tested on. Before sending models into production, it is a good practice to test the model on a variation of data. As Participant P3 mentioned, he/she would want to know how much variation is present in the data the model was trained and tested on. This variation could be data from different labs, data from different scanners, samples with perturbations, and more. Using this schema, users can get a picture of what the dataset looks like, its statistics, the biases present and more, all of which can contribute towards the performance of the model. With this information, users can determine the robustness of the model and how it will perform on different variations of data. *what or where is this entire training going wrong, because that would also help me evaluate what sort of augmentation should I use, and what sort of biases are embedded in the model or my data. (P4)*; *It seems that usually if you just train your model and you get the performance, but if you really want to debug your model and also want to investigate the data set you trained on. So with this schema or this rich information, you can identify what would be the factors that can affect the performance of the model and how to improve this issue maybe your dataset is just biased and which leads to the bad performance of the model or maybe is there some other problems in this data set that affect the performance. So with this data model, you can probably get some hints on what your dataset looks like and what are the drawbacks. (P6)* Users can use the information provided in the database to identify where the model fails and then improve on those parts. *In order to understand where the model fails and also to understand where and how I need to make the training better or the training data set better in order to get the best out of the model. P(4)* The problems may arise from the training data or model implementation. Thus, using the information provided by the data model, users can debug their models to find the core problems.

- **Model comparison** Participants also wanted a platform that stores model performances and can therefore compare models and make decisions on which one to use based on different characteristics (P4, P6, P13, P15, P18). One of the characteristics was model performance, where users can compare the commonly used evaluation metrics to determine which model fits their use case. *But in most cases you need to use uh commonly used evaluation metrics because you don't want to run every method you want to compare. (P18)*; *How different are the models' performance and also how good and some criteria to define like how this model is working and what is this model right? (P6)* There is no clear definition of a good model, but with the use of the data model, users can compare models based on criteria and choose the model(s) best suited for their task. Users can also compare model performances based on a dataset of interest. *A leaderboard of benchmarks that have already been set on that particular data set that hey, this is the research that that performs the best on this particular data set so that I already know what what works on this. (P4)* Thus, having this relationship is quite important to make these comparisons and also sets the data model apart from other available tools. These intrinsic relationships can be used to compare models for a dataset and also evaluate models based on different datasets. *the current projects are working on and we are trying to discover the relationships between models to models, models with data set and even data set to data set and we try to construct an ecosystem and to learn the relationship between these artifacts or data set models. If we know the intrinsic relationship, we can predict uh, which model is a better candidate for the new data set or what kind of models to use whenever we have a dataset to play and when you use the data sets to evaluate your* *models. (P6)* The data model can also be used to compare model performances based on the learnt concepts and attributes. This can be used to identify the granularity at which the model is making its predictions. *like the cloth texture you have then maybe there's some different kind of texture, maybe these are the concepts then you classify these texture or uh or concepts and you group them into different subsets and try to measure like how these models perform on each subset of this data set and you see OK, maybe this model is performing well on some yeah, in a more detailed, fine-grained way. (P6)* With the information provided by the data model, users can compare models based on the concepts they have learnt. For example, some models may be better at learning different textures while other models may be better at learning colors. Users can choose the model that better fits their dataset. Lastly, the model can be compared based on attributes present in the dataset. So the model's performance can be compared on a dataset with injected bias and then on the same dataset without any injected bias. With this, users will know whether the model learnt the injected bias, the influence it had on the model's performance, and how will it perform without the injected bias. *Then most of the images that you trained the birds were always accompanied by a cactus. So when you are trying to evaluate the performance of the model, whenever you use the images without the cactus, do they still receive a good performance or not? So whether this model is biased or not. So I think the cactus is the concept you want to identify or distinguish. (P6)*

- **Finding models** Participants also suggested that the data model can be used to find models for a particular dataset or an explainability model that is compatible with the CV model in use (P5, P6, P10, P13, P18). *The models I've implemented, I've had to edit this structure a little bit because oftentimes explainability models aren't compatible with that. (P13)*. Thus the data model can be used to find an explainability model that is compatible with the CV model in use. It can also be used to share this information so that another user who wants to use that CV model and explainability method has access to the adapted models. Oftentimes, research studies start with a baseline study as a line of comparison to their work. *And the baseline that you choose is something that's already been worked on the same data set that you're also using. (P18)* The data model can be used to find a baseline model, a model that is similar to the one a user is trying to implement or a model that has been evaluated on a dataset similar to the one a user is planning to use. Users can also explore the model to see the type of data it requires before using it. *We tried some other baseline model first and then it didn't fit for our project. So we give it up. And try to find out what other method we can use and then we can find the proper one. (P18)* This can save the user's time since he/she does not have to try different methods, but rather just skim through the model's details provided in the database and eliminate the ones that are not compatible with their dataset or model. Thus the data model can be used to find a model for a particular use case or for a certain dataset. *One of the challenges that we have is that, for example, you have a new dataset coming in and then probably you want to verify the performance of the model on this new dataset then how do you find a good model for it and in addition like, how do you find good models to finetune on this data set. (P6)* Using this data model, users can find models that can be used for a new dataset based on the model's characteristics or similar datasets the model has been trained or fine-tuned on.

- **Model access** Participants also found that building a model from scratch can be time-consuming, difficult, and requires users to understand various components, some of which may be out of their scope (P4, P7, P8). *Well, yeah, it's really the time that's the bottleneck, right? I mean, if as a user I need to read a new paper every time I want to extract something explainable, conceptualize it, generalize it, and then implement it, that takes I don't know, two or three man days. If there was a library that would just save a lot of time. (P4)* With this data model, users look through the details of the model to understand how it works and then pick the model he/she wants to implement and use. In comparison to reading every paper, simply reading the information provided by the data model is less time-consuming. These models also require hyperparameters to be set, but for many methods, especially explainable AI methods, it is not always clear what the hyperparameters do and how they can be correctly set. *XAI methods need hyperparams to be set in some way, but there is no clear understanding of how to do that; so having the entire model there makes it easier to use. (P7)* Including the code for each of the models along with the results, can help users to build their models, and pick the correct design and hyperparameters. The data model also includes the hyperparameters used for a model such that users can simply use those hyperparameters if the model matches their use case. Furthermore, many times models are not readily available on the web for others to access and use (P10, P14, P17). Users have to build their own models, by doing research and a lot of trials and errors to design a model that fits their product. *On one of my projects, we had to create all the models from scratch. There were very few that had repositories on the web. This is also a thing that would be quite better if there were a few more on the web. (P14)*; *So in our case, for instance, we currently use a heavy model, computationally heavy so you try to come up with solutions here and there you do have to do a bit of research and a lot of trial and error and finding proper architectures yourself so that can be a bit of a challenge to basically, design a model which perfectly fits for your products. So for us to give an example like we have a CV product which uses 2 cameras at the same time. So we want to have a model which can process two cameras to do that well, that's something which is not very common, so there's not a lot of open-source models, which is sufficient for this. (P10)* Users can populate the data model with all these models so that future users can use them as inspiration for their works. Having such a data model may reduce the amount of time required to find models, as users can find models that have been used for similar use cases and adapt them to their products. Moreover, the data model also provides users with information that can be used to access models and therefore reproduce studies.

Dataset and Sample

- **Custom dataset** Participants envisioned using the data model to create their own datasets (P2, P5, P6, P8, P10, P17, P19). These datasets can be customized based on the data already provided in the database. Firstly, the data model can be used to help users create diverse and balanced datasets. Bias in a model usually comes from the data it has been trained on [49] and one common reason is an unbalanced and not varied dataset. *I think that's the two most important things for me - diversity and equal representation of classes. (P5)* Using the data model, users can easily check the balance of classes in the dataset and its diversity. Datasets can also be created by selecting images with certain properties or images with certain concepts and/or attributes. *I might look for some possible data set and then I do some manual selections, bring some of the specimen in my own data set. (P8)* Participant P17 also created a dataset by manually adding images of dogs that did not have red eyes for example. *I wanted to have pictures of good-looking cats and dogs that I added manually in that because I thought the dataset from hugging face was it was all pictures like weird with dogs with red eyes and so I added them manually which took time for sure. (P17)*. All of this can be done by combining datasets, either entirely or using parts of the dataset as described above. Users can search for images with certain instances (eg. dogs with brown eyes) and curate their own dataset. One of the most common responses for characteristics of a dataset and constraints of datasets was the size of the dataset. *But they require a lot of images. In my case, we only have 18 images. So then we cannot get a good performance because we can only align the 10 images out of these 18. So then we can not use all of that, then that is not good. (P18)* To build a fair and robust CV model, a lot of data is necessary. With the use of the data model, datasets with similar characteristics can be combined to create bigger datasets. *So I use multiple data sets and I try to combine them. (P19)*

- **Explore datasets** Participants also wished to explore datasets before downloading and using them. They said that they can explore the characteristics of the dataset and the instances within the images through the data model (P6, P9, P12). *You know it's been a research topic an active research topic for many decades, right to create the metadata that describes what's in the data set. (P12)* It is useful to developers to know what images are in a dataset, their quality, instances within the dataset, and where the images were captured. Having this information in the database helps users to understand the data and therefore analyze and create better CV models. In this way, users know what to expect the model to learn and can also more easily pick up on any biases. Furthermore, some images require special datasets for preprocessing or to open the image. In one project, after participant P18 downloaded the dataset, he/she realized that it requires special software to view the images. The software was too large to run on his/her computer and therefore the dataset could not be used. Therefore, participants wished to explore the dataset and be informed about the software requirements before downloading the dataset. *I don't want to download a specific software, so just open it and then export it to another format. (P15)* Users can check the image extensions provided in the database or the extended datasheets for the software necessary to view/preprocess the images in the dataset.

