

POEM: Pattern-Oriented Explanations of CNN Models

by

Vargha Dadvar

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 2022

© Vargha Dadvar 2022

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

While Convolutional Neural Networks (CNN) achieve state-of-the-art predictive performance in applications such as computer vision, their predictions are difficult to explain, similar to other types of deep learning models. Different solutions have been proposed to explain CNNs, from explanations of individual image predictions, to interpretable models that approximate the predictions of the CNN model. A recent line of research focuses on explaining CNNs using semantic concepts in images, such as objects, shapes, or colors, which are easier to understand. We contribute to this line of research by proposing POEM, a framework that produces patterns of concepts to explain image classifier CNNs. POEM identifies patterns such as “If bed, then bedroom”, meaning that if an image contains a bed and the model pays attention to the bed, then the model classifies the image as a bedroom.

We first introduce the general pipelined framework used in POEM, which we also use to describe the current related solutions. Then we propose improvements in each of the pipeline steps for more accurate explanation of CNNs. We also create a web-based tool for interactive visual analysis of the patterns. Finally, we demonstrate the effectiveness of our solution using multiple use cases involving different CNN models and datasets.

Acknowledgements

I would like to thank my supervisor, Prof. Lukasz Golab, for his valuable guidance and support toward writing this thesis. I would also like to thank Dr. Divesh Srivastava for his constructive feedbacks during the course of this research.

Table of Contents

List of Figures	vii
List of Tables	ix
1 Introduction	1
2 Convolutional Neural Networks	6
3 Related Work	9
3.1 Concept Identification	9
3.2 Concept Attribution	12
3.3 Concept Pattern Mining	13
3.4 Concept Pattern Analysis and Visualization	15
3.5 Other Related Work	15
4 Methodology	19
4.1 Concept Identification	19
4.2 Concept Attribution	21
4.3 Concept Pattern Mining	23
4.4 Pattern Analysis Using the Web Interface	24
4.5 Implementation	27

5 Experiments	28
5.1 Experiment Design	28
5.2 Use Case 1: ResNet for Classifying Bedrooms, Kitchens and Living Rooms	30
5.2.1 Concept Identification	30
5.2.2 Concept Attribution	31
5.2.3 Concept Pattern Mining and Analysis	33
5.3 Use Case 2: VGG for Classifying Coffee Shops and Restaurants	36
5.3.1 Concept Identification	37
5.3.2 Concept Attribution	38
5.3.3 Concept Pattern Mining and Analysis	39
6 Conclusion and Future Work	43
References	45

List of Figures

1.1	Example input images on top, with their corresponding CAM saliency maps in the bottom [40]	2
1.2	Example feature visualizations showing what is learned by the last layer of the AlexNet CNN model [23]	2
1.3	General process of the surrogate approach for explaining black box models using interpretable methods	3
2.1	The architecture of a typical CNN model, including the feature extractor and classifier components [28]	7
2.2	Activation map created as a result of applying the filter over all parts of the input image or the activation maps of the previous layer [22]	8
3.1	Network dissection process for checking the overlap between concept segments and filter activations for each input image in the Broden dataset [4]	10
3.2	The process used in ACDTE for explaining the prediction of an image using the concepts found in a set of similar images [10]	11
3.3	Example decision tree created by ACDTE for explaining the concepts playing a role in the prediction of a place image [10]	13
3.4	Example decision tree (left) and corresponding sample images (right) created using the CNN2DT method [18]	14
3.5	Sample explanations from [6], where colored points show the pixels important in the recognition of digits	16
3.6	Sample decision tree built using [12] for explaining a digit recognition CNN	17
3.7	The clustering approach used in ACE to find the importance of different concepts for prediction of each class [14]	18

4.1	Overview of POEM’s pipeline	20
4.2	Concept identification process used in POEM based on Network Dissection method, which is repeated for each image, concept and filter	20
4.3	Concept attribution process used in POEM, which is repeated for each concept in each image	22
4.4	POEM web interface showing the patterns for bedroom vs. kitchen vs. living room classes	26
5.1	Concepts identified using the previous approach (a) and POEM (b) for Use Case 1 in different categories, with number of units (filters) mapped to each concept displayed through the vertical bars	31
5.2	Sample images with ‘bus’ concept high-activation areas highlighted, based on the previous approach	32
5.3	Sample images with ‘chair’ high-activation areas highlighted, based on the previous approach	33
5.4	Sample images with ‘sofa’ high-importance areas highlighted, based on POEM	34
5.5	Sample images matching pattern 2 in Table 5.1b, but wrongly-predicted by the model	36
5.6	Concepts identified using the previous approach (a) and POEM (b) for Use Case 2 in different categories	37
5.7	Sample images with ‘toilet’ concept high-activation areas highlighted, based on the previous approach	38
5.8	Sample images with ‘sea’ high-importance areas highlighted, based on POEM	39
5.9	Sample images with ‘swivel chair’ high-activation areas highlighted, based on the previous approach	40
5.10	Sample images with ‘red’ high-importance areas highlighted, based on POEM	41
5.11	Sample non-matching images of pattern 4 in Table 5.2b	41

List of Tables

5.1	Top patterns for Use Case 1 extracted using the previous approach (a) and POEM (b)	35
5.2	Top patterns for Use Case 2 extracted using the previous approach (a) and POEM (b)	42

Chapter 1

Introduction

Deep learning models have achieved state-of-the-art predictive performance in many applications, including computer vision and natural language processing. However, their predictions are not human-interpretable. This lack of explainability can be a barrier to more widespread adoption of deep learning models, especially in critical applications such as healthcare and law where the decisions should be transparent and justifiable [16]. In this work, we focus on explaining image classifier Convolutional Neural Networks (CNN), which are commonly used in computer vision applications involving classifying image inputs. We provide further background about CNNs in Chapter 2.

The approaches to explaining deep learning models can be generally categorized into *local* explanations, which explain single predictions of the model, and *global* explanations, which interpret the general decision-making of the model. In the case of CNNs, local explanations usually include identifying the importance of each pixel in the input image towards the prediction of the model, which forms the *saliency map* of the image [30, 36, 40, 29]. Figure 1.1 shows examples of saliency maps created using the *Class Activation Mapping (CAM)* method [40], where more red areas indicate higher importance for the prediction of the related image. For example, we can see how the upper part of a teapot is important toward detecting the image as a teapot. However, to find whether such an observation extends to all teapot images, we need to manually inspect the saliency maps for all the dataset images predicted as teapot.

For global explanations of CNNs, a common approach is to generate synthetic visualizations, also called *feature visualizations*, which can explain what is learned by each layer of a CNN model [23, 26, 24, 27]. Some examples of such visualizations are shown in Figure 1.2. Unlike saliency maps, feature visualizations can provide insights about what

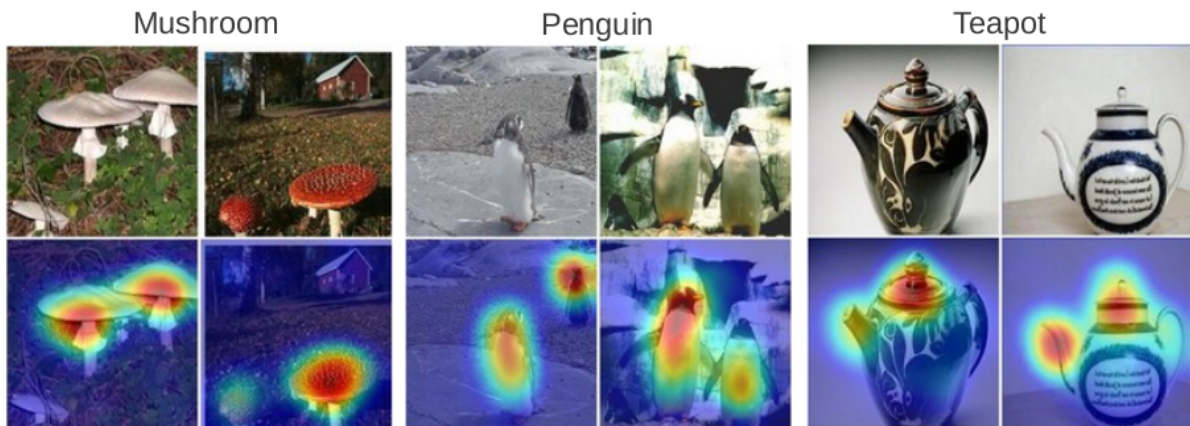


Figure 1.1: Example input images on top, with their corresponding CAM saliency maps in the bottom [40]

