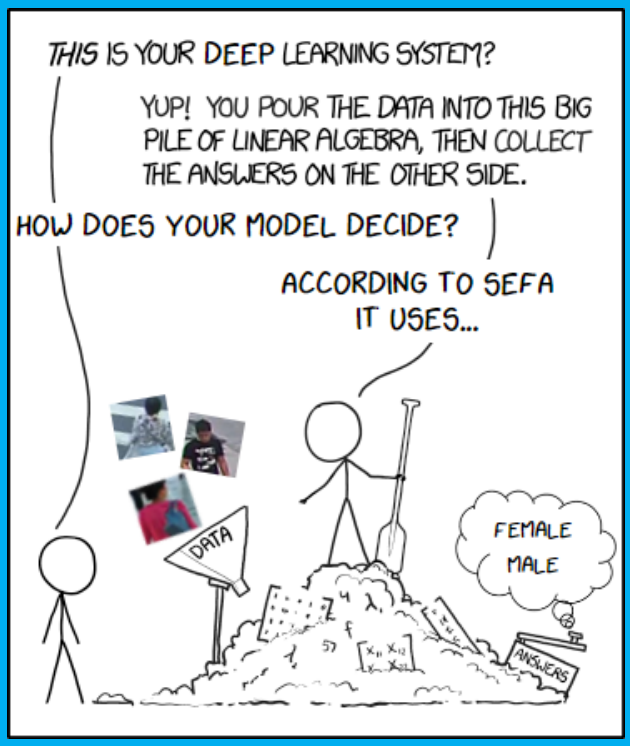
Global Interpretation of Image Classification Models via SEmantic Feature Analysis (SEFA)

Panagiotis Soilis



TU Delft

Global Interpretation of Image Classification Models via SEmantic Feature Analysis (SEFA)

Thesis

to obtain the degree of Master in Computer Science with Specialization in Data Science at the Delft University of Technology,
to defend in public on Wednesday 26 August 2020 at 11:00

by

Panagiotis Soilis
born in Athens, Greece.

Web Information Systems
Department of Software Technology
Faculty EEMCS, Delft University of Technology
Delft, Netherlands

Student number:	4921453
Project duration:	November 13, 2019 - August 26, 2020

Thesis committee:

Chair:	Prof. Dr. Alessandro Bozzon, Faculty EEMCS, TU Delft
University supervisor:	Prof. Dr. Alessandro Bozzon, Faculty EEMCS, TU Delft
Committee Member:	Dr. C. Lofi, Faculty EEMCS, TU Delft
Committee Member:	Dr. J. van Gemert, Faculty EEMCS, TU Delft

An electronic version of this thesis is available at
http://repository.tudelft.nl/.

[1] https://xkcd.com/1838/

Preface

Deep learning models have achieved state-of-the-art performance on several image classification tasks over the past years. Several studies claim to approach or even surpass human-levels of performance when using such models to classify images. However, these architectures are notoriously complex, thus making their interpretation a challenge. This limited interpretability, in turn, leads to several issues, such as restricting their applicability to critical domains like health care and finance.

Several methods in literature attempt to address this issue by providing local explanations which describe individual predictions or global ones that explain the model behaviour for a specific class. When focusing on global methods, we notice that they are limited with respect to the interpretability queries that they answer. For instance, consider we want to query whether the simultaneous presence of two objects is associated with predicting a specific class. To the best of our knowledge, there is no existing method that can tackle such a query type due to their limited expressivity. In this thesis, we address this limitation by answering the following research question: to what extent can image classification models be interpreted by analysing semantic features extracted from groups of salient image pixels?

We begin our study by investigating existing research work to devise the ideal characteristics that an interpretability method should adhere to. Our analysis highlights the aforementioned gap regarding the query complexity that existing methods cover. To address this limitation, we propose a new global interpretability method called SEmantic Feature Analysis (SEFA). To elaborate, it combines explanations of individual image predictions with semantic descriptions provided by human annotators about them, thus extracting the aforementioned semantic features. We argue that by analysing a structured data representation extracted out of semantic features will allow us to answer a wider range of interpretability queries compared to existing methods. The proposed method poses several challenges, such as identifying the number of image annotations required to obtain reliable results at a reasonable annotation cost.

Our results show that SEFA provides its users with the flexibility to answer several types of interpretability queries, including the ones that we found existing methods to be lacking. Further experimentation on its hyperparameters using three separate image classification tasks provides us with a set of suggested settings that one should use on similar datasets. Finally, we showcase the ability of SEFA to output semantic features relevant to the model classification behaviour by fine-tuning existing model architectures on biased datasets and evaluating whether the salient semantic features output describe the previous bias.

Panagiotis Soilis
Delft, August 2020

Acknowledgements

I would like to acknowledge everyone that worked with me towards completing my thesis over the past year. I am certain that without the support of many people along the way, my thesis would have remained just a vague idea.

In particular, I would like to thank my thesis supervisor Alessandro Bozzon for his precious feedback. Without his honest and direct approach, I would have been unable to pinpoint the areas I need to improve upon and how to better present my work. Moreover, I want to wholeheartedly thank Agathe Balayn for her time, willingness and patience throughout my thesis. She was there for me on a daily basis to provide structure to my thoughts, discuss my progress and to give me feedback. I also wish to acknowledge the other members of the committee, Christoph Lofi and Jan van Gemert, without whom I would not be able to graduate. Especially Christoph Lofi was present in several of my presentations giving me valuable feedback and input for my research work.

I also wish to thank my friends from university for making the last nine months a lot more fun. The study groups and breaks we had together helped me stay motivated throughout the whole thesis.

Finally, I would like to express my profound gratitude to my family to whom I owe my whole master experience. Without their help and support, I would not be able to embark on this wonderful knowledge journey.

Thank you.

Panagiotis Soilis
Delft, Netherlands
August 2020

Contents

List of Figures

List of Tables

1

Introduction

The use of Artificial Intelligence (AI) techniques and Machine Learning (ML) models is becoming more and more prominent in our daily lives [2]. A wide range of systems performing tasks such as classification and regression is used in several domains, namely commerce, legal and more. However, many of these models operate in a complex non-transparent way often referred to as "black box" [2] [24]. This lack of interpretability creates numerous issues. These include leading to unnecessary waste of model training effort [59], limiting the use of such methods in critical domains [29] like banking, leading to legal issues [23] [24] and more.

The aforementioned issues constitute interpretability as one of the essential characteristics of an ML system, both in research [2] and industry [22]. Interpretability within the ML field refers to explaining the model behaviour "in understandable terms to a human" [19]. Hence, understanding these behaviours would allow us to optimize the model training process, reduce the scepticism towards complex models in critical domains and provide the interpretability needed to avoid legal issues. For instance, being able to explain why a model classifies a CT scan image as "heart disease" would allow us to reason about the validity of the decision-making process and to build trust. While numerous methods have been proposed in literature, both to interpret existing systems and to create new models that have interpretability as a built-in component, several challenges remain.

One of these challenges is to explain the behaviour of state-of-the-art Deep Learning (DL) models used in image classification tasks [32]. In particular, existing DL studies claim to approach [18] or even surpass [26] human-levels of performance when classifying image data. However, these architectures are notoriously difficult to interpret meaning that we do not fully understand how they achieve such a high performance. For example, do they learn how a specific class is described in the real world or do they simply overfit on some dataset-specific bias? Dataset bias can correspond, for instance, to having specific background colours for each class, thus leading the model to classify images solely based on their background.

In this thesis project, we focus on interpreting the behaviour of these state-

1

of-the-art image classifiers. More specifically, we propose a new interpretability method that combines explanations of individual image predictions with semantic descriptions provided by human annotators to reason about the model for a class of interest. Its goal is to answer a wider range of interpretability queries regarding model behaviour compared to existing methods while using terms understandable to us humans.

The rest of the chapter is structured as follows: we first specify the problem statement of the research gap that we are addressing. Then, we define the research questions that we answer via this thesis project, we present the contributions of our work and conclude with an outline of this document.

1.1. Problem Statement

Interpretability methods can be divided into local and global depending on their goal when providing explanations. To be more exact, local interpretability refers to explaining a model's prediction for a specific sample while global interpretability corresponds to explaining a model's behaviour with respect to a specific class of interest. In order to make this distinction more clear, we provide a series of images from a gender classification task in Figure 1.1.

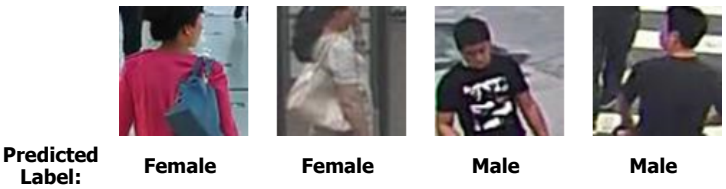


Figure 1.1: Image examples - Gender classification.

The previous images are representative samples of pedestrian images annotated with the person's gender. They depict humans in different orientations with diverse information in the image such as the objects present, their colour, the background and more. When training a deep learning model on this dataset, a few examples of interpretability queries that we would like to answer are the following:

- **Q1:** Why does the model classify the image above as male?
- **Q2:** Does the model associate the presence of a "bag" in the image with the female class?
- **Q3:** Does the model utilize the presence of "black" objects to classify as male?

Q1 is a typical example of a local interpretability method while Q2 and Q3 correspond to global interpretability queries. While the previous examples can be answered by existing methods, our literature review reveals that the current methods are unable to answer more complex questions, such as the ones available below:

- **Q4:** Is the **combination** of a "white bag" and "long hair" associated with the female class by the model?

- **Q5:** Does the model associate the **presence** of a "white bag" and **absence** of "long hair" with the male class?

Existing methods are unable to answer queries containing feature combinations (Q4) or combinations of feature presence and absence from the image (Q5). Therefore, we set out to design a method that is more expressive in that it can answer more complex global interpretability queries such as the previous ones. This new method is called **SEmantic Feature Analysis (SEFA)**. We hypothesise that obtaining groups of pixels that the model is sensitive to, on an image level, across several images and annotating those salient pixels with semantic descriptions will allow us to obtain the extra expressivity required. The idea is that we can structure the semantic description per image in a structured representation where the rows correspond to the images and the columns to the semantic data annotated. Then, we can analyse the extracted representation using traditional structured data analysis methods to answer more complex interpretability queries due to the flexibility that this structure provides.

Given that existing local interpretability methods can highlight the salient pixels on an image level (Figure 1.2), we can use them to get these groups of pixels across several images.

Figure 1.2: Local interpretability output.

Figure 1.3: Semantic features output. Left: short,black-hair|pale-ear. Middle: short,black-hair|white-road|black,white-shirt. Right: tan-neck|black-tshirt|pale-shoulder.

However, when working with image data a semantic gap arises since pixels are a too low-level representation from a semantic perspective for us humans, an issue that has been highlighted by several existing studies [29] [57] [59]. To be more specific, the highlighted pixels in these images provide no semantic meaning to humans when looked at a global interpretability level. For example, if pixel 200 is salient in the first image but not in the second one, it does not make us any wiser about the model behaviour as a whole. While such information makes sense on a local level, using the pixel numbers as features to aggregate their values globally does not provide any meaningful information about global interpretability.

To address this limitation we introduce a new notion called **semantic features**. We define them as groups of pixels that are extended with semantic meta-data. These can refer to different elements-objects in the images and their attributes-properties. Semantic features can be annotated using domain experts or potentially even crowd workers. Examples of semantic features for the previous images can be found in Figure 1.3. Moreover, we term the structured representation of images and semantic features mentioned previously as **semantic representation**.

1

1.2. Research Questions

Based on the problem statement presented in the previous section, our focus is on extending the output of local interpretability methods with semantic descriptions and analysing them to provide global explanations for image classification models. The *main research question* addressed in this work is the following:

Main Research Question

To what extent can image classification models be interpreted by analysing semantic features extracted from groups of salient image pixels?

The main idea is to develop a way to interpret image classification models by analysing semantic features extracted from local interpretations. More specifically, we hypothesise that using these semantic features to interpret an image classification model will allow us to provide more expressive global explanations that can answer more complex queries. In order to answer the main question, we break it down into the following *four research sub-questions*:

RSQ1: Which are the state-of-the-art interpretability methods for image classification models?

This question aims at finding the state-of-the-art interpretability methods and spotting their limitations with respect to global explanations. We answer this question via a detailed literature review which covers the definition of interpretability, its use cases and a categorization of existing methods. Finally, we lay out the characteristics of an ideal method, highlight the shortcomings of existing literature work according to them and showcase how SEFA addresses these limitations.

RSQ2: How can we use local interpretations to extract semantic features that allow for global interpretability?

This question refers to the methodology of extracting such semantic descriptions out of local explanations to enable for global explanations. The answer is given by presenting the SEFA methodology in detail, together with the design considerations and decisions made.

RSQ3: How do different design choices influence the method's ability to interpret the model's behaviour?

This question refers to the SEFA hyperparameters that can influence the global interpretability output. To answer this question we perform extensive experiments on three classification tasks derived from two separate datasets. The empirical evidence obtained enables us to reason about the optimal hyperparameter values.

RSQ4: To what extent does the analysis of semantic features enable us to answer a wider range of global queries for image classification models?

This question evaluates the extent to which SEFA allows us to answer more complex global interpretability queries for image classifiers compared to existing methods. The answer to it is given by applying the proposed methodology on two datasets, namely the pedestrian PA-100K [38] and the 1,000 class ImageNet ILSVRC-2012 [45], using the VGG16 [50] and Inception-V3 [55] deep learning image classifiers. A series of bias injection tests are also performed to evaluate the method's reliability and robustness.

1.3. Contributions

To summarize, the contributions of our work are the following:

C1: We provide an in-depth literature survey on image classification interpretability methods which answers **RSQ1**. In particular, we present a categorization of existing methods and discuss local interpretability methods that could be used to extract semantic features. Furthermore, we underline the limitations of existing methods compared to a set of ideal interpretability requirements. Our review highlights the challenge associated with creating an interpretability method for deep learning models due to the absence of a ground truth output. Therefore, one has to be inventive when evaluating the reliability of an interpretability method.

C2: We propose a new method which extracts semantic features from local interpretability output and structures them in a tabular representation. We then analyse this representation using existing structured data methods to answer complex global interpretability queries. This contribution addresses **RSQ2**.

C3: We implement SEFA, a system that takes trained models and a random subset of images as input and provides global explanations able to answer a wider range of queries as output. The SEFA implementation is required to answer three research questions, namely **RSQ2**, **RSQ3** and **RSQ4**.

C4: We evaluate the proposed method extensively. To elaborate, we experiment with the hyperparameter values under which it provides the optimal output and observe its behaviour in a variety of classification tasks and models to reason about the types of queries that it can answer. Contrary to existing global interpretability methods, we experiment with more than one image datasets and classification model architectures. That way, we ensure that our method can be applied to other datasets while we also analyse its behaviour in more depth with respect to its hyperparameters. For that purpose, we pre-processed the existing PA-100K [38] dataset to create a new gender classification dataset, termed as *PA-49K Gender*, which can be found on 4TU[2]. Inspired by our literature review, we came up with several bias injection experiments that can be used to evaluate the reliability of an interpretability method. The output of the SEFA evaluation is a discussion of the circumstances under which SEFA performs reliably, the types of queries that it can answer and its inherent limitations. This discussion enables us to answer research questions **RSQ3** and **RSQ4**.

[2] https://doi.org/10.4121/uuid:38dab37c-1179-495e-b357-0568b9aaaa7a

1

1.4. Outline

The rest of the thesis is structured as follows. In Chapter 2 we conduct an in-depth literature survey on image classification interpretability methods. Chapter 3 details the methodology and design choices made when creating SEFA. Following that, in Chapter 4 we report the experiments conducted to evaluate the proposed method and its hyperparameters, while we also discuss the findings arising from the empirical evidence collected. Finally, Chapter 5 concludes the thesis by summarizing our work, its limitations and proposing directions for future research. An overview of the thesis structure with respect to the research questions and contributions is available in Figure 1.4.

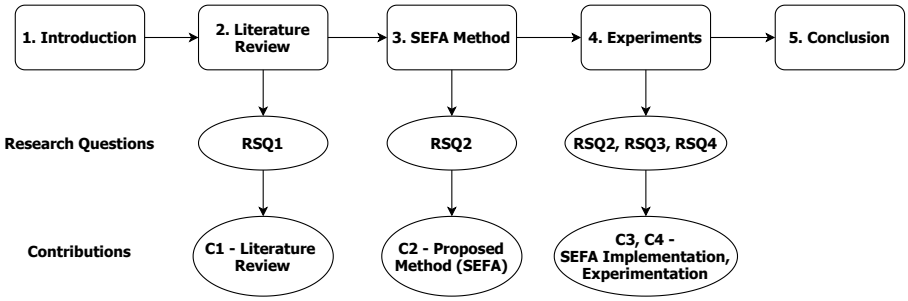


Figure 1.4: Thesis Outline - Chapters, Research Questions & Contributions.

2

Literature Review

This chapter presents the existing work on interpretability with a focus on image classification models. The goal is to answer our first research question (**RSQ1**), namely "Which are the state-of-the-art interpretability methods for image classification models?". To answer this question, we break it down to three sub-questions:

- **RSQ1.1:** How is interpretability defined and which are its uses?
- **RSQ1.2:** How can we categorise existing interpretability methods based on their characteristics?
- **RSQ1.3:** Which are the ideal requirements that an interpretability method should adhere to?

The rest of the chapter addresses these questions. In particular, we first define interpretability and present the uses that it can be applied for. Then, we present the different characteristics of an interpretability method, introduce the main ideas behind key existing methods and categorize them into a taxonomy based on the aforementioned characteristics. Finally, we introduce the ideal requirements of an interpretability method and analyse existing methods based on them.

Before providing the answers to the previous questions, we want to briefly explain our selection strategy behind the papers included in this survey. We started by studying the latest literature surveys on the topic of interpretability [2] [15] [24] [64]. These surveys enabled us to get an overview of the domain and to understand the categorization of different methods, such as their distinction between local and global methods. Following that, we looked into the most significant methods published in top-ranked machine learning and computer vision conferences on the topic, including NIPS[3], CVPR[4], ECCV[5], AAAI[6] and more.

[3] https://nips.cc/
[4] http://cvpr2020.thecvf.com/
[5] https://eccv2020.eu/
[6] https://www.aaai.org/

2.1. Interpretability Definition & Needs

To begin with, we consider it crucial to discuss how we define the term interpretability in our work before diving into the details of existing methods. While it is generally accepted that interpretability is a key aspect of machine and deep learning models [42] [44] [64], there is no common term used in literature. To elaborate, a variety of terms is used throughout existing studies, such as interpretability, explainability, comprehensibility, and intelligibility [24] [57].

The definitions of these words within the machine learning domain ranges from broad non-technical to specific technical ones. For instance, Bhatt et al. [9] describe it as "any technique that helps the user or developer of ML models understand" their behaviour. On the other hand, Ribeiro et al. [43] define interpretability in a more precise manner as providing "qualitative understanding between the input variables and the response" of the model. Zeiler and Fergus [61] describe it in a technical way as "interpreting the feature activity in intermediate layers".

While some researchers have attempted to disambiguate these terms, with the work of [15] being one such example, it is generally agreed that there still is no concrete definition [19] [37] of interpretability that is applicable across a range of studies and applications. Therefore, inspired by Doshi-Velez and Kim [19] and Du et al. [20], we define interpretability within the scope of our work as follows:

Interpretability Definition

The ability to explain the behaviour of a deep learning image classification model in terms that a human can understand.

We would like to clarify that the term behaviour can refer to either explaining why a specific prediction was made by the model or to explain which features the model uses to classify a class of interest.

Following the interpretability definition, we want to present the different needs that it covers and the corresponding use cases that it can be applied on. Based on existing literature [2] [19] [24] [46], we summarise the four main needs that interpretability fulfils, namely (1) system verification, (2) performance improvement, (3) legislation compliance and (4) knowledge gain. An overview of these needs coupled with example use cases for each one of them can be found in Table 2.1.

Interpretability Need	Use Case Example
System Verification [24] [46]	"Spot the background bias in the dataset."
Performance Improvement [2] [46]	"Understand AlexNet to improve its performance."
Legislation Compliance [2] [24] [46]	"Comply with the GDPR law."
Knowledge Gain [2] [19] [46]	"Improve our knowledge in the game of Go."

Table 2.1: Interpretability needs with use case examples.

System Verification

The first need is to verify the behaviour of a trained model. In particular, recent progress in machine and deep learning systems has been made possible by training

them on massive amounts of data [24]. However, these datasets often contain biases which can lead models to demonstrate unwanted behaviours [24] or to base their predictions on wrong features. The unwanted biases present in the datasets are then picked up or even amplified by the models, as shown by existing studies on text data [10] [14].

Ribeiro et al. [43] provide an example of this issue by training a logistic regression classifier on images with biased backgrounds and show that their method can detect the bias by highlighting the background artefacts. Similarly, Buolamwini and Gebru [12] discover that commercial gender classification systems have significant performance differences across users with different demographics. Therefore, the use of interpretability methods is crucial to understand the weaknesses and limitations of such models [24] [46] so that they can be addressed.

Performance Improvement

The second need for interpretability methods stems from our desire to continuously improve the performance of existing architectures. It is safe to assume that understanding the behaviour of a model will make our life much easier towards achieving this goal. Existing studies [2] [46] highlight that understanding a model and its weaknesses is a step in that direction. An example of such use case is the work of Zeiler and Fergus [61] who attempt to understand how AlexNet [35] achieved its performance on the ImageNet [45] benchmark dataset. Their method allowed them to visualise the intermediate feature layers of AlexNet which enabled them to come up with architectures that outperform the original model.

Legislation Compliance

Given the increasing presence of ML systems in our daily life, they are receiving more and more attention regarding their legal aspects [46]. One such example is the General Data Protection Regulation (GDPR) [23] law in the European Union. More specifically, it enables every individual to ask for an explanation regarding the decision taken by an automated system. Hence, interpretability methods that provide explanations for individual predictions could assist in addressing the GDPR requirements and other similar needs [2] [24] [46].

Knowledge Gain

Our need to extend human knowledge can be supported by the use of interpretability methods to explain existing models [2] [19] [46]. Doshi-Velez and Kim [19] highlight this need by claiming that "the human's goal is to gain knowledge". However, since some of the these ML systems are extremely complex for humans to perceive [46], designing methods that enable us to understand the behaviour of these models will allow us to extract further knowledge [2]. For example, explaining the strategy learned by the AlphaGo Zero [49] reinforcement learning algorithm would enable us to improve our knowledge at the game. Similarly, interpretability can potentially lead us to learn more about the hidden laws of nature in sciences such as physics, chemistry and biology [2] [46].

2.2. Interpretability Method Taxonomy

Having defined interpretability within the scope of our work and presented its use cases (**RSQ1.1**), we would like to summarise the existing methods in literature. To elaborate, we present existing interpretability methods that enable us to explain image classification models and structure them in a taxonomy (**RSQ1.2**). While our taxonomy is by no means exhaustive, we focus on discussing the main interpretability methods presented so far in terms of scientific impact.

2.2.1. Taxonomy Dimensions

Our taxonomy comprises of two dimensions: interpretability scope and model relation. The details of each dimension are presented below.

Interpretability Scope

One of the most common separations of interpretability methods made by existing studies [2] [22] [24] [29] [32] [59] is based on the method scope. To be more exact, the term scope refers to the goal that the method output attempts to achieve. Existing studies strive for two separate goals, local and global interpretability.

On the one hand, local interpretability methods [7] [43] [51] explain an individual prediction generated by the classification model for an image of interest. However, given that machine learning models are commonly trained on datasets containing millions of images, it is impractical for system designers to interpret their behaviour via individual image explanations [59]. Therefore, a range of global interpretability methods [22] [29] [59] has been proposed to address this limitation of local methods. Global methods attempt to explain the behaviour of an image classification model in terms that a human can understand. These methods allow us to interpret a model with respect to entire classes or sets of examples [32].

Model Relation

The second dimension of our taxonomy describes the relation of the interpretability method to the classification model [2] [24]. In particular, we refer to the range of models that an interpretability method can be used with. Based on existing work, methods can be classified as model-specific or model-agnostic.

Model-specific methods are designed with a specific image classification model in mind. To be more specific, the model is created with built-in interpretability, meaning that it can provide explanations about its predictions and its behaviour. For example, Zhang et al. [65] propose a Convolutional Neural Network (CNN) that can explain its features maps by using the same labeled training data as ordinary non-interpretable CNNs. On the contrary, model-agnostic interpretability methods can be applied to any existing deep learning architecture and are independent of the trained model.

An existing literature review by Adadi and Berrada [2] also proposes separating methods into intrinsic and post-hoc. Intrinsic methods refer to algorithms that are inherently interpretable while post-hoc correspond to methods that are applied to algorithms following their training without any change in their architecture [2]. However, we argue that intrinsic and post-hoc distinction matches the categorisation

made by the model relation while also having a similar overlapping meaning. Thus, we decided to exclude this dimension from our taxonomy.

2.2.2. Existing Methods

In this section, we summarise the main existing interpretability methods and describe how they fit in our taxonomy based on the two aforementioned dimensions. More specifically, we focus on methods that can be used to explain deep learning image classification models. To provide more structure, we group methods with similar characteristics. We should underline that although our focus is on methods explaining classification models, there is also the possibility to look into explanations of complex dataset distributions [31].

Gradient-Based

The first group of methods base their explanations on a gradient related computation of the activation function learned by the deep learning model. The gradient calculation allows us to test the effect of a small change in the image to the classification score output by the model [33].

Simonyan et al. [51] introduced the notion of saliency map which is currently one of the most popular ways to interpret an individual prediction of a deep neural network trained on image datasets [3] [33]. Given a specific image and a class of interest, they proposed to compute the gradient of the activation function for that class with respect to every image pixel input into the network. The calculated pixel intensity values can then be visualised to highlight the parts of the image that the model uses to discriminate the class of interest. Given that this method can be used with any trained model to explain individual image samples, it can be characterized as local and model-agnostic according to our taxonomy.

While gradient saliency maps seem to capture a correlation between the pixel input space and the label predicted by the model, their output can be particularly visually noisy [52]. Therefore, Smilkov et al. [52] proposed SmoothGrad which is a method that can sharpen the output of these gradient-based maps. More specifically, they sample several images from the original image by adding noise, then calculate the gradients for each of these images and obtained their average values for the resulting saliency map.

Another gradient-based method presented by Sundararajan et al. [54] is the Integrated Gradients, a local interpretability method similar to the original gradient approach that computes a saliency map. However, their approach computes the pixel intensities via a gradient integral along a straight-line path between the input image and a baseline image instead. In their work, they propose the use of a black image as a baseline when explaining image networks. Similarly to the "vanilla" gradients, their method can be applied to any deep neural network. However, Adebayo et al. [3] found that it has limited sensitivity to network weights by comparing the saliency map of trained and random network weights.

Signal-Based

This group of interpretability methods uses network signals instead of gradient values to compute the pixel intensities for the saliency maps. More specifically, a

relevance signal is backpropagated using the network's trained weights from the output neuron through each previous layer until it reaches the input pixel space [48]. Since these methods can be used with any trained neural network and their output is a saliency map similar to the gradient-based approaches, they can be characterized as local and model-agnostic.

Zeiler and Fergus [61] proposed the DeconvNet interpretability method motivated by the need to explain the state-of-the-art network [35] at the time. Contrary to Zeiler et al. [62] who experimented with deconvolutional networks, the DeconvNet of Zeiler and Fergus [61] focuses on explaining an already trained CNN. In particular, it visualises the input pixels that caused "a given activation in the feature maps" [61]. This functionality is achieved by performing typical CNN functions such as filtering and pooling in reverse, namely from the feature maps to the input pixel space. The process resembles backpropagating "a single strong activation" [61] instead of the usual network gradients.

That said, Springenberg et al. [53] observed that the aforementioned DeconvNet method does not perform well in the absence of a max-pooling layer. Therefore, they proposed a new variant of DeconvNet named Guided BackProp, which can visualise the salient pixels for higher CNN layers. Similarly to DeconvNet, their method backpropagates the trained network but only uses the top gradient signal when computing the gradient of a non linear function. Guided BackProp was found to perform well both for intermediate and higher network layers [53], thus addressing the aforementioned limitation of DeconvNet.

Another approach that takes advantage of network signals is the "Layer-Wise Relevance Propagation" (LRP) proposed by Bach et al. [7]. Contrary to the previous methods [53] [61], LRP starts from the final network layer and backpropagates relevance values through each prior layer until it reaches the input pixel space. As a result, it can compute the importance of each pixel for the model prediction, thus providing a similar output to gradient-based methods. Given the way the pixel intensities are computed, the LRP method provides information about pixels that contributed both negatively and positively towards a specific prediction. On the contrary, gradient-based methods only highlight the pixels that the model is sensitive to without an indication of positive or negative effect.