- **Dataset access** Participants also found that creating their own datasets can be quite challenging (P8, P10, P14). They have to look for datasets that may have classes or images that are relevant to them and then manually select these images to create their own dataset. Sometimes datasets are not representative of the real world, especially niche datasets. *We might want to gather a data set that not only taking you to know the videos coming out of a like a bike commuter in the Metropolitan City. Instead, we also want to take some footage, let's see when they are trying to bike in the mountains or when they park the bikes over a riverbank or something like that. (P8)* With the data model, users can find these images with rare settings to add to their dataset. Training CV models on such diverse datasets can lead to more reliable and robust models. Many times the datasets that they are interested in creating are already existing, but are inaccessible to them. *'I think in general that could be very useful indeed, because now basically if you if you have to search for datasets, it's this is a lot of trouble indeed' (P10)* Thus, having a platform where users can share datasets and access other (niche) datasets can be quite valuable and time-saving. The data model can also be used to find labelled datasets, especially since labelling and annotating datasets can be time-consuming and expensive. *For the CV task it's really just managing the data. What a nightmare. Trying to go through everything, especially with object detection where you have to go through each image and try and label them. (P13)* So having a platform where users can share this information will reduce redundant work.

- **Inspiration for dataset creation** Another struggle one of the participants (P17) encountered, was finding artificial bias that can be added to the dataset. He/She spent hours looking through multiple datasets of dogs and cats to get inspiration for what bias can be injected in the dataset. *I thought of focusing on cages by looking through multiple datasets and thinking about what we have in there. (P17)*. Users can find similar datasets and look for the artificial bias that was injected into those datasets. They can then use those datasets as inspiration for their dataset and also use images from those datasets to create their own datasets. They can also find inspiration for artificial bias by finding some of the most common concepts within images.

- **Data sharing** Participant P20 expressed a desire for a centralized hub encompassing both data and metadata about datasets, which could substantially economize users' time. This can help users find the data that they are looking for and also validate that it is the data that they are actually looking for. *A common hub of datasets maybe even with deeper explanations and insights into the data, like there's a lot of techniques to describe datasets like there's for example these datasheets approach. If that's automatically integrated into a hub, I think that would be super helpful for people to want to find the data that they need but also kind of validate that the data is actually what they're looking for. (P20)* The data model can be used to find datasets belonging to a certain domain, with specific classes, or other characteristics that will meet a user's needs. The model can also be used to find larger datasets especially since in many domains data is scarce and more data is always appreciated for CV applications. *For that project we only had 100 images. So I would use this to look for similar datasets to train the model. (P3)* The data model can also be used to find niche datasets or datasets that contain very specific data. *'very useful for finding those datasets which are less known Internet.; Use case so find very specific data sets and that becomes quite cumbersome, so it would be nice if there's a schema that allows you to find these more scarce or more unique data sets; I think like a month ago or something, I was looking for x-ray datasets on suitcases, for instance, and then you find all kinds of small ones, but it's quite hassle to find them and what indeed the characteristics are so if there's like a schema which would allow to more quickly find those find niche datasets that would be quite valuable. (P10)* Being able to find these niche or specific datasets already gives users an idea of what the images contain, the characteristics of the dataset, and in some cases also what they are expecting the model to learn. Participants found that the data model does not only have to be used to extract information but can also be used to publish their works and share data with other researchers (P2, P9, P13, P15, P19). *I would love to use it both for using it like you know having data but then also publishing back data in some way. (P15)*; *If I could see the quality of images, I would use the schema to look through all relevant datasets, download all the reasonable images and build my own dataset and* *then i can also upload this new dataset. (P19)* This will increase transparency and encourage reproducibility across CV applications. Users can also build more robust models as they will have access to more varied data rather than the few popular datasets that are always being used to train models. Furthermore, users can also share their knowledge about the datasets, thus giving other users enough information to use the data reliably and knowledgeably.

- **Image quality/relevancy** During the interview, participants were asked about the limitations of the current datasets that they were using or had access to. One of the limitations was the quality and relevancy of images in a dataset (P5, P18, P19). Participants complained that they had to manually scan datasets to handpick images that could be used for their use cases. *I mean that then sometimes the quality of the image was not so good. The images are blurred or the size of an image could be small or even the light or the angle and stuff like that (P5)*; *I had to categorize images. I had to semi-automatically sift through like all of these data sets to find the data that I find relevant. Where there is a small amount of patients because they're doing a certain type of treatment, they need to have their arms positioned above their head. (P19)*. Manually selecting images can take up a lot of the user's time, especially for larger datasets. Using queries, users can simply filter out unwanted images, for example, by filtering for clarity and existence of concepts in an image and image corrections. Knowing these details about the images in a dataset allows users to create more robust and fair models, as they can investigate the properties and clarity of the images the model predicted incorrectly.

Explanation

- **Human-in-the-loop concept-based explanations** Although not the intended purpose of the data model, participants used the model and interview to learn about a new topic - human-in-the-loop concept-based explanations. This essentially means that the concepts within an image are annotated, making it more interpretable to humans. Throughout the interviews, participants expressed an ongoing inclination for explanations for what the model has learnt to be comprehensible to human understanding (P1, P2, P3, P12, P14, P19). *We really needed to have better ways of well, first of all, explaining what exactly the model has learned in human concepts. (P12)* In our data model, we facilitate human understanding with mechanism, concept, and instance along with the saliency maps. We started designing our data model with a focus on saliency detection and slowly realized that saliency maps can sometimes be difficult to interpret, as shown in Figure 5.13. *From Imagenette which is a subset of Imagenet, yeah, and some of them are about fishes and it was quite hard to understand whether the, for example, the network was looking at the scales or the network looking at the color or the network was looking, I don't know at the side, the shape of the fish or things like that. (P14)*; *I think saliency maps are very easy to misinterpret the the reasoning why I don't see them as a good explainability method. (P2)*. Therefore, we extended the data model to also include entities that can translate these abstractly highlighted pixels into something more meaningful and easily understandable to humans. Including annotated concepts also makes it useful to query for images based on what is present within an image. *Having the concepts is useful because we want to know what's in the image. (P1)*

- **Mechanisms** Another unique element of the data model is the incorporation of mechanisms. As mentioned earlier, during the interviews, participants learnt about a new explainability method, concept-based explanations. Mechanisms can be used for several purposes. Firstly, mechanisms can be used to identify the reasons behind a model making a prediction (P5, P9). Sometimes, models make the right predictions but for the wrong reasons. A well-known example is the husky and wolf dataset (Figure 5.14). The CV model learnt that images of wolves contain snow in the background, while those of huskies do not. Therefore, when given an image of a husky in snow, it predicted it to be a wolf. With mechanisms, users can compare concepts that have been learnt by the model to the ones that they expect the model to learn (P1, P9, P10, P14). *It's not giving you the right prediction using the wrong reasons. But right so essentially what I do, I tend to consider data sets that do come with ground truth explanations that tell me what the machine should be looking at. And then I extract explanations from the machine's predictions using XAI tools, other existing ones, the state-of-the-art or something that I came up with and compare the gold standard with what the explanation that the machine gave me. (P9)* This comparison can be used to measure how robust a model is. The difference between the learnt and expected

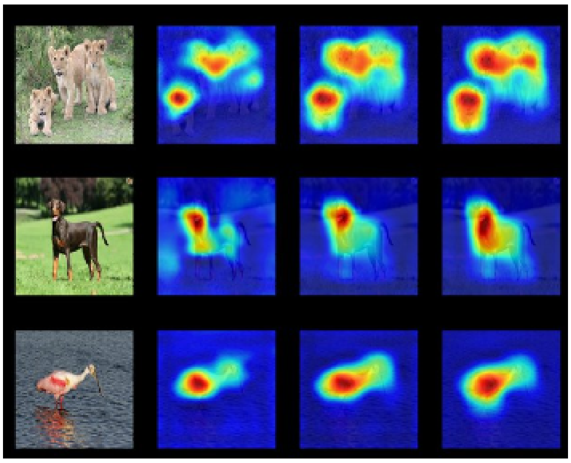


Figure 5.13: Examples of saliency maps which are not easily comprehensive to humans

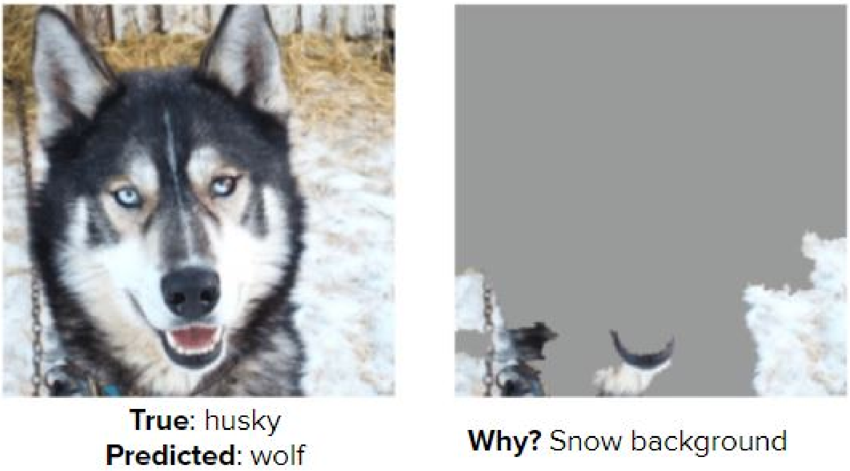


Figure 5.14: CV model bias - model predicts all images with snow backgrounds as a wolf

mechanisms can be used to debug the model (P2, P11). *So this relationship between what are the expected elements that should be there for a certain sample to be predicted as kitchen or not. Contrasting them with the ones learned through the saliency maps that match or mismatch could be also useful for debugging. (P11).* This helps a user know more about what the model is exactly looking at when making a prediction, whether it is looking at a part of a concept, an attribute, or something totally unrelated to the ground truth. Figure 5.15 shows an image of a Hairy Woodpecker and concepts within the image that the CV model learnt. It can be seen that the model learnt concepts related to the image background rather than the Hairy Woodpecker itself. Another importance of mechanisms is to identify parts of an image, which can help identify what the image constitutes and also allow users to select portions of data that contain similar concepts (P10, P13). *In this large database, let's say yeah, that's more like, maybe it goes a bit more into technical detail, but I want embeddings of all my images and I want to have certain clusters of images which are which are highly similar. I want to easily crawl within the data, and select portions of data. (P10)* Portions of data can be selected by writing a few queries to extract the necessary data. Lastly, mechanisms can be used to define a class (P4, P5, P10). There is no global definition for some classes. For example, living rooms in America are different from those in South Asia. Only the authors of the dataset have a clear definition of what the class is about and therefore can define it through mechanisms - a set of concepts, which together define a class.