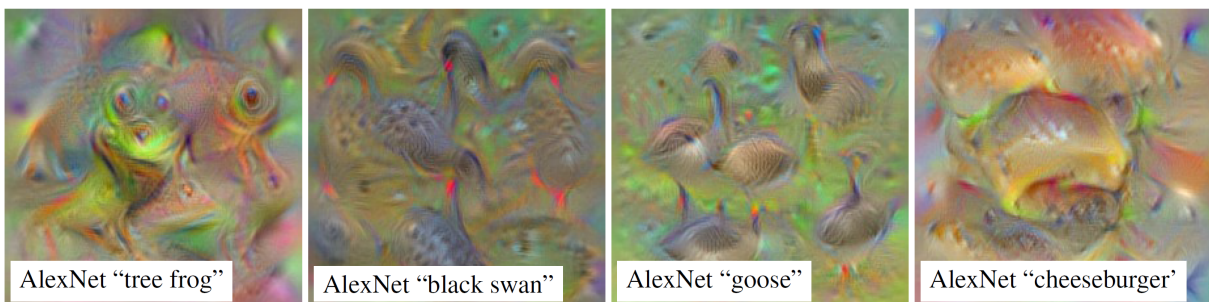


Figure 1.2: Example feature visualizations showing what is learned by the last layer of the AlexNet CNN model [23]

the CNN model learns in general. However, some of these visualizations may be hard to interpret. Moreover, we may have hundreds of such feature visualizations for each CNN model, which need to be analyzed manually to gain insights about what the model learns.

In response to the limitations of saliency maps and feature visualizations, some recent methods have focused on identifying the semantic concepts learned by a CNN model, which are easier to interpret, such as objects, shapes, colors and textures [4, 14, 19, 11, 9, 20]. Automated identification of such concepts learned by a CNN removes any need for burdensome human analysis of the results. The individual identified concepts, however, may not be enough to find out how different combinations of concepts can lead to prediction of different classes by the CNN. For example, we may want to know what class (bedroom



Figure 1.3: General process of the surrogate approach for explaining black box models using interpretable methods

or kitchen) the model predicts when the model pays attention to both a bed and a chair in an image.

In order to find interesting patterns relating the identified concepts to CNN model predictions, we can use the *surrogate* approach, also called *model distillation*. Using this approach, interpretable models such as decision trees or rule mining methods are used to approximate the predictions of opaque models such as CNNs. Figure 1.3 shows the general process used in the surrogate approach. In this process, the predictions of the black box model (e.g. the CNN) are used as the labels of the input data examples (e.g. images), instead of their ground-truth labels. Then this labeled dataset is used to train an interpretable model such as a decision tree, or to find patterns using a rule mining method. In the case of image classification using CNNs, image pixels are the input features that can be fed to the interpretable model along with the CNN predictions as labels. However, explanations based on image pixels may not be interpretable, and do not take into account what the internal components of the CNN are learning. To address these issues, a better alternative is to use the identified concepts learned by the CNN as the image features given to the interpretable model for explanation.

Based on this concept-based surrogate approach, we propose a new framework called *POEM* for explaining the image classifier CNN models using patterns of semantic concepts. The inputs given to POEM are a CNN model and a related target dataset, which is a dataset of images the CNN is trained to classify. The output is a set of patterns linking concepts in dataset images with CNN model predictions. For instance, in the case of a CNN model which classifies room images into bedrooms or kitchens, POEM may identify a pattern in the form “if bed, then bedroom”. This pattern indicates that if an image contains a bed and the model pays attention to the bed in the image during inference, then the model classifies the image as a bedroom rather than a kitchen. Such patterns can serve as explanations describing the prediction behavior of the CNN model based on the concepts in images.

In order to describe our solution and the improvements over the related work, we introduce the general framework used in POEM which includes the following three steps:

1. *Concept Identification* finds the set of important concepts which the model generally learns for classifying images.
2. *Concept Attribution* associates each input image with one or more of the identified concepts which the model pays attention to while classifying the image.
3. *Concept Pattern Mining* produces a set of interpretable rules or patterns linking concepts to model predictions.

As we explain in Chapter 3, a few related works have applied a similar rule-based surrogate approach to explaining CNNs using concepts [18, 32, 10, 38]. However, they have some shortcomings in their implementation of each of these steps, including how relevant and interpretable are the concepts identified, how much the concepts attributed to each image are actually important toward the model’s prediction, and how varied and informative are the patterns mined from the image concepts.

We address these issues by applying more effective methods in each step to have more accurate concepts identified and attributed to images. We also apply an ensemble of rule mining methods to find more informative patterns to explain the model. Two of these methods have not been applied previously for explaining CNNs. Furthermore, we create a web-based tool available online¹ which enables us to view the list of patterns and visually analyze their related data to gain insights about the CNN model. Further details of our approach in each step and the web-based tool can be found in Chapter 4. We then demonstrate the effectiveness of POEM compared to previous work through two use cases involving different CNN models and datasets, as explained in Chapter 5. Finally, Chapter 6 includes a summary of the limitations of our work and potential future research directions.

In summary, here are our contributions:

- We introduce a three-step framework for representing and comparing the surrogate approaches to explaining CNNs using patterns of concepts, including our method.
- We propose POEM as a more effective implementation of this framework based on improvements in each pipeline step, which results in more accurate explanations of CNNs.

¹<http://poem.lg-research-1.uwaterloo.ca> - Note that this is a temporary address which may not be available over long term

- We create a web-based tool for interactive visual analysis of the patterns and their related data.
- We perform experiments using two different use cases to demonstrate the effectiveness of POEM in explaining CNNs.

Chapter 2

Convolutional Neural Networks

Here we provide some background about the CNN architecture for image classification, as needed to understand our method and the related work. The input to a CNN is a matrix including pixel intensities of the input image, which is 2-dimensional for grayscale images and 3-dimensional for colour images. In this text, for brevity we consider the input image to be a 2D matrix.

A CNN model includes a *feature extractor* and a *classifier* component, as shown in Figure 2.1. The feature extractor receives the input image and passes it through multiple consecutive *convolutional* layers, which transform the input pixels to detect important features for the prediction. These transformations are done using multiple *filters* in each layer, and the output of each filter is a matrix called an *activation map* or *feature map*, as Figure 2.2 shows. Activation maps actually capture what a filter has learned from an image and passes forward in the network.

The activation maps from the last convolutional layer are then given to the dense fully-connected layers of the classifier to produce the model outputs. The output is a vector, where each element corresponds to the output value for one of the target classes. The class with the highest output value is selected as the prediction of the model.

During training, the training input images are passed through the model, and the convolutional filters and dense weights are updated by *backpropagating* the class prediction errors. The *gradients* of the prediction errors with respect to each of the filters are used to decide how much and in what direction each filter should be updated. In this way, the filters learn to identify those features in images which can minimize the overall prediction error of the model.

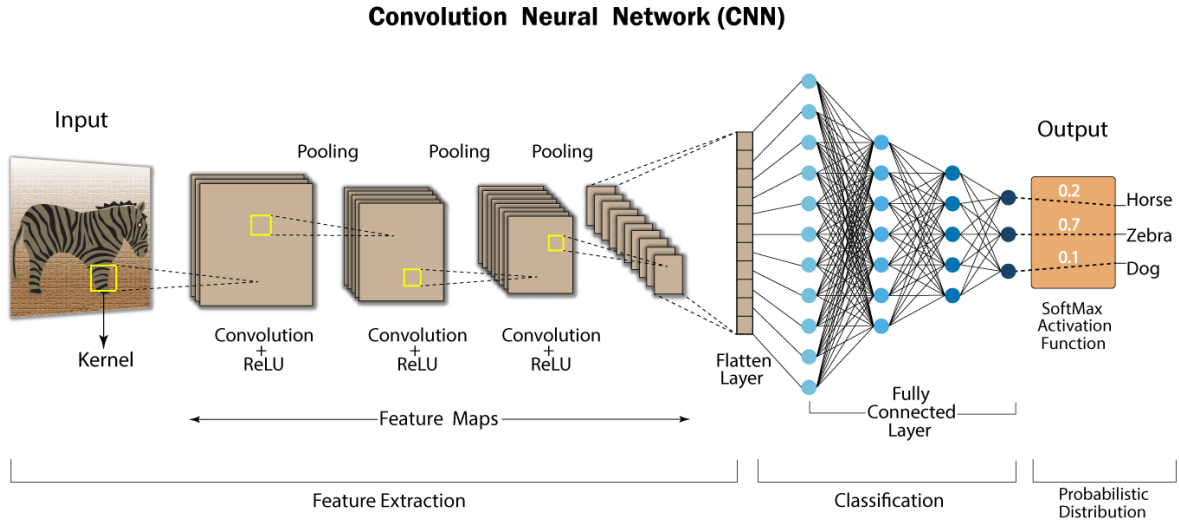


Figure 2.1: The architecture of a typical CNN model, including the feature extractor and classifier components [28]

Previous research shows that the first convolutional layers tend to identify simple features such as edges and corners in input images, while the deeper layers are more likely to detect high-level objects and shapes [23, 29, 4, 5]. Moreover, the output from the last convolutional layer includes what the model has learned from the input as passed to the final classifier part. For this reason, we focus on the filter activation maps of the last convolutional layer of the network for identifying the concepts, as they are more likely to identify high-level human-understandable concepts.