Finally, the "Deep Learning Important FeaTures" (DeepLIFT) method of Shrikumar et al. [48] is another interpretability method that computes the contributions of each pixel to the prediction for a specific image. Similarly to LRP, it backpropagates the contribution values through all the neurons of the trained neural network. However, its contributions scores are computed by comparing the neuron activations to "reference activations" and calculating their differences. The "reference activations" are based on an input that is selected per use case. For instance, for the MNIST dataset, [48] use a reference image of zero values since this is the background value of the digit images. As a result, DeepLIFT explains the differences of an image of interest versus the corresponding values of the "reference" one.

Local Approximation

The local approximation group of methods focuses on methods that attempt to interpret a complex model by approximating its behaviour locally in the pixel space.

The methods presented in this section can be used to reason about both local and global interpretability, and can be used with any trained model.

Ribeiro et al. [43] proposed the "Local Interpretable Model-agnostic Explanations" (LIME) method which provides explanations for individual images-predictions of any model. Their key idea is that a complex non-linear classifier can be interpreted by learning a linear interpretable model that locally approximates its prediction behaviour. In their work, they also modify the LIME method to provide global explanations about the model as a whole. In particular, they propose a submodular pick method, SP-LIME [43], which selects specific instances from the dataset that should be explained. The intuition is that by sampling multiple representative instances from the dataset and analysing their local explanations provided by LIME, they can reason about global interpretability. That said, they leave the sampling criterion of SP-LIME for image data as future work. Hence, we only consider the local variant LIME in our image data interpretability taxonomy.

Their work on LIME was followed up with Anchors [44] which is another local model-agnostic interpretability method that attempts to locally approximate the behaviour of complex models. Contrary to LIME, the desired behaviour is achieved via high-precision "if-then" rules instead. The motivation behind this new method is to address a limitation of LIME explanations, namely that they may not apply to unseen samples. Anchors are designed in such a way that their explanations are more faithfull by "adapting their coverage to the model's behaviour" [44].

Interpretable Models

The next group of existing interpretability methods focuses on designing models that are inherently interpretable and can interpret their own predictions. Hence, the methods in this group are characterized as local and model-specific.

Xu et al. [58] propose a model that is trained to describe image contents via a neural architecture that utilises attention. Their network can focus on salient objects in images and generate an output text sequence that describes these objects that are important for its prediction. The idea is that by visualising the attention layer of the model, they can provide explanations for specific outputs.

On the other hand, Zhang et al. [65] present a methodology that modifies existing CNNs to provide interpretable outputs. In particular, the filters in the higher convolutional layers of their modified network map to specific salient objects. The proposed method can be applied to modify numerous types of CNNs with different architectural choices to provide interpretable outputs.

Concept-Based

Finally, the group of concept-based methods resolves around providing global explanations via high-level concepts that are understandable by humans for any trained network. As such, they are characterized as global and model-agnostic.

The idea of concepts was first put forward by the work of Kim et al. [32]. To elaborate, they propose "Testing with Concept Activation Vectors" (TCAV), a method that aims to quantify the influence of high-level human concepts to neural network image classifiers with respect to a specific class. The Concept Activation Vectors (CAVs) described in their work enable us to interpret a network's internal state

in human concepts. This is done by collecting sample images that represent the human concept of interest, projecting the concept images coupled with random ones into a feature space learned by one of the model's intermediate layers and training a linear classifier to separate the concept from the random images. The CAV is then derived by using the orthogonal vector to the decision boundary. The importance of a concept is provided by a numeric value ranging from zero to one. That way, we can quantify, for instance, the influence of the concept "striped" to the predictions of class "zebra" for a trained neural network of interest.

However, TCAV only responds to user queries and requires significant workload to provide labeled images of the high-level human concepts [22]. To address its limitations, Ghorbani et al. [22] proposed "Automated Concept-based Explanation" (ACE), a framework that automatically extracts high-level concepts based on the dataset and trained model representation. This is achieved by utilizing image segmentation and clustering techniques, to extract image segments and to group them into similar concepts based on the representation of an intermediate network layer. Then, they use TCAV to compute the importance scores of the extracted concepts for the model with respect to the class of interest.

Another method that is based on the "concept-based" idea is "Global Interpretation via Recursive Partitioning" (GIRP) [59]. In particular, Yang et al. [59] proposed an "interpretation tree" that presents the decisions rules that a complex model uses for its predictions. The proposed tree is learned on top of a "contribution matrix" that quantifies the contribution of the input features for each sample to the model predictions. In the case of image data, they utilise an semantic segmentation algorithm to extract image "superpixels", a.k.a. concepts, for each image. Then, they use LIME [43] to compute the contribution of each concept for a classification, thus extracting the required contribution matrix.

While the aforementioned concept-based methods enabled us to answer a range of global interpretability queries that local methods were unable to, we argue that they struggle to provide answers to complex questions. In this work, we present a new model-agnostic interpretability method, called **SEmantic Feature Analysis (SEFA)**, which utilises human-understandable concepts named **semantic features**. The analysis of these features allows us to answer more complex queries compared to existing methods. While its primary goal is to provide global explanations, SEFA can also reason about individual predictions as a side-effect of the way its global interpretations are obtained. The differences between SEFA and the existing methods are discussed in more detail in Chapter 2.3.

2.2.3. Method Taxonomy

Using the descriptions of the interpretability methods provided in Chapter 2.2.2, we map them to our taxonomy dimensions from Chapter 2.2.1. The output of this process is presented in Table 2.2.

Based on the aforementioned table, it becomes clear that interpretability methods usually have either a local or a global scope. We argue that both of them are equally important since they address different needs from Chapter 2.1. To elaborate, local methods can explain individual predictions, thus enabling us to check

Interpretability Method	Interpretability Scope	Model Relationship
Gradient-Based [51], [52], [54]	Local	Model-agnostic
Signal-Based [7], [48], [53], [61]	Local	Model-agnostic
Local Approximation [43], [44]	Local	Model-agnostic
Interpretable Models [58], [65]	Local	Model-specific
Concept-Based [22], [32], [59]	Global	Model-agnostic
SEFA (ours)	Local, Global	Model-agnostic

Table 2.2: Taxonomy of existing interpretability methods.

whether a model makes unbiased decisions using features related to the classification task at hand. Similarly, they can visualise trained feature maps for specific images which give us a better understanding of trained models. Moreover, they address existing legislation regulations and can expand our understanding of certain tasks by observing how certain decisions are made.

That said, local methods provide limited information about the behaviour of the model as a whole, an issue that global interpretability methods address. More specifically, they allow us to interpret the sources of dataset bias and to evaluate whether the model uses the right features to discriminate a class of interest. What is more, they can help us understand the human concepts that the intermediate network layers have learned, thus enhancing our understanding of neural architectures and their behaviour. Finally, being able to explain complex high performing models can further our knowledge on certain tasks by providing us with new problem-solving strategies.

Moving on to the issue of the method's relation to the model, the majority of existing methods can be used with any existing trained network of choice. We strongly believe that having model-agnostic methods that are not inherently tied with a model is the research direction we should focus on. While model-specific interpretability offers simplicity, it cannot be used with existing models without retraining or modifying them, a process which can prove extremely costly for users with existing high performing models [32]. Therefore, our focus for the rest of the chapter is on model-agnostic methods.

2.3. Ideal Interpretability Requirements

Inspired by our reflections on existing literature and experiments of existing methods on the use case datasets of our work, we propose several interpretability requirements that any new method should adhere to (**RSQ1.3**). In this section, we evaluate how the existing methods and SEFA fare against them. The ideal requirements for an interpretability method can be summarised as follows:

- **Model Relation**: it should be able to explain any deep neural network without requiring to modify or retrain the model.
- **Interpretability Scope**: it should answer both local and global queries.

- **Complex Queries**: it should be able to answer complex queries concerning multiple entities in an image.
- **Concept Diversity**: it should allow us to reason both about elements in the images and their characteristics.
- **Convenience**: it should be relatively easy to use and able to provide explanations in a reasonable amount of time.
- **User Expertise**: it should not require a high level of expertise.

At this point, we would like to discuss the reasoning behind our requirements. The need to focus on model-agnostic methods stems from our literature review and is also discussed by the authors of TCAV [32]. Moving on to the interpretability scope, several studies underline that local and global methods answer different types of queries [2] [24], thus necessitating a different output per scope type. The ability to answer complex queries and to provide high concept diversity is a need that arose during our review of existing methods. We provide a more concrete example of this argument later in the section. As for the method's convenience, existing work [19] [24] has mentioned the need to take into account possible time restrictions. We argue that interpretability methods that are convenient to use and have a low computational cost are more desirable since they can tackle a wider range of use cases. Finally, given the wide range of interpretability stakeholders mentioned by [9] [19] [24], we believe that an interpretability method should not require high expertise to appeal to more target users, such as decision-makers, regulators and more. This point is especially important when considering that the user groups of several interpretability needs from Chapter 2.1 are non-experts.

In this work, we present SEFA, a new interpretability method that is designed with the previous requirements in mind. SEFA's goal is to explain a model's behaviour with respect to a specific class using **semantic features**. The main idea is to annotate groups of pixels highlighted by a local interpretability method with semantic descriptions to obtain the aforementioned semantic features. The semantic features extracted per image are then structured into our **semantic representation**, thus providing a structured representation similar to the contribution matrix of [59]. Concrete examples of our semantic representation are available in Chapter 3.5.4. The extracted representation can then be analysed using traditional structured data analysis methods to answer global interpretability queries. A comparison of the main model-agnostic methods presented in Chapter 2.2.2 and SEFA according to the ideal interpretability requirements can be found in Table 2.3.

According to the previous table, it becomes clear that SEFA is one of the first methods able to answer both local and global queries for image data, thus necessitating the use of just one method to answer both types of queries. As for the query complexity, while GIRP and SEFA provide the flexibility to answer complex interpretability queries, such as whether the presence of multiple elements leads to the classification of a specific class. TCAV and ACE are more limited since they can answer questions such as "does the presence of a desk point towards class office_room?" but fail to answer whether "the presence of a painting, a bed and

Interpretability Method	Local Queries	Global Queries	Complex Queries	Concept Diversity	Convenience	User Expertise
Local methods	Yes	No	Yes	Yes	High	Low
TCAV [32]	No	Yes	Limited	Yes	Low	Low
ACE [22]	No	Yes	Limited	Yes	Average	High
GIRP [59]	No	Yes	Yes	Limited	High	High
SEFA (ours)	Yes	Yes	Yes	Yes	Average	Low

Table 2.3: Comparison of existing methods to ideal interpretability characteristics.

a desk point towards class bedroom", contrary to GIRP and SEFA. Similarly, local methods can answer any type of query but only for individual images, a limitation which does not allow us to reason about the model behaviour as a whole.

Despite the aforementioned flexibility, a key difference between GIRP and SEFA is in how their structured representations are obtained. GIRP uses a semantic segmentation algorithm to obtain the features of its contribution matrix while SEFA takes advantage of human annotations to extract the features of its semantic representation. We argue that this key difference allows SEFA to answer a much wider range of queries concerning not only elements in images but also their attributes such as colour, shape and more. We should also mention that both TCAV and ACE can also answer queries related to the element attributes but are unable to perform combinations of them contrary to SEFA. Another distinction between GIRP and SEFA is on the main focus of their studies. While the former centres around the interpretation tree analysis method proposed, the latter focuses on the extraction of a semantic representation that can be analysed via existing analysis methods.

Moving on to the convenience of each method, local methods and GIRP require minimum input by their users. On the other hand, we argue that ACE is of average convenience given that the user has to carefully select the images that will be segmented and those that will be used to compute the TCAV scores. Similarly, SEFA requires the user to annotate a number of images with semantic features to obtain global explanations. TCAV is arguably the least convenient of the methods analysed since the user has to collect the images for all the concepts that will be evaluated, on top of creating the image folders needed for the score computation. As for the user expertise required to interpret the output of each method, we argue that SEFA, TCAV and local methods require no computer vision or machine learning background. On the contrary, GIRP and ACE users should already have some expertise to reason about their outputs. For instance, GIRP provides an output similar to a decision tree, meaning that users without any technical knowledge may find it counter-intuitive at first. Similarly, ACE requires knowledge of the method internals regarding the concept extraction to understand its output.

To conclude, this section makes clear what the contributions of SEFA are over the existing interpretability methods. To elaborate, it is a method that provides increased flexibility and expressivity in terms of the types of interpretability queries that it can answer. At the same time, it allows us to reason about both local and global interpretability in a convenient way that requires minimum user expertise, thus making it accessible to a wide range of stakeholders.

2.4. Summary

This chapter answers our first research question defined in Chapter 1.2. To achieve our aim, we broke down **RSQ1** into three sub-questions which we answered in order throughout the chapter. In particular, we first defined interpretability within this line of work and laid out the four main interpretability needs that current methods address (**RSQ1.1**). Then, we presented the main methods in literature and framed them within a taxonomy according to two dimensions, namely interpretability scope and model relation (**RSQ1.2**). Finally, we presented the ideal interpretability requirements stemming from our literature review and evaluated how existing methods and SEFA fair against those characteristics (**RSQ1.3**).

3

SEFA Method

In this chapter, we present the details of our proposed method. We first provide an overview of SEFA and its four components, and then we describe the design choices and considerations for each of them individually. By presenting SEFA in detail, we answer our second research question (**RSQ2**), namely "How can we use local interpretations to extract semantic features that allow for global interpretability?".

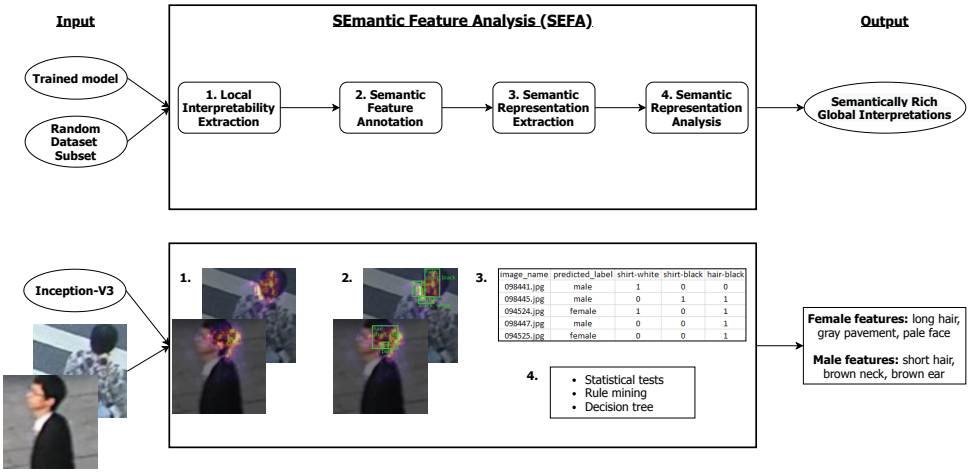


Figure 3.1: SEFA overview. Top: main steps of the method coupled with its input-output. Bottom: example output of each intermediate step for one of our use cases.

3.1. SEFA Overview

We begin by providing an overview of the proposed method in Figure 3.1. SEFA takes the trained model that is to be explained coupled with a random dataset

subset as input and outputs semantically rich global explanations. The global interpretability extraction is done in four steps which are summarised below:

1. **Local interpretability extraction**: we use a gradient-based local interpretability method to obtain a heatmap that highlights the pixels of each image that the model is sensitive to.
2. **Semantic feature annotation**: we annotate the highly sensitive areas of the heatmaps with semantic information using human annotators to extract the **semantic features** described in Section 1.1.
3. **Semantic representation extraction**: the semantic features obtained during the previous step are utilised to create the aforementioned **semantic representation**.
4. **Semantic representation analysis**: we analyse the semantic representation to answer complex queries about the model's behaviour with respect to a particular class. The analysis can be done using a plethora of techniques such as statistical tests, data mining techniques and fitting a decision tree.

In the following sections, we present the design details and considerations of each intermediate step in detail.

3.2. SEFA Input

As mentioned previously, SEFA needs a trained model and a subset of images as input. In particular, the trained model can be any deep learning network that takes the pixel values of an image as input and outputs a prediction for that sample. As for the image subset, it is randomly sampled from the dataset which we train, validate and test the model on. An important factor influencing the method's output is the number of images that are selected for annotation. The effect of this factor on SEFA is analysed in detail in Section 4.2.1.

3.3. Local Interpretability Extraction

The first step of SEFA is to extract the local explanations for each of the images sampled. In this section, we describe the design choices and considerations made regarding the method selected to extract these explanations.

3.3.1. Method Selection

Given the plethora of local interpretability methods presented in Chapter 2.2.2, a key question arises: "which interpretability method is more reliable and robust to extract local explanations?".

We have already mentioned in Chapter 2.2.2 that gradient-based local interpretability methods are one of the most popular ways to explain individual predictions of a deep learning image classification model. However, to further strengthen the reasoning behind this choice, we performed a series of bias injection experiments where we compared a gradient-based approach [51] with a signal-based

one [7] and a local approximation method [43]. The output of these experiments is available in Appendix A. Based on the empirical evidence obtained, gradient approaches seem to capture the biases introduced for the images in our study in a more reliable manner.

Despite narrowing down our choice, there are numerous gradient-based methods to choose from. To select the specific method that we should use, we compiled a list of requirements that it should adhere to in order to provide reliable and robust local explanations:

- It should be sensitive to the relationship between instances and labels. This characteristic ensures that the method captures the association between the model predictions and the images.
- It should be sensitive to the parameters of the trained model. This desired characteristic enables us to compare the classification behaviour of separate models trained on the same tasks.
- It should provide the same explanations for networks that process the images in identical manners. This requirement makes certain that the local interpretability method provides robust outputs.

The studies of [3] and [33] evaluate local interpretability methods based on the aforementioned characteristics. In particular, Adebayo et al. [3] perform a label randomization test and a network parameter randomization test, and evaluate different local methods based on our first two characteristics. Similarly, Kindermans et al. [33] evaluate the output of separate saliency methods when shifting the dataset by a constant vector. By comparing the methods that pass the sanity checks of both studies, we conclude that the "vanilla" gradient calculation [51] satisfies the three criteria that we defined and is appropriate for our study.

3.3.2. Vanilla Gradients

The "vanilla" gradient calculation is a straightforward way of computing a saliency map for a specific image. In particular, the sensitivity of the model with respect to each pixel in an image can be computed by the gradient of the activation function with respect to the class of interest for each pixel in the image [52]. Several authors have already proposed ways of mathematically computing these derivatives [8] [21] [51]. More specifically, the saliency map $M_c(x)$ of an image can be computed as:

$$M_c(x) = \frac{\partial S_c(x)}{\partial x} \tag{3.1}$$

where S_c the activation function of the model with respect to the class of interest c and x the image that is interpreted. The idea behind this computation is to find the pixels in the image that the model is most sensitive to when computing the activation score for class c. Another way to view the gradient values according to Simonyan et al. [51] is as "which pixels need to be changed the least to affect the class score the most". As a result, a saliency map for a specific image highlights

the pixels in the image that are discriminative for the model given a specific class of interest [51].

That said, this particular method of extracting local explanations has an important limitation. More specifically, gradient saliency maps are known to be particularly noisy [3] [52]. This issue is something that we were also able to observe during our initial experiments. To showcase the issue at hand, we provide examples of noisy saliency maps from the paper of Smilkov et al. [52] and our work in Figures 3.2 and 3.3 respectively.

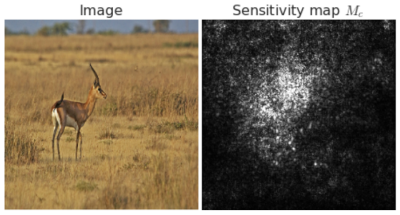


Figure 3.2: "Vanilla" gradients noise issue example provided by Smilkov et al. [52].

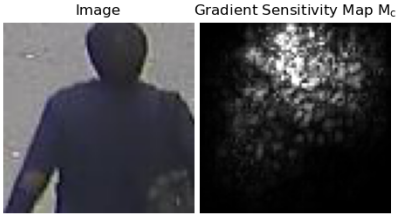


Figure 3.3: "Vanilla" gradients noise issue observed during our experiments.

In the next section, we describe how we addressed the issue of gradient noise while maintaining the use of "vanilla" gradients for the saliency map computation, a method which satisfies the selection criteria defined in Chapter 3.3.1.

3.3.3. SmoothGrad

Smilkov et al. [52] proposed SmoothGrad, a method that directly addresses the gradient noise issue. Their proposal is based on the idea that smoothing the partial derivative of the activation function $\partial S_c(x)$ with a Gaussian kernel will reduce the influence of the saliency map to gradient fluctuations. However, since computing its local average in a high-dimensional pixel input space would be intractable, they resorted to an approximation [52]. To be more precise, they create random samples in the neighbourhood of the input image by adding noise and then compute the saliency map by taking the average of all the sampled maps. The aforementioned procedure can be computed as follows:

$$\overline{M}_c(x) = \frac{1}{n}\sum_{1}^{n} M_c(x + \mathrm{N}(0, \sigma^2)) \tag{3.2}$$

where n the number of random samples obtained and $\mathrm{N}(0, \sigma^2)$ the amount of Gaussian noise added. The effectiveness of applying SmoothGrad to the gradient computation becomes visible when comparing saliency maps using "vanilla" gradients versus their SmoothGrad counterparts in Figure 3.4.

While SmoothGrad does indeed reduce gradient noise, an important question arises: "does it maintain the same desired characteristics as the "vanilla" gradients?". Kindermans et al. [33] found that SmoothGrad inherits the sensitivity of the method used to compute the pixel attribution in the first place. As a result,

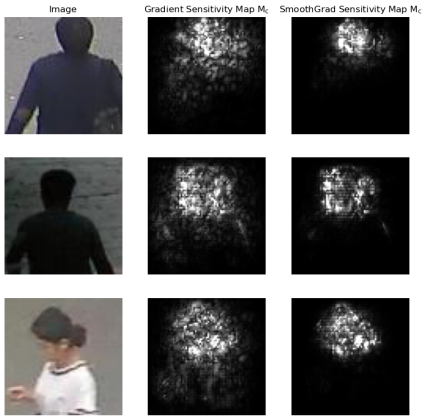


Figure 3.4: Saliency maps: "vanilla" gradients vs Smoothgrad.

given that the "vanilla" gradients were found to have the desirable behaviour, we are confident that SmoothGrad maintains it and is suitable for our use case.

Finally, we should underline that SmoothGrad has two hyperparameters that need to be tweaked: the number of random samples that we average over and the standard deviation σ which defines the amount of noise added. We suggest tweaking these values accordingly for each dataset used. The values used in our use cases are presented in detail in Chapter 4.1.3.

3.4. Semantic Feature Annotation

The next step in the SEFA methodology is to annotate semantically the groups of salient pixels in each image. To that end, we design an annotation task that can be used either by domain experts or crowd workers to provide the semantic features necessary to answer complex global interpretability queries. More specifically, we create a task that takes the image and its SmoothGrad [52] saliency map as input, and enables users to provide semantic information for the groups of pixels that the trained model is most sensitive to. The rest of this section presents the design and technical considerations taken when developing this task.

3.4.1. Annotation Task Input

The first step during the task design was to prepare the task input, i.e. the content that human annotators will annotate. The SmoothGrad implementation[7] used allowed us to interpret the classification of an image in three separate ways:

1. Visualise the pixel sensitivity with its grayscale value.
2. Plot the grayscale values using a colormap that resembles a heatmap.
3. Depict the x% most salient pixels in the image.

[7] https://github.com/PAIR-code/saliency

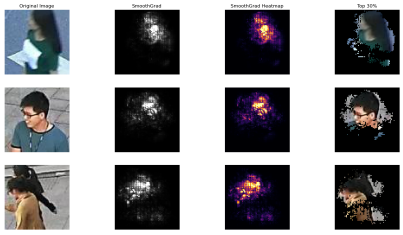


Figure 3.5: Smoothgrad visualisation options.

Examples of these three different options are provided in Figure 3.5. Given these options, we chose to use the heatmap version since the areas of higher intensity are more clearly visible compared to the grayscale version while it also provides information about every part of the image, contrary to the top x% option. Furthermore, we decided to overlay the original image with the heatmap to make it easier for the annotators to understand which semantic feature is highlighted in an image. For that reason, we experimented with different opacity levels to understand which one is more suitable. The different levels tested can be found in Figure 3.6.

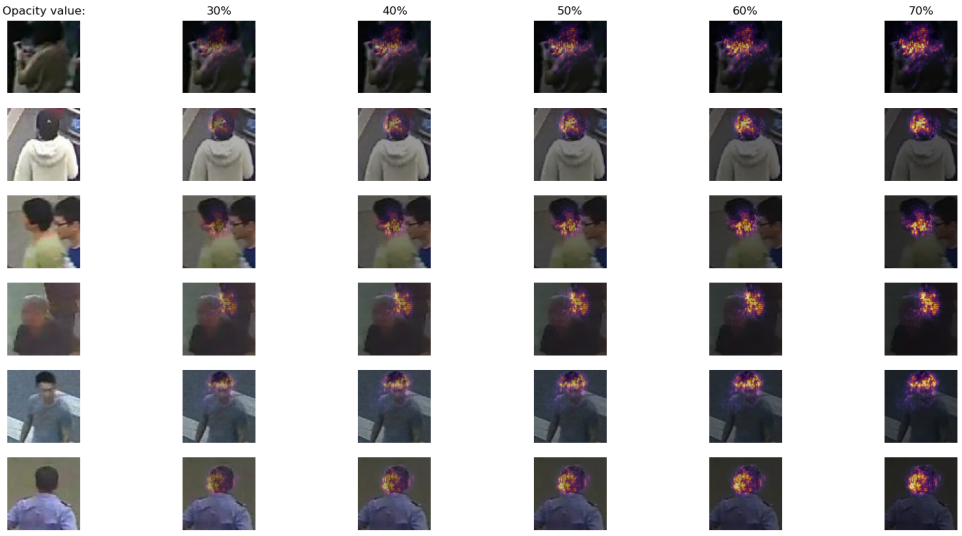


Figure 3.6: Image/Heatmap Overlay - Opacity levels.

Based on the previous opacity level tests, we concluded that a level of 50% opacity strikes a balance between the clarity of the original image and the colour intensity of the heatmap.

Finally, for datasets where the original image is of low resolution, we propose upsampling to make the image content clearer to the human annotators. A comparison of an upsampled image which has a resolution that is three times larger than its original 75x75 counterpart is available in Figure 3.7. By observing the figure, it becomes clear that increasing the resolution makes the blurry parts in the original image smoother, thus allowing the annotators to understand what these pixel groups represent in an easier manner.

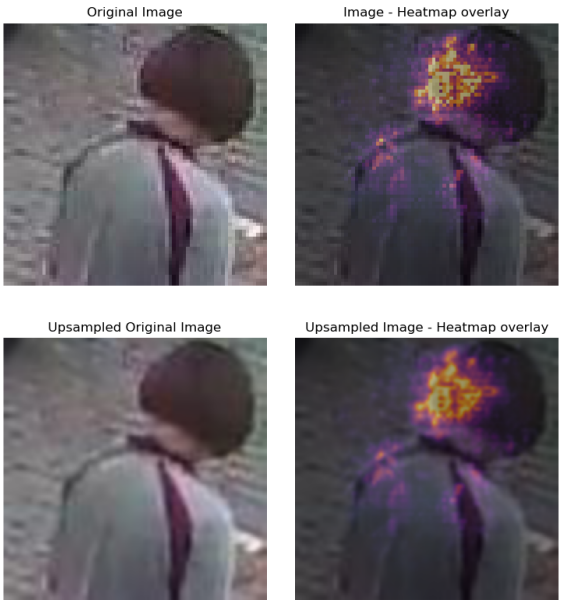


Figure 3.7: Upsampled vs original image - 50% opacity.

3.4.2. Human Annotator Characteristics

The next design step concerns the human annotator characteristics. In particular, we assumed that the task will be used by lay men/women without any Information Technology (IT) or Computer Science (CS) background. The reason for this choice is that we wanted the task to be suitable both for domain experts with IT/CS expertise and crowd workers from a crowdsourcing platform, such as *MTurk*[8]. This way, we enable SEFA users the option to scale out the annotation process with non-expert annotators. Therefore, the task instructions were kept as simple as possible while we also avoided the use of terminology and jargon.

3.4.3. Annotation Task User Interface

With the previous considerations in mind we designed the User Interface (UI) of the semantic feature annotation task. The UI is made up of four components: framing, instructions, examples and annotation task. An overview of the designed task can be found in Figure 3.8.