- **Ground truth explanation** Participants also wished that more datasets included ground truth

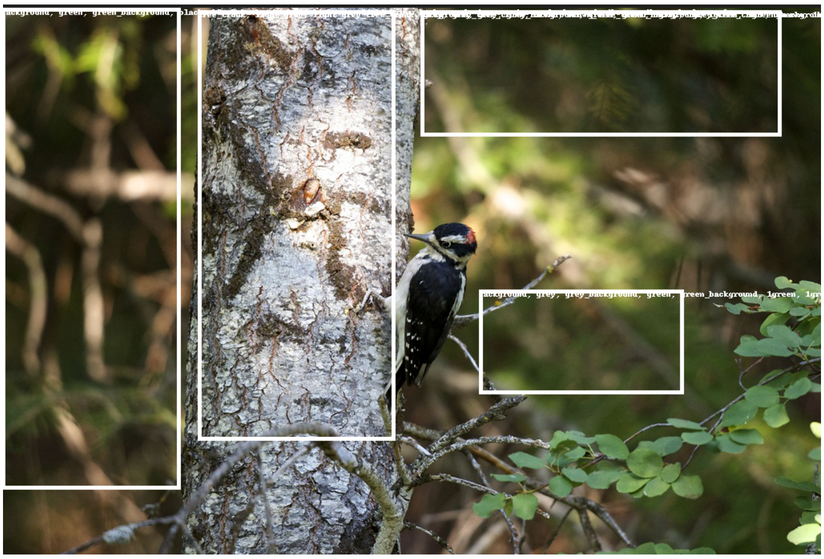
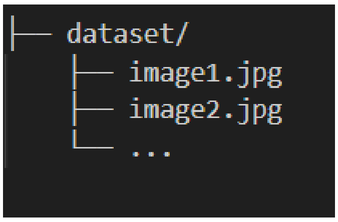

ackground, green, green_background, blac
ackground, grey, grey_background, green, green_background, 1green, 1gr

Figure 5.15: Sample of a Hairy Woodpecker where the model learnt instances of the background rather than the bird

explanations so that they know what they can expect the model to look at and what a good explanation should provide (P1, P5, P6, P9, P12, P15). *It's useful for defining the problem right? What a good explanation should provide. (P12)* The lack of ground truth explanations for datasets results in models overfitting to the small number of datasets that contain ground truth explanations. *There are very few datasets that come with ground truth explanations, right because that requires effort really. So there aren't that many out in the wild, meaning that of course algorithms that tend to focus on those and tend to overfit those in the sense that if you look at them, people tend to optimize performance on very few data sets, right? (P9)* Ground truth explanations can also help users to diagnose for what cases the model is being biased. Users can compare the ground truth explanations to what the model has learnt and then determine the contribution of the additional concepts learnt and the degree of concern for the lack of concepts learnt. The data model includes expected mechanisms for datasets, which can be used as ground truth labels. Although they are quite expensive, adding this information to the data model will enrich it and allow others to also have access to it. This can result in a larger range of datasets with ground truth explanations which can be used to train and evaluate models.

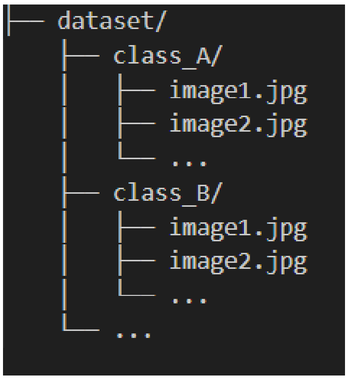
Structure, Documentation, Definitions

- **Data Structure** Participants also wished that there was a platform that stored datasets, their metadata, the way the data was preprocessed, and explanations for the datasets in a structured manner (P2, P3, P6, P13, P15, P18, P20). Having this structure can contribute to time-saving for many users. For example, *if the structure is the same, then you don't need different codes to preprocess datasets. (P18)* datasets can be structured in many different ways. For example, the Birds dataset was stored in a single folder (Figure 5.16), Imagenette and Colored MNIST were stored according to the class labels (Figure 5.17, and Adience was stored according to user id and annotations (Figure 5.18. These are three among many ways in which a dataset can be structured. Furthermore, participants held the viewpoint that the inclusion of additional details regarding the dataset could prove to be beneficial and enhance informativeness. Most datasets only contain information about the classes that belong to the dataset and the samples within each class. Figures 5.19 and 5.20 show the overview of a dataset sourced from Kaggle. It can be seen that the dataset overview simply shows the labels included in this dataset and its potential use cases. However, there is no explanation for how the data was collected or what can be expected in the folder 'Other' or 'Master Folder'. *having a repository where there is some structure on the kind of metadata that you want to have, let's say. That would be great because there are repositories like this that exist, but I think they don't have this kind of thing. This is more for everything. (P15)* Participant P20 also believed that including some structure that is modular so that users can choose the information that they want depending on the task at hand -*want*



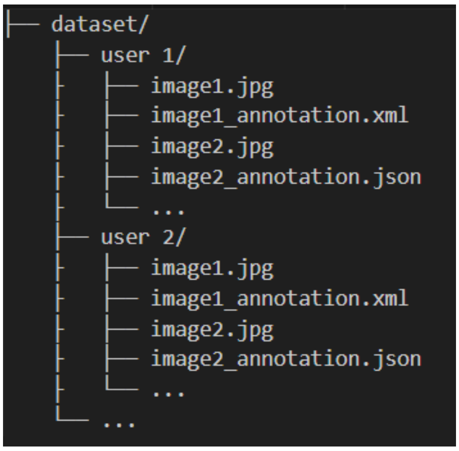
```
├── dataset/
    ├── image1.jpg
    ├── image2.jpg
    └── ...
```

Figure 5.16: Dataset stored in a single folder structure



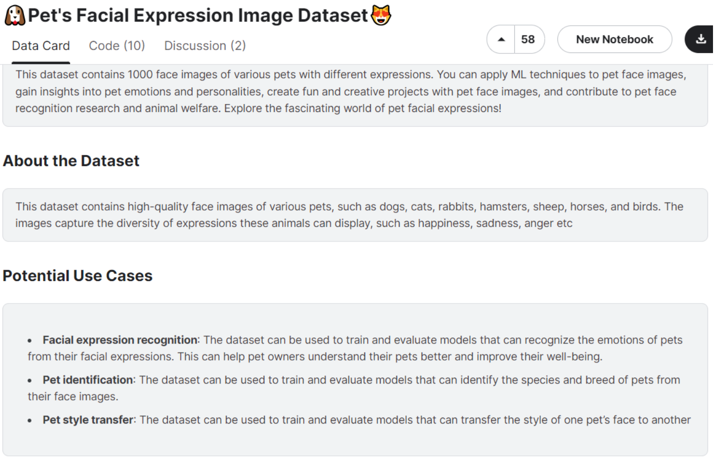
```
├── dataset/
    ├── class_A/
    │   ├── image1.jpg
    │   ├── image2.jpg
    │   └── ...
    ├── class_B/
    │   ├── image1.jpg
    │   ├── image2.jpg
    │   └── ...
    └── ...
```

Figure 5.17: Dataset stored in a class folder structure



```
├── dataset/
    ├── user 1/
    │   ├── image1.jpg
    │   ├── image1_annotation.xml
    │   ├── image2.jpg
    │   ├── image2_annotation.json
    │   └── ...
    ├── user 2/
    │   ├── image1.jpg
    │   ├── image1_annotation.xml
    │   ├── image2.jpg
    │   ├── image2_annotation.json
    │   └── ...
    └── ...
```

Figure 5.18: Dataset stored in a user id - annotation structure



Pet's Facial Expression Image Dataset

This dataset contains 1000 face images of various pets with different expressions. You can apply ML techniques to pet face images, gain insights into pet emotions and personalities, create fun and creative projects with pet face images, and contribute to pet face recognition research and animal welfare. Explore the fascinating world of pet facial expressions!

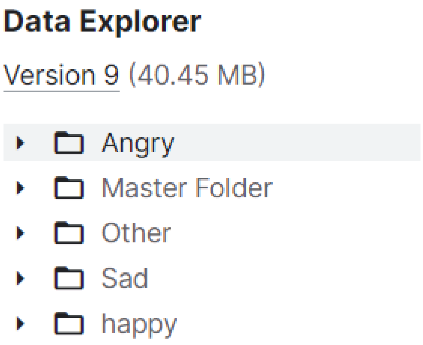
About the Dataset

This dataset contains high-quality face images of various pets, such as dogs, cats, rabbits, hamsters, sheep, horses, and birds. The images capture the diversity of expressions these animals can display, such as happiness, sadness, anger etc

Potential Use Cases

- **Facial expression recognition**: The dataset can be used to train and evaluate models that can recognize the emotions of pets from their facial expressions. This can help pet owners understand their pets better and improve their well-being.
- **Pet identification**: The dataset can be used to train and evaluate models that can identify the species and breed of pets from their face images.
- **Pet style transfer**: The dataset can be used to train and evaluate models that can transfer the style of one pet's face to another

Figure 5.19: Overview of a dataset from Kaggle



Data Explorer

Version 9 (40.45 MB)

- Angry
- Master Folder
- Other
- Sad
- happy

Figure 5.20: Structure of the dataset from Kaggle

to be able to structure the data differently depending on the task. So data is usually structured for training, and often if you want to visualize or explain that data, you need to restructure it which is difficult. (P20) Storing data in a data model adds structure to the data, but also allows users to easily reconstruct the data by picking certain attributes. Adding structure to the data also makes it easier to combine and compare datasets. *That would be very nice to combine and compare datasets. It opens a lot of new research opportunities: if they have the same schema, they probably are very comparable. You could query what is the data of the Netherlands and Belgium, and compare them in different regards. (P2)*; *I think it would be useful because if it is structured then you can easily compare different datasets before you're even loading it. (P6)* This also makes it easier to reuse datasets on different methods due to the modularity. *Data that is structured it's of course easier to reuse data sets on different methods because you can make them more modular. (P2)*

- **Defining explainability, fairness, and robustness** One of the most reoccurring problems was that there is no clear definition of explainability, fairness, and robustness (P7, P9, P15, P16, P17). These factors can be quite subjective and we do not have ways to measure any of these factors. There is no clear definition of what is meant by 'the model is understandable' (P2). *For explainability, because explainability happens now by we thinking by ourselves that well. Is this explainable? Is this unexplainable? We have a lot of people that have different perceptions. We understand some people might feel this is explainable. Some people might feel this is not explainable. (P16)*; There is no metric which can measure explainability. The data model serves the purpose of allowing researchers to compare what the model has learnt to what it was expected to learn, at a global and local level, thus adding some measure of explainability. There is also no standard definition of fairness in CV. *In fairness, in machine learning, I think it's also very much debated what fairness means. (P7)*. When speaking with Participant P14, he/she said that one of the struggles of the task was to come to a clear conclusion of what concepts in the image were considered to be protected attributes - some might say that a church is a protected attribute because it may reveal the person's religion, while others may disagree with this. The authors of datasets and models are asked to fill in such information in the data model, thus having a clear definition for each of these factors at least for the given dataset or model. The data model also serves as a platform where dataset creators can share this information giving other users more information about the dataset and its uses.