Finally, when we mention *target dataset* in this text, we mean a dataset of images which is related to a CNN model's goal. For instance, if a CNN model is trained to distinguish between different types of indoor places, a dataset of indoor place images is a target dataset for this model. The target dataset may be the same as the training set of the model, though we use a test set as the target dataset in our use case experiments in order to avoid any bias in the pipeline process.

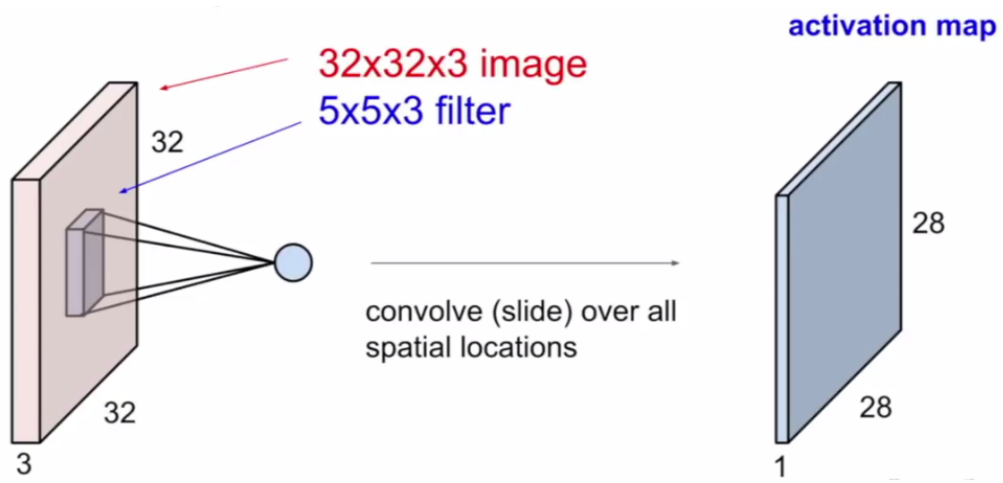


Figure 2.2: Activation map created as a result of applying the filter over all parts of the input image or the activation maps of the previous layer [22]

Chapter 3

Related Work

In this chapter, we review the related works based on their approach to each of the steps in the pipeline we proposed, and highlight their differences with our work. We focus mostly on related works which, similar to POEM, use a concept-based rule mining approach to explain CNNs. In the last section of this chapter, for completeness we briefly review some of the methods which use a surrogate approach for explaining CNNs without using concepts, as well as the significant methods which provide concept-based explanations of CNNs without taking advantage of an interpretable surrogate model.

3.1 Concept Identification

The goal of concept identification is to identify the concepts learned by the CNN model, which is usually done using a dataset of images as input. The output from this process can be either a set of concepts or filter-concept mappings which show what concept is mostly learned by each filter in a target convolutional layer of the CNN.

Multiple concept identification methods have been proposed recently. Some of these methods analyze a pretrained CNN model in order to identify the concepts learned without modifying or retraining the model [4, 14, 11, 19]. Other methods have proposed modifications to the architecture of the CNN model in order to direct it toward revealing the concepts learned [9, 38, 20]. Those related works which use a rule mining approach similar to POEM, have applied some of the mentioned concept identification methods or their variants as the first step in their process. We explain some of these methods here, while in Section 3.5 we review some of the other concept identification methods which generate independent concept-based explanations without a surrogate approach.

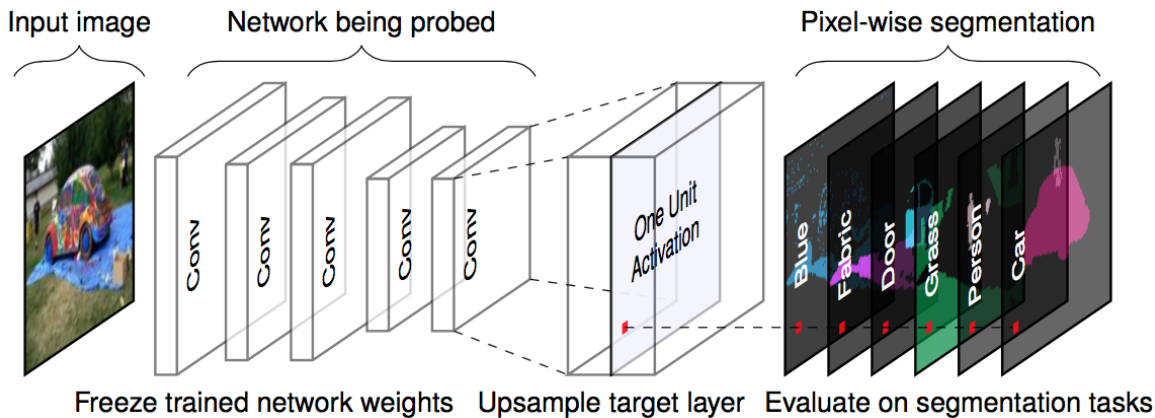


Figure 3.1: Network dissection process for checking the overlap between concept segments and filter activations for each input image in the Broden dataset [4]

A system relatively similar to our pipeline is *CNN2DT* [18], which uses *Network Dissection* [4] for concept identification. In this method, images from a fixed pre-segmented dataset called *Broden* are passed through the CNN to find out which segments (concepts) in images lead to higher activations in each filter of the last convolutional layer, as shown in Figure 3.1. Then the overlap between each concept and each filter’s high-activation areas are measured. Finally, each filter is mapped to a concept it has the most overlap with over all the images. As we show in our experiments in Chapter 5, a limitation of this method is that some of the concepts identified may not be relevant to the task and classes of the CNN model. This happens because of using a fixed secondary dataset rather than the target dataset of the CNN.

In *ACDTE* [10], a concept identification method is proposed for local explanation of single image predictions. As Figure 3.2 shows, this method finds and segments a set of images similar to the image to be explained. It then groups the filter activations of image segments into different clusters, where each cluster potentially represents a single concept. Using this approach, human inspection is required to map each cluster to the corresponding concept. Moreover, each cluster is not guaranteed to represent only a single concept, and may include image segments of multiple different concepts. *ERIC* [32], which is a method for mining rules relating filters to model predictions, has the same shortcoming, as it requires manual inspection of images and their activations to identify filter-concept mappings.

Finally, methods such as [37] add new convolutional layers to the CNN model and use an optimization process to direct the newly-added filters to reveal the concepts learned by