The UI of the task was designed with the following considerations in mind:

[8] https://www.mturk.com/

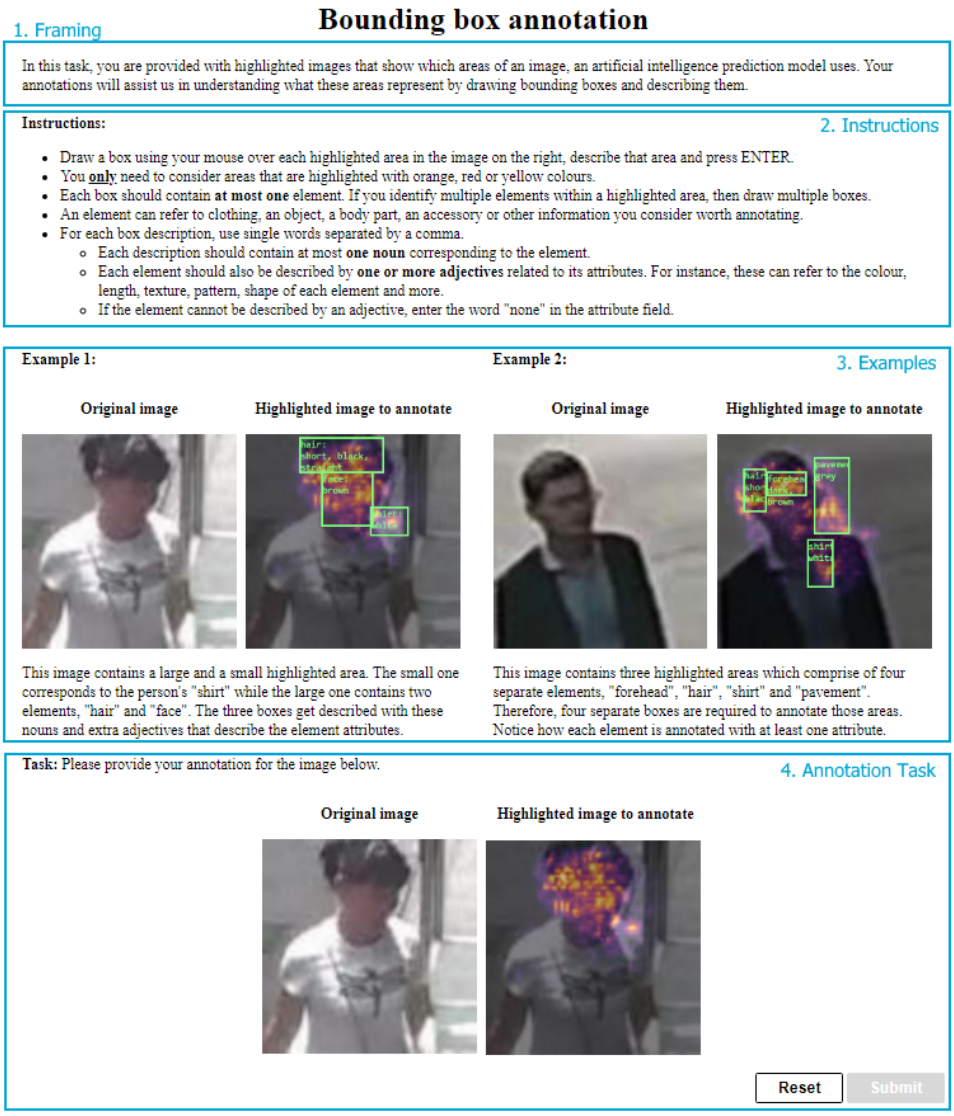


Figure 3.8: Overview of the human annotation task.

General Considerations

- Domain experts and crowd workers are already used to performing bounding box and image tagging annotations. Therefore, we created a task that is easily identifiable by such a user group and avoided too much information to minimise their cognitive load.

- The images are randomly presented to each annotator to avoid potential dif-

ficulty bias. For instance, having more difficult samples to annotate first can lead them to lose motivation.

Framing/Instructions

- We added a short framing introduction since quality is known to improve when workers are aware of how their annotations contribute [16]. While we considered adding extra information, such as what task the system performs or how the results will be used, we believe that it would make the combination of framing, instructions and examples sections too long. More importantly, it would run the risk of biasing the workers by having them only look at task specific parts of the image, such as gender characteristics in this example.

- Terminology such as sensitive, saliency maps, classification etc., was avoided since we assume that the annotators do not have any CS experience.

- We do not show any classifier related information to the annotator, such as the predicted class or the classification confidence to avoid biasing their predictions. Besides, such information is not related to the goal of the task.

Examples

- Human annotators are provided with two example annotations together with a small explanation for each one of them. While we understand that this may bias them towards our interpretations, we believe that the task is not trivial and would be confusing if no examples were provided.

Annotation Task

- We provide annotators with both the image and the heatmap overlay that needs to be annotated. This is done because the overlay can be vague or even mask important information, such as the actual colour of the element.

- The annotation of each semantic feature is made by drawing a bounding box and providing a description. The description is broken down into a single "Element" and its "Attributes" to encourage annotators to describe element properties such as length, colour, texture etc. This idea was inspired by [6] that distinguish the information annotated by crowd workers in saliency maps as "Saliency-Features" and "General-Attributes". An example of the element-attributes breakdown is found in Figure 3.9.

Figure 3.9: Human annotation task - Worker input

- One of the most crucial design choices made was whether we should provide the annotators with a list of options or open text fields. We moved ahead with the latter option since we felt that providing our pre-filled options would

heavily bias the study and we would potentially miss out on important information. What is more, these pre-filled options would have to be modified for each dataset/task. That said, this decision poses several aggregation challenges which are addressed in the next section.

- We ask the workers to provide descriptions with single words separated by comma. This is done to discourage long sentence inputs while it also simplifies the annotation post-processing step.

3

3.4.4. Annotation Quality Control

Moving on to the annotation quality, we made three considerations, namely User Interface (UI) quality checks, annotator selection and annotation post-processing.

UI Quality Control

We check the input in the UI before allowing the annotation to go through. In particular, the submit button is disabled until the worker provides at least one valid bounding box annotation per image. What is more, we check the descriptions provided and allow them only if the user fills both the "Element" and "Attributes" areas, and provided they described the element with a single word. Examples of the aforementioned checks are available in Figures 3.10 and 3.11.

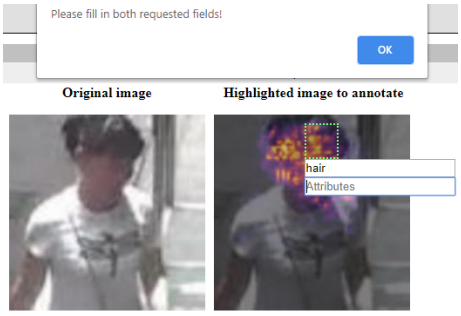


Figure 3.10: Description annotation - completion check.

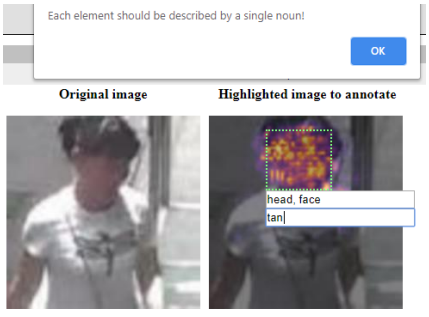


Figure 3.11: Element annotation - single word check.

Annotator Selection

For SEFA users that want to use crowd workers for the human computation step, we suggest filtering workers that have completed at least 5,000 HITs with a 95% HIT approval rate, as suggested by MTurk[9]. Moreover, the location filter provided can be utilised to limit the task to workers from USA and UK. This way, the user can ensure that the task is completed by native English speakers.

[9] https://blog.mturk.com/qualifications-and-worker-task-quality-best-practices-886f1f4e03fc

Post-processing

In cases of more than one annotators, we suggest the use of majority voting [36] for the elements and attributes provided. We also considered adding honeypot questions but the absence of a commonly agreed ground truth makes it challenging. An alternative could be to purposefully inject an image twice in a batch and check if the workers provide a similar annotation between these two image instances.

3.4.5. Pilot Study

The aforementioned task was validated by performing a pilot study of 20 images from a gender classification dataset using ten human annotators. More specifically, we randomly selected ten female and ten male images from the dataset and used five lay-men/women and five workers with a CS background. This selection was made to receive more in-depth feedback from the latter group and to judge the task effectiveness on end-users without any expert knowledge from the former.

Following the annotation of the 20 images, each annotator that participated in the pilot study was asked to fill in a short feedback questionnaire containing the following questions:

- Was the task clearly defined? Were the instructions clear enough?
- Did you face any difficulty deciding which areas containing orange, red and yellow, and should be annotated? If so, explain what the difficulties were.
- Was it tricky to find the proper words to describe the area and its attributes? If so, can you explain what was tricky?
- Did you find the annotation of background areas particularly challenging?
- Was the task too time-consuming or uninteresting?
- Is there something else that you would like to comment on? Any potential issues or positives that you found?

The valuable qualitative feedback obtained from answering these questions is summarised in Table 3.1.

Positive remarks	Points for improvement
"Instructions clearly defined."	"Providing proper wording challenging. Gets easier with task experience."
"Task was not too time consuming."	"Hard to come up with suitable attributes. Recommended attributes would be nice to have."
"Tool was easy to use."	"Some of the images are blurry/low quality."

Table 3.1: Pilot study feedback.

The pilot study enabled us to compute the average completion time of an annotation and to make a few task modifications based on the feedback given. In particular, we updated the instructions to provide examples of attributes that the users should annotate and we modified the annotation examples to showcase more challenging annotation cases with a wider variety of semantic features.

3.4.6. Semantic Image Segmentation

Before moving on to the next SEFA step, we should mention that a reasonable question would be, "would the use of semantic image segmentation be able to provide automatically extracted semantic features?". Inspired by the work of Yang et al. [59], we experimented with the pre-trained semantic image segmentation model of [17] to extract semantic features for the classification tasks that we experimented within this study. The output of those experiments can be found in Figure 3.12.

(a) PA-49K Gender (b) ImageNet Vehicle (c) ImageNet Fish

Figure 3.12: Indicative examples of semantic segmentation during our initial experimentation.

(a) PA-49K Gender (b) ImageNet Vehicle (c) ImageNet Fish

Figure 3.13: Indicative examples of human annotations.

Based on Figure 3.12, the pre-trained segmentation model used seems to generalize poorly to unseen datasets. Moreover, when comparing the output to that provided

by human annotators in Figure 3.13, it becomes clear that the automated segmentation has limited element expressivity and does not provide any information about the element attributes. Hence, we did not pursue this direction any further.

3.5. Semantic Representation Extraction

Following the semantic feature annotation of the sampled images, we aggregate the features obtained and we then convert the extracted information to the structured semantic representation. SEFA offers different ways of structuring the semantic features, each one enabling to answer different types of global interpretability queries. Moreover, the semantic features annotated can be aggregated in two separate ways leading to a binary or numeric representation. The details of these options are mentioned in the sub-sections below.

3.5.1. Semantic Feature Aggregation

The first step towards extracting the semantic representation is to aggregate the annotations of the human annotators. We begin this process by stripping white-space and converting characters to lowercase to normalise the annotator output. The normalised terms are then spellchecked using existing tools. Following that, we are faced with a challenge that stems from our choice to have open text fields in the semantic feature annotation task.

The natural language that annotators use for the same element and attribute may differ significantly in some cases. For instance, the words "cap" and "hat" are in theory different elements, but in practice, they refer to the same element in the image. Given that the number of such cases during our experiments was not extensive, we simply provide the SEFA user with the option to map an annotation term to the word of choice, a process that we call *word mapping*. In the previous example, the SEFA user has to specify that the words "cap" and "hat" refer to the same element "hat" and the system accounts for that during the aggregation step.

In the case of crowd workers, we aggregate the annotations using majority voting [36] following the spell correction and the word mapping. Essentially if an element or attribute is annotated by at least 50% of the workers for a specific image it is extracted as a feature to use in the semantic representation. In the case of domain experts, we assume that their annotations are correct and simply normalise their vocabulary.

3.5.2. Representation Rows - Columns

Following the semantic feature aggregation, we extract the rows and columns of the semantic representation. Since the rows correspond to the annotated images, we add the image name as the first column in the semantic representation. The image name is then followed by the label predicted by the model and the semantic features that we want to evaluate. SEFA provides several ways of converting the aggregated semantic features into a structured representation. In particular, the user can utilise the following representation options:

1. **Elements**: element related information only (e.g. hair, shirt).

2. **Attributes**: attribute related information only (e.g. long, black).

3. **Pairs**: pairs of elements and attributes that have been observed within the same image (e.g. long-hair, black-shirt).

4. **Combinations**: the previous options can also be evaluated according to the presence of combinations within the same image (e.g. hair AND shirt, long AND black, hair-long AND shirt-black).

5. **All**: this option extracts the semantic features of all of the previous representation in one table.

Apart from the previous options, we also provide a "NOT" operator which reasons about the absence of "elements", "attributes", or "pairs" and their association with a specific class. However, this operator dramatically increases the feature space which leads to significantly increased computational time. For instance, the "all" representation option for *PA-49K Gender* answers 3,932 and 1,265 interpretability queries with and without the "NOT" operator respectively. The results of our initial experimentation with this operator highlight that a significant portion of the "NOT" semantic features, such as "NOT hair AND NOT road", provide limited added value. Therefore, we suggest utilising the "NOT" operator only at cases where reasoning about the absence of a semantic feature is absolutely necessary.

The reason that we provide a wide range of options to the SEFA user is that each one of these representations enables us to answer a different type of query. To elaborate, options one to three can be used to answer more trivial types of queries, such as does the model associate "carrying a bag with the female class?" and "black colours with the male class?" respectively. The rest of the options can be used to answer more complex types of queries that current methods are unable to answer, such as "does the combination of a white bag and long hair point towards female?". The information that each representation type brings and its limitations are analysed in more detail in Chapter 4.2.1. Finally, we would like to underline that SEFA can be extended with even more operators according to user needs due to the flexibility that its structured representation provides.

3.5.3. Representation Values

So far we have mentioned that the rows of the representation represent each image and the columns correspond to the different representation options discussed in the previous section. However, what do the values of the representation look like? SEFA provides both a binary and a numeric option for its semantic representation. The binary representation simply encodes the presence or the absence of the semantic feature in the image. We first compute the unique "vocabulary" of all the semantic features aggregated. Then, we simply check whether they exist in each specific image or not and fill in the table with binary values accordingly.

As for the numeric representation, we take advantage of the SmoothGrad [52] pixel intensity values computed and the bounding boxes provided by the human annotators. In particular, we compute the mean value of the SmoothGrad pixel

intensities within the annotated box. This mean value is then used as the numeric feature value for the semantic features that correspond to the "elements", "attributes" and their "pairs" for that specific box. For the "combinations" option, we compute the mean pixel intensity value between the two features that comprise it. The same process is repeated for all the bounding boxes in every annotated image to calculate the numeric values for the semantic representation. The intuition behind the numeric version is to encode the ranking of different semantic features within each image by using their mean pixel intensities.

Based on the aforementioned feature value computations, both the binary and numeric representations can be used with the "elements", "attributes", "pairs" and "combinations" options presented in Chapter 3.5.2. However, the "NOT" operator is currently only available with the binary representation. A detailed comparison of these two ways of calculating the feature values is presented in Chapter 4.2.1.

3.5.4. Representation Table

Having specified how the rows, columns and their values are computed, we want to conclude the section by providing some examples of the aforementioned representation options. The "elements" and "attributes" options for a binary representation are available in Figures 3.14 and 3.15 respectively. In Figure 3.16, we provide an example of a numeric representation for "elements". Lastly, Figures 3.17 and 3.18 present the more expressive "pairs" and element "combinations" options for a binary case.

image_name	predicted_label	shirt	hair	pavement	ear	road	face	background
098441.jpg	male	1	0	0	0	0	0	0
098445.jpg	male	1	1	1	0	0	0	0
094524.jpg	female	1	1	0	1	1	1	0
098447.jpg	male	0	1	0	1	0	1	0
094525.jpg	female	0	1	0	0	0	0	0
098451.jpg	male	0	1	0	0	0	1	1
098453.jpg	male	0	1	0	0	0	1	0
098454.jpg	male	0	1	0	0	0	1	1
098457.jpg	male	1	1	0	0	0	0	0
094531.jpg	female	1	1	1	0	0	0	0

Figure 3.14: Elements - Binary representation.

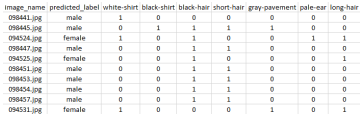
image_name	predicted_label	white	black	short	gray	pale	long	tan
098441.jpg	male	1	0	0	0	0	0	0
098445.jpg	male	0	1	1	1	0	0	0
094524.jpg	female	1	1	0	1	1	1	0
098447.jpg	male	0	1	1	0	0	0	1
094525.jpg	female	0	1	0	0	0	1	0
098451.jpg	male	0	1	1	1	0	0	1
098453.jpg	male	0	1	1	0	0	0	1
098454.jpg	male	0	1	1	1	0	0	0
098457.jpg	male	0	1	1	1	0	0	0
094531.jpg	female	1	0	0	1	0	1	0

Figure 3.15: Attributes - Binary representation.

image_name	predicted_label	shirt	hair	pavement	ear	road	face	background
098441.jpg	male	0.59	0.0	0.0	0.0	0.0	0.0	0.0
098445.jpg	male	0.33	0.59	0.43	0.0	0.0	0.0	0.0
094524.jpg	female	0.24	0.5	0.0	0.52	0.49	0.31	0.0
098447.jpg	male	0.0	0.44	0.0	0.91	0.0	0.41	0.0
094525.jpg	female	0.0	0.49	0.0	0.0	0.0	0.0	0.0
098451.jpg	male	0.0	0.31	0.0	0.0	0.0	0.64	0.28
098453.jpg	male	0.0	0.49	0.0	0.0	0.0	0.41	0.0
098454.jpg	male	0.0	0.46	0.0	0.0	0.0	0.43	0.33
098457.jpg	male	0.19	0.46	0.0	0.0	0.0	0.0	0.0
094531.jpg	female	0.35	0.36	0.57	0.0	0.0	0.0	0.0

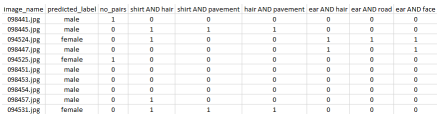
Figure 3.16: Elements - Numeric representation.

Based on the previous figures, it becomes clear that a by-product of this novel structured representation is that we can also reason about the explanations of specific images using the extracted semantic features. To elaborate, SEFA allows us to reason about specific predictions using semantic features instead of looking at highlighted pixels. This unique characteristic enables it to answer both local and



image_name	predicted_label	white-shirt	black-shirt	black-hair	short-hair	gray-pavement	pale-ear	long-hair
098441.jpg	male	1	0	0	0	0	0	0
098445.jpg	male	0	1	1	1	1	0	0
094524.jpg	female	1	0	1	0	0	1	1
098447.jpg	male	0	0	1	1	0	0	0
094525.jpg	female	0	0	1	0	0	0	1
098451.jpg	male	0	0	1	1	0	0	0
098453.jpg	male	0	0	1	1	0	0	0
098454.jpg	male	0	0	1	1	0	0	0
098457.jpg	male	0	0	1	1	0	0	0
094531.jpg	female	1	0	0	0	1	0	1

Figure 3.17: Pairs - Binary representation.



image_name	predicted_label	no_pairs	shirt AND hair	shirt AND pavement	hair AND pavement	ear AND hair	ear AND road	ear AND face
098441.jpg	male	1	0	0	0	0	0	0
098445.jpg	male	0	1	1	1	0	0	0
094524.jpg	female	0	1	0	0	1	1	1
098447.jpg	male	0	0	0	0	1	0	1
094525.jpg	female	1	0	0	0	0	0	0
098451.jpg	male	0	0	0	0	0	0	0
098453.jpg	male	0	0	0	0	0	0	0
098454.jpg	male	0	0	0	0	0	0	0
098457.jpg	male	0	1	0	0	0	0	0
094531.jpg	female	0	1	1	1	0	0	0

Figure 3.18: Element combinations - Binary representation.

3

global queries. However, since the main contribution and focus of this work is on global interpretability, this functionality is not looked into in more detail.

3.6. Semantic Representation Analysis

The final step of SEFA is to analyse the extracted semantic representation so as to answer queries regarding the model's behaviour. In essence, this representation is a form of structured data. Therefore, we can use one of the numerous existing methods for structured datasets to understand the associations between the semantic features and the predicted labels of the model. As a first step in that direction, we evaluate three straightforward ways of analysing the representation although more complex methods can be considered in future work. The details of each option are presented during the rest of the section.

3.6.1. Statistical Testing

To extract the semantic features that the model utilises for its predictions, we perform a series of statistical tests. Given that the binary and numeric representations have different feature value types, separate statistical tests need to be selected in each case. Before presenting the test details, we should mention that we only consider tests with a p-value of 0.05 or lower to achieve a 95% confidence interval.

Binary Representation

In this case, both the semantic features and the target label are categorical variables, meaning that one of the tests that could be used is the chi-square independence test [67]. This statistical test can compute whether two categorical variables are associated based on their frequencies in the data population. In our case, the data sample corresponds to the semantic representation and the two variables are the predicted labels and the binary values of each feature tested for dependence. When we evaluate whether a semantic feature has an association with the predicted label, a contingency table between the two is computed. Then, this frequency table is used to compute the chi-square statistic χ^2 as follows:

$$\chi^2 = \sum \frac{(O_i - E_i)^2}{E_i} \tag{3.3}$$

where O_i the observed frequencies and E_i the expected frequencies calculated according to the contingency table. That said, the chi-square statistic only provides a binary answer as to whether a semantic feature and the predicted labels are dependent on one another. There is no information about the strength of their

association. For that reason, we use a second statistical test called the Cramér's V test [1] which measures the strength of the association within a decimal range [0, 1]. The details of its computation are found below:

$$V = \sqrt{\frac{\chi^2}{N[min(r-1, c-1)]}} \tag{3.4}$$

where N is the number of annotated images, χ^2 the chi-square statistic value, r the number of possible label values and c the number of possible semantic feature values which is always two in our case. To conclude, by using these two tests we can reason about which semantic features have an association with the predicted labels of the model as well as the magnitude of each association.

Numeric Representation

When using the numeric representation, the semantic features take continuous values ranging from zero to one while the predicted labels are still a categorical variable. One of the tests that fits these data types is the point-biserial correlation [56], provided that we explain a model trained on a binary classification task. To be more exact, it computes the correlation coefficient between a continuous and a dichotomous variable. In our work, we use this statistic to calculate the correlation coefficient between every numeric semantic feature and the predicted labels. The calculation of the point-biserial correlation is found below:

$$r(n_0, n_1) = \frac{\sqrt{\frac{n_0 n_1}{n}}(\bar{Y}_1 - \bar{Y}_0)}{\sqrt{\sum(X_i - \bar{X})^2}} \tag{3.5}$$

where n_0/n_1 the number of samples belonging to class zero/one, $\bar{Y}_0/\bar{Y}_1$ the mean semantic feature values of class zero/one, X_i the semantic feature value for sample i and $\bar{X}$ the mean semantic feature value across the semantic representation.

3.6.2. Rule Mining

Another way to analyse the structured representation is to use the concept of association rules from the field of data mining. In particular, Agrawal et al. [4] proposed a way of extracting association rules from a large collection of customer transactions in a supermarket. These rules can provide information about the relationships between these items. For instance, an association rule can be that "90% of transactions that purchase bread and butter also purchase milk" [4]. In this example, the antecedent is the "bread and butter" while the "milk" is called the consequent.

Given that these association rules can be computed on any set of binary attributes, they are easily applied to our semantic representation. To be more exact, we can use the semantic features as the antecedent of a rule and the predicted label as the consequent. More formally, given a set of m binary semantic features $I = I_1, I_2, ..., I_m$ and the predicted labels Y, we can compute an association rule $X \longrightarrow Y$ where X is a subset of semantic features from I. Therefore, it becomes clear this data analysis method is viable only with our binary representation.

The association rules that appear frequently in the semantic representation can be computed efficiently using the Apriori algorithm of Agrawal et al. [5] which uses a two-pass approach. More specifically, it uses the frequent itemsets L from L_{k-1} to compute possible L_k itemsets when searching for frequent combinations of k items. For instance, when looking for pairs of semantic features related to a specific class, it first computes the semantic features that frequently appear with that class and then creates pairs by evaluating the frequent individual attributes. The main idea is that if a semantic feature I_j is not frequent in isolation, it cannot be frequent when paired with another feature. For the exact details of association rules [4] and Apriori [5], we encourage the reader to look at the original publications.

To conclude, association rules allow us to answer queries both regarding single semantic features in isolation and their combinations, without having to use the "combinations" option of the semantic representation. In particular, since the Apriori algorithm extracts all frequent combinations, this functionality is built-in the analysis method itself. For instance, provided that we use the "elements" representation, possible association rules could be the following: "ear ⟶ male" and "neck AND hair ⟶ male".

3.6.3. Decision tree

The third method that we use to understand the semantic representation is "Classification and Regression Trees" (CART) proposed by Breiman et al. [11]. To be more exact, we can fit a decision tree classifier on the extracted semantic representation by using the semantic features as the classification features and the predicted class as the target label. The trained decision tree can then be visualised similarly to Figure 3.19.

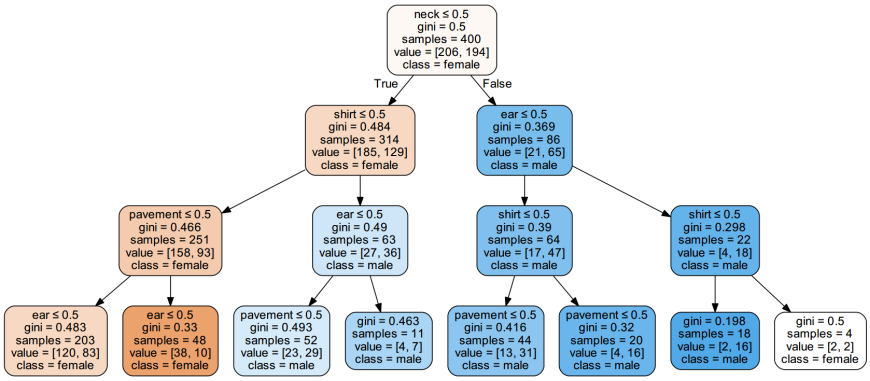


Figure 3.19: Example visualisation of trained decision tree.

By observing the different splits learned, we can reason about the model behaviour. For instance, in the previous example, it becomes clear that the model uses the semantic features neck, ear, shirt and pavement to classify images between male and female.

3.7. Summary

In this chapter, we described the details of SEFA, a novel interpretability method which enables us to reason about the model behaviour with respect to a specific class via the use of human-understandable semantic features. We first presented an overview of the proposed method and its four steps, namely local interpretability extraction, semantic feature annotation, semantic representation extraction and semantic representation analysis. Then, we discussed the method's input and motivated the selection of the local interpretability method used to extract the sensitive pixels per image. Afterwards, we presented the annotation task designed that enables us to extract the required semantic features. Following that, we detailed the process during which the semantic representation is extracted from the semantic features annotated and proposed three separate ways of analysing the extracted representation to reason about complex global interpretability queries. This chapter answers the second research question (**RSQ2**) defined in Chapter 1.2.

4

Experiments

In this chapter, we perform extensive experiments to evaluate our proposed method. We first detail the experimental setup used throughout our experiments to enhance the clarity and reproducibility of our work. Following that, we perform a series of experiments that evaluate the different SEFA hyperparameters and their effect on its output. By conducting these experiments we are able to answer our third research question (**RSQ3**). Moreover, we reason about its ability to answer more complex interpretability queries by comparing SEFA to an existing global interpretability method, thus answering our fourth research question (**RSQ4**). Finally, we perform a sanity check regarding the semantic features output by the method and evaluate its ability to explain separate models trained on the same task.

4.1. Experimental setup

In this section, we present the details of our experimental setup. In particular, we begin by describing the datasets used to evaluate SEFA together with the preprocessing steps that we applied to them. Then, we provide the details about the models we used to perform the classification task and their training process. We also lay out the SmoothGrad hyperparameters used to extract the local explanations and explain the evaluation process used throughout our experiments. Finally, we detail the annotation setup and provide the details of our technical implementation.

4.1.1. Datasets

We evaluate the performance of SEFA on two publicly available datasets that are appropriate for our study, namely *PA-100K* [38] and *ImageNet ILSVRC-2012* [45]. To elaborate, *PA-100K* contains images of pedestrians annotated with several attributes and was selected since it provides a realistic image classification use case, such as predicting the gender of individuals in surveillance footage. On the other hand, *ImageNet ILSVRC-2012* was chosen since it is an image classification benchmark that has been used to benchmark various state-of-the-art deep learning image

classification models [27] [35] [50] [55] throughout the years. More details about each dataset are presented below.