- **Documentation, traceability, and validation** Participants proposed that the data model can be used for documentation and traceability purposes. Users can manage their work and keep track of decisions they have made (P1, P13). *Just a machine learning model training. Just managing that, making sure there's traceability and for documentation purposes as well. If you've got all the stuff in one place, you can see how you got to a certain place and see what you've done. (P13)* Having all the data in one place also makes it easier to reimplement studies, especially since the data model contains all the information that was used for a classification task. This makes it easy to reproduce studies. *It's beneficial to say, like you have a database or my experiments and with this database maybe others can relatively easily reimplement my study. (P1)* He/She also found that it can be useful to help researchers organize data and make conclusions about experiments. *To say help explainable AI researchers to organize data or say like make consistent study conclusions and stuff. (P1)* The data stored in the data model can also help researchers validate other studies. Many papers tend to show results of only a few good examples and omit to mention the downsides of the model. Having all the information necessary to reproduce a study can allow users to explore and validate new methods themselves. *I think what I would do if I have this I will then build the benchmark to validate some of the most popular exponent methods to see what exactly it explains what are the things they you know leave out right. (P12)* Furthermore, this is a rapidly developing field, so it becomes difficult to keep up with the new methods. *So each year there is an increasing number of papers, so it's hard to keep pace with all the things that are new because there is really an enormous amount of work because of the trend of it also. (P17)*; *'I'm a less domain-specific indeed and more CV broad and it's so broad and so much is happening, it's more difficult to keep track of everything. (P10)* Having the data model constantly updated can make it slightly easier to keep up with new research and methods related to CV.

5.2.2. Evaluation of Model Usability

In the second round of interviews, participants were asked to interact with the data model to evaluate the usability of the model. Participants had a choice of populating the database with their own dataset or extracting information from the database. We recruited two participants for the experiment and both chose to extract information from the database. The database was populated with the four datasets described in Section 3.2. Before the interviews, the participants were given access to the data model so that they could explore the data and relationships that it contains. They were also given access to the code and file structures in the case that they opted to populate the database with their own dataset. Within this section, we describe how both participants chose to interact with the model, along with the ease at which these interactions were done.

Participants' uses of the model During the first round of interviews, many participants stated that they did not see themselves using the entire data model for their tasks (P10, P18). For example, Participant P18 said that he/she will not be using the saliency maps as that is not related to his/her work. Participant P10 also said that he/she will be focusing most on the dataset, model, and samples part of the data model. This was also seen in the second round of interviews. Both participants chose to interact with different parts of the data model. Participant P21 focused on the dataset and CV model part of the data model. Figure 5.21 shows the part of the model the participant focused on during his/her interaction. As a machine learning expert, Participant P21 does a lot of data exploration before uploading the data to a cloud infrastructure. During the interview, he/she first started by exploring the data model to see what content it contains. He/She then went into searching for the datasets present in the

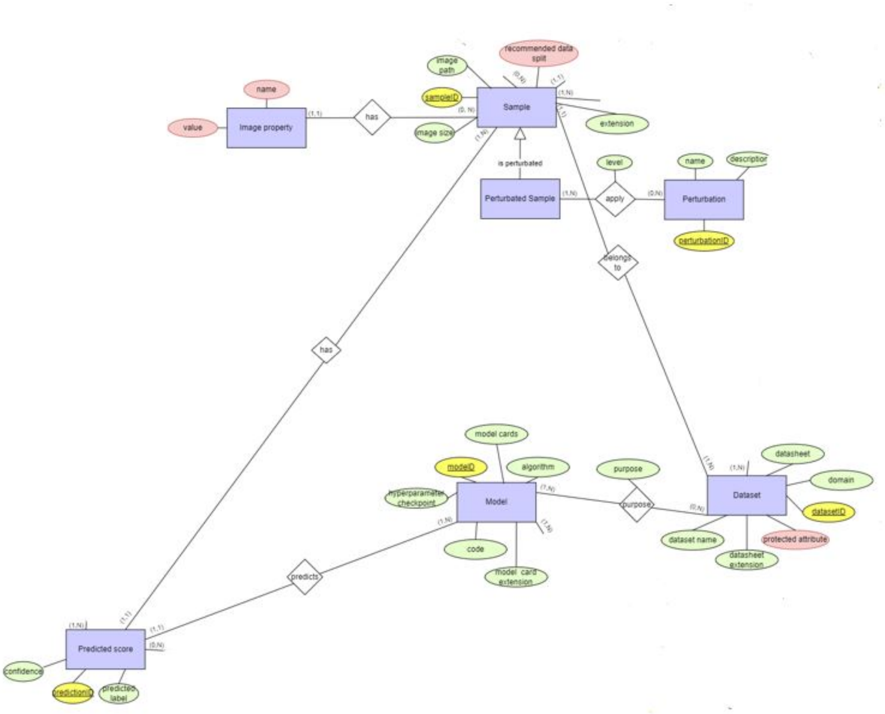


Figure 5.21: Part of data model participant P21 focused on

database, their domains, whether they are labelled, and classes of the dataset. These tasks were done by writing simple queries: `SELECT dataset_name FROM dataset;`, `SELECT class.class_name FROM dataset INNER JOIN ON dataset.dataset_id = class.dataset_id WHERE dataset_name = ’birds’;`. He/She continued exploring the images within the Birds 3.2.1 dataset. As part of his/her daily tasks, he/she writes Python scripts to extract image properties from the image, such as the size of the image and its color stream. He/She also goes through each image to check for its clarity. *Since all this information is here, I will not have to write scripts to analyze images in a dataset; This gives an overview of what I want to check; It reduces the time I spend studying the data. (P21)* He/She also mentioned that currently he/she uses Kaggle for data exploration and viewing images before downloading them, but this data model gives more information about what's in the dataset and can help save quite some time. He/She also said that they focus on data drifts so he/she envisions him-/herself using the data model to compare old data to the new data. *I can collect this data, add it to the database and then compare the data to the previous data. (P21)* He/She would also want to use the data model to join smaller datasets based on their features to make larger datasets which are necessary for the CV models he/she uses. During the interaction, he/she said that he/she would also use the data model to search for and explore CV models. There are many different versions of models, each taking in images of different sizes. Thus he/she would use the data model to explore the features and attributes of a model to find the ones that would match his/her use case and dataset. *Instead of checking for models on my own, I can just go to the database and see the available models, check how they perform, their attributes, and the hyperparameters that were used to train the model. (P21)*

Participant P22 used a majority of the data model. His/Her focus was on finding the LEMs, LLMs, and saliency maps of samples and then comparing the LLMS and LEMS to see what the model has learnt, what it was not able to learn, and what other important concepts the model learnt that were not initially in the LEMs. An example query used to find the LEMs of sample 47178 was `SELECT * FROM sample JOIN sample_mechanism ms ON ms.sample_id = sample.sample_id JOIN mechanism m ON m.mechanism_id = ms.mechanism_id JOIN type_of ON type_of.mechanism_id = m.mechanism_id JOIN mechanism_type mt ON mt.mech_type_id = type_of.mech_type_id`

`WHERE mt.mech_type = 'LEM' and sample.sample_id = 47178;`. He/She also liked that all the entities were separated. *This allows users to choose what type of explanation and information they want to extract from the data model. (P22)*

Practitioner vs Researcher It was observed that the practitioner, like Participant 21, was more interested in the aspects that were related to the machine learning pipeline and preprocessing such as the properties of the image, its size, and the models that were used on the dataset. Participant P22, who was a researcher, was more interested in the explanability part, trying to understand what the model learnt and did not learn in comparison to what we expected the model to learn. This comparison can help the participant make various conclusions about the model, such as whether it picked up on bias in the dataset, whether it was making random predictions, or the granularity at which it was making predictions. However, due to the small number of participants this was observed for, we cannot generalize this observation across all practitioners and researchers.

Difficulties faced by participants In general there was a mixed review on the ease of using the database. Participant P21 who had a low proficiency in writing SQL queries found it a little difficult to use. Throughout the interview, he/she recommended adding an interface to make it easier to use. Writing short queries where information was only required from one table was easy, but joining tables became challenging. Participant P22, on the other hand, enjoyed using the data model and writing queries. He/She is more experienced with databases and SQL, thus writing longer queries with many table joins was interesting and enjoyable. These conducted interviews yielded insights indicating that users lacking proficiency in composing SQL queries might refrain from utilizing the data model.

This was also a concern of participants from the first round of interviews. They found that the data model was restricted to people who have database experience (P1, P4, P17, P20). To extract information or even populate the database, one needs to know how to write queries. *This is restricted for people who have database experience. (P17)*; *Not everyone has knowledge base experience and knows how to write queries, so maybe use some standard APIs so that the queries are very simple and it can be used by everyone. (P1)* Participants requested to make this toolkit easier to use by adding a UI to make it more interactive and include some sort of visualization. The complexity of writing long queries may make it difficult to use and users may avoid using it. *If it's not understandable and easy to use, then it'll be difficult to broadcast it and get users to use it. (P1)*

Another concern was about the amount of data necessary to populate the database. Initially, participant P22 opted to populate the database with his/her own dataset. However, the resources to obtain the annotations were time-consuming and expensive. *It was a little difficult to populate because the concepts and mechanisms require a lot of information and it is time-consuming. I could have skipped that but it seems like the most important part of the model. (P22)*

Data model accessibility The data model can be published publicly or company-based. Both of these have their pros and cons. Publishing the data model publicly allows researchers from all around to populate it with data, extract information, and encourage transparency and reproducibility of all research. However, users may have concerns about the reliability of the data. This problem can be solved if the data model is kept in-house, where users can keep track of who is populating the database and with what information. However, keeping it at a company level also means that the database will contain less data and outside perspectives.