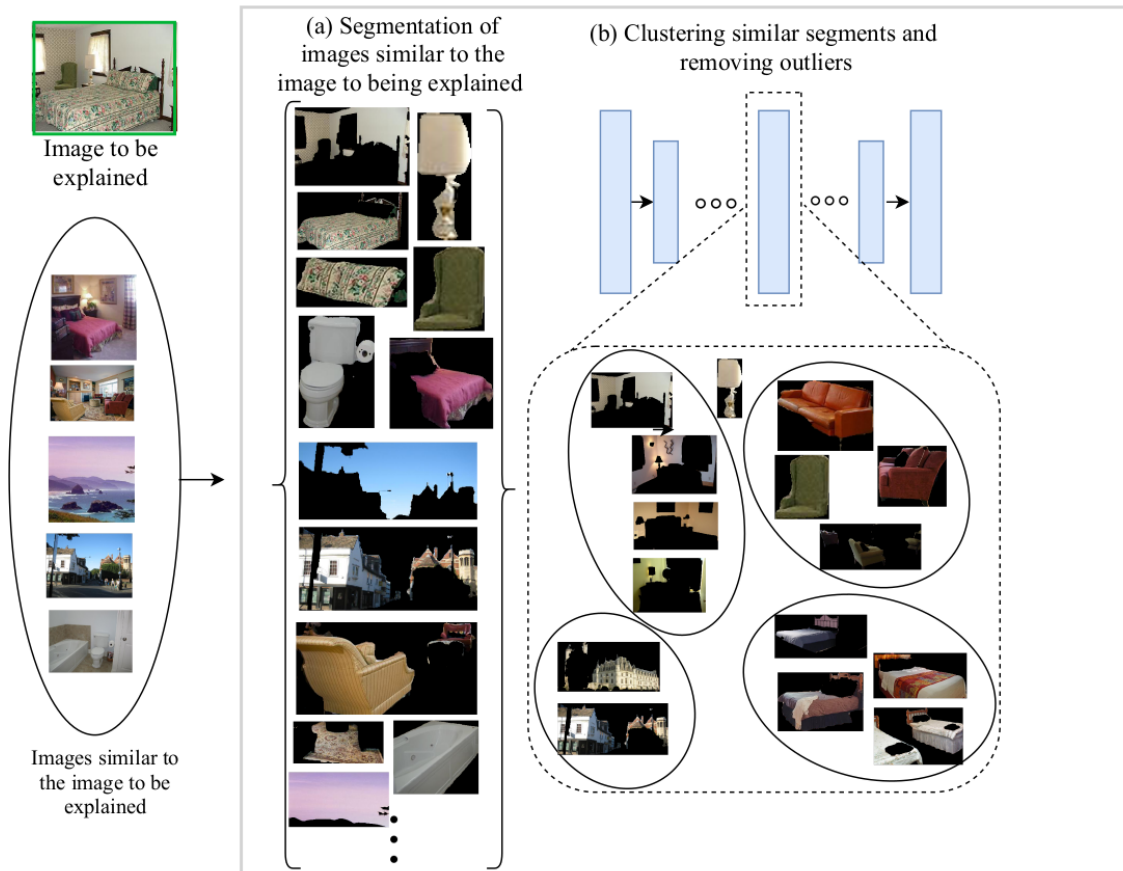


Figure 3.2: The process used in ACDTE for explaining the prediction of an image using the concepts found in a set of similar images [10]

the model. This requires modifying or partial retraining of the model analyzed, which may not be always desirable.

As we explain further in Section 4.1, we address these issues by using a more recent version of Network Dissection [5]. This method enables us to identify a wide range of concepts automatically from any CNN model and target dataset, without needing to use a secondary dataset and to modify or retrain the model. This also leads to identifying concepts which are relevant to the task of the CNN model and the related classes.

3.2 Concept Attribution

After the concepts learned by the filters of the model are identified, the next step is to associate each input image in the target dataset to a subset of these concepts that played a role in the model’s prediction for that image.

Towards this goal, a naive solution is to only check those concepts which are present in an image. This is similar to the classic surrogate approach where the features of input data are used directly as the inputs to the surrogate model. The problem with this approach is that, only knowing that a concept is present in the image does not mean the model is looking at the concept for its prediction. In order to reach such a conclusion, the internal components of the model (i.e. the filters in the case of CNNs) need to be analyzed.

In most related work [18, 10, 32], those filters are considered for concept attribution of an image which are highly activated when the image passes through them. Using the filter-concept mappings from the concept identification step, such an approach can identify the important concepts for an image. While high activation can be an indication that a filter pays attention to an image, it does not necessarily mean that the filter’s activation map has actually played a significant role in the final prediction of the model. Furthermore, we are not sure if the high-activation region in the filter’s activation map actually corresponds to the location of the learned concept in the image.

As we explain in Section 4.2, by employing a semantic segmentation model that segments an image into its constituting concepts, we make sure that the high-activation area overlaps with the concept in the image. We also check the gradients of the model to ensure the concept actually plays a role in the final predicted class for the image. As we demonstrate in Chapter 5, these checks lead to more accurate patterns explaining the CNN model’s behavior. Also because fewer concepts are linked with each image, insignificant concepts with few occurrences over all images can be filtered out to have faster pattern mining and more concise patterns in the next step.

An alternative approach for concept attribution can be based on *counterfactual* or *perturbation-based* explanations [34, 15, 1]. Instead of analyzing the filter activations and gradients in a CNN model, methods using the counterfactual approach perturb an image to find out removing which parts or concepts from an image leads to a different prediction by the model. As we explain in Chapter 6, we will explore a counterfactual approach for concept attribution in a future work.

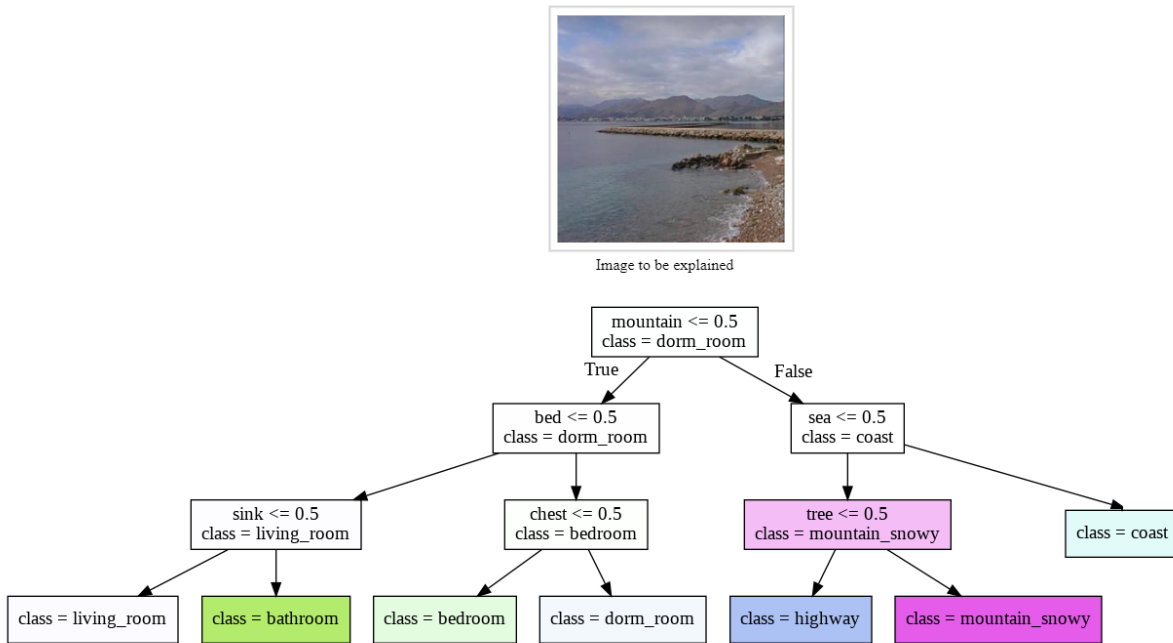


Figure 3.3: Example decision tree created by ACDTE for explaining the concepts playing a role in the prediction of a place image [10]

3.3 Concept Pattern Mining

The goal of this final step is to find interesting patterns linking the concepts attributed to images to specific CNN model predictions. Such patterns can provide potential insights about the model’s prediction rationale and help to audit the model and the target dataset.

Almost all related methods have used some form of decision trees as surrogate models to explain the model predictions using concepts [38, 18, 10, 32]. This is mostly because decision trees are generally intuitive and fast to build. Also it is possible to interpret each root-to-leaf path in the tree as a rule or pattern to analyze. Figure 3.3 shows an example decision tree created using ACDTE [10] for explaining the classification of a place image. Another decision tree example created by CNN2DT [18] can be seen in the left side of Figure 3.4, where each splitting node of the tree represents a concept.

The downside with the rules extracted from decision trees is that, as the tree grows deeper, the rules become more narrow and less concise. Moreover, the order of concepts used to split the tree nodes limits the variety of the patterns. For example, the patterns “If



Figure 3.4: Example decision tree (left) and corresponding sample images (right) created using the CNN2DT method [18]

bed, then bedroom” and “If chair, then bedroom” cannot coexist in a tree, because only one concept can have a chance to be chosen as the root node of the tree. Even further, decision trees do not allow overlapping patterns. For instance, the tree cannot include both the patterns “If bed, then bedroom” and “If bed and chair, then bedroom”. These properties limit the variety and conciseness of the patterns extracted from a decision tree.

To have more varied and informative patterns, we apply an ensemble of rule mining methods. In addition to CART (Classification and Regression Trees) [7], we use two recent rule mining methods which have not been used previously for explaining CNN models: *Explanation Tables* [13] and *Interpretable Decision Sets* [21]. Both methods find concise and potentially-overlapping patterns which can be used for explaining the model. We should emphasize that we use these patterns independently for explanation of the model predictions over the target dataset, rather than building an interpretable classifier which approximates the CNN model predictions for new data examples.