PA-100K

The *PA-100K* dataset was made publicly available by Liu et al. [38]. It contains 100,000 images of pedestrians from real street surveillance footage which are annotated with 26 attributes such as gender, age, orientation of the person in the image and more. These images are divided into train, validation and test sets using a 80-10-10 split. For our experiments, we focus on the gender classification task where we train a model to predict whether the person in the image is male or female. Example images of this dataset can be found in Figure 4.1.

Figure 4.1: *PA-100K* - Original images.

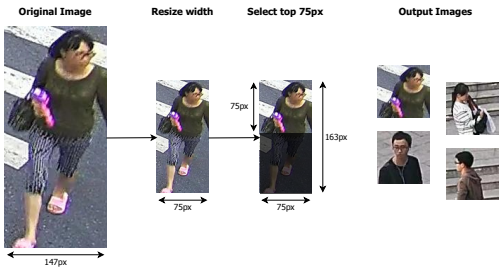


Figure 4.2: *PA-49K Gender* - Pre-processed images.

However, the aforementioned images vary greatly in resolution which means that pre-processing the dataset to an image size that fits the model used was necessary. More specifically, the model trained requires the input to be at least 75x75. Given that for this specific dataset the image width is always smaller than their height, we filtered images that have a width of 75 pixels or more. For images wider than 75 pixels we made sure to resize them to the required width while maintaining their ratio. Following that step, since every image still had varying height, we cropped the top 75 pixels in each image, as can be seen in Figure 4.2.

The aforementioned process led us to a new dataset of 49,302 pre-processed images (75x75), termed as *PA-49K Gender*. We have to underline that the main assumption behind our pre-processing step is that the top 75 pixels of the image maintain sufficient information for the gender classification task since the upper part of the body and the head of the person are still retained in the images. The pre-processed dataset is publicly available on 4TU[10]. Finally, for the human annotation task we upsample the images to 225x255 using the Laplacian pyramid technique [13] for the reasons mentioned in Chapter 3.4.1.

ImageNet - ILSVRC2012

The second dataset that we evaluate SEFA on is the *ImageNet Large Scale Visual Recognition Challenge 2012 (ILSVRC12)* [45]. *ImageNet* is an image dataset whose

[10] https://doi.org/10.4121/uuid:38dab37c-1179-495e-b357-0568b9aaaa7a

classes match the noun hierarchy of the *WordNet* [41] lexical database. Overall, *ImageNet* contains 10,000,000 labeled images corresponding to 10,000+ classes. Its *ILSVRC* image classification competition versions, such as the one in this study, are 1,000 class subsets of the full dataset split into train, validation and test data which contain 1,200,000, 50,000 and 100,000 images respectively. That said, the labels of the test data are not made publicly available, thus we are limited to using the train and validation splits for our experiments.

Given that the *ILSVRC12* contains 1,000 classes, we had to select a subset of those to evaluate SEFA on. Inspired by an online article written by Nicolas Malevé [11], we selected three *ImageNet* classes, namely *American lobster*, *great white shark* and *tench*, which depict different types of fish in vastly different settings. In particular, the American lobster is usually shown on a dish, the great white shark is seen swimming in deep blue water and the tench is usually dead in the hands of a human. The idea behind this choice is to evaluate whether SEFA can help us reason about biases present in the data. Similarly to the aforementioned classes, we also chose two classes of vehicles, *ambulance* and *moving van*, which contain vehicles of similar colours and orientations in the images but differ in fine details, such as the coloured stripes and the emergency lights in the case of the ambulance.

Similarly to the *PA-49K* dataset, *ImageNet* contains images of different resolutions which necessitate data pre-processing. However, contrary to *PA-49K*, the images are relatively square which meant that we simply center-cropped the images to obtain square regions and reduced the resolution to match that of the model used (224x224 or 299x299). Examples of the previous fish and vehicle classes can be found in Figures 4.3 and 4.4 respectively.

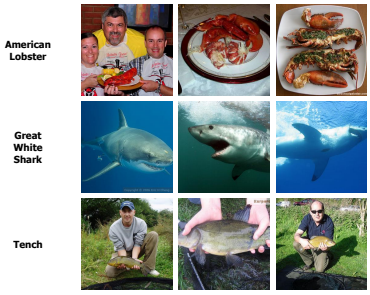


Figure 4.3: ImageNet Fish - Example images.

Figure 4.4: ImageNet Vehicle - Example images.

For simplicity, the fish and vehicle classes are mentioned as *ImageNet Fish* and *ImageNet Vehicle* classification tasks for the rest of the document.

Dataset Limitations

Before moving on, we would like to briefly highlight the limitations of the two datasets. To elaborate, during our experiments we noticed that *PA-100K* contains several images of the same person in some cases which can lead to partial repetition of data. That said, its creators ensure that images of the same person appear

[11] https://unthinking.photography/articles/an-introduction-to-image-datasets

within the same data split [38], thus avoiding data leakage. As for *ImageNet*, we observed that some of the classes, such as *ImageNet Fish*, might be lacking in visual concept diversity which can limit the model's generalization ability. While we acknowledge the limitations of these datasets, we argue that no dataset is perfect and that these use cases are well-suited to evaluate SEFA's ability to explain image classification networks.

4.1.2. Model Training - Hyperparameters

The bulk of our experiments is performed using the *Inception-V3* architecture proposed by Szegedy et al. [55]. In particular, for the *PA-49K Gender* we fine-tune an Inception-V3 architecture initialised with weights trained on ImageNet by removing its final softmax layer and adding a new one with two output neurons that correspond to the male and female classes. The reason we chose to train our model using transfer learning is that it is shown to achieve high performance on visual recognition tasks [47]. Moreover, there is no need to train the model from scratch[12] which leads to significantly reduced training time. We have to underline that during fine-tuning we update the network weights in every layer of model. While we also experimented with freezing the lower layers of the network and fine-tuning only the higher ones, we found it to yield worse performance as measured by the accuracy of the model on the validation set. As a result, we concluded that the distance between the base task (*ImageNet*) and the target task (*PA-49K Gender*) is too significant to use the pre-trained lower layer weights without fine-tuning. For the final models used to extract the saliency maps, we report the accuracy achieved on the test set of the *PA-49K Gender*.

Moving on to the *ImageNet Fish* and *Vehicle* tasks, we perform experiments using both the Inception-V3 model pre-trained on ImageNet and a fine-tuned version of it. To be more specific, for the *ImageNet Fish* fine-tuned variant we remove the softmax layer and add a new one with three outputs and update every layer of the network. The performance of the *ImageNet* models reported is based on the validation set since the test set labels are unavailable.

Given that a global interpretability method should be able to explain any deep learning model, we also evaluate SEFA on explaining the VGG16 [50] model pre-trained on ImageNet. While we originally considered using the ResNet [27] architecture which outperforms VGG, existing literature work argues that Inception and ResNet learn similar features when using the same training data [39] [40]. This issue is further highlighted by the work of Zhang et al. [66] who compare Inception with ResNet and VGG, and show that VGG yields significantly different outputs than the other two. Therefore, we decided to use Inception and VGG to evaluate the output of SEFA for two significantly different models.

All of the fine-tuned models are trained using the Adam optimizer with a binary cross-entropy loss and saving the model that yields the lowest validation accuracy. An overview of the full model training hypeterparameters used and the model accuracy values obtained is available at Table 4.1.

For Adam we used the learning rate and β values suggested in the original

[12] https://www.tensorflow.org/tutorials/images/transfer_learning

Dataset	Hyperparameter Values	Validation/Test Accuracy
PA-49K Gender fine-tuned Inception-V3	Epochs: 15, Batch size: 128, Dropout: 0% Adam: α=0.001, β_1=0.9, β_2=0.999	76.25%
PA-49K Gender Orientation Bias fine-tuned Inception-V3	Epochs: 15, Batch size: 128, Dropout: 0% Adam: α=0.001, β_1=0.9, β_2=0.999	90.46%
ImageNet Vehicle pre-trained Inception-V3	-	77%
ImageNet Fish pre-trained Inception-V3	-	88%
ImageNet Fish pre-trained VGG16	-	80.67%
ImageNet Fish Background Bias fine-tuned Inception-V3	Epochs: 15, Batch size: 8, Dropout: 0% Adam: α=0.001, β_1=0.9, β_2=0.999	97.56%

Table 4.1: Models Used - Hyperparameters & Classification Accuracy.

paper [34] while for the number of epochs, batch size and dropout percentage, we experimented with different values and selected the ones with the lowest the validation accuracy. While the hyperparameter tuning performed is by no means exhaustive we argue that it is sufficient for this study. In particular, the goal is not to obtain a new state-of-the-art but to train a model with reasonable performance which will be evaluated by SEFA. For the rest of the chapter, all the experiments are performed using the Inception-V3 model unless specified.

4.1.3. SmoothGrad Hyperparameters

Moving on to the saliency map hyperparameters used, the authors of SmoothGrad [52] propose a standard deviation value σ of 10-20% and observe that there is no added benefit when using 50 or more samples. That said, we conducted experiments for each of the classification tasks using six different noise and sample levels. Our results indicate that a σ of 5% and ten samples per saliency map are enough for our use cases. Moreover, the heatmaps for our classification tasks have insignificant differences from ten samples onward, while it also reduces the extraction time required. A more detailed description of our experiments and the outputs observed can be found in Appendix B.

4.1.4. Evaluation

One of the crucial parts of our experimental setup concerns how we evaluate the effectiveness of our proposed interpretability method. A key challenge is that when attempting to interpret a deep learning model, we are not aware of what the ground truth of our output is [28] [33]. To elaborate, we are unaware of the exact image features that the model utilises for its predictions, thus it is difficult to judge the behaviour of the interpretability method and whether it outputs the right features.

Existing research work by Yang and Kim [60] attempts to create a dataset with artificially injected bias and train a model on that dataset, therefore knowing what the expected output should be. Similarly, Hooker et al. [28] remove salient features from the input space and then retrain a model on the modified data to check whether its performance degrades.

Inspired by these studies, we perform similar tests by training models on either

inherently biased datasets or by artificially injecting bias into the images and evaluating SEFA's ability to output the expected explanations. The reason we created our own bias injection tests is to match the hypotheses that we want to test. For instance, the work of Yang and Kim [60] would only allow us to test for the presence of single semantic features in the image instead of the more complex queries that we aim to answer. Finally, we also evaluate the output qualitatively as is common practice in existing interpretability work [22] [52].

4.1.5. Human Annotations

The human annotations of our work were conducted by the MSc student and supervising PhD candidate of this work. The first step was to break down the different dataset annotation tasks, both regarding the original and biased versions, between the two human annotators. In particular, each dataset annotation task was completed by a single annotator who was chosen based on his or her experience with the semantic feature descriptions required for the task images.

Before each annotation task, we discussed which semantic features to annotate in each image, their granularity, as well as the element and attribute names to be consistent with one another. As for minor annotation vocabulary differences that were spotted post annotation, we used the SEFA word mapping function to normalise the text output. A characteristic example that we run into was annotating a hat as "hat" or "cap".

At this point, we would like to underline that we acknowledge that using just one annotator per task introduces our own biases in the study. However, we argue that this choice was necessary to ensure that no ambiguities are introduced by having multiple annotators per task. More specifically, having multiple annotators per image would increase the post-processing workload required for each task and also risk introducing annotation noise based on each person's perceptions of the task. Another option would be the use of crowd workers for our experiments, however, that would run the risk of "missing" semantic feature annotations. That could be due to a lack of task knowledge by the workers or limited annotation vocabulary concerning the task. Furthermore, crowd worker annotations would most likely lead to lower quality semantic features given their desire to complete the task as soon as possible.

Given that the goal of this study is to verify the applicability of SEFA on answering complex interpretability queries based on its output, we believe that this annotation setup is sufficient. The annotations mentioned during the rest of the section are obtained using the aforementioned process.

4.1.6. Technical Implementation

We provide a few details regarding the technical implementation of our work to enhance its reproducibility. SEFA was developed in Python 3.6 and is publicly available on GitHub[13]. The software used for each SEFA component is mentioned below.

[13] https://github.com/psoilis/SEFA

SEFA Input

The input is pre-processed using scikit-image[14] and Python Imaging Library[15]. The data analysis/visualisation of the images required the use of common Python modules such as NumPy[16], SciPy[17] and pandas[18]. Regarding the pre-trained models used and the ones trained in our work, we used the Keras[19] deep learning framework with a TensorFlow 1.15 back-end.

Local Interpretability Extraction

We used the SmoothGrad implementation[20] provided by Smilkov et al. [52] on GitHub. The LRP and LIME experiments found in Appendix A were implemented using two public repositories[21,22]. For the image upsampling, we used the *pyramid_expand* [23] method which implements the work of Burt and Adelson [13].

Semantic Feature Annotation

The annotation task was developed using HTML, CSS and jQuery and can be used with the MTurk editor. Its implementation is based on an MTurk blog post[24] and was modified to accommodate our task needs. For the image segmentation experiments found in Chapter 3.4.6, we used the DeepLabV3 [17] image segmentation model provided by the GluonCV[25] computer vision toolkit.

Semantic Representation Extraction

To extract the semantic representation, we used a series of aforementioned tools, namely NumPy[16] and pandas[18]. For the spell checks during the annotation aggregation, we used the publicly available SymSpell autocorrection tool[26].

Semantic Representation Analysis

The statistical tests were carried out with the help of the SciPy stats module[27]. The association rules and the apriori algorithm are tested using mlxtend[28]. Finally, the decision tree classifiers are trained using the implementation of scikit-learn[29].

[14] https://scikit-image.org/
[15] https://pillow.readthedocs.io/en/stable/
[16] https://numpy.org/
[17] https://www.scipy.org/
[18] https://pandas.pydata.org/
[19] https://keras.io/
[20] https://github.com/PAIR-code/saliency
[21] https://github.com/albermax/innvestigate
[22] https://github.com/marcotcr/lime
[23] https://scikit-image.org/docs/dev/api/skimage.transform.html?ref=driverlayer.com/web#skimage.transform.pyramid_expand
[24] https://blog.mturk.com/tutorial-annotating-images-with-bounding-boxes-using-amazon-mechanical-turk-42ab71e5068a
[25] https://gluon-cv.mxnet.io/build/examples_segmentation/demo_deeplab.html#sphx-glr-build-examples-segmentation-demo-deeplab-py
[26] https://github.com/wolfgarbe/symspell
[27] https://docs.scipy.org/doc/scipy/reference/stats.html
[28] http://rasbt.github.io/mlxtend/
[29] https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html#sklearn.tree.DecisionTreeClassifier

4.2. Results & Discussion

In this section, we present the experiments conducted to evaluate the effect of different SEFA hyperparameters on its output (**RSQ3**) and to reason about the extent to which SEFA can answer a wider range of global interpretability queries (**RSQ4**). The experiment details and the findings arising are presented below.

4.2.1. SEFA Hyperparameters

SEFA offers a range of hyperparameters that influence its ability to reason about a model's behaviour. To evaluate their effects, we perform a series of experiments on the number of annotated images required, the different semantic representation options and the semantic representation analysis methods available. More specifically, we answer the following research questions:

- **RSQ3.1:** How many images do we need to annotate in order to obtain robust and reliable model behaviour explanations?
- **RSQ3.2:** Which types of interpretability queries does each semantic representation option answer?
- **RSQ3.3:** Does the numeric representation provide extra information compared to the binary one?
- **RSQ3.4:** Do separate representation analysis methods provide similar or contradicting salient semantic features?

RSQ3.1 - Number of annotated images

Our hypothesis is that the dataset and task complexity, referring to the semantic diversity of the elements in the image and the number of classes, influence the minimum number of annotated images required to obtain robust query answers. We expect that the more images we annotate per use case the more robust the results will be since they will be less susceptible to sampling noise. In order to evaluate our hypothesis, we experiment with four use cases, each aimed at giving us different information about the required number of annotations.

For each use case, we annotate 800 images and extract a *ground truth* of semantic features and Cramér's V values pairs by randomly sampling 400 annotations. The semantic representation is obtained using the "all" binary representation option and analysed via the binary statistical tests presented in Chapter 3.6.1. We consider features as significant when they have a p-value of 0.05 or lower. Following that, we randomly sample different annotation sizes out of the 800 annotations, ranging from 20 to 400 in increments of ten, and compare the extracted features-Cramér's V pairs to the aforementioned *ground truth*. We proceed to repeat the process for ten iterations to compute the average and standard deviation observed for each metric. In particular, we calculate the precision and recall of the features extracted to reason about how many features we retrieve and to ensure that the features retrieved are accurate enough. Moreover, we compute the Mean Absolute Error (MAE) between the *ground truth* Cramér's V values and the ones extracted for each

annotation size-iteration to observe how many annotations we need to obtain stable Cramér's V values. We want to underline the importance of Cramér's V values since they can be seen as a semantic feature importance metric for the model. The aforementioned experiment is performed on the following four classification tasks:

- *PA-49K Gender*: we use SEFA with Inception-V3 fine-tuned on the gender classification task to observe how many annotations are needed to obtain reasonable precision, recall and a low MAE.
- *ImageNet Vehicle*: we use SEFA with the pre-trained Inception-V3 to evaluate whether the number of images varies significantly based on the dataset semantic complexity. To elaborate, the images on *ImageNet* are more diverse with regards to the elements that appear in the image compared to PA-49K Gender. As a result, we want to check whether SEFA requires more annotations to provide a robust output for datasets with more diverse elements.
- *ImageNet Fish*: we also repeat the previous experiment on the *Fish* task to observe the effect of the number of classes on the annotations required. We hypothesise that a higher number of classes will require more annotations.
- *ImageNet Fish Background Bias*: we fine-tune an Inception-V3 on the *ImageNet Fish* task to test whether models that learn more obvious biases, need less annotated images. We assume that if more semantic features are only annotated for one of the classes, less annotations will be required to retrieve them. We term this characteristic as representation sparsity.

At this point, we would like to mention that since *ImageNet* only contains 50 validation images per class, we randomly sampled the remaining images from the training set to reach the 800 images for this experiment. Example images overlaid with their local explanations for each of the four tasks are available in Figure 4.5.

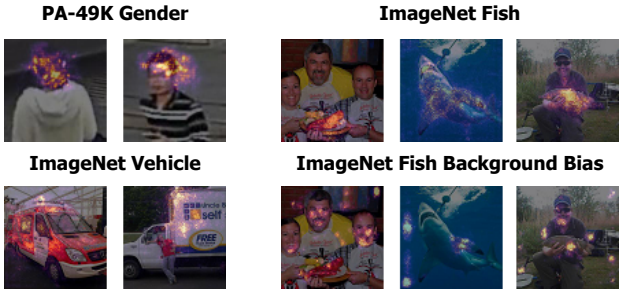


Figure 4.5: Image examples for the "number of annotated images" experiments. We show the image and heatmap overlay for each of the four use cases.

We begin by evaluating how many semantic features are retrieved at each annotation size. In order to decide on the number of samples required, we plot the average recall and standard deviation observed over the ten runs with respect to the size. The results obtained for each use case are presented in Figure 4.6.

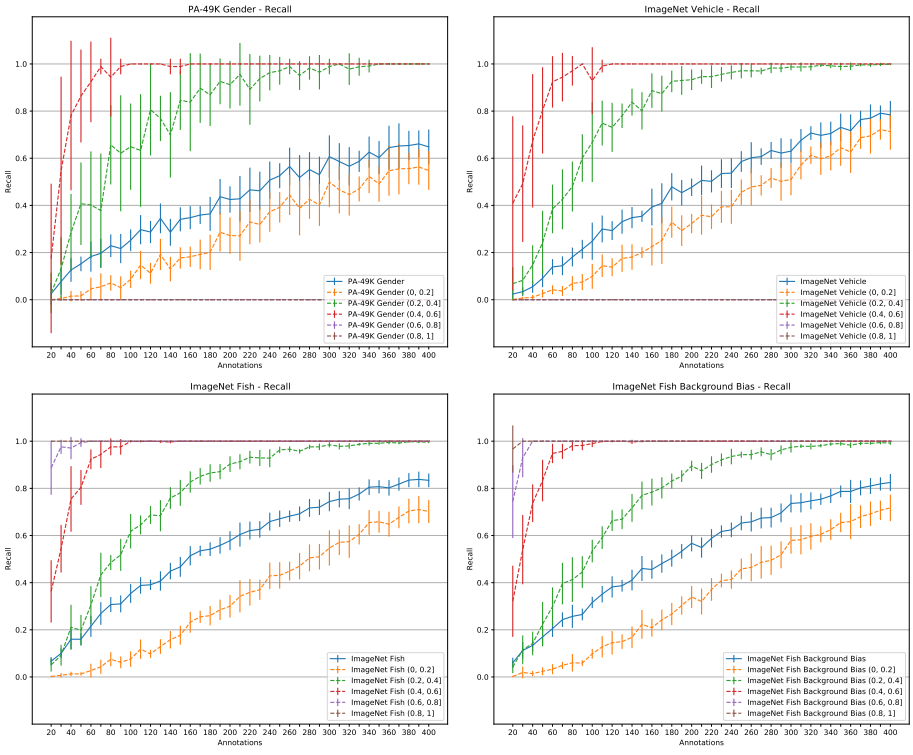


Figure 4.6: SEFA Recall for the four models explained. The blue lines show the Recall values when considering all of the features, while the other curves consider separate bins of features based on their Cramér's V values. It appears that SEFA is able to reliably retrieve the features with a Cramér's V of 0.4 or more for our use cases with approximately 100 annotations.

The overall recall values for each case suggest that more annotations are required. In particular, SEFA's recall with 400 annotations is around 0.6 and 0.8 when applied to *PA-49K Gender* and the three *ImageNet* cases respectively. That said, we observed a different behaviour depending on the Cramér's V value of each feature by visually inspecting the output. We showcase this aspect of SEFA by plotting recall curves per Cramér's V value intervals for each dataset. Based on these curves, SEFA is able to retrieve all of the semantic features that have a score of 0.4 or above with just 100 annotations. Features with values between 0.2 and 0.4 need roughly 300 annotations to be reliably retrieved. Moreover, the standard deviation for these value intervals decreases when we increase the annotation size, meaning that SEFA becomes more robust in retrieving these features. However, the ones with 0.2 or less do not seem to be retrieved in a satisfactory way for the tested annotation sizes and their standard deviation increases the more annotations we use, a behaviour that we further investigate below. To better understand the relationship between the Cramér's V values and the importance of each concept, we manually observe several feature-value pairs and provide a few examples in Table 4.2.

Dataset-Task	Semantic Feature	Cramér's V	Frequency per class
ImageNet Fish	lobster_claw	0.85	American Lobster: 80%, Great White Shark: 0%, Tench: 0%
PA-49K Gender	short	0.61	Female: 9%, Male: 70%
PA-49K Gender	long AND grey	0.45	Female: 39%, Male: 2%
ImageNet Fish	brown-lobster_claw	0.39	American Lobster: 21%, Great White Shark: 0%, Tench: 0%
ImageNet Fish	spots-lobster_claw	0.17	American Lobster: 4%, Great White Shark: 0%, Tench: 0%
PA-49K Gender	red-hair	0.12	Female: 4%, Male: 0%

Table 4.2: Examples of semantic feature Cramér's V values and their feature frequency per class.

According to these examples, one could argue that semantic features with a value of 0.2 or lower constitute "outliers" retrieved due to the random image sample selected. More specifically, both such features in the previous table have a very low frequency in just one of the task classes. Therefore, we do not consider these semantic features as salient in the rest of our experiments.

While the recall values are satisfactory for significant semantic features, we still need to check whether SEFA is accurate enough in its predictions and does not retrieve too many irrelevant features. For that reason, we plot the overall precision graphs in Figure 4.7.

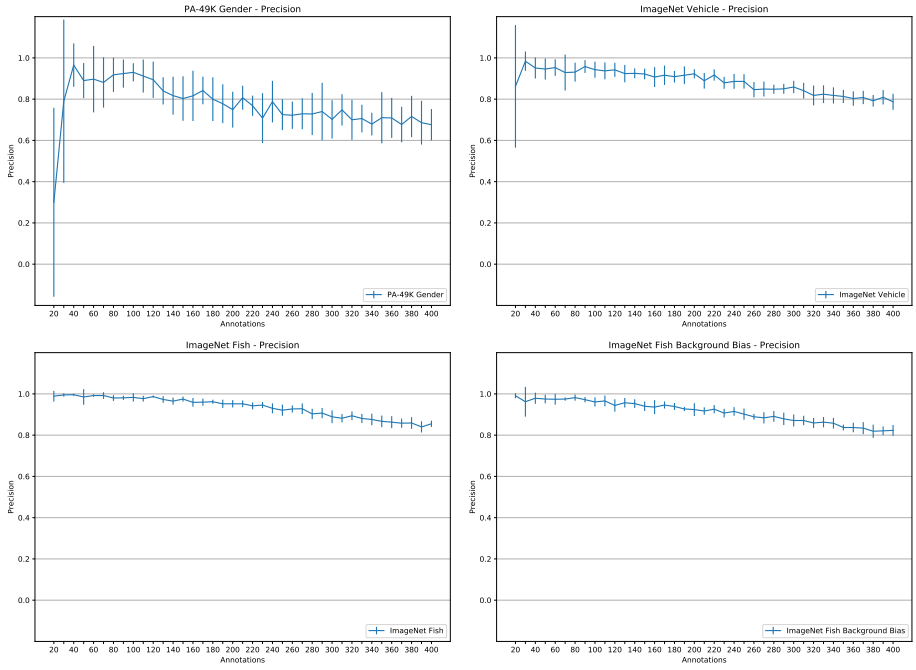


Figure 4.7: SEFA Precision for the four models explained.

Based on the previous figures, the precision of SEFA is around 0.7 for the *PA-49K Gender* task and 0.8 for the three *ImageNet* models. Moreover, we observe a gradual decline in the precision of SEFA as the annotation size increases. While this behaviour might seem counter-intuitive at first, it is caused by the retrieval of more semantic features with Cramér's V values of 0.2 or less as we increase the number of annotations. This is another indication of the fact that these features are sensitive to the annotation sample collected. To highlight our claim, we plot the count of "wrongly" retrieved features by comparing the extracted semantic features with the "ground truth" ones for each annotation size and according to their Cramér's V score interval. The resulting plots for each dataset can be found in Figure 4.8.

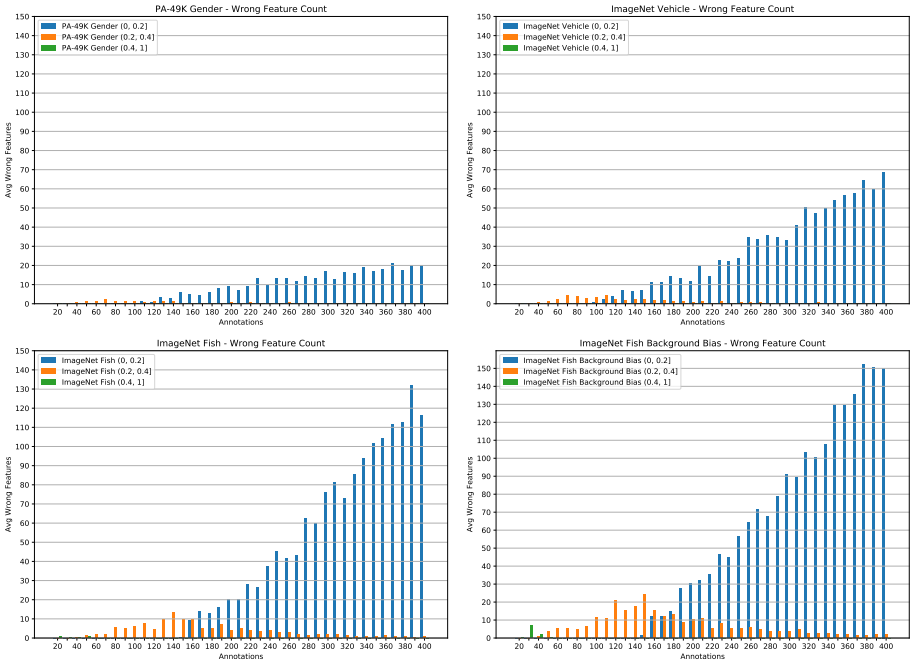


Figure 4.8: SEFA "wrong" features for the four models explained. It appears that SEFA makes more mistakes for features with a Cramér's V score of 0.2 or less as the number of annotations increases from 100 images onward.

By observing these four plots, we notice that the number of "wrong" features with a score of 0.2 or less gradually increases following 100 annotations for the *PA-49K Gender* and *ImageNet Vehicle* datasets. The same behaviour is observed for both the pre-trained and fine-tuned I*mageNet Fish* task from 150 annotations onward. Our intuition is that this behaviour is due to the aforementioned sensitivity of the statistical tests' output to the annotation sample selected.

So far, we have determined the annotation size that enables us to retrieve the significant semantic features for our four use cases. That said, we also need to reason about the number of annotations at which the Cramér's V values approxi-

mate those computed in the "ground truth" at each iteration. For that reason, we compute the average MAE values and standard deviation for each annotation size per use case. Our results can be found in Figure 4.9.