6
Discussion and Conclusion

In addition to the identified opportunities embraced by participants concerning the data model, this chapter delves into the concerns expressed by participants, as discussed in Section 6.1. Furthermore, we have acknowledged certain constraints within the design of the data model and the experimentation process, which are elaborated upon in Section 6.2. Throughout the interviews, participants drew comparisons between our data model and other established platforms and artifacts. These comparative insights are outlined in Section 6.3. Finally, Section 6.4 encapsulates the study's summary and concluding remarks, offering glimpses into potential future avenues for research in this domain.

6.1. Data Model Concerns

While the participants had many positive use cases for the data model, there were also a few concerns regarding this model. The concerns have been grouped into concerns regarding the data, concerns regarding the schema, and concerns for users who intend to use the data model. In this section, we will highlight the concerns brought out by the participants.

Data

- **Richness of data** A recurring concern about the data model was about the richness of the data (P1, P2, P9, P20). They believe that the database is as rich as the data it contains. *It can be useful depending on how rich the repository is. (P9)* This depends on who is populating the database and their expertise in that field. For example, a doctor annotating and labelling medical images will be of higher value than a lay worker doing the same. They are concerned about the injected bias present in the dataset and the explanations provided. *It really depends, of course, always how well the data is in there. Like I mean you you include injected bias, but like how well did you analyze the bias? Is it gonna convince me that this is the only bias and that I don't have to look for myself for any more bias? Basically the same for explanability. (P2)* Their concern is that if incorrect data is added to the database, users may become too dependent on the information provided in the database and use incorrect information in their research. This can be a concern for many researchers. Many papers showcase good results and omit the parts that did not work well. Researchers and practitioners who want to use that research must always verify it before putting it into use. This is the same for the data model. In the case that the data model is used publicly, users who intend to extract information from the data model should scan through the data first before putting it into use. However, it is also the duty of those populating the database to add correct information or truthfully fill out the 'Annotator' details so that other users know the amount of trust and reliability they can put in the data.
- **Data from different domains** Another concern regarding the data was that the data model may not be generalized to all domains (P1, P19). For example, for some tasks in the medical domain, the segmentation of images is used, which requires additional attributes to be supported by the data model. *If you would expand it to my domain, yeah. If you could make a version that has the NN unit stands for no new unit and it's the nature paper from the German Cancer Research Center.*

Consequent	Mechanism
Kitchen	fridge AND gas stove
	white fridge AND silver gas stove
	fridge AND gas stove AND oven AND 4 bar stool

Table 6.1: Mechanisms of different granularity and generalizability

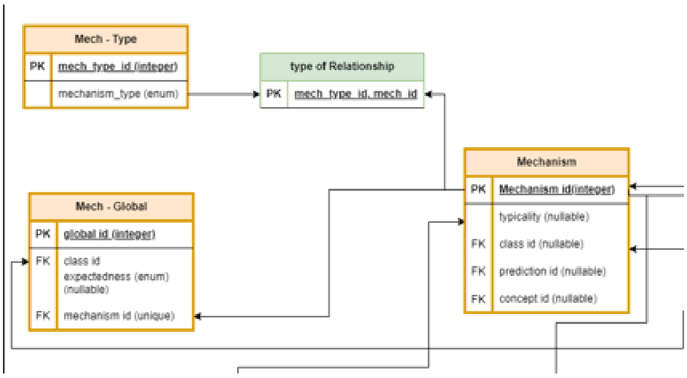


Figure 6.1: Tables representing mechanisms in the database

And it basically gives you a state-of-the-art segmentation on, like hundreds of classes. (P19) A few attributes for each domain may be necessary to add to the domain to make it more generalizable to all domains. This version of the data model was designed for CV datasets in general and no particular domain. Thus we focused on attributes that are important to CV applications in general. After publishing this version of the data model, different versions of the data model can be designed which are tailored to other domains.

- **Storage and complexity** Another concern was about the storage and complexity of the mechanisms in the data model. Mechanisms can be very specific to a dataset and each sample may have several mechanisms, depending on the granularity and generalizability of the mechanism. For example, Table 6.1 shows mechanisms for a *Kitchen*. The first mechanism is quite generalizable and may be present in all samples of a kitchen. The second and third are more specific to a certain sample of a kitchen. *But like this way I think it's very detailed and very data set specific. If you need to define some rules for each data set, I think you also store it in a very complex way. It may make it very hard to say, generalize and unify.* While this is true, including mechanisms in the data model can encourage users to annotate their data and can also provide ground truth explanations for datasets, a desire from many participants. We did try to separate entities as much as possible to avoid duplicates. For example, the first mechanism in Table 6.1 can be a GEM, GLM, LLM, and LEM. Instead of saving it four times in the database, we made a separate table for the type of mechanism, such that the mechanism is only stored once and has a relationship with each of the types of mechanism it is a type of (shown in Figure 6.1).

- **Data Collection** Having a detailed and informative table comes at a cost. Over the years, it has been discovered that data collection is expensive, and collecting annotations is even more expensive. This was a concern of participants P9 and P20. *How do you get the data? It's a lot of data. (P20)*; *Ground truth explanations are expensive. (P9)* One of the intentions of this data model was to share data. Researchers are already collecting a lot of data for their research, so populating the database with this data can help other researchers with their research.

Schema

- **Innovation** There were a few concerns about the innovation of the data model (P4, P6, P7). Other tools provide users with access to datasets, such as Kaggle [1] and Torchvision [2]. However, these do not provide users with metadata or (human-understandable) explanations. *Not sure*

[1] https://www.kaggle.com/datasets

[2] https://pytorch.org/vision/

how innovative it is because torch vision also provides you with access to data but it misses the metadata. (P6) This data model is different because it provides users with more information, not only about the dataset but also about the models used with the dataset, along with explanations that are understandable to humans. The explanations are presented at both a global and local level. Another concern was that the data model may not be useful for popular datasets because metadata about those datasets is already available. *May not be useful for readily available and frequently used datasets such as ImageNet for example, because metadata about those datasets is available all over the internet. However, a schema like this is useful for newly created datasets, where we can add all the data and keep track of all the statistics that are generated in each iteration of generating new data samples. (P4)* If metadata on popular datasets is readily available, it may not take too much of the user's time to add it to the database. Including these datasets in the database can help users make comparisons with other datasets and models used on these datasets.

- **Maintenance** A huge concern was about the maintenance of the data model. The data model can be expensive to maintain because it needs to be updated frequently. Hence the developer needs to maintain a continual engagement with research endeavors to keep the model up to date. *A structure like this needs to be updated quite frequently so the developer needs to always be checking literature and updating this. (P14)* With the constant research going on in the CV field, participants were also afraid that the data model would not exist for a long period due to the shift in research. *May be useful in the short term but not sure about the long term because the resource may or exist if the research focus has been shifted. (P9)* While these are valid, we still believe that the data model can be very useful for this current period and can guide other researchers in the direction of creating more structure, and encouraging transparency and reproducibility. We hope that the data model can be used as an avenue for future works around data modelling and machine learning applications.

Users

- **Usability** Participants also found that the data model was restricted to people who have database experience (P1, P4, P17, P20). A UI will make the data model easier to use and also allow more users to use it since not everyone who works with CV applications has database knowledge. The creation of a user interface constitutes a forthcoming task of this study. This concern is further highlighted in Section 5.2.2.

- **Relevance across different stakeholder cohorts** Explanations are not only provided for developers but they should also be provided for end users of the application. They should also be aware of how the model is making decisions to build their trust in the models. A concern by Participant P2 was that the information provided in this data model may not all be relevant to the end users (eg. doctors and lawyers) of these applications. *May not be usable to other stakeholders (medical) because doctors are not interested in global and local mechanisms or any bias. (P2)* This is also a valid concern. This version of the data model was created to aid developers with the explanability, fairness and robustness of CV applications. Expanding the data model to provide information and explanations for end users of the models remains an avenue for potential future development.

6.2. Limitations

Besides, for some of the concerns participants had for the data model, we also found some limitations throughout the process of interviews, designing the data model, and modelling it. Many of these limitations have already been addressed in previous sections of the report. In this section, we will address these limitations and some potential solutions which can be implemented in the future. We have grouped the limitations in terms of the design of the data model and the experimentation process.

Design

- **Saliency detection** As of now, the data model is limited to saliency detection as the explainability method for CV. We chose to use this method of explanation as it is commonly used for explainability in CV models [102], [19]. This data model was designed to create a platform for explainability, fairness and robustness in CV applications. Users are encouraged to tailor this data model to support their daily tasks, which may include different explanability methods. *May want to make it more generic rather than just saliency maps. (P9)* The data model is designed such that all entities are modular and detangled, which makes it easier to add or remove entities.

- **Single label dataset** The data model currently supports single-label datasets. This can be seen as a limitation to some because there are CV datasets that are multilabeled. *Single class can be a bottleneck. (P9)* One example is the Adience dataset, which has labels for age and gender. For this research, the dataset was used to train models based on gender and stored in that way as well. In the future, multilabel datasets can be supported by making a few changes to the design, for example by changing the cardinalities.
- **Modality** Participants also said that there are tasks in which multiple modalities are required to do classification. For example, Participant P3 mentioned that for one of his/her projects, they used both visual and textual data to predict data. For this research, our focus was on visual data. With more research, the data model can be extended to support multiple modalities as well.
- **Database knowledge** As previously stated, the data model is restricted to CV researchers and practitioners who have knowledge about data management. Based on the interviews, it was found that many CV professionals lack proficiency in the management of databases for the storage and retrieval of information. In the future, we hope to design a user interface that showcases all the entities and relationships of the data model. Ideally, this interface will enhance the usability of the data model for experts in the field of CV.