3.4 Concept Pattern Analysis and Visualization

After a set of concept patterns are found, further analysis is required to extract potential insights about the CNN model and the target dataset. Most related works have provided limited analysis of the resulting trees or rules extracted and the data examples related to each rule [10, 32, 38]. In [18], it is possible to visualize the decision tree, and check the most activated images matching each concept (node) in the tree, as shown in Figure 3.4. However, the rules and their matching data are not analyzed further to evaluate the model’s rationale based on concepts. Also because only the top activated images are visualized, potential problems in the concept identification and attribution processes are not revealed.

We analyze the patterns further by visual exploration of different images matching a pattern’s concepts. This includes those images matching or not matching the model prediction stated in a pattern, as well as those images having a ground-truth label different than the model’s prediction. Looking at such interesting subspaces of data helps us to evaluate the model, and identify the potential characteristics of the data examples related to each pattern, including any mislabeled examples. Some of these data categories related to each pattern have been explored previously in *RuleMatrix* [25] for analysis of the patterns extracted from neural networks trained on tabular data. As we explain in Section 4.4, we extend this to the context of CNNs and image data by creating a web-based tool for interactive analysis of concept patterns and their related image examples.

3.5 Other Related Work

There are other related works which use a surrogate approach to explaining CNNs, but without relying on concepts as the input image attributes fed to the surrogate model [12, 8, 6, 2]. The feature attributes of patterns in such methods are either specific image regions or CNN filters, both of which provide explanations which are not easily interpretable. For example in [6], a set of rules are generated to link the input pixels to the convolutional filters, and another set of rules are created to relate the convolutional filters to the model output. Chaining these two sets of rules helps to identify those image pixels important towards the prediction of the model, as the green and red points in Figure 3.5 show for the problem of digit recognition. Such rules can only provide local explanations for single images, because pixel locations are not consistent among different images and cannot lead to meaningful global explanations of the CNN.

In methods such as [12], the filter activations are used to generate a binary decision tree. Figure 3.6 shows an example of such a tree for the digit recognition problem. We

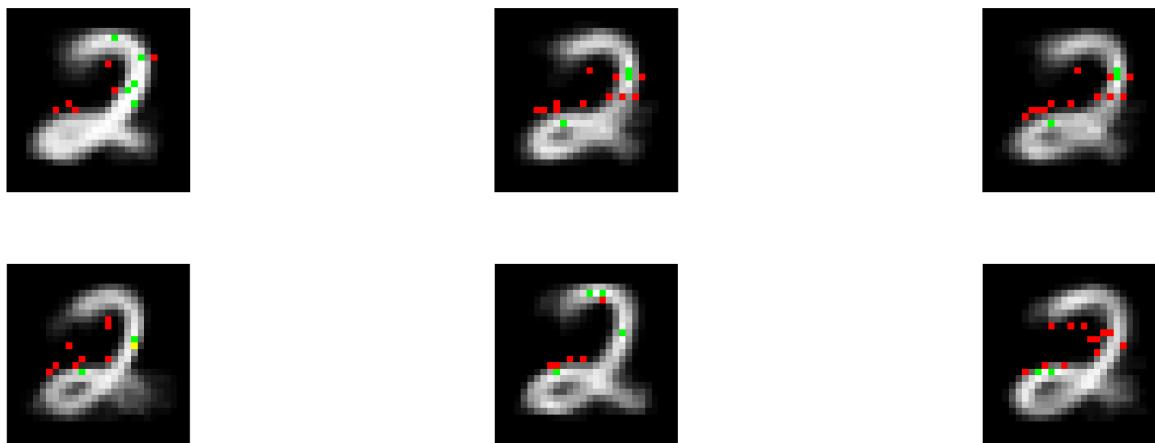


Figure 3.5: Sample explanations from [6], where colored points show the pixels important in the recognition of digits

can see that the filter outputs used as nodes in the tree are not easily interpretable when compared with specific concepts such as shapes, textures and colors in the images.

Another group of related work are those concept-based explanation methods which do not use any surrogate interpretable model for explaining CNNs [14, 11, 19, 9]. The outputs from these methods are usually in the form of importance weights assigned to different concepts for the prediction of each class. For example, *TCAV* [19] trains different concept detector classifiers to test whether a CNN filter activation map represents a specific concept of interest or not. Then it passes the images of a specific class (e.g. bedroom) through the CNN and classifies the filter activations using the concept detectors. As a result, the importance of each concept for the prediction of the class can be estimated. *TCAV* requires a dataset of examples for each of the concepts of interest to train the concept detectors.

Another method providing concept scores is *ACE* [14], which uses a clustering approach similar to *ACDTE*, but for global explanations. As Figure 3.7 shows, the images are first segmented in different resolutions. Then the segmentations are passed through the CNN model, and their last layer filter activations are grouped into different clusters. Finally, assuming that each cluster corresponds to a specific concept, the importance score of each concept is determined using the *TCAV* method. *ACE* has some limitations similar to *ACDTE*, including the manual analysis required to map each cluster of segments to the corresponding semantic concept.

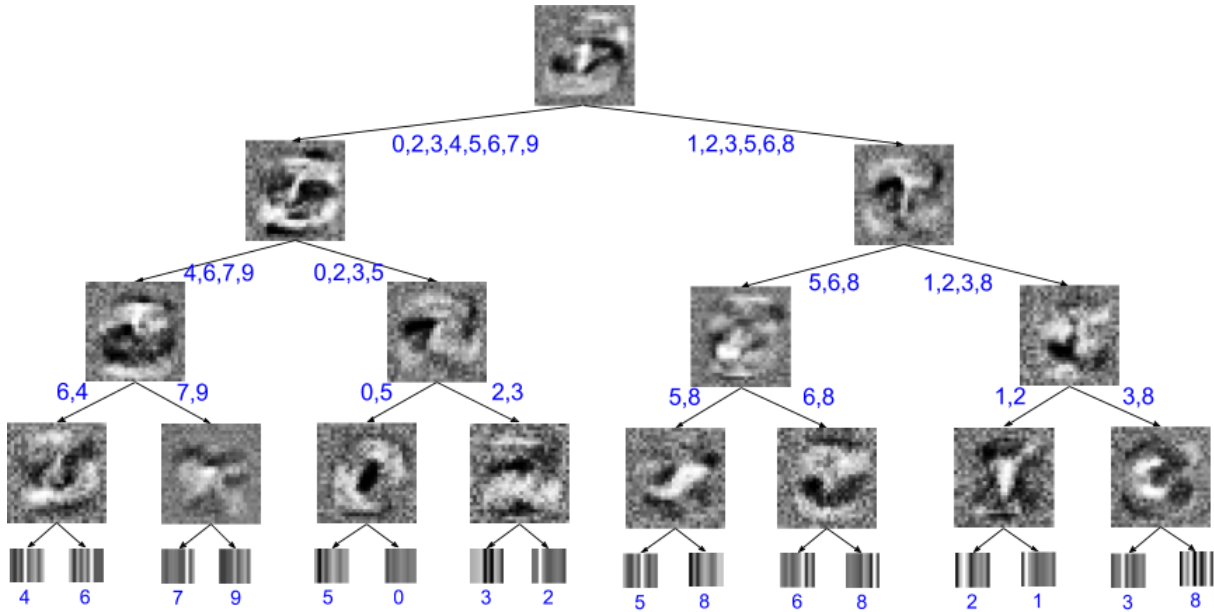


Figure 3.6: Sample decision tree built using [12] for explaining a digit recognition CNN

A general limitation of TCAV, ACE, and other related approaches is that using these methods, automating the process of producing concept-based explanations for any CNN model of interest is not straightforward, and requires knowing the concepts of interest beforehand and having access to concept datasets. Moreover, such methods usually do not reveal how different combinations of concepts (not just single concepts) are related to the model predictions. We may want to know how the CNN model behaves when multiple concepts of interest exist in images. This is where a rule-based surrogate approach similar to POEM which provides patterns of concepts can be helpful.