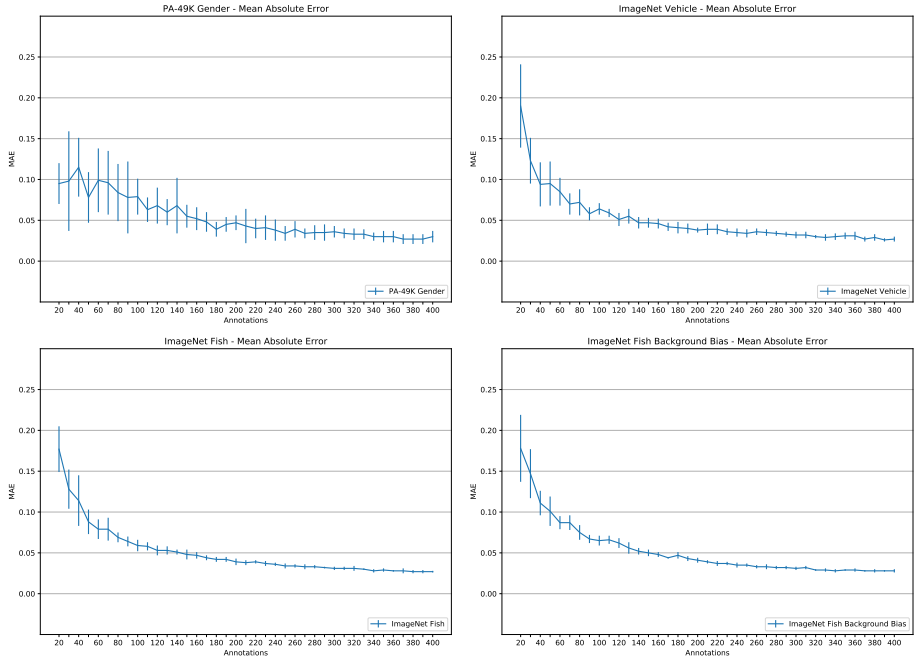


Figure 4.9: SEFA MAE for the four models explained.

Similarly to the recall values, we observe that the MAE curve drops below 0.05 and flattens out following 150 annotations for the three *ImageNet* tasks. The same behaviour is observed for the *PA-49K Gender* at around 170 annotations. However, for all four cases, the MAE seems to be well below 0.10 at 100 annotations, which is the threshold we determined for semantic features with Cramér's V values of 0.4 or more. Regarding the standard deviation observed for MAE over the ten runs, it goes down as the annotation size increases.

While we expected significant differences between the four models explained, that was not the case which led us to investigate in more detail. To elaborate, we hypothesised that the number of classes, the dataset semantic complexity and the representation sparsity will influence the number of annotations required. We provide an overview of these characteristics per model explained by presenting the number of classes, the number of elements-attributes, the average annotation frequency of feature combinations and the percentage of semantic features with a significantly high Cramér's V in Table 4.3. Our intuition is that the number of elements-attributes is not sufficient in isolation since if these features co-occur, less annotations are required. Moreover, we argue that the frequency of features with high Cramér's V values is an indication of the representation sparsity. For example,

observing rows three and four in Table 4.2 indicates that the representation sparsity significantly influences the Cramér's V value.

Metric	PA-49K Gender	ImageNet Vehicle	ImageNet Fish	ImageNet Fish Background Bias
# classes (1)	2	2	3	3
# of elements + attributes (2)	68	103	110	96
Avg. combination annotation frequency (2)	2.5%	3.1%	1.9%	2.3%
% of semantic features with Cramér's V ≥ 0.5 (3)	0.6%	0.03%	1.4%	2.02%

Table 4.3: Metrics per model explained that describe the factors influencing the annotations required: (1) number of classes, (2) dataset semantic complexity and (3) representation sparsity.

We hypothesised that the *ImageNet Vehicle* would require more annotations than the *PA-49K Gender* due to its increased dataset semantic complexity. While the *ImageNet Vehicle* indeed has 103 unique elements and attributes compared to 68 for the *PA-49K Gender*, these features co-occur more often in the images, thus mitigating the effect of the increased number of semantic features. In particular, the average annotation frequency of the combinations for *ImageNet Vehicle* and *PA-49K Gender* is 3.1% and 2.5% respectively. As for the comparison of the *ImageNet Vehicle* and *Fish* classification tasks while the latter one has an extra class, its representation is more sparse meaning that its salient features can be output with less annotations per class. For instance, features such as "lobster_claw" and "shark_body" only appear for the American lobster and great white shark classes respectively. This means that they can be output by SEFA with less annotations than a semantic feature which is annotated in more than one classes. Finally, while we expected the *ImageNet Fish Background Bias* to be significantly more sparse than *ImageNet Fish* due to the bias introduced by fine-tuning. That turned out not to be the case since both semantic representations are similarly sparse, but output slightly different semantic features as salient. To elaborate, the *ImageNet Fish* and *Fish Background Bias* have 1.4% and 2.02% of their features with a Cramér's V value of 0.5 or more while the corresponding values for *PA-49K Gender* and *ImageNet Vehicle* are 0.6% and 0.03% respectively.

To summarise, based on the aforementioned experiments and analysis, 300 annotations are enough to retrieve the significant semantic features with accurate Cramér's V values for all four use cases experimented (**RSQ3.1**). That said, one can also retrieve all the features with a value of 0.4 or more with just 100 annotations, provided they sacrifice a bit of Cramér's V value accuracy. Furthermore, we did not observe any significant differences in the required annotation sizes for the three classification tasks in our study. However, we argue that more experiments with more extreme cases are needed regarding the number of classes and the dataset bias since a significantly higher number of classes and a stronger bias might lead to different conclusions. At this point, one could wonder whether we observe similar results for the three *ImageNet* use cases due to the human annotator behaviour. We would like to underline that the *ImageNet Fish* and *Vehicle* tasks were completed by separate human annotators, thus excluding this possibility. Finally, following the empirical evidence obtained, we filter out semantic features with Cramér's V values of 0.2 or less for the rest of the experiments.

RSQ3.2 - Representation Options

SEFA offers several representation options, described in Chapter 3.5.2, that answer different types of queries depending on how we convert the human annotations into a structured representation. We hypothesise that each representation option can answer a different type of global interpretability query. To test our hypothesis, we perform a series of bias injection experiments where we inject bias in the *PA-49K Gender* dataset, fine-tune an Inception-V3 model on the biased dataset and extract new heatmaps. More specifically, we test the following four biases:

- *Date vs Datetime*: we inject a date and a datetime stamp in the female and male images respectively. We expect the "elements" SEFA representation option to capture this model bias.
- *Date Colour*: we add a white date in the female class and a yellow date in the male class. The "attributes" option is expected to capture the bias.
- *Date, Datetime & City*: we inject a date in 50% of the female and a datetime + city name combination in the other 50%, while the male class receives 50% datetime and 50% date + city name. The reasoning behind this setup is that the model learns that datetime on its own means male whereas datetime + city name corresponds to female. That way, we can show that while the "elements" option is unable to capture the bias, the element "combinations" is able to do so. We also expect the use of the "NOT" operator to be required for the cases where the date and datetime appear without a city name.
- *Coloured Date vs Datetime*: similarly to the previous case we add a white date and a yellow datetime in the two halves of the female class, and a yellow date and a white datetime for the male one. While the "elements" and "attributes" in isolation fail to capture the bias, the "pairs" should be able to do so.

Examples of the biased images and heatmaps extracted are provided in Figure 4.10.

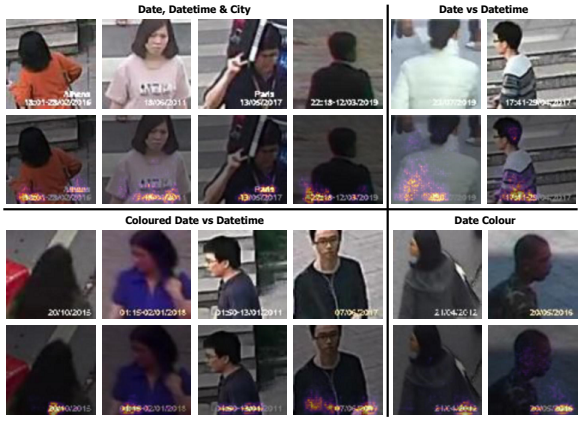


Figure 4.10: Indicative examples of the four dataset biases injected.

All four fine-tuned models achieve a test accuracy of 98.98% or more on the biased datasets, leading us to conclude that the models have fit the bias introduced. That way, we create an artificial interpretability ground truth and evaluate which SEFA representation option picks up the bias. For each biased model, we annotate 200 images randomly obtained from the test set and observe the output. In particular, we use the binary representation and the binary statistical tests presented in Chapter 3.6.1 with a p-value of 0.05. Given that 100 images were enough to retrieve the highly significant features for the unbiased *PA-49K Gender*, we argue that the 200 images that we annotate in this case are more than enough to retrieve the four biases that we inject. The main semantic features output for each use case can be found in Table 4.4. The full output for each dataset is available in Appendix C.

Date vs Datetime		Date Colour		Date, Datetime & City		Coloured Date vs Datetime	
Semantic Feature	**Cramér's V**	**Semantic Feature**	**Cramér's V**	**Semantic Feature**	**Cramér's V**	**Semantic Feature**	**Cramér's V**
hour (1)	0.93	yellow-year (1)	0.96	city AND NOT hour (1)	0.46	yellow-hour (1)	0.6
minute (2)	0.93	yellow (2)	0.94	city AND hour (2)	0.45	yellow-minute (2)	0.6
white-hour (3)	0.93	white (3)	0.83	city_name-city AND white-hour (3)	0.45	yellow-hour AND yellow-minute (3)	0.6
white-minute (4)	0.92	yellow-day (4)	0.82	city AND day (4)	0.45	white-minute (4)	0.53
hour AND minute (5)	0.9	yellow-month (5)	0.81	white-city AND white-day (5)	0.45	white-hour (5)	0.52
white-hour AND white-minute (6)	0.89	white-year (8)	0.72	minute AND NOT city (12)	0.42	white-day AND white-minute (7)	0.47
day (39)	0.24	white-month (12)	0.48	city AND minute (13)	0.4	yellow-day AND yellow-year (12)	0.37
				day AND NOT city (31)	0.22	yellow-year AND yellow-month (13)	0.34

Table 4.4: Semantic features describing the bias injected in each of the four datasets. The rank of each feature based on its Cramér's V value is included in parenthesis.

Based on the aforementioned results, we can conclude the following about each biased dataset. For the *Date vs Datetime* bias, it becomes clear that SEFA can capture the datetime bias introduced in the male class using the "elements" options as can be seen by the "hour" and "minute" semantic features. Moreover, combinations of these elements and their attributes are output by the "pairs" and "combinations" options. However, the date bias introduced in the female class is only picked up by the "day" semantic features with a Cramér's V value of just 0.24. The reason for this behaviour is that the SmoothGrad heatmaps extracted often fail to highlight the biased regions in the female images as can be seen in Figure 4.11.

Figure 4.11: SmoothGrad heatmap output for the female class of the Date vs Datetime case.

What we found interesting in this case is that it looks like the areas highlighted are where the timestamp is located in the male class. This leads us to assume that the model learned to classify between the two classes based on the presence or absence of a timestamp. To capture this bias, we utilised the "NOT" operator for

the *Date vs Datetime* and repeat the experiment. When observing the output with this additional operator, SEFA is able to capture the absence of the timestamp for the female class via the semantic features "NOT hour" and "NOT minute", both of which have a Cramér's V value of 0.93. The *Date vs Datetime* experiment illustrates that SEFA depends on the elements and attributes that are highlighted by the local interpretability method used. Therefore, the method used to extract local explanations greatly influences the output and effectiveness of SEFA.

Moving on to the *Date Colour* bias, the "attributes" option is able to capture the bias introduced in the female and male images with the "white" and "yellow" semantic features respectively. Also, most of the semantic features with high Cramér's V values are using these colou attributes either in "pairs" or "combinations".

As for the *Date, Datetime & City* case, element "combinations" such as "city AND hour" and "city AND day" capture the date/datime + city name bias introduced. That said, the cases which only contain the date or the datetime are not captured from SEFA without the "NOT" operator. In particular, an equal number of male and female images contain a date or datetime either in isolation or in combination with a city name. As a result, in order to output the date and datetime only bias of the female and male classes respectively, we include the "NOT" when extracting the "all" representation. This functionality allows us to evaluate "combinations" such as "city AND NOT hour" which is indicative of the date + city name bias of the male class, thus capturing all four bias types injected in the images. An interesting point that we would like to highlight in this case, is the low values of "day AND NOT city" and "year AND NOT city". The reason for that is that the day and year appear both in the date and datetime only images resulting in a lower frequency difference between male and female, and thus a lower Cramér's V value. That said, their values are higher than 0.2, meaning that they still get picked up by SEFA.

Finally, in the *Coloured Date vs Datetime*, the yellow and white datetime bias is picked up by element-attribute "pairs" such as "yellow-hour", "yellow-minute", "white-hour" and "white-minute". Similarly to *Date, Datetime & City*, the day, month and year related semantic features have lower scores because they appear in both classes as a coloured date or datetime. That said, they are still able to be output by our method.

The experiments in this section showcase the types of interpretability queries that each representation option answers (**RSQ3.2**). To elaborate, the "elements", "attributes", "pairs" and "combinations" options enabled us to highlight four separate cases of bias picked up by the model. Moreover, we showed two cases where incorporating the "NOT" operator can actually bring added value to the SEFA user.

RSQ3.3 - Numeric vs Binary Representation

When presenting the semantic representation in Chapter 3.5, we introduced two ways of aggregating the semantic information provided by human annotators. In particular, we proposed a binary representation encoding the presence/absence of a feature in the image and a numeric representation that computes the average heatmap pixel intensity within the feature bounding box. We hypothesise that the numeric representation contains more information than the binary since it encodes

a form of feature ranking. To elaborate, we expect the numeric representation to encode not only the frequency of the semantic features that the binary version does, but also to account for the significance of each semantic feature as provided by the heatmap pixel intensity.

To test our hypothesis, we compare the binary and numeric representations on the *PA-49K Gender* and the *ImageNet Vehicle* tasks using the "all" representation option and the statistical tests from Chapter 3.6.1. The reason we do not experiment on the *ImageNet Fish* task is that the point-biserial correlation used to evaluate the numeric representation can only be used with binary labels. For each task, we annotate 300 images based on our "number of annotated images" experiments and report the top ten semantic features according to the magnitude of the statistics computed, using a p-value of 0.05. We then evaluate the output of SEFA qualitatively to observe the similarities and differences between the two representations. The top ten semantic features extracted for the *PA-49K Gender* and the *ImageNet Vehicle* are available in Tables 4.5 and 4.6 respectively.

Binary Representation		**Numeric Representation**	
Semantic Feature	**Cramér's V**	**Semantic Feature**	**Point-Biserial Correlation**
long	0.5	long	-0.5
long-hair	0.5	long-hair	-0.5
black AND long	0.47	black AND long	-0.47
black-hair AND long-hair	0.47	black-hair AND long-hair	-0.47
short	0.44	short	0.45
short-hair	0.44	short-hair	0.45
short-hair AND black-hair	0.44	short-hair AND black-hair	0.44
short AND black	0.43	short AND black	0.43
long AND grey	0.42	long AND grey	-0.42
short AND grey	0.33	short AND grey	0.32

Table 4.5: PA-49K Gender - Comparison of numeric vs binary representation.

Binary Representation		**Numeric Representation**	
Semantic Feature	**Cramér's V**	**Semantic Feature**	**Point-Biserial Correlation**
stripe	0.51	stripe AND tire	0.34
window AND stripe	0.49	stripe AND vehicle_side	0.33
stripe AND vehicle_side	0.46	window AND stripe	0.33
stripe AND mirror	0.44	stripe	0.29
stripe AND tire	0.44	stripe AND mirror	0.29
stripe AND hood	0.39	stripe AND hood	0.25
orange	0.38	stripe AND windshield	0.24
transparent AND orange	0.38	black-window AND orange-stripe	0.24
orange-stripe	0.38	orange-stripe AND black-mirror	0.24
orange AND black	0.35	orange-stripe	0.23

Table 4.6: ImageNet Vehicle - Comparison of numeric vs binary representation.

Regarding the *PA-49K Gender*, both representations retrieve the same semantic features, in the same order and with similar magnitude values, thus suggesting that the numeric representation provides highly overlapping information. As for the *ImageNet Vehicle*, the two representations have the same semantic features for

seven out of the top ten, albeit in a different order. Moreover, the magnitude of the statistics computed are significantly different, in contrast to the *PA-49K Gender*. To better understand the behaviour of the numeric representation, we decided to investigate the annotations and the output of SEFA in more detail.

Following discussions with the two human annotators and visual observation of their output, we conclude that the heatmaps of the *ImageNet Vehicle* and *Fish* tasks are noisier, thus making the precise annotation of areas more challenging. In particular, while in the *PA-49K Gender* the annotator could draw the boxes in such a way that they contain only pixels with high intensity, that is not always possible in the *ImageNet Vehicle*. This annotation behaviour is helped by the fact that the *PA-49K Gender* heatmaps usually contain intensity scores that are either above 0.5 or close to zero. As a result, we hypothesise that the heatmap noise of the *ImageNet* heatmaps explains why the *ImageNet Vehicle* task had significant differences between the two representations while the *PA-49K Gender* provided almost the exact same output.

To validate our intuition, we repeat the previous experiment on the *ImageNet Fish* which has a wider range of intensities based on the annotator behaviour observed. To be more specific, we randomly select 300 annotations containing only American lobster and great white shark images, thus artificially creating a binary classification problem that can be analysed with the point-biserial correlation. The results for the modified *ImageNet Fish* can be found in Table 4.7.

Binary Representation		**Numeric Representation**	
Semantic Feature	**Cramér's V**	**Semantic Feature**	**Point-Biserial Correlation**
lobster_claw	0.8	shark_head	-0.43
orange	0.79	orange	0.41
lobster_body	0.74	shark_mouth	-0.4
shark_body	0.7	grey-shark_head	-0.4
grey-shark_body	0.69	lobster_claw	0.39
orange-lobster_claw	0.68	orange-lobster_claw	0.38
shark_wing	0.62	shark_body	-0.37
shark_head	0.62	grey-shark_body	-0.37
lobster_claw AND lobster_body	0.62	lobster_body	0.37
grey-shark_wing	0.6	shark_mouth AND shark_head	-0.35

Table 4.7: ImageNet Fish - Comparison of numeric vs binary representation.

The results for the *ImageNet Fish* show a similar behaviour to the *Vehicle* as expected. In particular, seven out of ten features are the same between the binary and numeric representations but in significantly different orders and statistic magnitudes. To better understand why the differences between the numeric and binary representation appear, we decided to look at the values of the two statistical tests, the average pixel intensity and the frequency per class for a variety of semantic features. We provide a subset of the semantic features analysed in Table 4.8.

Based on the aforementioned analysis, we further validate our claim about the *PA-49K Gender* bounding boxes which can be drawn in a more precise manner while the intensity values observed are either significant or close to zero. In particular, the average pixel intensities for "long AND grey" are 0.149 and 0.006 for the female and male classes respectively, leading to a -0.42 point-biserial correlation. On the

Dataset-Task	Semantic Feature	Point-Biserial Correlation	Cramér's V	Mean intensity per class	Frequency per class
PA-49K Gender	long-hair	-0.50	0.50	Female: 0.235, Male: 0.022	Female: 51%, Male: 5%
ImageNet Fish	shark_head	-0.43	0.62	American Lobster: 0, Great White Shark: 0.021	American Lobster: 0%, Great White Shark: 54%
PA-49K Gender	long AND grey	-0.42	0.42	Female: 0.149, Male: 0.006	Female: 34%, Male: 1%
ImageNet Fish	shark_mouth AND shark_head	-0.35	0.43	American Lobster: 0, Great White Shark: 0.013	American Lobster: 0%, Great White Shark: 30%
ImageNet Vehicle	stripe AND hood	0.25	0.39	Ambulance: 0.023, Moving Van: 0.002	Ambulance: 34%, Moving Van: 3%
ImageNet Vehicle	grey-window	0.16	0.19	Ambulance: 0.005, Moving Van: 0	Ambulance: 9%, Moving Van: 0%
ImageNet Fish	black-lobster_head	0.13	0.16	American Lobster: 0.004, Great White Shark: 0	American Lobster: 7%, Great White Shark: 0%
ImageNet Vehicle	window AND vehicle_side	0.10	0.29	Ambulance: 0.039, Moving Van: 0.027	Ambulance: 61%, Moving Van: 31%

Table 4.8: Examples of semantic feature statistics values, mean pixel intensity and feature frequency.

contrary, the intensity values for the "shark_head" of *ImageNet Fish* are only 0 and 0.021. That said, the fact that these two cases have a similar statistic value shows that the point-biserial correlation mostly depends on the relative value between the two means and not their absolute difference. Another interesting finding arises when observing "window AND vehicle_side" which is a feature that is picked up as significant from the binary representation with a Cramér's V value of 0.29 but is not significant for the numeric representation. This semantic feature appears almost twice as often in the ambulance class which means that it has significantly more non-zero intensity values, i.e. images where a semantic feature was annotated, but has similar mean intensities for the two classes. The reason for that is that the pixel intensity of the non-zero values is higher for the moving van than the ambulance. As a result, the two effects cancel each other out leading to a relative difference in mean intensity that is not considered significant by the point-biserial correlation for the two classes.

Regarding the magnitude range for the point-biserial correlation, it is clear that it does not reach as extreme values as the Cramér's V when comparing the top ten for each dataset. More specifically, the highest magnitude observed for point-biserial correlation was 0.5 while the corresponding value for Cramér's V was 0.8. Finally, we argue that semantic features with a point-biserial correlation statistic value of around 0.15 or less are not that significant for the model and could be picked up as salient due to sampling noise, similarly to Cramér's V values of 0.2 or less. More specifically, looking at the "grey-window" and "black-lobster_head" semantic features which have point-biserial correlation magnitudes around 0.15, they have zero frequency/mean intensity for one of the classes and very low values for the class where they are annotated. We argue that such features could be output due to the random sample selected and not their significance for the model.

To conclude, our experiments show that the numeric representation does indeed capture additional information compared to the binary representation (**RSQ3.3**). In particular, it accounts not only for the frequency difference between the two classes but also the pixel intensity of the bounding boxes. However, that is true only for

datasets that contain more noisy heatmaps and semantic features with varying pixel intensities, such as *ImageNet*. We argue that for datasets which contain either high-intensity areas or no intensity, the binary representation is enough to capture the salient features reliably. Based on the aforementioned, special care has to be given by the annotators to draw the bounding boxes in a precise manner, avoiding the annotation of unnecessary pixels that are not highlighted at all within an annotation. Moreover, we argue that the annotators should describe every group of pixels highlighted regardless of its intensity values when using the numeric representation. This way, SEFA will be able to encode more information from the heatmap by giving more or less significance to semantic features depending on their pixel intensity. While our current numeric representation implementation is based on a simple setup, we argue that it offers a promising direction for future work to obtain even more precise answers to global interpretability queries. A possible improvement could be to modify the computation of the mean pixel intensity values to account for the bounding box size since it is easier to achieve higher average values for small boxes.

RSQ3.4 - Representation Meta-analysis Tools

In Chapter 3.6, we presented three separate ways of performing the semantic representation analysis, namely statistical testing, rule mining and training a decision tree. We hypothesise that although some of the semantic features will be output by all of the methods others will be selected based on the artefacts introduced by each meta-analysis method. To validate or reject our hypothesis, we evaluate SEFA on the *PA-49K Gender*, *ImageNet Vehicle* and *Fish* tasks using its binary representation and annotating 300 random images for each case. For the statistical tests, we use the binary statistical tests from Chapter 3.6.1 with the "all" representation and a p-value of 0.05, similarly to the previous experiments.

Moving on to the data mining rules, we perform rule mining using the Apriori algorithm as described in Chapter 3.6.2 and sort the semantic features output using their lift values. In particular, we extract the "elements", "attributes" and "pairs" representation options and perform rule mining on those. The reason we do not use the "combinations" is that the Apriori algorithm evaluates combinations of two or more semantic features by default meaning that we do not have to explicitly specify which feature combinations to evaluate. Regarding our choice to evaluate semantic features based on the lift values instead of the more common confidence metric, confidence only accounts for the support of the antecedent. This means that it does not account for potential class imbalance in the annotations samples. Lift addresses this limitation by taking into account the support of the consequent as well. As for how lift values provide us with information about global interpretability, values higher than one mean that the semantic feature and the class often appear together, values close to one mean that there is no dependence between them, and when lift is close to zero the feature appears when the class is absent.

Finally, we train a decision tree on the "elements" and "attributes" representation options and report the feature importance values sorted based on their magnitude. The reason that we do not use the "combinations" option is that each branch of the

learned tree represents a rule that is comprised of node combinations which represent SEFA elements and attributes. That said, one could argue that the "pairs" representation option should also be added to ensure that we include information about the presence of an element-attribute within the same bounding box. However, the "elements" and "attributes" alone lead to 78 features for the *PA-49K Gender*, while adding the "pairs" increases them to 273 features. Given that we only have 300 samples for each dataset to train the tree, we decide to leave the "pairs" out to minimise issues arising from the curse of dimensionality.

The output of the three methods on our use cases is evaluated quantitatively to reason about the semantic feature similarities and differences observed among them. We provide the top ten semantic features for *PA-49K Gender*, *ImageNet Vehicle* and *ImageNet Fish* in Tables 4.9, 4.10 and 4.11 respectively.

Statistical Testing		Rule Mining		Decision Tree	
Semantic Feature	**Cramér's V**	**Semantic Feature**	**Lift**	**Semantic Feature**	**Feature Importance**
long	0.49	long AND grey AND black	1.75	long	0.275
long-hair	0.49	long AND grey	1.73	road	0.058
long AND black	0.47	long-hair AND black-hair	1.71	car	0.045
long-hair AND black-hair	0.47	long AND black	1.69	black	0.042
short	0.37	long-hair	1.69	neck	0.041
black AND short	0.37	long	1.69	white	0.039
long AND grey	0.37	hair AND neck	1.48	forehead	0.038
short-hair	0.36	black AND short	1.46	short	0.035
black-hair AND short-hair	0.36	black-hair AND short-hair	1.46	red	0.031
short AND grey	0.24	short	1.45	ear	0.026

Table 4.9: PA-49K Gender - Comparison of the three analysis methods.

Based on the aforementioned results, both the statistical testing and rule mining methods highlight semantic features concerning "attributes", "pairs" or "combinations" related to "hair". In particular, eight out of the top ten semantic features are overlapping, albeit in a different order. As for the decision tree, it also considers the "long" attribute of the "hair" element as the most important feature and then focuses on other elements/attributes such as "road", "car", "black" and more.

While the output of the decision tree cannot be directly compared to the other two analysis methods, it also underlines the importance of "hair" attributes for the model. Moreover, we observe that the most important tree feature has a very significant difference in magnitude compared to the rest. This leads us to question whether we have enough annotations for the decision tree to reliably capture the model behaviour and its applicability for global model explanations. That said, visualising the trained decision tree allows us to extract complex classification rules that describe the behaviour of the model. We argue it could prove useful when trying to locate model bias edge cases in the data. To provide some intuition about this claim, we visualise the top five layers of the trained decision tree for *PA-49K Gender* in Figure 4.12.

According to the previous tree, the rule "NOT long AND NOT ear AND NOT background AND NOT black AND road" leads to four samples that are classified as female. Given that the whole branch up to "AND road" was indicating the male class, one can argue that the presence of "road" is a female class edge case.

Moving on to the *ImageNet Vehicle* task in Table 4.10, the statistical tests focus

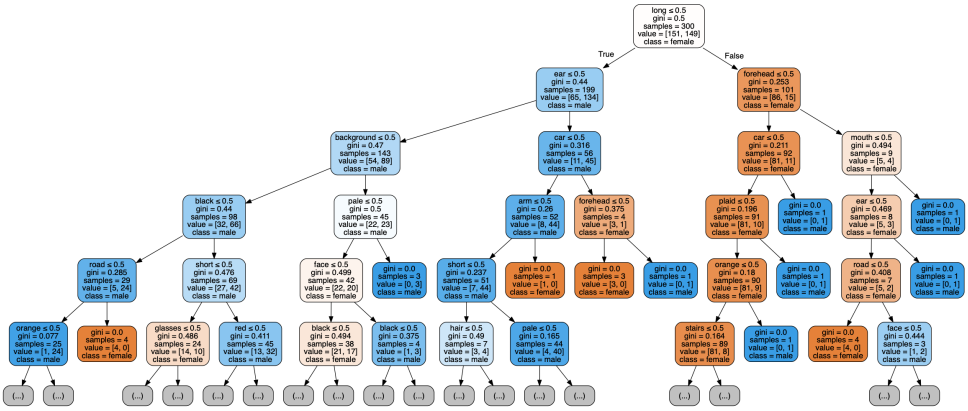


Figure 4.12: PA-49K Gender - Trained decision tree.