Experimentation

- **Scalability** While the data model can be used to store large amounts of data, we did notice that there is some temporal complexity. Table 4.1 shows the amount of time taken to load each dataset. Color MNIST took the most time (2 hours) while Adience was relatively faster, taking only 20 minutes. The scripts for populating the database can be optimized for the future, but we hope that users can use them to leverage their usage of the data model.
- **Number of participants** For the first round of interviews, we interviewed 20 participants. We found that to be enough as we reached a point of saturation where participants' suggestions and intended uses of the model were quite similar. For the second round of interviews, we only interviewed two participants, both of whom participated in extracting data from the database. This did not give us concrete on the usage of the data model, especially when populating the data model. Furthermore, we found that the practitioner preferred using the data model for proprocessing and the machine learning pipeline and the researcher was more interested in the explanations provided. However, we cannot generalize this for all practitioners and researchers. In the future, we hope to interview more participants to get an unbiased view of the different ways in which practitioners and researchers use the data model.

6.3. Comparison to Other Platforms and Artefacts

Before seeing the contents of the data model, many participants questioned its innovation in comparison to other platforms and artefacts. In this section, we compare the data model to other platforms and artefacts addressing the non-functional requirements: explainability, fairness, and robustness in CV applications.

Kaggle and HuggingFace are known as central hubs for datasets and models. TorchVision is also a popular library that is used for building and training machine learning models. These platforms store information about datasets and/or models. When describing the data model as a central hub that stores data and metadata for datasets, participants wondered what made it different from these existing platforms. The data model does not only give users access to the dataset and model but also provides them with an immense amount of metadata, such as how the data was collected, the tasks for which the dataset or model can be used, biases that may be presented in the dataset and more. The dataset overviews on these platforms contain information about the classes in the dataset, while the data model gives users further insights into the dataset such as the concepts present in the images in the datasets, parts of the images they expect models the learn, and model performances on the datasets that go beyond widely used metrics (eg. accuracy). With the data model, users can compare model performances in terms of what the model has learnt in contrast to what it was expected to learn. This is further explained in Section 6.1.

There have also been studies that focused on developing toolkits to aid with explainability [5], [53], [48], fairness [85], [17], [14] and robustness of machine learning models [45]. These toolkits act as a central hub that stores multiple methods and metrics relating to one or two of these non-functional requirements. The data model presented in this research addresses all three requirements as they require a lot of overlapping information. While these toolkits contain several methods and models, they do not contain the relationships between these models to different datasets, making it difficult to compare models for different datasets or explainability methods for CV models and datasets. This is further explained in Chapter 2.

6.4. Conclusion

The field of Computer Vision is experiencing rapid growth, with algorithmic researchers continually developing new methods to assess the explainability, fairness, and robustness of CV models. Meanwhile, researchers in Human-computer Interaction are evaluating these methods. However, our investigation has uncovered significant challenges faced by these researchers in their daily work. These challenges include the difficulty of finding and accessing relevant, comprehensive datasets and models, as well as the time and cost associated with experiments due to the absence of structured data.

Recognizing the absence of suitable tools to assist researchers in overcoming these obstacles, we designed a data model that comprehensively captures the essential information pertaining to explainability, fairness, and robustness in regard to Computer Vision. The data model emerged through a mixed-method approach, obtaining information from the literature and insights gathered through 20 semi-structured interviews with researchers.

The data model features key entities (e.g. Sample, Model, Saliency, Concept, Class) each with associated attributes. Within this structured framework with the entities and relationships between entities, researchers can use database queries to extract meaningful information. Moreover, researchers have the capability to contribute their novel methods to the database.

Our findings indicate a high level of enthusiasm among participants for utilizing this data model. Its potential applications encompass promoting transparency and reproducibility in research, speeding up the research process, and facilitating the evaluation of methods. Additionally, the model inadvertently introduced participants to the concept of explainability methods based on semantic explanations, which they found to be more interpretable for human understanding.

In the future, we plan on enhancing the model's usability, expanding its scope to encompass domain-specific use cases and diverse datasets (e.g., multi-labeled data, multiple modalities), and further validating its usability through user studies where researchers populate the database with their own datasets or extract information from the database.

Bibliography

[1] Julius Adebayo et al. “Sanity checks for saliency maps”. In: *Advances in neural information processing systems* 31 (2018) (cit. on pp. 5, 16).

[2] Mahmoud Afifi and Michael S Brown. “What else can fool deep learning? Addressing color constancy errors on deep neural network performance”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 243–252 (cit. on p. 16).

[3] Muhannad Alkaddour and Usman Tariq. “Investigating the effect of noise on facial expression recognition”. In: *Advances in Computer Vision: Proceedings of the 2019 Computer Vision Conference (CVC), Volume 2 1*. Springer. 2020, pp. 699–709 (cit. on p. 16).

[4] Ariful Islam Anik and Andrea Bunt. “Data-centric explanations: Explaining training data of machine learning systems to promote transparency”. In: *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 2021, pp. 1–13 (cit. on pp. 1, 16, 33, 50).

[5] Vijay Arya et al. “AI Explainability 360 Toolkit”. In: *Proceedings of the 3rd ACM India Joint International Conference on Data Science & Management of Data (8th ACM IKDD CODS & 26th COMAD)*. 2021, pp. 376–379 (cit. on pp. 11, 71).

[6] Aharon Azulay and Yair Weiss. “Why do deep convolutional networks generalize so poorly to small image transformations?” In: *arXiv preprint arXiv:1805.12177* (2018) (cit. on p. 9).

[7] Agathe Balayn, Lorenzo Corti, and Jie Yang. “ARCH: A Data- and Knowledge-driven, Human-centered, Reasoning-based Tool for Diagnosing Computer Vision Models”. In: 2023 (cit. on pp. 16, 30, 31, 38–40, 48).

[8] Agathe Balayn et al. “Faulty or Ready? Handling Failures in Deep-Learning Computer Vision Models until Deployment: A Study of Practices, Challenges, and Needs”. In: *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 2023, pp. 1–20 (cit. on pp. 2, 12, 16).

[9] Agathe Balayn et al. “How can Explainability Methods be Used to Support Bug Identification in Computer Vision Models?” In: *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. 2022, pp. 1–16 (cit. on pp. 2, 13, 16, 17, 53).

[10] Agathe Balayn et al. “Ready Player One! Eliciting Diverse Knowledge Using A Configurable Game”. In: *Proceedings of the ACM Web Conference 2022*. 2022, pp. 1709–1719 (cit. on p. 16).

[11] Agathe Balayn et al. “What do you mean? Interpreting image classification with crowdsourced concept extraction and analysis”. In: *Proceedings of the Web Conference 2021*. 2021, pp. 1937–1948 (cit. on pp. 8, 16, 17, 30, 36, 48).

[12] Gabriel Diniz Junqueira Barbosa et al. “Investigating the relationships between class probabilities and users’ appropriate trust in computer vision classifications of ambiguous images”. In: *Journal of Computer Languages* 72 (2022), p. 101149 (cit. on p. 5).

[13] Solon Barocas, Moritz Hardt, and Arvind Narayanan. “Fairness in machine learning”. In: *Nips tutorial* 1 (2017), p. 2017 (cit. on p. 16).

[14] Rachel KE Bellamy et al. “AI Fairness 360: An extensible toolkit for detecting and mitigating algorithmic bias”. In: *IBM Journal of Research and Development* 63.4/5 (2019), pp. 4–1 (cit. on pp. 1, 2, 12, 71).

[15] Cynthia Bennett and Os Keyes. “What is the point of fairness?” In: *Interactions* 27.3 (2020), pp. 35–39 (cit. on p. 16).

[16] Srinadh Bhojanapalli et al. “Understanding Robustness of Transformers for Image Classification”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. Oct. 2021, pp. 10231–10241 (cit. on p. 16).

[17] Sarah Bird et al. “Fairlearn: A toolkit for assessing and improving fairness in AI”. In: *Microsoft, Tech. Rep. MSR-TR-2020-32* (2020) (cit. on pp. 1, 12, 71).

[18] Shreyan Biswas et al. “CHIME: Causal Human-in-the-Loop Model Explanations”. In: *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*. Vol. 10. 1. 2022, pp. 27–39 (cit. on p. 16).

[19] Angie Boggust et al. “Saliency Cards: A Framework to Characterize and Compare Saliency Methods”. In: *Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency*. 2023, pp. 285–296 (cit. on pp. 1, 5, 11, 16, 27, 37, 42, 45, 46, 48, 70).

[20] Angie Boggust et al. “Shared interest: Measuring human-AI alignment to identify recurring patterns in model behavior”. In: *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. 2022, pp. 1–17 (cit. on pp. 16, 39).

[21] Tolga Bolukbasi et al. “Man is to computer programmer as woman is to homemaker? debiasing word embeddings”. In: *Advances in neural information processing systems* 29 (2016) (cit. on p. 8).

[22] Caridad F Brito. “Demonstrating experimenter and participant bias.” In: (2017) (cit. on p. 9).

[23] Lennart Brocki and Neo Christopher Chung. “Concept saliency maps to visualize relevant features in deep generative models”. In: *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*. IEEE. 2019, pp. 1771–1778 (cit. on p. 41).

[24] Michael L Brodie. “On the development of data models”. In: *On Conceptual Modelling: Perspectives from Artificial Intelligence, Databases, and Programming Languages*. Springer, 1984, pp. 19–47 (cit. on p. 13).

[25] Joy Buolamwini and Timnit Gebru. “Gender shades: Intersectional accuracy disparities in commercial gender classification”. In: *Conference on fairness, accountability and transparency*. PMLR. 2018, pp. 77–91 (cit. on pp. 1, 9, 16, 19, 29, 31, 33–35, 39, 41).

[26] Alexandra Chouldechova and Aaron Roth. “The frontiers of fairness in machine learning”. In: *arXiv preprint arXiv:1810.08810* (2018) (cit. on p. 16).

[27] Ekin D. Cubuk et al. “AutoAugment: Learning Augmentation Strategies From Data”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2019 (cit. on p. 16).

[28] Piotr Dabkowski and Yarin Gal. “Real time image saliency for black box classifiers”. In: *Advances in neural information processing systems* 30 (2017) (cit. on p. 16).