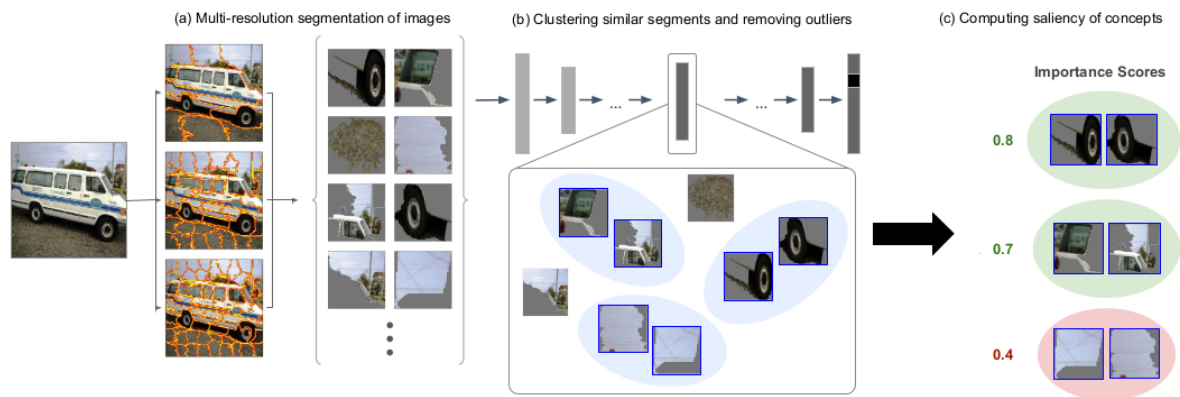


Figure 3.7: The clustering approach used in ACE to find the importance of different concepts for prediction of each class [14]

Chapter 4

Methodology

POEM includes three main modules illustrated in Figure 4.1, which correspond to the three steps in our framework.

4.1 Concept Identification

In the concept identification module, the goal is to identify the concepts detected by the filters in the last convolutional layer of the CNN model. Figure 4.1 shows three example filters being mapped to their corresponding concepts ‘bed’, ‘stove’ and ‘headboard’ as a result of concept identification. As mentioned earlier, we use a new version of Network Dissection for concept identification [5].

Figure 4.2 shows the concept identification process performed for a specific filter and the ‘bed’ concept in a sample input image. First, each image in the target dataset needs to be segmented to its constituent concepts. For this purpose, we use a semantic segmentation model called *UPerNet* based on the *Unified Perceptual Parsing* approach [35], which is pretrained on the *Broden* dataset [4] to identify a wide range of concepts in different categories, such as objects, object parts, materials and colors. The Broden dataset itself includes multiple segmented scene and object datasets. This semantic segmentation model can potentially be pretrained on other segmented datasets in order to segment images based on any other specific set of concepts which are meaningful in a specialized target domain such as medical image analysis.

After identifying the concepts present in each image, we pass each image through the CNN and measure the pixel overlap between the concepts and high-activation areas of each

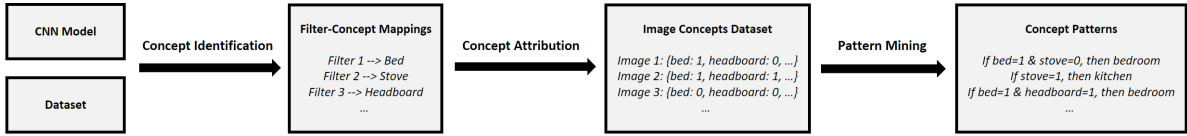


Figure 4.1: Overview of POEM’s pipeline

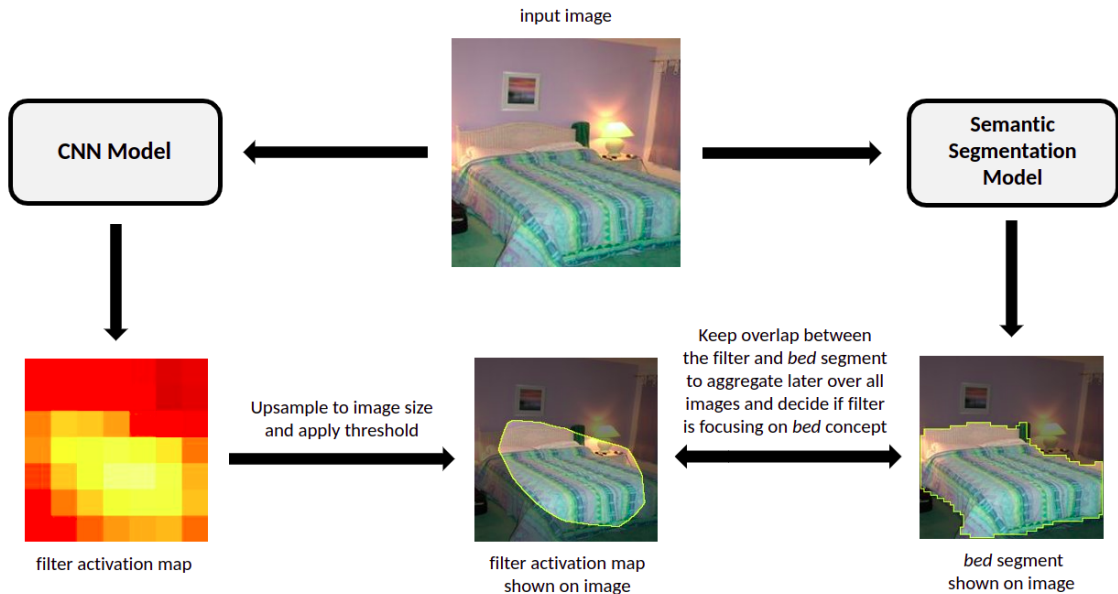


Figure 4.2: Concept identification process used in POEM based on Network Dissection method, which is repeated for each image, concept and filter

filter’s activation map. For this purpose, each activation map needs to be *upsampled* (i.e. resized) to the size of the input image. High-activation areas for a filter are those activation values which are higher than 99% of the filter’s activation values over all the images.

After repeating this process for all the images, we map each filter to the most likely concept it is detecting. This is the concept having the most overlap with high-activation areas of the filter over all images. This overlap is computed using *Intersection over Union (IoU)*, which measures the ratio of the total intersection between concepts and high-activation areas over their union. As in the Network Dissection method, we ignore those weak filter-concept mappings which have an IoU lower than 0.04. While each filter is only mapped to a single concept it is learning the most, multiple filters may be potentially mapped to the

same concept.

4.2 Concept Attribution

The goal of concept attribution is to attribute to each image the most likely concepts that play a role in the model’s prediction decision. Figure 4.1 shows three examples as the output of concept attribution. The process we use to decide if a concept should be attributed to an image or not is shown in Figure 4.3 for the ‘bed’ concept and a sample image predicted as ‘bedroom’ by the CNN. We check the following conditions for attributing concepts to images:

1. The concept should be present in the image.
2. A filter mapped to the concept (as a result of concept identification step) has played a role in the model’s prediction for the image, as indicated by the filter’s activations and gradients.
3. There is a significant overlap between the concept’s location in the image and the high-importance area in the related filter activation map.

To check condition 1 above, we use the semantic segmentation model explained earlier, which allows us to locate the concept mapped to each filter in the image.

We check condition 2 using a process inspired by *Grad-CAM* [29], which is a method for explaining single image predictions based on the CNN model gradients. As we explained in Chapter 2, a CNN classifier outputs a value for each of the classes to be predicted, and the class with the highest output value is selected as the predicted class. Similar to Grad-CAM, we compute the gradients of the predicted class output for the image with respect to each of the filter’s activation values. We then multiply the gradients by the corresponding activation values element-wise, to take into account the effect of both the gradients and the activations. We also apply the *ReLU* function to the results in order to focus on those activation values which have a positive effect on the predicted class. We finally have an activation-gradient map for each filter, which we simply call the *filter saliency map*, and has a similar size to the filter’s activation map. The values in each filter’s saliency map are good indicators of the importance of each activation value for the model’s prediction.