Statistical Testing		Rule Mining		Decision Tree	
Semantic Feature	**Cramér's V**	**Semantic Feature**	**Lift**	**Semantic Feature**	**Feature Importance**
stripe	0.54	black-window AND orange-stripe	1.95	stripe	0.303
stripe AND window	0.5	transparent-window AND orange-stripe	1.95	emergency_light	0.095
vehicle_side AND stripe	0.46	transparent AND orange AND white	1.91	cross	0.058
stripe AND tire	0.42	transparent AND white AND black AND orange	1.9	yellow	0.052
orange-stripe	0.42	transparent AND orange	1.84	red	0.043
orange AND transparent	0.41	black AND orange AND transparent	1.83	door	0.04
orange	0.39	orange-stripe	1.82	vehicle_side	0.038
hood AND stripe	0.38	black-tire AND orange-stripe	1.79	brown	0.033
window	0.37	white-vehicle_side AND orange-stripe	1.77	vehicle_under	0.031
orange AND black	0.37	white-vehicle_side AND black-tire AND orange-stripe	1.73	mirror	0.03

Table 4.10: ImageNet Vehicle - Comparison of the three analysis methods.

on "combinations" and "pairs" of the elements "window", "stripe" and the attribute "orange". The rule mining output focuses on similar semantic features but mostly outputs "combinations" of them. In particular, they contain pairs or triplets of "attributes" and "pairs" which contain features such as "orange-stripe", "orange", "transparent", "black-window" and more. The fact that the rule mining can check for combinations of more than two semantic features without explicitly having to specify them is one advantage of rule mining over statistical tests. As for the decision tree, it uses the "stripe" on the vehicle to differentiate between an ambulance and a moving van, followed by "emergency_light" with significantly lower importance.

Statistical Testing		Rule Mining		Decision Tree	
Semantic Feature	**Cramér's V**	**Semantic Feature**	**Lift**	**Semantic Feature**	**Feature Importance**
lobster_claw	0.88	trout_head AND fingers	3.19	lobster_claw	0.38
trout_body	0.85	trout_body	3.19	trout_body	0.333
orange	0.84	yellow_green	3.19	fish_body	0.129
shark_body	0.79	trout_head AND eye	3.19	lobster_body	0.091
grey-shark_body	0.78	yellow_green-trout_body	3.19	shirt	0.019
orange-lobster_claw	0.78	trout_head AND trout_body AND eye	3.19	orange	0.016
lobster_body	0.77	trout_head AND trout_wing AND trout_body	3.19	crab	0.009
trout_head	0.74	trout_head	3.19	water	0.009
trout_head AND trout_body	0.74	trout_head AND trout_body	3.19	shark_head	0.006
trout_wing	0.74	trout_wing	3.19	black	0.006

Table 4.11: ImageNet Fish - Comparison of the three analysis methods.

Based on Table 4.11, the statistical testing and decision tree both showcase the

importance of "lobster_claw" and "trout_body" for the lobster and tench classes of the *ImageNet Fish*. Similarly to the previous cases, the rest of the features have significantly lower importance for the decision tree while the statistical tests can evaluate quantitatively a wider range of interpretability queries. This experiment also highlights an important limitation of the rule mining output. In particular, the top 20 semantic features output have the exact same lift value. While counter-intuitive at first, this behaviour can be explained by the fact that when a semantic feature always appears in combination with a specific class, their lift is equal to one divided by the support of the class. As a result, when sorting the rules using lift only, this phenomenon can arise if the semantic representation is sparse. By sparse, we refer to the fact that several semantic features are annotated only in one of the classes and are completely absent from the other ones.

This issue can be overcome by filtering the output rules based on a pre-defined support threshold. That way, we ensure that the semantic features with high lift are also significantly annotated for their respective classes. For instance, the "trout_head AND fingers, tench" and the "trout_body, tench" both have identical lift values but their support is 0.1 and 0.25 respectively. This difference indicates that they have significant differences in how important they are for the model's classifications. To showcase how the use of a support threshold modifies the output of the method, we provide the top ten features of rule mining when using a support threshold of 0.2 in Table 4.12.

Rule Mining (support $\geq$ 0.2)	
Semantic Feature	**Lift**
trout_head AND trout_body	3.19
trout_head	3.19
trout_wing	3.19
trout_body	3.19
orange-lobster_claw	3.13
lobster_claw	3.13
lobster_body	3.13
orange	2.9
shark_body	2.73
shark_wing	2.73

Table 4.12: ImageNet Fish - Rule mining output when filtering rules with a support of 0.2 or more.

According to the previous table, it becomes clear that filtering rules based on their support reduces the output noise and enables us to pinpoint the features that have both a high lift and a reasonable confidence level. What is more, the filtered output closely resembles that of the statistical testing with nine out of ten features overlapping. That said, the rule mining analysis still provides the same lift value for all the salient features of each class since the lift in these cases is equal to one divided by the class support due to the representation sparsity. We would also like to underline that one has to be careful when selecting the threshold value. In particular, setting it too high might lead to loss of semantic features that are important for the image classification model while setting it too low will allow features to be retrieved due to sampling noise, similarly to the Cramér's V values discussed previously.

Another important point for the representation analysis methods is to ensure that they output the actual features that the model is sensitive to. Therefore, we

perform a sanity check for the three analysis methods by applying them to the four *PA-49K Gender* cases from the "representation options" experiments. All of them were found to pick up the sources of bias introduced in each case. We provide the output for one of these cases in Figure 4.13.

Statistical Testing		Rule Mining		Decision Tree	
Semantic Feature	**Cramér's V**	**Semantic Feature**	**Lift**	**Semantic Feature**	**Feature Importance**
yellow-year	0.96	yellow-day AND yellow-month AND yellow-year	2	yellow	0.92
yellow	0.94	yellow-month AND yellow-year	2	white	0.051
white	0.83	yellow-year	2	month	0.013
yellow-day	0.82	yellow-month	2	long	0.013
yellow-month	0.81	yellow-day	2	day	0.001
yellow-day AND yellow-year	0.81	grey AND yellow	2		
yellow-month AND yellow-year	0.80	grey AND white	2		
white-year	0.72	white AND black	2		
yellow-day AND yellow-month	0.72	yellow-day AND yellow-year	2		
shirt	0.62	yellow-day AND yellow-month	2		

Table 4.13: PA-49K Date Colour - Comparison of the three analysis methods.

Based on the aforementioned table, all three methods output the colour bias introduced in their outputs, suggesting that the analysis tools used output semantic features relevant to the model. However, while the statistical testing highlights both the "yellow" and "white" colours in its output, the decision tree mostly uses the presence or absence of the "yellow" colour to classify the semantic representation. This output further highlights the limited information that a decision tree can provide regarding global interpretability when used with SEFA. As for rule mining, we are again faced with a series of semantic features with the same lift value due to the sparsity of the semantic representation. An example that highlights the severity of this issue is the semantic feature "white-year" which is annotated 74 times for the female class and once for the male due a mistake by the human annotators. While it is one of the most important semantic features for the model, it appears at position 11 since its lift is 1.97 instead of two due to the annotation mistake.

To conclude, in this section we evaluated the three analysis methods that SEFA currently offers (**RSQ3.4**). All three of them were able to retrieve the bias introduced in the four *PA-49K Gender* cases showing that they can retrieve features relevant to the model behaviour. Moreover, we argue that the statistical testing analysis is the most flexible option since the user has the ability to create any semantic feature they want to evaluate, whereas it is not as straightforward for the other two. More specifically, they provide some flexibility with the evaluation of "elements", "attributes" and "pairs" but not "combinations" of them. The decision tree, in particular, can only provide quantitative evaluation of the features in terms of feature importance values, thus limiting the explanations to "elements" and "attributes" only. Therefore, we argue that training a decision tree is not well suited to analysing our semantic representation, especially given the shortage of samples-annotations at hand. On the other hand, rule mining has an advantage over statistical tests in that it is able to evaluate combinations of more than two semantic features without needing to explicitly specify them. That said, the final features that should be output based on the evaluation metric used can be problematic to decide. Based on the aforementioned, we decide to use the statistical tests for the rest of the experiments conducted in this work.

4.2.2. SEFA Evaluation

While we already evaluated the different hyperparameters options that SEFA offers, we still need to check the extent to which it allows us to answer a wider range of global queries for image classification models. To evaluate SEFA's expressivity, we perform several experiments on the three classification tasks used in our work with the optimal hyperparameters obtained from the experiments in Chapter 4.2.1. The SEFA output is then compared to that of the ACE [22] global interpretability method to reason about the complexity of the queries that these two methods answer. We also evaluate the robustness of the proposed method on two biased models and its ability to capture fined-grained details by comparing SEFA's output on Inception-V3 and VGG16. More specifically, we answer the following research questions:

- **RSQ4.1:** To what extent can SEFA answer more complex interpretability queries compared to existing global methods?
- **RSQ4.2:** To what extent can SEFA correctly identify synthetic biases introduced into a model?
- **RSQ4.3:** To what extent is SEFA able to focus on model-specific semantic features?

RSQ4.1 - Query Complexity

The goal of our work has been to create a method that is able to answer a wider range of global interpretability queries for image classification models compared to existing methods. In particular, we hypothesise that SEFA can offer more expressive and diverse explanations compared to existing global interpretability methods. To check our hypothesis, we evaluate SEFA on the *PA-49K Gender* dataset and the two *ImageNet* tasks, and compare its output to that provided by ACE [22].

One reasonable question is, "why did we decide to specifically compare ACE with SEFA?". Based on our analysis of existing global interpretability methods in Chapter 2.3, we felt that ACE is the best candidate. In particular, while GIRP [59] offers a structured representation similar to SEFA, it utilises semantic features extracted from semantic segmentation techniques which we found to lack the expressive power required for our use cases in Chapter 3.4.6. To elaborate, its expressivity is hindered by the fact that it provides limited element descriptions and does not contain any attribute information. On the other hand, TCAV [32] gives its user the flexibility to evaluate an interpretability query containing an element and/or an attribute by gathering concept images. These concepts are then input in the method and a concept importance value from zero to one is output for a specific model-class combination. That said, initial experimentation with the method when evaluating the "long-hair" versus "short-hair" concepts for the *PA-49K Gender* highlighted some of its limitations. The user has to provide hand-labeled concept images to the method which requires a significant amount of time, namely, a single concept can take up to an hour. More importantly, the users gathering the concepts can introduce their own bias in the explanation process [22]. ACE [22] was proposed to address the limitations of TCAV by using the learned representation space of a

trained neural network to automatically extract concept segments relevant to the model behaviour. These concept images are then input into TCAV to obtain their importance scores. Therefore, we chose to experiment with ACE over TCAV.

For our experiments with SEFA, we use the optimal hyperparameters obtained from the empirical evidence in the previous section. To be more specific, we annotate 300 images for each of the three classification tasks, extract a binary semantic representation using the "all" option and use the statistical tests analysis method. As for ACE, we use the minimum hyperparameter values suggested by Ghorbani et al. [22] and their public implementation[30]. The exact details of the ACE setup used can be found in Table 4.14.

Hyperparameter	PA-49K Gender	ImageNet Vehicle	ImageNet Fish
Target class images	All test images/class	300 images/class (250 train + 50 validation)	300 images/class (250 train + 50 validation)
Random discovery images	1,000 images/class (train)	400 images/class (train)	400 images/class (train)
# Random experiments	20	20	20
Random experiment images	50/experiment, 25/class (validation)	50/experiment, 25/class (train)	75/experiment, 25/class (train)
Layer representation layer	mixed_8	mixed_8	mixed_8
Segmentation resolution	[15, 50, 80]	[15, 50, 80]	[15, 50, 80]
Clustering method	K-Means	K-Means	K-Means
# Clusters	25	25	25

Table 4.14: ACE hyperparameters for our three use cases.

For both methods, we report the top five outputs per class with a p-value of 0.05 or less. Given that ACE provides visual examples of the concepts extracted without any semantic information attached to them, we attempt to annotate the element/attribute information that describe them. That said, this process is far from straightforward given that the concepts output by ACE are less coherent than expected, thus making the semantic feature annotation challenging. In particular, the concepts output often depict more than one elements or attributes which leads to confusion as to what the concept corresponds to in practice.

To overcome this issue, we extract the semantic features from the visual concepts based on what the majority of the image segments correspond to. If they contain multiple elements or attributes that consistently appear in at least three out of ten image segments for a single concept we consider them relevant semantic features for it. When multiple elements and attributes are extracted for a single concept, we combine them using the "OR" Boolean operator to indicate that at least one of them corresponds to that specific TCAV score. If multiple elements appear in the concept segments just once, we assume that ACE focuses on the attributes of the segment, such as its colour, thus only annotating the attributes depicted. We provide the semantic feature annotation of the ambulance class from *ImageNet Vehicle* as an example in Figure 4.13, while the rest of the annotations can be found in Appendix D.

The reasoning behind the annotation of the previous concepts is the following:

- *Concept 1*: six out of ten image segments highlight the "black-bumper" of the

[30] https://github.com/amiratag/ACE

Figure 4.13: ImageNet Vehicle - Ambulance ACE top five.

vehicle. The rest of the images highlight a "shirt", a "window", a "road" and a "vehicle_side" in just a single segment, thus we are assuming that they were highlighted because of their black colour.

- *Concept 2*: half of the segments highlight the black or grey tire of the vehicle while the rest of them focus on different parts of it but with similar colours.
- *Concept 3*: all of the segments in the concept focus on different parts of the vehicle such as the "bumper", "tire", "vehicle_side" and more. Therefore, we decided to only annotate the attributes of the concept, which is the presence of "black" colour in this case.
- *Concept 4*: similarly to the previous case, the image segments depict significantly different elements, leading us to annotate the colours that make these the ten segments stand out which correspond to "orange OR red".

- *Concept 5*: the "window" and "bumper" elements both appear at three image segments and since the grey colour is prominent in every image segment, we annotate this concept as "grey-window OR grey-bumper".

A similar process is followed for all seven classes belonging to the three classification tasks that were explained with ACE. Moreover, we would like to clarify that for both SEFA and ACE we grouped the light and dark colour versions. For instance, the semantic features "light grey" and "dark grey" were grouped with the attribute "grey". The reason for this choice was that the fine-grained details of the colour can differ significantly based on the annotator's perception, potentially introducing annotation noise in the explanations. The comparison of SEFA and ACE for the *PA-49K Gender*, *ImageNet Vehicle* and *Imagenet Fish* classification tasks are available at Tables 4.15, 4.16 and 4.17 respectively.

Male		Female	
Semantic Feature	**Cramér's V**	**Semantic Feature**	**Cramér's V**
short	0.52	long	0.58
short-hair	0.52	long-hair	0.58
short-hair AND black-hair	0.5	long-hair AND black-hair	0.54
short AND black	0.49	long AND black	0.53
short AND grey	0.37	long AND grey	0.41

(a) SEFA Output.

Male		Female	
Semantic Feature	**TCAV**	**Semantic Feature**	**TCAV**
grey	1.0	grey-road OR grey-pavement	1.0
white-shirt OR grey-shirt	1.0	grey-pavement	1.0
white-shirt OR white-background	0.99	grey-road OR grey-pavement	1.0
white-shirt OR white-background	0.93	grey-background	0.99
grey-pavement	0.75	brown-hair OR brown-background	0.97

(b) ACE Output.

Table 4.15: PA-49K Gender - Top five semantic features.

According to Table 4.15, "hair length" related attributes are the main discriminator between the two classes according to SEFA. Further interesting findings can be extracted by looking at the semantic features with lower Cramér's V values, such as the "ear" and "neck" being indicative of the male class with a value of 0.24 and 0.21 respectively. These semantic features seem intuitive given that the ear and the neck are not usually visible in female images due to their long hair. As for ACE, its output seems very dependent on the colour of its concepts-segments. In particular, grey is considered salient for both classes while white is important for male classifications and brown for female ones.

A likely reason why ACE is unable to capture coherent element semantic features in the *PA-49K Gender* is the network weight values. To elaborate, when looking at the trained weights for a sample of network layers, we observed that there is a significant number of zero values. This weight behaviour is potentially due to only having two classes for the fine-tuned network compared to the 1,000 of ImageNet

for which the Inception-V3 was designed. As a result, we argue that only a fraction of its expressive power is needed to classify a two-class classification problem, thus leading to a series of zero weights. This behaviour hypothetically limits the information of the feature representation space that ACE uses both to extract the segments and to compute the TCAV scores, thus leading to limited visual concepts.

Ambulance		Moving Van	
Semantic Feature	**Cramér's V**	**Semantic Feature**	**Cramér's V**
window AND stripe	0.51	vehicle_under	0.3
stripe	0.49	vehicle_side AND vehicle_under	0.27
stripe AND vehicle_side	0.4	vehicle_top AND vehicle_under	0.26
stripe AND mirror	0.39	black-vehicle_under	0.25
stripe AND tire	0.38	black-vehicle_under AND black-tire	0.22

(a) SEFA Output.

Ambulance		Moving Van	
Semantic Feature	**TCAV**	**Semantic Feature**	**TCAV**
black-bumper	1.0	black-vehicle_under	0.69
black-tire OR grey-tire	1.0	black OR grey	0.18
black	1.0	tire	0.15
orange OR red	1.0	white-sky	0.05
grey-window OR grey-bumper	0.99	orange-letters OR red-letters	0.01

(b) ACE Output.

Table 4.16: ImageNet Vehicle - Top five semantic features.

Moving on to the *ImageNet Vehicle* task, SEFA mainly focuses on the presence of "stripe" for the ambulance class while it uses the underside of the vehicle for the moving van. The importance of the coloured stripes for the ambulance class is further highlighted by the semantic features "orange-stripe" and "red-stripe" which have a Cramér's V value of 0.35 and 0.27 respectively. On the other hand, ACE is centered around describing colour related information, combined with some elements. Interestingly it extracts the feature "orange OR red" with a TCAV value of one for the ambulance class suggesting that it also picks up on the stripe colour importance, similarly to SEFA. Other than that, it also highlights the importance of the vehicle underside for the moving van.

By comparing SEFA and ACE based on the types of queries they answer, an important difference is observed. To elaborate, both of them can reason about "elements", "attribute" and element-attribute "pairs", but SEFA is also able to answer queries containing "combinations" of them. The usefulness of such an option is apparent when looking at the "vehicle_side" semantic feature in the SEFA output. When the side of the vehicle is highlighted in combination with a coloured stripe it indicates the ambulance class whereas when both the side and the underside of the vehicle are highlighted, it is classified as moving van. ACE and the other existing global interpretability methods are unable to answer such an interpretability query.

As for the ImageNet Fish output in Table 4.17, SEFA captures elements and combinations of them that are specific to each class, such as "lobster_claw", "lobster_claw AND lobster_body", "shark_body" and "trout_body". Moreover, it asso-

American Lobster		Great White Shark		Tench	
Semantic Feature	**Cramér's V**	**Semantic Feature**	**Cramér's V**	**Semantic Feature**	**Cramér's V**
lobster_claw	0.9	shark_body	0.78	trout_body	0.84
orange	0.82	grey-shark_body	0.76	trout_wing	0.68
lobster_body	0.81	shark_wing	0.72	trout_head	0.68
orange-lobster_claw	0.77	grey-shark_wing	0.7	trout_head AND trout_body	0.67
lobster_claw AND lobster_body	0.76	shark_wing AND shark_body	0.64	trout_wing AND trout_body	0.65

(a) SEFA Output.

American Lobster		Great White Shark		Tench	
Semantic Feature	**TCAV**	**Semantic Feature**	**TCAV**	**Semantic Feature**	**TCAV**
orange-lobster_body	1.0	grey_blue-water OR grey_blue-shark_body*	1.0	blue-background OR grey-background OR green-background	1.0
orange-lobster_head OR orange-lobster_claw	1.0	grey-shark_wing OR grey-shark_head*	1.0	yellow-trout_body OR grey-background OR yellow-background	1.0
orange-lobster_claw	0.99	water OR shark_stomach*	1.0	grey-shirt OR grey-trout_body	0.96
white-dish	0.86	blue-water OR blue-shark_body OR grey-shark_body*	1.0	beige-fingers OR green_yellow-trout_body	0.91
beige-lobster_claw OR orange-lobster_claw	0.49	blue-shark_body OR blue-water*	1.0	brown-trout_body	0.45

(b) ACE Output.

Table 4.17: ImageNet Fish - Top five semantic features.

ciates the "orange" and "grey" colours with the American lobster and great white shark respectively. Another interesting finding provided by SEFA's combinations is that the combinations of separate fish parts, such as "shark_wing AND shark_body", have one of the highest Cramér's V values for their classes. Therefore, we can assume that large parts of the fish are highlighted in the heatmaps, hinting that the Inception-V3 pre-trained on *ImageNet* is sensitive to large parts of the fish body instead of fine-grained details such as the eyes, mouth and more.

On the contrary, TCAV outputs surprising similar semantic features to SEFA concerning the fish parts highlighted but pairs them with several colours. Furthermore, it highlights some extra semantic features related to the background bias of *ImageNet Fish* discussed previously, such as "white-dish", "blue-water" and "beige-fingers". At this point, we want to highlight that all eight concepts output for the great white shark by ACE have the exact same values with a TCAV of one and a p-value of 0.066. Also, the p-values fail to satisfy the 0.05 threshold. The aforementioned issues, lead us to view the ACE concepts for the shark class with some scepticism. That said, these concepts are the only output ACE provides and since their p-values are only marginally higher than 0.05, we decided to report them in our study.

When observing the differences and similarities between the two methods for our three use cases, we have to keep in mind that they attempt to achieve slightly different interpretability goals. For instance, when considering *ImageNet Vehicle*, SEFA answers which semantic features discriminate an ambulance from a moving van and the other way around. On the other hand, ACE provides information about the concepts that make an ambulance belong to that class out of the 1,000 classes that the pre-trained ImageNet model classifies. To make this different behaviour clearer, we compare the behaviour of the two methods for features that are only

highlighted by ACE in Table 4.18.

Semantic Feature	TCAV	Cramér's V	Frequency per class
grey | male	1.0	0.08	female: 0.59 | male: 0.8
grey-background | female	0.99	0.08	female: 0.22 | male: 0.3
black | ambulance	1.0	0.09	ambulance: 0.83 | moving van: 0.75
tire | moving van	0.15	0.02	ambulance: 0.64 | moving van: 0.61

Table 4.18: SEFA class frequencies for metrics that are only output by ACE.

According to the previous table, it is clear that some of the features with high TCAV scores were not considered as salient by SEFA because they had similarly high frequencies for both classes. We argue that these examples underline our previous claim that the output of two methods answer a slightly different question. That said, SEFA could be easily modified to reason in a similar manner to ACE. To be more exact, simply printing semantic features sorted by their class frequency, regardless of their annotations for the other class, would allow us to achieve that.

The experiments conducted in this section show that both SEFA and ACE can answer global interpretability queries regarding "elements", "attributes" and element-attribute "pairs". We also noticed that some of those semantic features are output by both methods suggesting that they are indeed salient for the model. That said, SEFA can also reason about "combinations" of the aforementioned query types which allows us to define more complex interpretability questions about the model behaviour, thus providing increased expressivity (**RSQ4.1**). We argue that SEFA provides higher flexibility compared to existing methods due to the structured representation it provides. In particular, SEFA users need a specific number of annotations and then they can define any type of global interpretability query. Apart from the query types and operators evaluated in this study, users can also defined any other question based on their needs, such as checking for combinations of three or more semantic features, using the "OR" operator and more. We suggest experimenting with more query types aimed at different use cases, such as locating bias edge cases, as a promising direction for future work.

Finally, we want to highlight that the absence of a benchmark or ground truth within the interpretability domain makes it almost impossible to say with confidence which interpretability method closer describes the model behaviour. Nonetheless, we perform two rigorous experiments in the next section that evaluate the robustness of our proposed method with respect to its ability to capture dataset biases which we assume that image classification models learn.

RSQ4.2 - Output Robustness

As mentioned in Chapter 4.1.4 we are unfortunately not aware of what the ground truth of our output should be when interpreting a deep learning model. However, since we want to evaluate the reliability of SEFA to output the right semantic features as salient, we train two Inception-V3 models on datasets where we introduce synthetic bias and evaluate the method's output quantitatively. The idea is that by injecting the bias, we create a form of ground truth that we expect SEFA to capture. To be more exact, we perform the following experiments:

- *PA-49K Gender - Orientation Bias*: the *PA-100K* comes with annotations regarding the orientation of the person in the image, namely "front", "back" and "side". We take advantage of this fact and sample all of the male images with "front" orientation and the female ones with "back" orientation. As a result, when we fine-tune an Inception-V3 model on this biased dataset, we expect it to predict the gender by checking whether the image contains "hair" or "facial" characteristics.
- *ImageNet Fish - Background Bias*: by observing the background of the three classes in *ImageNet Fish* task, we noticed that the white shark class is always depicted in the sea, whereas the lobster is served on a dish and the tench is held by humans. By taking advantage of this inherent dataset bias, we fine-tune the model on the biased data and observe whether SEFA picks up the aforementioned biases.

We provide examples of how the heatmap outputs change when we fine-tune the models on the aforementioned biased datasets in Figure 4.14.

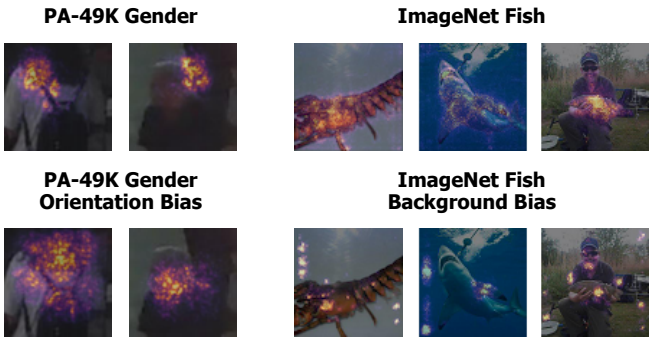


Figure 4.14: Image examples for the robustness experiments. We show how the heatmap outputs change when we fine-tune the models on the biased datasets.

By observing these examples we can see how the focus of the heatmaps shifts from other elements in the image to the hair and the facial characteristics of the individuals in the orientation bias *PA-49K Gender*. Similarly for *ImageNet Fish*, the biased model focuses more on image details that are related to the background bias of the dataset. We evaluate SEFA's ability to describe the biases introduced by annotating 300 images, using the binary representation with the "all" option and the statistical tests with a p-value of 0.05. The top ten semantic features observed for each biased dataset and their original counterparts can be found in Tables 4.19 and 4.20 for PA-49K Gender and *ImageNet Fish* respectively.

Based on the aforementioned results we conclude that SEFA is able to capture the synthetic biases introduced in both cases. More specifically, for the *PA-49K Gender*, our method is able to output that "hair" related semantic features, such as "long-hair", "black-hair" and more, are related to the female class. On the contrary, elements such as "neck" and "cheek" are associated with the male class since they

PA-49K Gender		PA-49K Gender Orientation Bias	
Semantic Feature | Class	**Cramér's V**	**Semantic Feature | Class**	**Cramér's V**
long | female	0.48	hair | female	0.71
long-hair | female	0.48	black-hair | female	0.69
long-hair AND black-hair | female	0.45	neck | male	0.61
long AND black | female	0.44	black | female	0.53
short | male	0.43	long | female	0.51
short-hair | male	0.43	long-hair | female	0.51
black AND short | male	0.43	long AND black | female	0.49
black-hair AND short-hair | male	0.41	long-hair AND black-hair | female	0.49
grey AND long | female	0.39	cheek | male	0.48
short AND grey | male	0.3	cheek AND neck | male	0.44

Table 4.19: PA-49K Gender - Comparison of original vs model with orientation bias.

ImageNet Fish Pre-trained		ImageNet Fish Background Bias	
Semantic Feature | Class	**Cramér's V**	**Semantic Feature | Class**	**Cramér's V**
lobster_claw | lobster	0.9	trout_body | tench	0.9
trout_body | tench	0.86	lobster_claw | lobster	0.83
shark_body | shark	0.82	orange | lobster	0.79
grey-shark_body | shark	0.81	grey-trout_body | tench	0.76
orange | lobster	0.8	orange-lobster_claw | lobster	0.73
orange-lobster_claw | lobster	0.79	lobster_body | lobster	0.71
lobster_body | lobster	0.75	blue-water | shark	0.71
shark_wing | shark	0.69	green | tench	0.7
grey-shark_wing | shark	0.69	beige | tench	0.7
trout_head | tench	0.68	water | shark	0.7

Table 4.20: ImageNet Fish - Comparison of pre-trained vs fine-tuned ImageNet model.

are only visible in male images due to the bias introduced in the data. Similarly, for the *ImageNet Fish* case, only five out of the top ten semantic features are the same between the original pre-trained network and the fine-tuned biased one. In particular, SEFA is able to pick up semantic features that correspond to the background bias of each class, such as the "green" grass in the background of tench images and the "water" that the white sharks are depicted in.