[29] Roxana Daneshjou et al. “Lack of transparency and potential bias in artificial intelligence data sets and algorithms: a scoping review”. In: *JAMA dermatology* 157.11 (2021), pp. 1362–1369 (cit. on p. 16).

[30] Kanjar De and Marius Pedersen. “Impact of Colour on Robustness of Deep Neural Networks”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*. Oct. 2021, pp. 21–30 (cit. on p. 16).

[31] Derek Doran, Sarah Schulz, and Tarek R Besold. “What does explainable AI really mean? A new conceptualization of perspectives”. In: *arXiv preprint arXiv:1710.00794* (2017) (cit. on p. 1).

[32] Nathan Drenkow et al. “A systematic review of robustness in deep learning for computer vision: Mind the gap?” In: *arXiv preprint arXiv:2112.00639* (2021) (cit. on p. 16).

[33] Cynthia Dwork et al. “Decoupled classifiers for group-fair and efficient machine learning”. In: *Conference on fairness, accountability and transparency*. PMLR. 2018, pp. 119–133 (cit. on p. 16).

[34] Eran Eidinger, Roee Enbar, and Tal Hassner. “Age and Gender Estimation of Unfiltered Faces”. In: *IEEE Transactions on Information Forensics and Security* 9.12 (2014), pp. 2170–2179. DOI: `10.1109/TIFS.2014.2359646` (cit. on pp. 16, 19).

[35] Andre Esteva et al. “Dermatologist-level classification of skin cancer with deep neural networks”. In: *nature* (2017) (cit. on p. 1).

[36] Clare Garvie. *The perpetual line-up: Unregulated police face recognition in America*. Georgetown Law, Center on Privacy & Technology, 2016 (cit. on p. 8).

[37] Timnit Gebru et al. “Datasheets for datasets”. In: *Communications of the ACM* 64.12 (2021), pp. 86–92 (cit. on pp. 1, 2, 10, 11, 16, 27, 29, 33, 42, 43, 48, 52).

[38] Robert Geirhos et al. “ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness”. In: *arXiv preprint arXiv:1811.12231* (2018) (cit. on pp. 16, 34).

[39] Amirata Ghorbani et al. “Towards automatic concept-based explanations”. In: *Advances in Neural Information Processing Systems* 32 (2019) (cit. on pp. 7, 8, 16, 29, 34, 39, 53, 54).

[40] Naman Goel, Mohammad Yaghini, and Boi Faltings. “Non-discriminatory machine learning through convex fairness criteria”. In: *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*. 2018, pp. 116–116 (cit. on p. 16).

[41] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. “Explaining and harnessing adversarial examples”. In: *arXiv preprint arXiv:1412.6572* (2014) (cit. on p. 9).

[42] Nina Grgic-Hlaca et al. “The case for process fairness in learning: Feature selection for fair decision making”. In: *NIPS symposium on machine learning and the law*. Vol. 1. 2. Barcelona, Spain. 2016, p. 11 (cit. on p. 10).

[43] Cornelia Gyorödi, Robert Gyorödi, and Roxana Sotoc. “A comparative study of relational and non-relational database models in a Web-based application”. In: *International Journal of Advanced Computer Science and Applications* 6.11 (2015), pp. 78–83 (cit. on p. 27).

[44] Benjamin Haibe-Kains et al. “Transparency and reproducibility in artificial intelligence”. In: *Nature* 586.7829 (2020), E14–E16 (cit. on p. 10).

[45] Michaela Hardt et al. “Amazon sagemaker clarify: Machine learning bias detection and explainability in the cloud”. In: *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 2021, pp. 2974–2983 (cit. on pp. 1, 12, 71).

[46] Moritz Hardt, Eric Price, and Nati Srebro. “Equality of opportunity in supervised learning”. In: *Advances in neural information processing systems* 29 (2016) (cit. on p. 16).

[47] Josh Harguess, Diego Marez, and Nancy Ronquillo. “An investigation into strategies to improve optical flow on degraded data”. In: *Geospatial Informatics, Motion Imagery, and Network Analytics VIII*. Vol. 10645. SPIE. 2018, pp. 110–119 (cit. on p. 16).

[48] Anna Hedström et al. “Quantus: An explainable ai toolkit for responsible evaluation of neural network explanations and beyond”. In: *Journal of Machine Learning Research* 24.34 (2023), pp. 1–11 (cit. on pp. 2, 11, 71).

[49] Thomas Hellström, Virginia Dignum, and Suna Bensch. “Bias in Machine Learning–What is it Good for?” In: *arXiv preprint arXiv:2004.00686* (2020) (cit. on p. 58).

[50] Dan Hendrycks and Thomas Dietterich. “Benchmarking neural network robustness to common corruptions and perturbations”. In: *arXiv preprint arXiv:1903.12261* (2019) (cit. on pp. 1, 9, 35, 41).

[51] Kenneth Holstein et al. “Improving fairness in machine learning systems: What do industry practitioners need?” In: *Proceedings of the 2019 CHI conference on human factors in computing systems*. 2019, pp. 1–16 (cit. on pp. 2, 13).

[52] Andreas Holzinger, André Carrington, and Heimo Müller. “Measuring the quality of explanations: the system causability scale (SCS) comparing human and machine explanations”. In: *KI-Künstliche Intelligenz* 34.2 (2020), pp. 193–198 (cit. on p. 16).

[53] Brian Hu et al. “XAITK: The explainable AI toolkit”. In: *Applied AI Letters* 2.4 (2021), e40 (cit. on pp. 11, 71).

[54] Lingxiao Huang and Nisheeth Vishnoi. “Stable and fair classification”. In: *International Conference on Machine Learning*. PMLR. 2019, pp. 2879–2890 (cit. on p. 16).

[55] Yixing Huang et al. “Some investigations on robustness of deep learning in limited angle tomography”. In: *Medical Image Computing and Computer Assisted Intervention–MICCAI 2018: 21st International Conference, Granada, Spain, September 16-20, 2018, Proceedings, Part I*. Springer. 2018, pp. 145–153 (cit. on p. 16).

[56] Shashank Mohan Jain. “Hugging face”. In: *Introduction to Transformers for NLP: With the Hugging Face Library and Models to Solve Problems*. Springer, 2022, pp. 51–67 (cit. on p. 12).

[57] Ray Jiang et al. “Wasserstein fair classification”. In: *Uncertainty in artificial intelligence*. PMLR. 2020, pp. 862–872 (cit. on p. 16).

[58] Jungseock Joo and Kimmo Kärkkäinen. “Gender slopes: Counterfactual fairness for computer vision models by attribute manipulation”. In: *Proceedings of the 2nd international workshop on fairness, accountability, transparency and ethics in multimedia*. 2020, pp. 1–5 (cit. on pp. 1, 16, 20).

[59] Faisal Kamiran and Toon Calders. “Classification with no discrimination by preferential sampling”. In: *Proc. 19th Machine Learning Conf. Belgium and The Netherlands*. Vol. 1. 6. Citeseer. 2010 (cit. on p. 16).

[60] Andrei Kapishnikov et al. “Xrai: Better attributions through regions”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 4948–4957 (cit. on pp. 5–7, 16).

[61] Been Kim et al. “Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav)”. In: *International conference on machine learning*. PMLR. 2018, pp. 2668–2677 (cit. on pp. 7, 16, 31, 34, 40).

[62] Jang-Hyun Kim, Wonho Choo, and Hyun Oh Song. “Puzzle mix: Exploiting saliency and local statistics for optimal mixup”. In: *International Conference on Machine Learning*. PMLR. 2020, pp. 5275–5285 (cit. on p. 16).

[63] Emmanouil Krasanakis et al. “Adaptive sensitive reweighting to mitigate bias in fairness-aware classification”. In: *Proceedings of the 2018 world wide web conference*. 2018, pp. 853–862 (cit. on p. 16).

[64] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. “Imagenet classification with deep convolutional neural networks”. In: *Advances in neural information processing systems* 25 (2012) (cit. on p. 34).

[65] Alfred Laugros, Alice Caplier, and Matthieu Ospici. “Addressing neural network robustness with mixup and targeted labeling adversarial training”. In: *Computer Vision–ECCV 2020 Workshops: Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16*. Springer. 2020, pp. 178–195 (cit. on p. 16).

[66] Alfred Laugros, Alice Caplier, and Matthieu Ospici. “Are adversarial robustness and common perturbation robustness independant attributes?” In: *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*. 2019, pp. 0–0 (cit. on p. 16).

[67] Heeseok Lee. “Justifying database normalization: a cost/benefit model”. In: *Information processing & management* 31.1 (1995), pp. 59–67 (cit. on p. 50).

[68] Jin-Ha Lee et al. “Smoothmix: a simple yet effective data augmentation to train robust classifiers”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*. 2020, pp. 756–757 (cit. on p. 16).

[69] Yi Li and Nuno Vasconcelos. “Repair: Removing representation bias by dataset resampling”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 9572–9581 (cit. on pp. 16, 19).

[70] Wei Liu et al. “Ssd: Single shot multibox detector”. In: *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*. Springer. 2016, pp. 21–37 (cit. on p. 18).

[71] Jonathan Long, Evan Shelhamer, and Trevor Darrell. “Fully convolutional networks for semantic segmentation”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 3431–3440 (cit. on p. 34).

[72] Ninareh Mehrabi et al. “A survey on bias and fairness in machine learning”. In: *ACM computing surveys (CSUR)* 54.6 (2021), pp. 1–35 (cit. on p. 16).

[73] Christian Meske and Enrico Bunde. “Transparency and trust in human-AI-interaction: The role of model-agnostic explanations in computer vision-based decision support”. In: *Artificial Intelligence in HCI: First International Conference, AI-HCI 2020, Held as Part of the 22nd HCI International Conference, HCII 2020, Copenhagen, Denmark, July 19–24, 2020, Proceedings 22*. Springer. 2020, pp. 54–69 (cit. on pp. 1, 5).

[74] Tim Miller. “Explanation in artificial intelligence: Insights from the Social Sciences”. In: *Artificial Intelligence* 267 (Oct. 2018), pp. 1–38. DOI: `10.1016/j.artint.2018.07.007` (cit. on pp. 16, 29, 34).