Our approach is different from Grad-CAM in that we do not aggregate the saliency maps of different filters to create a single saliency map for the image. The reason is that

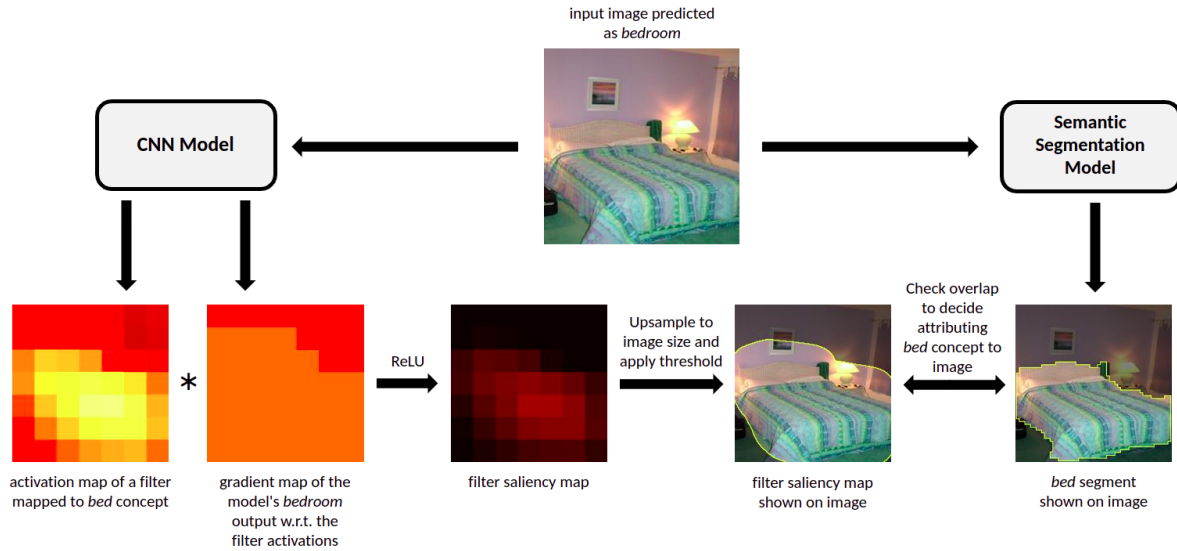


Figure 4.3: Concept attribution process used in POEM, which is repeated for each concept in each image

we are interested in evaluating the importance of each single filter (concept) toward the prediction of the model. Moreover unlike Grad-CAM, we do not average the gradients of a filter before multiplying by the activations. This is because we need to know the importance of each single value in a filter’s activation map when later we check their overlap with the concept segment in the image.

Now in order to decide the high-importance parts of the saliency map for each filter, we use as threshold the 95th percentile of all the activation-gradient values over all the filters for the current image. Activation-gradient values higher than this threshold have the most impact on the prediction of the model. Our experiments show that most values below this threshold are very close to zero and have little importance toward the prediction.

Having the high-importance mask for each filter, we can check its overlap with the location of the concept mapped to the filter in the image, which is condition 3 mentioned above. To check this overlap, we need to upsample the filter’s activation-gradient map to the size of the input image. We then check that at least 50% of the high-importance area of the filter is covered by the concept found in the image. In this case, our conditions for attributing the concept to the image are satisfied, and we can set the concept value to 1 for this image. Otherwise, we set the concept to 0, which means that we do not have reliable evidence that this concept is important toward the model’s prediction of this

image. It may be possible to give different weights to the attributed concepts based on their level of overlap with high-importance filter areas or the number of related filters activated. However, for simplicity we only consider binary concept attributes here and leave addressing such complexities for future work.

After we finish concept attribution for all the images in the target dataset, we discard the weakest concepts, which are those concepts associated with less than a certain fraction of the images. Furthermore, we only keep the top concepts having the highest mutual information with the model’s prediction. These are important concepts which provide the most information about the distribution of model predictions. This filtering of weak concepts not only helps to have faster and more scalable pattern mining in the next step, but also leads to more concise patterns based on the most important concepts.

Finally, we have a new dataset of images, represented by their important concepts as feature attributes, and the CNN model predictions as the outcome. The set of concepts attributed to an image can serve as a local explanation for interpreting the prediction of the image. However, we go further in the next step by finding the patterns over all images that can provide insights into the general decision-making of the model.

4.3 Concept Pattern Mining

As mentioned in Section 3.3, we use an ensemble of rule mining methods to find patterns linking concepts to model predictions, namely *Classification and Regression Trees (CART)*, *Explanation Tables*, and *Interpretable Decision Sets (IDS)*. This helps to look at a varied set of informative patterns that cover different subspaces of data. Figure 4.1 shows three examples of such patterns.

Using CART, each tree node represents a concept, which is split based on the 0 or 1 value of the concept. We use entropy as the criteria to choose the order of concepts in the tree. Each root-to-leaf path in the tree represents a pattern of concepts. We control the size of the tree by setting the minimum samples per leaf. This is equivalent to the minimum support of each pattern, which is the number of data examples matching the concepts of a pattern.

Explanation Tables finds a potentially-overlapping set of patterns that together provide the most information gain about the distribution of the outcome attribute, which is the model’s predicted class in our case. The potential overlap and focus on information gain in the patterns found by Explanation Tables helps to explain the relation between the concepts and the CNN model predictions. This method first generates a set of candidate

patterns based on a random sample of data examples. Then iteratively finds the next pattern from the candidates that can increase the overall information gain the most. We modified this process by introducing a minimum support threshold that is used to filter out low-support candidate patterns.

IDS finds a set of rules by taking into account several optimization criteria such as support, confidence and conciseness, which are all important for interpretable and accurate explanation of CNNs based on concepts. IDS first generates a larger set of candidate rules, and then iteratively chooses rules which can maximize the optimization criteria. The minimum support of the candidate patterns and the weights of different optimization criteria can be changed as the parameters of the method. While IDS was designed to be an interpretable classifier similar to decision trees, we use each rule in the set independently for explanation.

Finally, we combine all the patterns found by these methods into a single set of patterns. We remove redundant patterns found by multiple methods. Also we order the patterns based on a score computed as $\frac{\text{support} \times \text{confidence}^2}{\text{size}^2}$, and we discard the patterns having a score lower than a threshold. Support is the fraction of images matching the concepts of a pattern, confidence is the fraction of supporting images which also match the CNN prediction stated in the pattern, and size is the number of concepts in a pattern. This ordering helps to only keep more concise and confident patterns which cover a larger set of images and can explain the CNN model predictions better.

4.4 Pattern Analysis Using the Web Interface

While the patterns themselves can reveal how different concepts are related to the predictions of the CNN model, visual analysis of the image examples related to each pattern provides further insights into the weaknesses and strengths of the model and potential data quality issues. For this purpose, we can analyze multiple interesting categories of data for each pattern, including the *matching*, *non-matching*, and *wrongly-predicted* images related to the pattern, which we explain below.

Matching images are those which match both the concepts and the prediction (i.e. the CNN predicted label) stated in the pattern. For example in a pattern like “If bed, then bedroom” found for a CNN which decides between bedrooms and kitchens, the matching images are those attributed to the concept ‘bed’ and predicted as bedrooms by the CNN. Looking at these images, with their high-importance areas highlighted, helps to verify the correct attribution of concepts to images based on the model behavior.

Non-matching images match the concepts of a pattern, but not the prediction. In the example pattern mentioned above, the non-matching images are those attributed to the concept ‘bed’ but predicted as kitchen. Non-matching images exist when the confidence of a pattern is less than 100%, which means that the model’s predictions were sometimes different from the prediction stated in the pattern. For instance, we may want to know whether there are uncommon images of kitchens including beds in the target dataset, or there are specific images of bedrooms which the model predicts incorrectly despite the existence of beds.

Wrongly-predicted images are those matching the concepts and the model’s prediction in the pattern, but having a different ground-truth label in the dataset. For example, wrongly-predicted images for the pattern mentioned earlier are those which are attributed to the concept ‘bed’ and were predicted as bedrooms, but are labelled in the target dataset as kitchens. If we define the *accuracy* of a pattern to be the fraction of images matching the concepts and prediction of a pattern that also have the same label as the prediction, wrongly-predicted images exist when the accuracy of a pattern is less than 100%. Looking at such examples can help us identify the characteristics of images that the model predicts incorrectly, as defined by their concepts. For instance, some of these images may include both the concepts usually found in bedrooms and kitchens. Including more similar examples in the training data of the model may be one way to improve the predictive performance of the model. Some wrongly-predicted images may even be mislabeled as kitchens while they actually show bedrooms, which is an example of how analyzing this data category can also help to identify data quality issues.

In order to visually analyze the mentioned data categories for each pattern, we have created a web-based tool for POEM. This tool serves as a proof of concept for visual analysis of the patterns to gain further insights about a CNN model. It currently allows selecting a few CNN models with their corresponding target datasets. In future, this tool can be extended to allow extracting and analyzing pattern-based explanations from any image classifier CNN model and target dataset of interest.