To summarise, the experiments conducted in this section indicate that SEFA is robust enough to capture synthetic biases introduced in datasets (**RSQ4.2**). However, the local interpretability method used to extract the heatmaps needs to highlight the image regions containing the bias for SEFA to include it in its salient semantic features. The reason for that, is that human annotators can only describe the bias if it is highlighted in the individual image heatmaps as salient for the model. For instance, if the "water" where the great white shark is depicted in the *Background Bias* case is not highlighted, then the annotators will be unable to describe it and it will not be included in the SEFA output. Therefore, it is crucial that the local interpretability method captures the association between the model predictions and the input images, as discussed in Chapter 3.3.1.

RSQ4.3 - Model Sensitivity

An interpretability method should be able to focus on the image details that each model uses to classify a specific class. Hence, such a method needs to be sensitive to the parameters of an image classification neural network. We hypothesise that when two separate model architectures are trained on the same task, SEFA will be able to capture the subtle differences in their behaviour.

To test our hypothesis, we run SEFA on the unbiased *ImageNet Fish* task using the VGG16 model and compare its output to that of Inception-V3 architecture used throughout the rest of our experiments. As already mentioned in Chapter 4.1.2, the reason why we chose the VGG16 architecture over other higher-performing architectures, such as the ResNet-50, is that it has been found to focus on significantly different pixels compared to the Inception-V3 [66]. This fact is also evident by the validation accuracy values of Inception-V3 and VGG16 which achieve an accuracy of 88% and 80.67% respectively on *ImageNet Fish*. The subtle differences in the pixels that these two models focus on can be found in Figure 4.15.

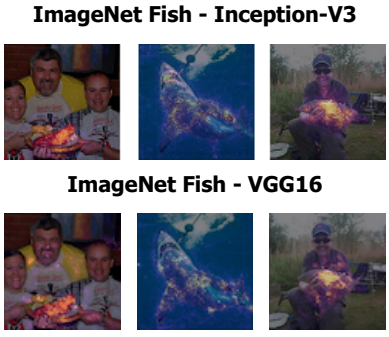


Figure 4.15: ImageNet Fish - Inception-V3 vs VGG16 heatmap examples.

While at first glance the two architectures seem to focus on roughly the same areas, upon closer inspection we observe some subtle differences. For example, the Inception-V3 focuses more on the mouth and teeth of the shark than VGG16 in the previous example images. We run SEFA on the *ImageNet Fish* with these two models by annotating 300 random images and taking advantage of the binary representation with the "all" option. Then, we analyse the representation using the binary statistical tests with a p-value of 0.05. The top ten semantic features output by SEFA for Inception-V3 and VGG16 are available in Table 4.21.

According to the previous table, SEFA is able to pick up on the different semantic features that the two image classification architectures focus on. In particular, seven out of the top ten features are the same for both models but with a significantly different ranking and Cramér's V values. For instance, while features such as "grey-shark_wing" and "trout_head" are in positions nine and ten for Inception-V3 they only appear at positions 19 and 17 for VGG16 respectively. By further analysing the top 40 outputs for the two models we observed the same findings, namely, there is a high number of overlapping semantic features for the two models but in significantly

Inception-V3		VGG16	
Semantic Feature | Class	**Cramér's V**	**Semantic Feature | Class**	**Cramér's V**
lobster_claw | lobster	0.9	trout_body | tench	0.99
trout_body | tench	0.86	lobster_claw | lobster	0.93
shark_body | shark	0.82	grey-trout_body | tench	0.92
grey-shark_body | shark	0.81	orange | lobster	0.91
orange | lobster	0.8	lobster_body | lobster	0.85
orange-lobster_claw | lobster	0.79	shark_body | shark	0.83
lobster_body | lobster	0.75	grey-shark_body | shark	0.83
shark_wing | shark	0.69	orange-lobster_claw | lobster	0.82
grey-shark_wing | shark	0.69	shark_head | shark	0.8
trout_head | tench	0.68	grey-shark_head | shark	0.79

Table 4.21: ImageNet Fish - Comparison of salient semantic features for Inception-V3 vs VGG16.

different rankings and Cramér's V scores. We argue that this behaviour is expected given that the VGG16 has a significantly lower accuracy compared to Inception-V3 but not that much lower to indicate that it focuses on completely different parts of the image.

To conclude, the Inception-V3 versus VGG16 comparison in this section indicates that SEFA can indeed focus on model-specific semantic features (**RSQ4.3**). Similarly to the robustness experiments, we have to underline that the output of SEFA greatly depends on the ability of the local interpretability method to pick up the differences between separate models on an image level. Given that one of the reasons why we selected SmoothGrad gradients is that they were shown by Adebayo et al. [3] to be sensitive to network parameters, we are confident that they can pick up such differences among separate models. That said, we argue that if these differences are too subtle to be noticed by humans during the semantic feature annotation, then SEFA might fail to output their differences in such cases.

4.3. Summary

In this chapter, we investigated the extent to which the analysis of semantic features enables us to answer a wider range of global queries for image classification models. To answer this question, we first conducted a series of experiments evaluating different SEFA hyperparameters to better understand its behaviour in practice and the options that one should use to obtain reliable and robust explanations (**RSQ3**). Following that, we compared SEFA with ACE, one of the latest global interpretability methods, to reason about the types of queries that these methods can answer and to evaluate whether SEFA indeed offers more expressive explanations (**RSQ4**). Finally, we tested SEFA on datasets with artificial bias and on two separate model architectures to reason about its ability to output semantic features that are important for the model and its sensitivity to network parameters respectively.

RSQ3.1: How many images do we need to annotate in order to obtain robust and reliable model behaviour explanations?

We tested SEFA with different annotation sizes, ranging from 20 to 400, for four separate Inception-V3 models. Our results suggest that 100 annotations are enough to capture the semantic features that are of absolute importance for the model. Annotating an extra 200 annotations can also provide information about semantic features that are less commonly present in the images of each class. That said, we argue the choice to annotate more than 100 images depends on the cost-quality trade-off decided by each SEFA user and their needs. We would like to underline that the aforementioned annotation sizes are suggested for datasets with a relatively small number of classes, i.e. two or three. Further experiments are required to reason about the number of annotations required for datasets with significantly higher numbers of classes.

RSQ3.2: Which types of interpretability queries does each semantic representation option answer?

SEFA offers several representation options, namely "elements", "attributes", "pairs" and "combinations". We performed experiments on four artificially biased datasets derived from the *PA-49K Gender* dataset. Our results indicate that each representation option is able to answer different types of queries that capture separate cases of dataset bias. Moreover, they highlight that some use cases may necessitate the use of the "NOT" logical operator which checks about the absence of a semantic feature in an image. Based on the aforementioned empirical evidence, we suggest the use of the "all" option which evaluates the semantic features extracted by all four representation options to evaluate the widest range of queries that SEFA offers. The additional use of the "NOT" operator should be decided according to the user needs since it significantly increases the computational complexity of the method.

RSQ3.3: Does the numeric representation provide extra information compared to the binary one?

We compared the output of the numeric and binary representations on three separate classification tasks, namely *PA-49K Gender*, *ImageNet Vehicle* and *ImageNet Fish*. Our results suggest that when the heatmaps of a dataset-model are not noisy and their areas either have a high pixel intensity or no intensity at all, then the binary representation is sufficient to explain the model. However, if the maps are noisy and have areas that are highlighted with various pixel intensity values, then using the numeric representation leads to more reliable explanations. That said, this option comes with a higher annotation cost since the human annotators need to describe all the highlighted areas in the image based on their pixel intensity.

RSQ3.4: Do separate representation analysis methods provide similar or contradicting salient semantic features?

We also compared the output of the three available analysis methods on the three classification tasks mentioned before. The results indicate that the statistical tests provide meaningful quantitative evaluations of the semantic features requested by the user. On the other hand, while the rule mining allows the method to evaluate

combinations of more than two semantic features without the need to explicitly specify them, we showed that it requires a combination of at least two evaluation metrics, lift and confidence. We argue that its output analysis is more challenging than statistical testing. As for the decision tree, it provides limited quantitative information about the semantic features in the form of feature importance values for the trained tree. Therefore, we propose the use of the statistical tests since they are flexible, more intuitive and can be easily evaluated via their magnitude values.

RSQ4.1: To what extent can SEFA answer more complex interpretability queries compared to existing global methods?

As for SEFA's ability to answer a wider range of queries compared to existing methods, we compared its output versus ACE. In particular, we used the optimum hyperparameters according to our previous experiments and evaluated the two methods on the three classification tasks used in our work. Our results showcase that both methods are able to answer queries regarding "elements", "attributes" and "pairs" of semantic features. That said, ACE provides a fixed output where you have no option to perform extra queries of your choice while SEFA gives the user the flexibility to create different types of queries based on their needs. Even more importantly SEFA provides increased expressivity that allows to reason about "combinations" of the aforementioned semantic features, thus allowing us to answer interpretability queries that current methods are unable to.

RSQ4.2: To what extent can SEFA correctly identify synthetic biases introduced into a model?

We experimented with SEFA on two purposefully biased datasets to reason about whether it outputs semantic features related to the model behaviour. More specifically, we fine-tuned two Inception-V3 models on a modified *PA-49K Gender* dataset where the images of each class are filtered according to the orientation of the person in the image and on *ImageNet Fish* which is inherently biased with the background of each class. In both cases, SEFA was able to modify its output in comparison with the unbiased cases to accommodate for the artefacts in the data which we assume that the model picked up during training.

RSQ4.3: To what extent is SEFA able to focus on model-specific semantic features?

We compared SEFA's output when using Inception-V3 and VGG16 trained on *ImageNet Fish*. We were able to observe significant differences between the two models while keeping in mind that they achieve relatively similar accuracy. These experiments suggest that SEFA does indeed retrieve semantic features relevant to model classifications and that it can modify its output according to network parameters.

5

Conclusion

In this chapter, we summarise the work conducted in this thesis, draw the conclusions that answer our main research question and discuss the method limitations. Following that, we lay out directions for future work based on the conclusions and limitations of our study.

5.1. Summary

In this section, we summarise the focus of our work, the proposed methodology, the conclusions reached and the limitations of our method.

5.1.1. Work Focus

Deep learning is achieving state-of-the-art performance on a series of image classification tasks. However, these models suffer from limited interpretability which creates several issues, such as their limited applicability in mission-critical domains. Several interpretability methods have been proposed that attempt to explain the behaviour of these models with respect to a class of interest. We argue that such a method should be able to answer complex interpretability queries, such as whether the combination of two objects in an image is associated with the prediction of a class. To the best of our knowledge, no existing method can provide the expressivity required to answer these types of queries. To address this gap, we proposed a novel global interpretability method, called **SEmantic Feature Analysis (SEFA)**, that utilises human annotations to provide the semantic information and structure needed to answer such questions.

5.1.2. Methodology Approach

The SEFA method includes four main steps: (1) local interpretability extraction, (2) semantic feature annotation, (3) semantic representation extraction and (4) semantic representation analysis. To elaborate, we first extract explanations for individual predictions by highlighting the pixels that the model is most sensitive to. Then, we

use human annotators to describe these areas by drawing bounding boxes and describing the corresponding element and its attributes. The elements and attributes annotated per image are termed **semantic features**. The next step is to extract a structured representation where the rows correspond to the annotated images and the columns to the semantic features. The idea is that by giving semantic structure to the salient areas in an image, we can create the proposed **semantic representation** which can be analysed using traditional structured data methods. We argue that this representation provides us with the flexibility and expressivity required to answer any type of complex interpretability query.

5.1.3. Conclusions

Given the limited expressivity and query complexity of existing methods, we designed and developed SEFA to address these issues. In particular, we studied the extent to which the extraction and analysis of semantic features allows us to interpret image classification models (**MRQ**). Firstly, we conducted a literature review (Chapter 2) on the existing interpretability methods and the current state of that domain (**RSQ1**). Then, we designed (Chapter 3) and implemented a new interpretability method that extracts the aforementioned semantic features from local interpretability methods and allows us to answer global interpretability queries (**RSQ2**). Finally, we performed extensive experimentation of SEFA on three classification tasks and two separate deep learning models (Chapter 4). Our experiments allowed us to obtain more insight into the hyperparameters under which SEFA performs best (**RSQ3**) and to reason about the extent to which it can answer more complex interpretability queries compared to existing methods (**RSQ4**). The main contributions of our work can be summarised as follows:

C1: literature review. We performed an in-depth literature review on existing interpretability methods and the state of the field as a whole. The review enabled us to define interpretability within the score of our work and to understand the needs that such methods address. Furthermore, we obtained a better understanding of the existing methods, their limitations and their experimental setups. In particular, it enabled us to reason about the characteristics that an interpretability method should adhere to and highlighted the limitation of existing methods to answer complex queries. Moreover, we selected SmoothGrad [52] as the method to extract the salient pixels per image based on analysis of existing local interpretability methods. Finally, we understood that one of the main challenges of the field is method evaluation. To elaborate, since we are unaware of the image areas that the models we are interpreting are sensitive to, there is no concrete ground truth to reason about the reliability and robustness of the method. That said, we were able to come up with some bias injection experiments inspired by existing literature work.

C2: SEFA design. We designed a new global interpretability method, called SEFA. This process included choosing the local method used to extract the heatmaps for the human annotations and their visualisation during the annotation process. Furthermore, we designed a user interface which can be used with minor modifications

for any dataset and allows human annotators to extract semantic features. Following that, we designed the process of extracting the structured representation and its four different representation options, each one aimed at answering a different type of interpretability query. Finally, we selected three existing structured data methods that can be used to answer global interpretability queries.

C3: SEFA implementation. We implemented the proposed method to allow for its experimentation. We also make the method publicly available on GitHub[31] for re-use by fellow researchers in future work.

C4: SEFA experimentation. We performed extensive experimentation of SEFA on three classification tasks and two model architectures, both using the original datasets and artificially biased ones. During our experimentation, we pre-processed an existing dataset to create a new gender classification dataset, term as *PA-49K Gender*, which can be found on 4TU[32]. The conducted experiments provided us with the suggested hyperparameter settings under which SEFA performs optimally. We argue that apart from the required number of annotations, these findings can generalise to any image classification dataset. As for the annotation size, it is sufficient for classification tasks with two or three classes while further experimentation is needed for multi-class datasets with a significantly higher number of classes. Furthermore, we showed that our method can answer all of the interpretability queries that current methods cover and more, such as the combinations between different semantic features. The main benefit of our method is its flexibility to define any type of query a user wants, no matter how complex it is. All its users need to do is annotate a set of random dataset images. For the three classifications tasks in our study, 300 annotations were enough to obtain reliable results. Given that the annotation of these images required three to five hours depending on the dataset, we argue that it is a reasonable time investment for a model user to understand its behaviour. To put it into context, the evaluation of a single query for a single class using TCAV [32] requires roughly two hours of runtime on commodity hardware without factoring in the time required to gather the image concepts.

5.1.4. Method-Study Limitations

The absence of an interpretability ground truth greatly limits our ability to reason about SEFA's reliability and to directly compare its output to existing methods. The fact that we are not aware of the image areas that the model utilises, in reality, is an inherent issue with all existing deep learning interpretability methods that limits our understanding of them. For instance, in our study, we noticed that in some images only a part of the person's hair is highlighted for the *PA-49K Gender* dataset. While the annotators described these parts with semantic features such as "long-hair" and "short-hair", we do not know if the model is sensitive to the hair length in reality or if the human annotators introduced their own bias in the study.

[31] https://github.com/psoilis/SEFA
[32] https://doi.org/10.4121/uuid:38dab37c-1179-495e-b357-0568b9aaaa7a

Another important point is that SEFA is significantly affected by the local interpretability output, meaning that human annotators can only describe areas highlighted in the heatmap output. This fact can lead the method to fail in cases with fine-grained differences. For instance, if we want to compare the behaviour of two models that are sensitive to almost the same image parts, their fine-grained differences might not get output by SEFA if they are not noticed by the annotators.

Finally, we did not evaluate our method on datasets with a high number of classes, i.e. tens or hundreds, meaning that we cannot reason about the number of annotations required in those cases. In particular, we expect SEFA to require significantly more annotations as the appearances of the same semantic features are spread across more classes. In this case, the option of scaling out the human computation step with crowd workers might be required to speed up the process.

5.2. Future Work

In this section, we propose directions for future work based on the findings and limitations of our study.

5.2.1. Interpretability Benchmark

One of the most interesting directions for future work is the creation of a common benchmark for interpretability methods. While we understand that it is by no means a trivial task and that a universal benchmark approach applicable to every method would be almost impossible, we argue that more work is needed to agree on a common evaluation framework for methods with similar outputs. The presence of a benchmark would allow us to further evaluate new methods, such as SEFA, and to directly compare existing methods with one another. It would also help us understand the usability of existing methods in practice and to come up with possible additional requirements that they should adhere to.

To the best of our knowledge, the only relevant study (under review) is the preliminary work of Yang and Kim [60] who propose an evaluation framework for interpretability methods. However, without significant modifications in the bias injection process and the metrics of the framework, it only allows to evaluate queries referring to single concepts and not the more complex ones evaluated in our study, such as combinations of concepts. Therefore, the aforementioned framework did not match our requirements.

5.2.2. Human Annotation

One key step in our methodology is the annotation of semantic features. While the use of a domain expert yielded promising outputs, such a process can be time-consuming and costly. We argue that a promising line of work would be to attempt to scale this step out using crowd workers. That said, we expect several challenges regarding the quality and aggregation of their annotations. A potential first step could be to evaluate the option of having pre-filled options of semantic features on a dataset basis instead of the open text fields in this study.

5.2.3. Query Types

While our work is able to answer a wider variety of interpretability queries, such as conjunctive queries on semantic concepts (SEFA "combinations"), we argue that there is scope for future work. To be more exact, the structured representation introduced provides great flexibility to create any type of query based on the user needs. An example of such a case is the "OR" operator that could answer whether the presence of at least one of the semantic features in a query is associated with a specific class. Moreover, SEFA can answer significantly different types of questions, such as answering queries regarding bias edge cases, with minor modifications. In that case, one could evaluate the rule mining output using a different metric or analyse the decision tree graph.

5.2.4. Meta-Analysis Methods

SEFA comes with three meta-analysis methods, but can easily be extended with extra ones that can potentially allow us to reason about the model behaviour more reliably. Therefore, we provide a few interesting alternatives that arose during our study. While the work of Yang et al. [59] has been mainly discussed as a global interpretabilty method so far, they provide an "interpretation tree" which could be used instead of the decision tree to overcome its limitations. To elaborate, their tree also provides information about the "classification accuracy" on each tree node, thus making it more intuitive to interpret. As a result, one can use their tree method with our more expressive structured representation to reason about global interpretability. Another interesting direction is the use of data mining [25] or causal reasoning [30] [63] methods to reason about the presence of discrimination bias.

Bibliography

[1] Alan C Acock and Gordon R Stavig. A measure of association for nonparametric statistics. *Social Forces*, 57(4):1381–1386, 1979.

[2] Amina Adadi and Mohammed Berrada. Peeking inside the black-box: A survey on explainable artificial intelligence (xai). *IEEE Access*, 6:52138–52160, 2018.

[3] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. In *Advances in Neural Information Processing Systems*, pages 9505–9515, 2018.

[4] Rakesh Agrawal, Tomasz Imieliński, and Arun Swami. Mining association rules between sets of items in large databases. In *Proceedings of the 1993 ACM SIGMOD international conference on Management of data*, pages 207–216, 1993.

[5] Rakesh Agrawal, Ramakrishnan Srikant, et al. Fast algorithms for mining association rules. In *Proc. 20th int. conf. very large data bases, VLDB*, volume 1215, pages 487–499, 1994.

[6] Ahmed Alqaraawi, Martin Schuessler, Philipp Weiß, Enrico Costanza, and Nadia Berthouze. Evaluating saliency map explanations for convolutional neural networks: a user study. *arXiv preprint arXiv:2002.00772*, 2020.

[7] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10 (7):e0130140, 2015.

[8] David Baehrens, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and Klaus-Robert MÃžller. How to explain individual classification decisions. *Journal of Machine Learning Research*, 11(Jun):1803–1831, 2010.

[9] Umang Bhatt, Alice Xiang, Shubham Sharma, Adrian Weller, Ankur Taly, Yunhan Jia, Joydeep Ghosh, Ruchir Puri, José MF Moura, and Peter Eckersley. Explainable machine learning in deployment. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 648–657, 2020.

[10] Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *Advances in neural information processing systems*, pages 4349–4357, 2016.

[11] Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. *Classification and regression trees*. CRC press, 1984.

[12] Joy Buolamwini and Timnit Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on fairness, accountability and transparency*, pages 77–91, 2018.

[13] Peter Burt and Edward Adelson. The laplacian pyramid as a compact image code. *IEEE Transactions on communications*, 31(4):532–540, 1983.

[14] Aylin Caliskan, Joanna J Bryson, and Arvind Narayanan. Semantics derived automatically from language corpora contain human-like biases. *Science*, 356 (6334):183–186, 2017.

[15] Supriyo Chakraborty, Richard Tomsett, Ramya Raghavendra, Daniel Harborne, Moustafa Alzantot, Federico Cerutti, Mani Srivastava, Alun Preece, Simon Julier, Raghuveer M Rao, et al. Interpretability of deep learning models: a survey of results. In *2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, pages 1–6. IEEE, 2017.

[16] Dana Chandler and Adam Kapelner. Breaking monotony with meaning: Motivation in crowdsourcing markets. *Journal of Economic Behavior & Organization*, 90:123–133, 2013.

[17] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[18] Dan Ciregan, Ueli Meier, and Jürgen Schmidhuber. Multi-column deep neural networks for image classification. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3642–3649. IEEE, 2012.

[19] Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*, 2017.

[20] Mengnan Du, Ninghao Liu, and Xia Hu. Techniques for interpretable machine learning. *Commun. ACM*, 63(1):68–77, December 2019. ISSN 0001-0782. doi: 10.1145/3359786. URL https://doi.org/10.1145/3359786.

[21] Dumitru Erhan, Yoshua Bengio, Aaron Courville, and Pascal Vincent. Visualizing higher-layer features of a deep network. *University of Montreal*, 1341(3): 1, 2009.

[22] Amirata Ghorbani, James Wexler, James Y Zou, and Been Kim. Towards automatic concept-based explanations. In *Advances in Neural Information Processing Systems*, pages 9273–9282, 2019.

[23] Bryce Goodman and Seth Flaxman. European union regulations on algorithmic decision-making and a "right to explanation". *AI magazine*, 38(3):50–57, 2017.

[24] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. A survey of methods for explaining black box models. *ACM computing surveys (CSUR)*, 51(5):1–42, 2018.

[25] Sara Hajian and Josep Domingo-Ferrer. A methodology for direct and indirect discrimination prevention in data mining. *IEEE transactions on knowledge and data engineering*, 25(7):1445–1459, 2012.

[26] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[27] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[28] Sara Hooker, Dumitru Erhan, Pieter-Jan Kindermans, and Been Kim. A benchmark for interpretability methods in deep neural networks. In *Advances in Neural Information Processing Systems*, pages 9734–9745, 2019.

[29] Mark Ibrahim, Melissa Louie, Ceena Modarres, and John Paisley. Global explanations of neural networks: Mapping the landscape of predictions. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pages 279–287, 2019.

[30] Niki Kilbertus, Mateo Rojas Carulla, Giambattista Parascandolo, Moritz Hardt, Dominik Janzing, and Bernhard Schölkopf. Avoiding discrimination through causal reasoning. In *Advances in Neural Information Processing Systems*, pages 656–666, 2017.

[31] Been Kim, Rajiv Khanna, and Oluwasanmi O Koyejo. Examples are not enough, learn to criticize! criticism for interpretability. In *Advances in neural information processing systems*, pages 2280–2288, 2016.

[32] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, and Rory sayres. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (TCAV). In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2668–2677, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR. URL http://proceedings.mlr.press/v80/kim18d.html.

[33] Pieter-Jan Kindermans, Sara Hooker, Julius Adebayo, Maximilian Alber, Kristof T Schütt, Sven Dähne, Dumitru Erhan, and Been Kim. The (un) reliability of saliency methods. In *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, pages 267–280. Springer, 2019.

[34] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[35] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[36] Ludmila I Kuncheva, Christopher J Whitaker, Catherine A Shipp, and Robert PW Duin. Limits on the majority vote accuracy in classifier fusion. *Pattern Analysis & Applications*, 6(1):22–31, 2003.

[37] Zachary C Lipton. The mythos of model interpretability. *Queue*, 16(3):31–57, 2018.

[38] Xihui Liu, Haiyu Zhao, Maoqing Tian, Lu Sheng, Jing Shao, Shuai Yi, Junjie Yan, and Xiaogang Wang. Hydraplus-net: Attentive deep features for pedestrian analysis. In *Proceedings of the IEEE international conference on computer vision*, pages 350–359, 2017.

[39] David McNeely-White, J Ross Beveridge, and Bruce A Draper. Inception and resnet features are (almost) equivalent. *Cognitive Systems Research*, 59:312–318, 2020.

[40] David G McNeely-White, J Ross Beveridge, and Bruce A Draper. Inception and resnet: Same training, same features. In *Biologically Inspired Cognitive Architectures Meeting*, pages 352–357. Springer, 2019.

[41] George A Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine J Miller. Introduction to wordnet: An on-line lexical database. *International journal of lexicography*, 3(4):235–244, 1990.

[42] Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*, 73:1–15, 2018.

[43] 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*, pages 1135–1144. ACM, 2016.

[44] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Anchors: High-precision model-agnostic explanations. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[45] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

[46] Wojciech Samek, Thomas Wiegand, and Klaus-Robert Müller. Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models. *arXiv preprint arXiv:1708.08296*, 2017.

[47] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 806–813, 2014.

[48] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3145–3153. JMLR. org, 2017.

[49] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *Nature*, 550 (7676):354–359, 2017.

[50] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1409.1556.

[51] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In Yoshua Bengio and Yann LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Workshop Track Proceedings*, 2014. URL http://arxiv.org/abs/1312.6034.

[52] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017.

[53] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014.

[54] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3319–3328. JMLR. org, 2017.

[55] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[56] Robert F Tate. Correlation between a discrete and a continuous variable. point-biserial correlation. *The Annals of mathematical statistics*, 25(3):603–607, 1954.

[57] Daniel S Weld and Gagan Bansal. The challenge of crafting intelligible intelligence. *Communications of the ACM*, 62(6):70–79, 2019.

[58] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057, 2015.

[59] Chengliang Yang, Anand Rangarajan, and Sanjay Ranka. Global model interpretation via recursive partitioning. In *2018 IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, pages 1563–1570. IEEE, 2018.

[60] Mengjiao Yang and Been Kim. Benchmarking attribution methods with relative feature importance. *arXiv*, pages arXiv–1907, 2019.

[61] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.

[62] Matthew D Zeiler, Graham W Taylor, and Rob Fergus. Adaptive deconvolutional networks for mid and high level feature learning. In *2011 International Conference on Computer Vision*, pages 2018–2025. IEEE, 2011.

[63] Lu Zhang, Yongkai Wu, and Xintao Wu. Causal modeling-based discrimination discovery and removal: Criteria, bounds, and algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 31(11):2035–2050, 2018.

[64] Quan-shi Zhang and Song-Chun Zhu. Visual interpretability for deep learning: a survey. *Frontiers of Information Technology & Electronic Engineering*, 19 (1):27–39, 2018.

[65] Quanshi Zhang, Ying Nian Wu, and Song-Chun Zhu. Interpretable convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8827–8836, 2018.

[66] Zijian Zhang, Jaspreet Singh, Ujwal Gadiraju, and Avishek Anand. Dissonance between human and machine understanding. *Proceedings of the ACM on Human-Computer Interaction*, 3(CSCW):1–23, 2019.

[67] Minhaz Fahim Zibran. Chi-squared test of independence. *Department of Computer Science, University of Calgary, Alberta, Canada*, 2007.

A

Local Interpretability Evaluation

We compared SmoothGrad [51] (gradient-based) with LRP [7] (signal-based) and LIME [43] (local approximation) on three separate uses cases to evaluate their applicability for our study. To elaborate, we wanted to evaluate which of these methods is better able to highlight the features that the model is sensitive to. Given that there is no interpretability ground truth when explaining deep learning models, we decide to perform three bias injection experiments on the *PA-49K Gender* [33] dataset and to fine-tune an Inception-V3 [55] architecture on these biased datasets. The idea is that the bias injected creates an artificial ground truth that we can use to check whether the methods are able to capture it reliably. The test accuracy of the models fine-tuned exceeded 99.84% in all three datasets, indicating that they fit the bias injected. The three types of bias injected in the female class of the *PA-49 Gender* are the following:

1. **Square box**: we injected a square box of concrete color at the bottom right of each female image. This case was chosen as a very obvious bias case, expecting all three methods to be able to highlight it to some extent.