[75] Margaret Mitchell et al. “Model cards for model reporting”. In: *Proceedings of the conference on fairness, accountability, and transparency*. 2019, pp. 220–229 (cit. on pp. 1, 11, 16, 27, 30, 37, 42, 44, 45, 48, 52).

[76] Sina Mohseni, Jeremy E Block, and Eric Ragan. “Quantitative evaluation of machine learning explanations: A human-grounded benchmark”. In: *26th International Conference on Intelligent User Interfaces*. 2021, pp. 22–31 (cit. on p. 16).

[77] Norman Mu and Justin Gilmer. “Mnist-c: A robustness benchmark for computer vision”. In: *arXiv preprint arXiv:1906.02337* (2019) (cit. on pp. 1, 9, 16, 35).

[78] Tiago Santana de Nazare et al. “Color quantization in transfer learning and noisy scenarios: an empirical analysis using convolutional networks”. In: *2018 31st SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*. IEEE. 2018, pp. 377–383 (cit. on pp. 16, 35).

[79] Isar Nejadgholi et al. “Towards Procedural Fairness: Uncovering Biases in How a Toxic Language Classifier Uses Sentiment Information”. In: *arXiv preprint arXiv:2210.10689* (2022) (cit. on p. 10).

[80] David W. Nickerson and Todd Rogers. “Political Campaigns and Big Data”. In: *Journal of Economic Perspectives* 28.2 (May 2014), pp. 51–74. DOI: `10.1257/jep.28.2.51`. URL: `https://www.aeaweb.org/articles?id=10.1257/jep.28.2.51` (cit. on p. 1).

[81] Kexin Pei et al. “Towards practical verification of machine learning: The case of computer vision systems”. In: *arXiv preprint arXiv:1712.01785* (2017) (cit. on p. 9).

[82] Vikram V Ramaswamy, Sunnie SY Kim, and Olga Russakovsky. “Fair attribute classification through latent space de-biasing”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, pp. 9301–9310 (cit. on p. 9).

[83] Marco Tulio Ribeiro, Sameer Singh, and Carlos 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*. 2016, pp. 1135–1144 (cit. on pp. 1, 5, 6, 16, 18, 19, 30, 37).

[84] Lawrence A Rowe and Michael Stonebraker. “The POSTGRES Data Model.” In: *vldb*. Vol. 87. 1987, pp. 83–95 (cit. on p. 13).

[85] Pedro Saleiro et al. “Aequitas: A bias and fairness audit toolkit”. In: *arXiv preprint arXiv:1811.05577* (2018) (cit. on pp. 2, 12, 71).

[86] Ramprasaath R Selvaraju et al. “Grad-cam: Visual explanations from deep networks via gradient-based localization”. In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 618–626 (cit. on pp. 5, 16).

[87] Ruoxi Shang, KJ Kevin Feng, and Chirag Shah. “Why am I not seeing it? Understanding users’ needs for counterfactual explanations in everyday recommendations”. In: *2022 ACM Conference on Fairness, Accountability, and Transparency*. 2022, pp. 1330–1340 (cit. on p. 16).

[88] Shahin Sharifi Noorian et al. “What Should You Know? A Human-In-the-Loop Approach to Unknown Unknowns Characterization in Image Recognition”. In: *Proceedings of the ACM Web Conference 2022*. 2022, pp. 882–892 (cit. on pp. 16, 30, 31, 36, 38, 40).

[89] Shubham Sharma, Jette Henderson, and Joydeep Ghosh. "Certifai: A common framework to provide explanations and analyse the fairness and robustness of black-box models". In: *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*. 2020, pp. 166–172 (cit. on p. 5).

[90] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. "Deep inside convolutional networks: Visualising image classification models and saliency maps". In: *arXiv preprint arXiv:1312.6034* (2013) (cit. on p. 16).

[91] Richa Singh, Mayank Vatsa, and Nalini Ratha. "Trustworthy ai". In: *Proceedings of the 3rd ACM India Joint International Conference on Data Science & Management of Data (8th ACM IKDD CODS & 26th COMAD)*. 2021, pp. 449–453 (cit. on p. 5).

[92] Vivek K Singh et al. "Female librarians and male computer programmers? Gender bias in occupational images on digital media platforms". In: *Journal of the Association for Information Science and Technology* 71.11 (2020), pp. 1281–1294 (cit. on p. 9).

[93] Daniel Smilkov et al. "Smoothgrad: removing noise by adding noise". In: *arXiv preprint arXiv:1706.03825* (2017) (cit. on pp. 5, 17).

[94] Il-Yeol Song, Mary Evans, and Eun K Park. "A comparative analysis of entity-relationship diagrams". In: *Journal of Computer and Software Engineering* 3.4 (1995), pp. 427–459 (cit. on p. 13).

[95] Jost Tobias Springenberg et al. "Striving for simplicity: The all convolutional net". In: *arXiv preprint arXiv:1412.6806* (2014) (cit. on p. 19).

[96] Rod Stephens. *Beginning database design solutions*. John Wiley & Sons, 2009 (cit. on p. 50).

[97] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. "Axiomatic attribution for deep networks". In: *International conference on machine learning*. PMLR. 2017, pp. 3319–3328 (cit. on p. 6).

[98] Christian Szegedy et al. "Intriguing properties of neural networks". In: *arXiv preprint arXiv:1312.6199* (2013) (cit. on p. 9).

[99] Christian Szegedy et al. "Rethinking the inception architecture for computer vision". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 2818–2826 (cit. on pp. 17–19).

[100] Mark Theunissen and Jacob Browning. "Putting explainable AI in context: institutional explanations for medical AI". In: *Ethics and Information Technology* 24.2 (2022), p. 23 (cit. on pp. 39, 49).

[101] Minghui Tian, Shouhong Wan, and Lihua Yue. "A color saliency model for salient objects detection in natural scenes". In: *Advances in Multimedia Modeling: 16th International Multimedia Modeling Conference, MMM 2010, Chongqing, China, January 6-8, 2010. Proceedings 16*. Springer. 2010, pp. 240–250 (cit. on p. 37).

[102] Inam Ullah et al. "A brief survey of visual saliency detection". In: *Multimedia Tools and Applications* 79 (2020), pp. 34605–34645 (cit. on p. 70).

[103] Berk Ustun, Yang Liu, and David Parkes. "Fairness without harm: Decoupled classifiers with preference guarantees". In: *International Conference on Machine Learning*. PMLR. 2019, pp. 6373–6382 (cit. on p. 16).

[104] Himanshu Verma et al. "Rethinking the role of AI with physicians in oncology: revealing perspectives from clinical and research workflows". In: *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 2023, pp. 1–19 (cit. on pp. 2, 13).

[105] Tom Viering et al. "How to manipulate cnns to make them lie: the gradcam case". In: *arXiv preprint arXiv:1907.10901* (2019) (cit. on p. 45).

[106] Shunxin Wang, Raymond Veldhuis, and Nicola Strisciuglio. "The Robustness of Computer Vision Models against Common Corruptions: a Survey". In: *arXiv preprint arXiv:2305.06024* (2023) (cit. on p. 16).

[107] Xiang Wang et al. "Multi-view stereo in the deep learning era: A comprehensive review". In: *Displays* 70 (2021), p. 102102 (cit. on p. 1).

[108] Zeyu Wang et al. "Towards fairness in visual recognition: Effective strategies for bias mitigation". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 8919–8928 (cit. on pp. 1, 9, 16, 29, 34, 35).

[109] Blake Woodworth et al. "Learning non-discriminatory predictors". In: *Conference on Learning Theory*. PMLR. 2017, pp. 1920–1953 (cit. on p. 16).

[110] Yongkai Wu, Lu Zhang, and Xintao Wu. "Fairness-aware classification: Criterion, convexity, and bounds". In: *arXiv preprint arXiv:1809.04737* (2018) (cit. on p. 16).

[111] Tian Xu et al. "Investigating bias and fairness in facial expression recognition". In: *Computer Vision–ECCV 2020 Workshops: Glasgow, UK, August 23–28, 2020, Proceedings, Part VI 16*. Springer. 2020, pp. 506–523 (cit. on p. 16).

[112] Yu Yang et al. "Enhancing fairness in face detection in computer vision systems by demographic bias mitigation". In: *Proceedings of the 2022 AAAI/ACM Conference on AI, Ethics, and Society*. 2022, pp. 813–822 (cit. on p. 16).

[113] Roshan Reddy Yedla and Shiv Ram Dubey. "On the performance of convolutional neural networks under high and low frequency information". In: *Computer Vision and Image Processing: 5th International Conference, CVIP 2020, Prayagraj, India, December 4-6, 2020, Revised Selected Papers, Part III 5*. Springer. 2021, pp. 214–224 (cit. on p. 16).

[114] Chih-Kuan Yeh et al. "On completeness-aware concept-based explanations in deep neural networks". In: *Advances in neural information processing systems* 33 (2020), pp. 20554–20565 (cit. on pp. 7, 16).

[115] Dong Yin et al. "A fourier perspective on model robustness in computer vision". In: *Advances in Neural Information Processing Systems* 32 (2019) (cit. on p. 16).

[116] Muhammad Bilal Zafar et al. "Fairness constraints: Mechanisms for fair classification". In: *Artificial intelligence and statistics*. PMLR. 2017, pp. 962–970 (cit. on p. 16).

[117] Matthew D Zeiler and Rob Fergus. "Visualizing and understanding convolutional networks". In: *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13*. Springer. 2014, pp. 818–833 (cit. on pp. 16, 31, 41).

[118] Songtao Zhang et al. "Corruption-robust enhancement of deep neural networks for classification of peripheral blood smear images". In: *Medical Image Computing and Computer Assisted Intervention–MICCAI 2020: 23rd International Conference, Lima, Peru, October 4–8, 2020, Proceedings, Part V 23*. Springer. 2020, pp. 372–381 (cit. on p. 16).

[119] Stephan Zheng et al. "Improving the robustness of deep neural networks via stability training". In: *Proceedings of the ieee conference on computer vision and pattern recognition*. 2016, pp. 4480–4488 (cit. on p. 16).

[120] Bolei Zhou et al. "Interpretable basis decomposition for visual explanation". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 119–134 (cit. on pp. 16, 29, 31, 34, 41).