2. **Horizontal line**: we injected a horizontal line of concrete color at the top of each female image. The idea was that such as bias is more subtle than the box, thus potentially more challenging to be highlighted by each method.

3. **greyscale colors**: we converted all of the female images to greyscale while maintaining the male ones as coloured. This image wide bias was injected to check whether the method outputs will shift their focus to black, white and grey colored pixels to differentiate between the two classes.

We provide an indicative output of SmoothGrad, LRP and LIME for each of the aforementioned bias cases in Figures A.1, A.2 and A.3 respectively.

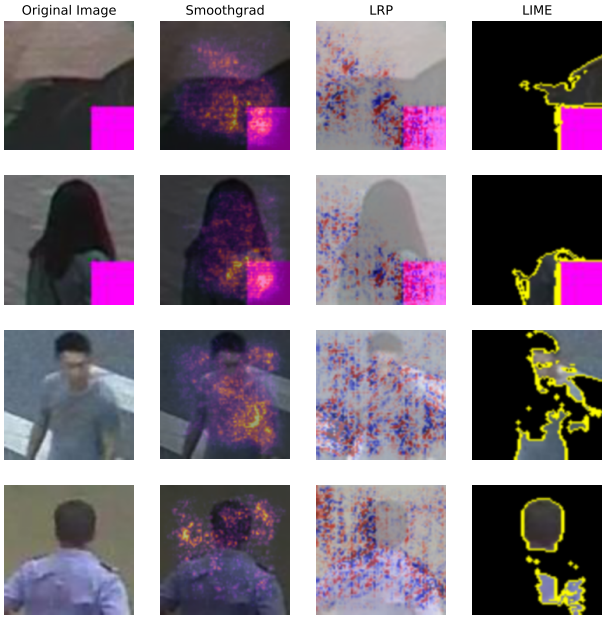


Figure A.1: SmoothGrad vs LRP vs LIME - Square box injected bottom right of female images.

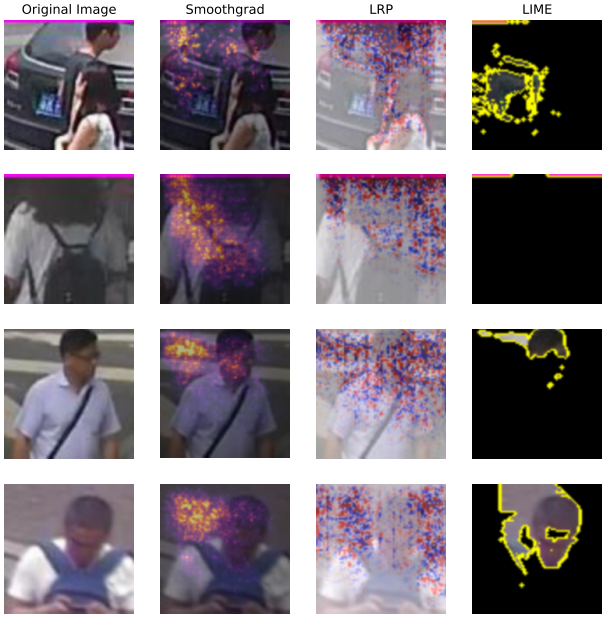


Figure A.2: SmoothGrad vs LRP vs LIME - Horizontal line injected at the top of female images.

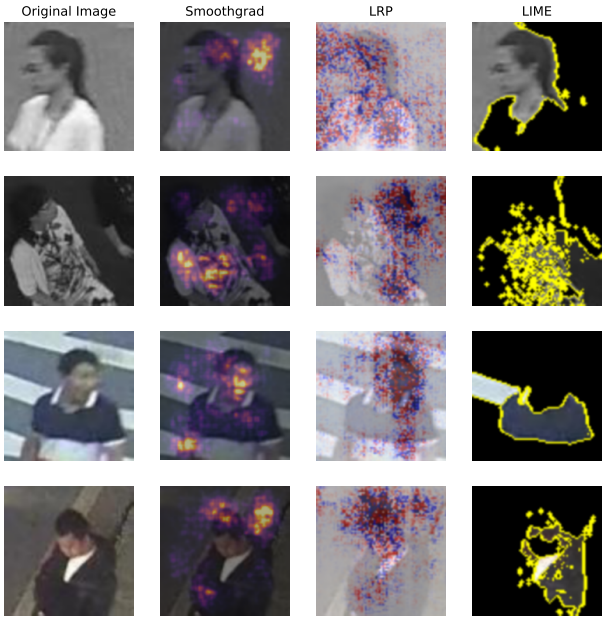


Figure A.3: SmoothGrad vs LRP vs LIME - greyscale female images.

According to the previous experiments, all three methods are able to capture the box and line biases introduced in the female images to some extent. Moreover, they highlight black, grey and white pixels in the greyscale bias case, indicating that they output the salient pixels for the model. However, the output of LRP suffers from noise making it challenging to understand the exact semantic feature highlighted. Moreover, it outputs pixels that have both a positive and negative influence on the classification, thus increasing the complexity of the annotation task. As for LIME, it segments relevant image areas but does not provide any visualisation of the importance of each pixel in these segments. As a result, all of the pixels visualized seem equally important for the annotator. We also found LIME to be particularly sensitive to its hyperparameter settings since the output changes quite dramatically when they are modified.

On the other hand, SmoothGrad is able to provide less noisy outputs than LRP and can output separate pixel intensities for each image area when compared with LIME. Furthermore, we found it has significantly fewer changes in its output when its hyperparameters are modified. Based on the aforementioned findings, we argue that SmoothGrad is more suited to the images used in this study. That said, we cannot claim which method should be used universally since it depends on the images, the model and hyperparameters selected.

[33] https://doi.org/10.4121/uuid:38dab37c-1179-495e-b357-0568b9aaaa7a

B

SmoothGrad Hyperparameters

We experimented with separate SmoothGrad [51] noise levels and number of samples for the three classification tasks in our study, namely *PA-49K Gender*, *ImageNet Vehicle* and *ImageNet Fish*. To be more exact, we experimented with the following hyperparameters:

- **Noise level:** 5%, 10%, 20%, 30%, 40%, 50%. The sample size used during these experiments was 25 since it is the default value proposed by the public implementation[7] of the SmoothGrad authors.
- **Sample size:** 2, 5, 10, 25, 50, 100. The noise level selected was the one that yielded the optimal output from the previous experiments.

For each of the aforementioned hyperparameter setups, we visualized the output for six random images from the dataset and evaluated them qualitatively to choose the optimal values. The results for all three classification tasks show that 5% noise and ten samples are sufficient to extract the heatmaps for human annotations. Furthermore, our experiments seem to suggest that the hyperparameters selected can introduce their own artefacts in the explanations. However, the absence of an interpretability ground truth does not allow us to reason about which setup provides the most reliable and faithful interpretations. The results of each setup per classification task are presented below.

B.1. PA-49K Gender

The noise level and sample size experiments for the *PA-49K Gender* dataset can be found in Figures B.1 and B.2 respectively.

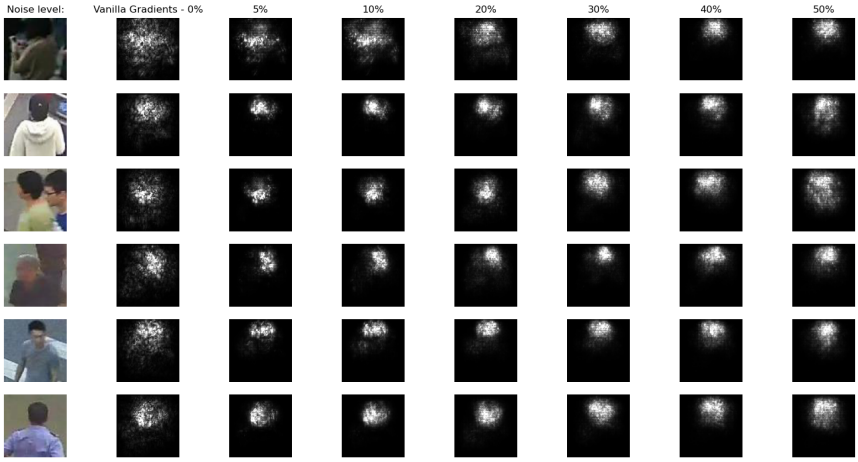


Figure B.1: SmoothGrad noise levels comparison - sample size 25.

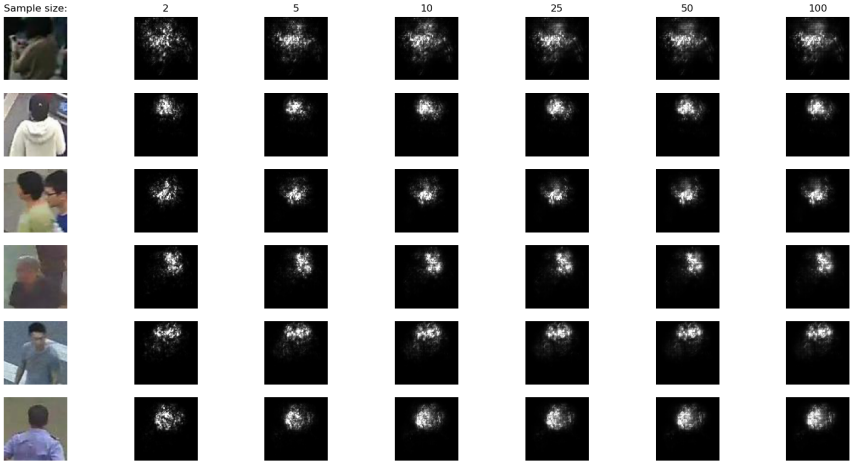


Figure B.2: SmoothGrad number of samples comparison - noise level 5%.

Figure B.1 indicates that 5% noise is already clear enough for every image apart from the first one which interestingly seems to become less noisy the more noise you add. When using a noise level of 5%, we observe insignificant differences in the explanations output from ten samples onward in Figure B.2.

B.2. ImageNet Vehicle

The noise level and sample size experiments for the *ImageNet Vehicle* classification task can be found in Figures B.3 and B.4 respectively.

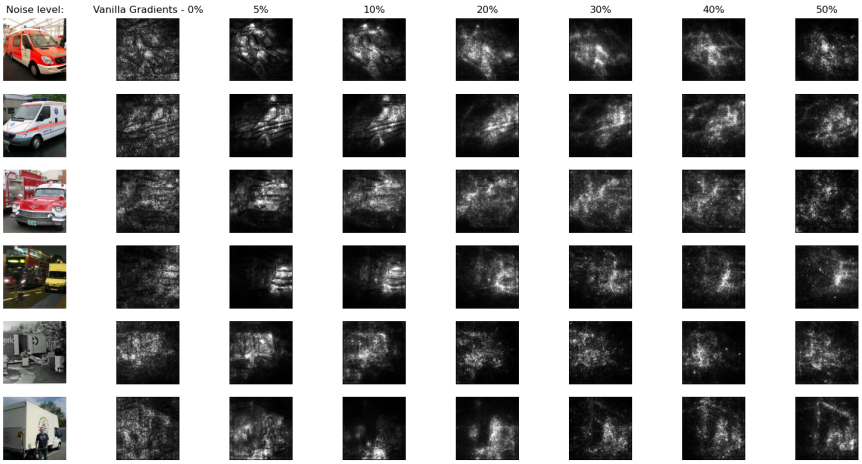


Figure B.3: SmoothGrad noise levels comparison - sample size 25.

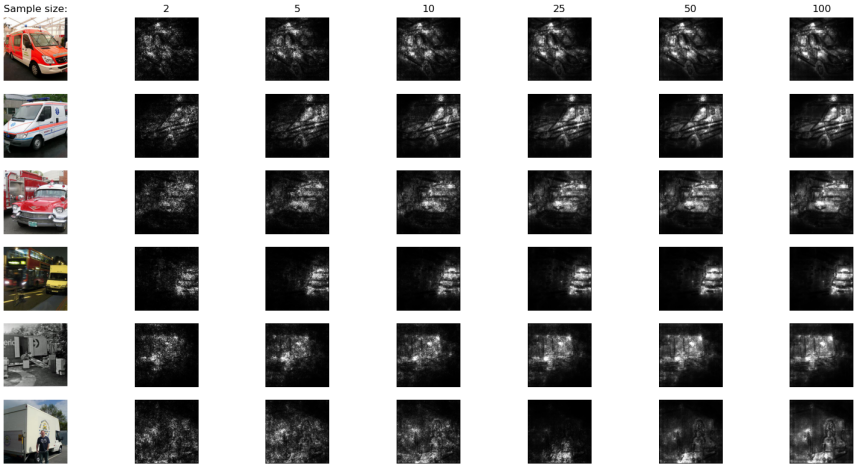


Figure B.4: SmoothGrad number of samples comparison - noise level 5%.

Similarly to *PA-49K Gender*, the resulting explanations in Figures B.3 and B.4 indicate that a noise level of 5% and ten samples are enough to provide saliency maps with significantly less noise compared to "vanilla" gradient computations.

B.3. ImageNet Fish

The noise level and sample size experiments for the *ImageNet Fish* classification task can be found in Figures B.5 and B.6 respectively.

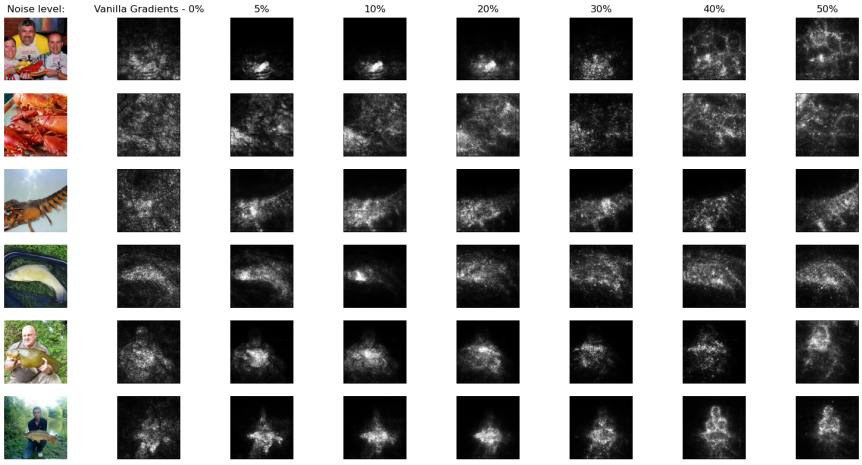


Figure B.5: SmoothGrad noise levels comparison - sample size 25.

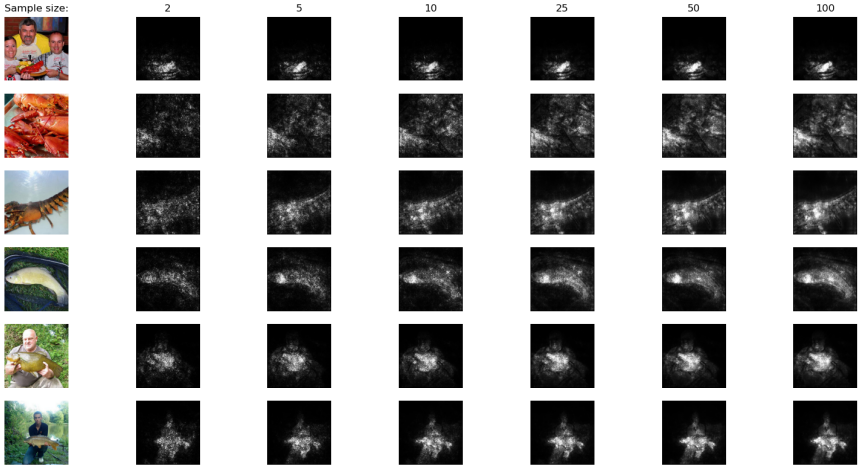


Figure B.6: SmoothGrad number of samples comparison - noise level 5%.

The resulting explanations in Figures B.5 and B.6 suggest that a noise level of 5% and ten samples are enough to produce less noisy outputs than "vanilla" gradients for the *ImageNet Fish* classification task.

C

Representation Options - Full Results

The full SEFA output for the four biased *PA-49K Gender* datasets presented in the representation options experiments (Chapter 4.2.1) can be found below.

Date vs Datetime		Date Colour		Date, Datetime & City		Coloured Date vs Datetime	
Semantic Feature	**Cramér's V**	**Semantic Feature**	**Cramér's V**	**Semantic Feature**	**Cramér's V**	**Semantic Feature**	**Cramér's V**
hour	0.93	yellow-year	0.96	city AND NOT hour	0.46	yellow-hour	0.6
minute	0.93	yellow	0.94	city AND hour	0.45	yellow-minute	0.6
white-hour	0.93	white	0.83	city_name-city AND white-hour	0.45	yellow-hour AND yellow-minute	0.6
white-minute	0.92	yellow-day	0.82	city AND day	0.45	white-minute	0.53
hour AND minute	0.9	yellow-month	0.81	white-city AND white-day	0.45	white-hour	0.52
white-hour AND white-minute	0.89	yellow-year AND yellow-day	0.81	city_name-city AND white-day	0.45	white-minute AND white-hour	0.51
minute AND shirt	0.7	yellow-year AND yellow-month	0.8	city AND day	0.45	white-day AND white-minute	0.47
hour AND shirt	0.69	white-year	0.72	white-city AND white-day	0.45	white-day AND white-hour	0.46
minute AND day	0.63	yellow-month AND yellow-day	0.72	city_name-city AND white-day	0.45	yellow-minute AND yellow-day	0.43
hour AND day	0.63	shirt	0.62	white-city AND white-hour	0.44	yellow-hour AND yellow-day	0.42
white-hour AND white-day	0.63	year AND day	0.52	hour AND NOT city	0.42	yellow-year	0.37
white-minute AND white-day	0.62	white-month	0.48	minute AND NOT city	0.42	yellow-day AND yellow-year	0.37
white	0.61	day AND month	0.46	city AND minute	0.4	yellow-year AND yellow-month	0.34
grey-shirt AND white-minute	0.57	white-year AND white-month	0.46	white-city AND white-minute	0.4	hour AND shirt	0.31
grey-shirt AND white-hour	0.57	white AND black	0.45	city_name-city AND white-minute	0.4	year	0.3
minute AND background	0.57	day	0.45	city AND NOT minute	0.39	white-month AND white-minute	0.3
white-minute AND grey-background	0.55	white-day	0.45	city AND NOT day	0.38	minute AND shirt	0.29
hour AND background	0.55	year AND shirt	0.43	day AND NOT year	0.33	yellow-month	0.29
white-hour AND grey-background	0.53	month	0.41	day AND NOT shirt	0.32	yellow-day AND yellow-month	0.29
white AND grey	0.44	year AND month	0.41	minute AND month	0.26	year AND shirt	0.29
hour AND hair	0.42	white AND grey	0.39	white-minute AND white-month	0.26	white-month AND white-hour	0.29
minute AND hair	0.42	white-year AND white-day	0.37	year AND NOT hour	0.25	yellow AND black	0.28
white-hour AND short-hair	0.4	yellow AND grey	0.37	year AND NOT minute	0.25	day AND year	0.27
white-minute AND short-hair	0.4	year	0.36	shirt AND NOT minute	0.25	yellow-day AND black-shirt	0.25
white AND short	0.34	white-day AND white-month	0.33	shirt AND NOT month	0.24	month AND year	0.24
white-hour AND grey-hair	0.29	black-shirt	0.31	day AND NOT coat	0.24	black	0.24
white-minute AND grey-hair	0.29	black	0.3	year AND NOT background	0.23	yellow-hour AND white-shirt	0.22
short	0.28	long	0.3	shirt AND NOT hour	0.23	yellow-minute AND white-shirt	0.22
short-hair	0.28	long-hair	0.3	shirt AND NOT background	0.23	yellow-year AND black-shirt	0.22
white AND black	0.28	white AND long	0.3	month AND NOT year	0.23	black-shirt	0.21
white-hour AND black-hair	0.28	long AND black	0.28	day AND NOT city	0.22		
white-minute AND black-hair	0.28	white-shirt	0.27	day AND NOT bag	0.21		
grey AND short	0.27	shirt AND hair	0.27	day AND NOT arm	0.21		
day AND shirt	0.27	long-hair AND black-hair	0.27	year AND NOT city	0.21		
white-hour AND black-shirt	0.27	month AND shirt	0.25	shirt	0.21		
white-minute AND black-shirt	0.27	day AND shirt	0.25	shirt AND NOT hair	0.21		
white-day AND grey-shirt	0.26	shirt AND background	0.25	shirt AND NOT face	0.21		
day AND background	0.25	white-year AND black-hair	0.25	shirt AND NOT nothing	0.21		
day	0.24	white-year AND white-shirt	0.24	shirt AND NOT jacket	0.21		
white-day	0.24	white-year AND long-hair	0.24	shirt AND NOT dress	0.21		
long	0.24	yellow AND black	0.24	shirt AND NOT arm	0.21		
long-hair	0.24	white-year AND black-shirt	0.23	shirt AND NOT costume	0.21		
grey AND long	0.24	hair	0.22	shirt AND NOT neck	0.21		
short-hair AND white-day	0.24	yellow-year AND grey-background	0.22				
white-day AND grey-background	0.23	yellow-year AND grey-pavement	0.22				
face AND hour	0.23	yellow-month AND grey-background	0.21				
face AND minute	0.23	yellow-month AND grey-pavement	0.21				
hour AND coat	0.22						
beige-face AND white-hour	0.22						
beige-face AND white-minute	0.22						

Table C.1: Full SEFA output for the four *PA-49K Gender* biased datasets created.

D

ACE Output - Top Five

In this section, we provide the annotated semantic features for the ACE experiments conducted in Chapter 4.2.2.

D.1. PA-49K Gender

The annotations provided for the male and female classes are visualized in Figures D.1 and D.2 respectively.

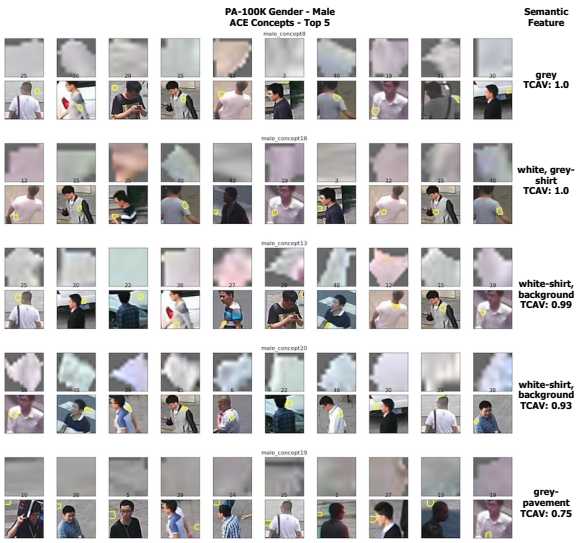


Figure D.1: PA-49K Gender - Male ACE top five.

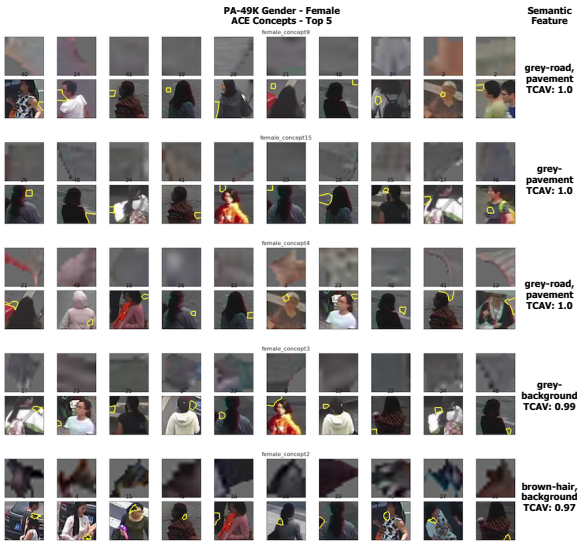


Figure D.2: PA-49K Gender - Female ACE top five.

D.2. ImageNet Vehicle

The semantic features annotated for the moving van can be found in Figure D.3.

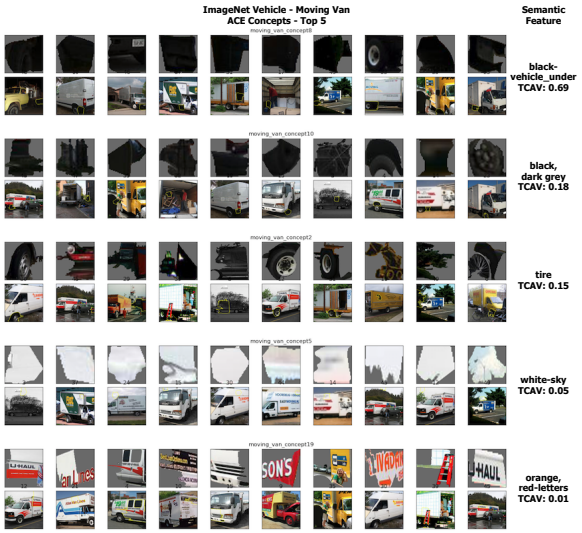


Figure D.3: ImageNet Vehicle - Moving van ACE top five.

The corresponding output for the ambulance class has already been visualized in the experiments section and can be found in Table 4.13.

D.3. ImageNet Fish

The semantic features provided for the three *ImageNet Fish* classes are visualized in Figures D.4, D.5 and D.6.

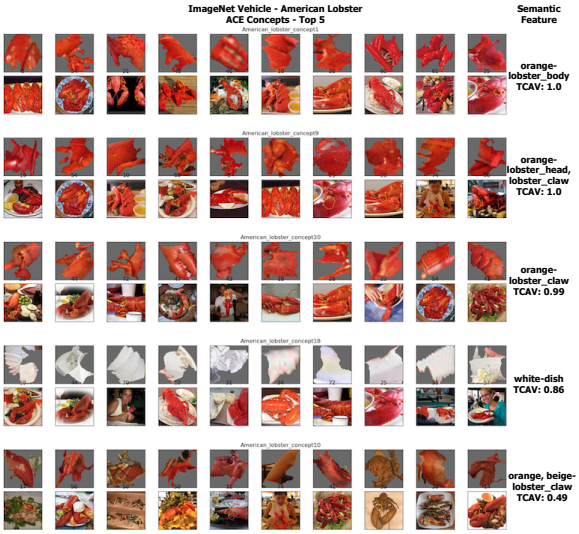


Figure D.4: ImageNet Fish - American lobster ACE top five.

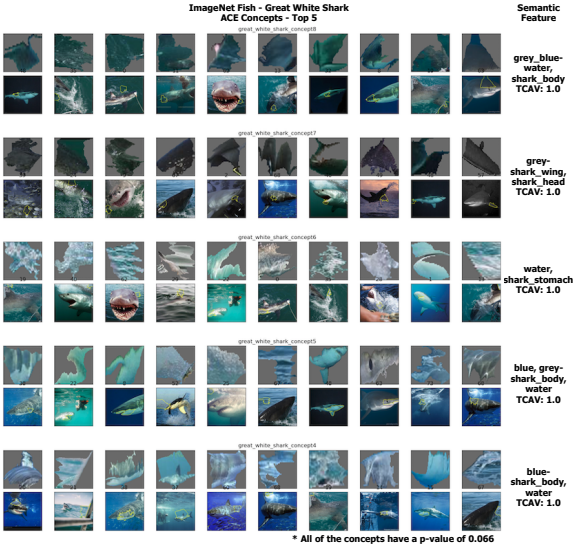


Figure D.5: ImageNet Fish - Great white shark ACE top five.

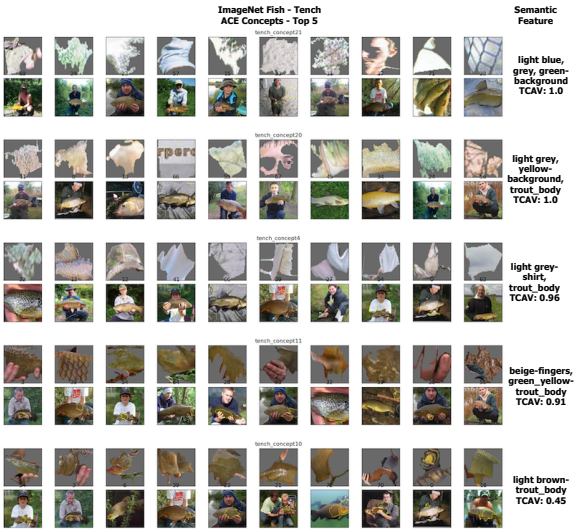


Figure D.6: ImageNet Fish - Tench ACE top five.