Figure 4.4 shows this web interface. It allows selecting the target dataset and the CNN model from a set of options using the *settings* panel in the left, as well as the pattern mining methods and their minimum support ratio parameters. It shows the patterns in a tabular format in the *patterns* panel on the top, which includes the concepts of each pattern, the related CNN model prediction, support, confidence, accuracy, score, and the method(s) used to compute the pattern. By default the patterns are ordered based on their computed score, but they can also be ordered based on their prediction, support, confidence, accuracy or method. For example in Figure 4.4, pattern 1 found by IDS is “If bed, then bedroom”, which states that 19% of the images (as shown by support) are

POEM: Pattern-Oriented Explanations of CNN Models

Dataset: Places (Bedroom, Kitchen, Livingroom)

CNN Model: Resnet-18

Pattern Mining Methods:

- Explanation Tables
- Interpretable Decision Sets (IDS)
- CART Decision Trees

Parameters of Pattern Mining Methods:

Exp. Tables min support:

IDS min support:

CART min support (min samples per leaf):

Compute Patterns


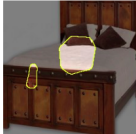
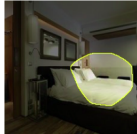

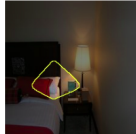
No.	bed	orange	plant	sofa	stove	tile	window	work surface	Prediction	Support	Confidence	Accuracy	Score	Method	Options
1	Yes								bedroom	0.19	0.99	0.99	0.19	IDS	Matching Non-Matching Wrongly-Predicted
2				Yes					livingroom	0.07	0.95	0.97	0.07	IDS	
3								Yes	kitchen	0.03	1	0.99	0.03	IDS	
4						Yes			kitchen	0.02	0.86	1	0.02	IDS	
5	Yes						Yes		bedroom	0.02	1	1	0.01	CART	
6					Yes				kitchen	0.01	1	1	0.01	Exp, IDS	

Choose a concept to see its activation images:

bed None

Show Activation Images

Images matching the concepts and prediction of pattern 1, with bed activations highlighted

Predicted bedroom, Labeled bedroom
Concept bed highlighted (filter 248)

Predicted bedroom, Labeled bedroom
Concept bed highlighted (filter 412)

Predicted bedroom, Labeled bedroom
Concept bed highlighted (filter 412)

Predicted bedroom, Labeled bedroom
Concept bed highlighted (filter 454)

Predicted bedroom, Labeled bedroom
Concept bed highlighted (filter 7)

Figure 4.4: POEM web interface showing the patterns for bedroom vs. kitchen vs. living room classes

associated with a ‘bed’ concept. Moreover, as indicated by a confidence of 99%, having the bed concept results in a prediction of a bedroom 99% of the time, and 99% of such examples are predicted correctly as bedrooms by the model, as shown by the accuracy.

It is also possible to select a pattern, and then choose one of the options to load a category of related images, including matching (green button), non-matching (yellow button), and wrongly-predicted images (red button). The related images are then displayed in the bottom *images* panel, with the ability to highlight the high-importance area related to a concept on the images. For example in Figure 4.4, the pattern “If bed, then bedroom” is selected at the top, and its matching images are displayed in the bottom, with their high-importance ‘bed’ concept areas highlighted.

4.5 Implementation

We implemented our pipeline mostly using Python and the PyTorch package. We developed the POEM web interface using the VueJS¹ framework as frontend and NodeJS² as backend.

We used and modified the Network Dissection code from the project’s Github page³. For semantic segmentation based on Unified Perceptual Parsing, we used the related Github code⁴.

We obtained the code of Explanation Tables from the authors. For IDS, we used the code from its Github page⁵, and for CART we used the implementation from the Scikit-Learn package⁶.

¹<https://vuejs.org>

²<https://nodejs.org>

³<https://github.com/davidbau/dissect>

⁴<https://github.com/CSAILVision/unifiedparsing>

⁵https://github.com/lvhimabindu/interpretable_decision_sets

⁶<https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html>

Chapter 5

Experiments

In this chapter, we conduct multiple experiments using two different CNN models and datasets to demonstrate the improvements of our pipeline compared with the previous related work.

5.1 Experiment Design

In our experiments, we use an approach similar to CNN2DT [18] to represent the best previous work for concept-based explanation of CNNs using the surrogate approach. Because we did not have access to the code for CNN2DT or any of the other main related work, we use the following choices made in CNN2DT in each step of the pipeline to implement the previous approach:

- For concept identification, we use the old Network Dissection method [4], which uses a secondary dataset to identify the concepts learned by the filters of a CNN model. We use the code for old Network Dissection from its Github repository¹.
- For concept attribution, we only check that a filter mapped to a concept includes a high-activation area when an image is passed through the model. We use the 99% percentile of the filter’s activation values over all images as the threshold for high activation, which is the same threshold used in Network Dissection.

¹<https://github.com/CSAILVision/NetDissect-Lite>

- For pattern mining, we only use the root-to-leaf paths from decision tree (CART), which is the common choice in most related work. We only keep the top 10 patterns based on their score.

As we explained in Chapter 4, our approach in POEM includes the following different choices in the pipeline steps:

- For concept identification, as we explained in Section 4.1, we use a more recent version of Network Dissection, which uses a semantic segmentation model and the target dataset itself for identifying the concepts learned by the CNN. The segmentation model is pretrained to identify a wide range of concepts in object, object part, material, and color categories.
- For concept attribution, we check all the three conditions mentioned in Section 4.2. We also filter out the weakest concepts attributed to less than 1% of the images, and then keep at most the top 10 concepts as measured by their mutual information with the model predictions.
- For pattern mining, we combine the patterns extracted by CART, Explanation Tables and IDS, as we explained in Section 4.3. We only keep the top 10 patterns as measured by the score.

We experiment with two different deep CNN models, namely *ResNet-18* [17] and *VGG-16* [31], which both have achieved state-of-the-art predictive performance in image classification tasks in a certain period during the last few years. We focus on the 512 filters of the last convolutional layer in each model. We analyze these models for the task of place classification, because place images usually consist of many interpretable concepts such as objects and object parts, which makes them a common choice for demonstrating concept-based explanation methods [4, 5, 9, 32]. In each of the use cases, we use a small subset of the classes in the *Places* dataset [39], which includes a total of 365 different place classes with 5000 images in each class. When using patterns for explanation, it makes sense to focus on a few target classes of interest to see how the model distinguishes between these classes based on their concepts.

For each use case, we use the CNN model pretrained on the entire *Places* dataset, but we modify its classification layer to match the number of target subset of classes. We then fine-tune the classification layer on the training images of the target classes, while freezing the rest of the network. We use 70% of the images in each class for training, and use the remaining data as the target dataset for explanation using POEM.

For all the three pattern mining methods, we use 0.03 as the minimum support threshold to allow more number of candidate patterns to be evaluated by each method. In Explanation Tables, we use the sample size of 16 as recommended by the authors. In IDS, we apply default uniform weights for the optimization criteria.

For each use case, we qualitatively analyze the concepts identified, the concepts attributed to images, and the patterns mined, in order to compare the previous approach with POEM.

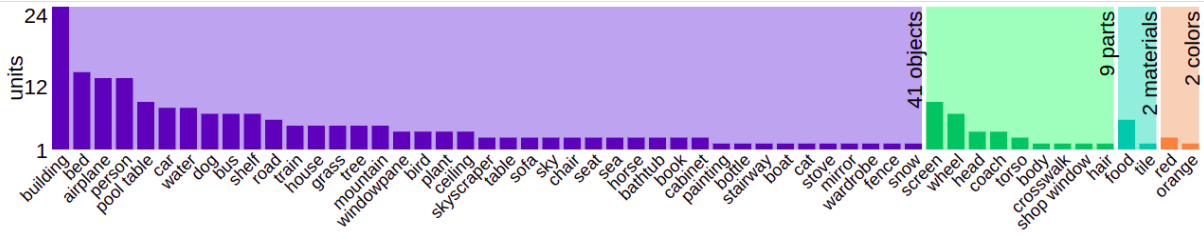
5.2 Use Case 1: ResNet for Classifying Bedrooms, Kitchens and Living Rooms

For this use case, we analyze the ResNet-18 CNN model for deciding between the bedroom, kitchen and living room classes of the Places dataset. The adapted model has a 92.4% prediction accuracy in distinguishing between these three target classes.

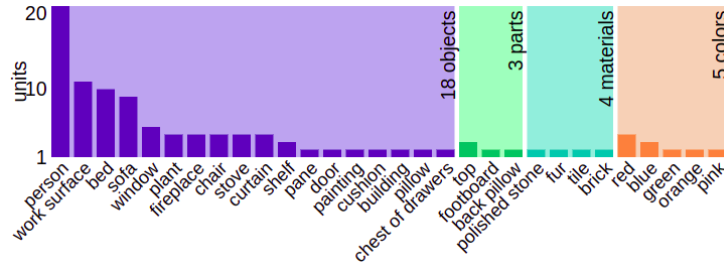
5.2.1 Concept Identification

Figure 5.1 shows the concepts identified using the previous approach (top) and POEM (bottom). The concepts from different categories are shown with different colors, and the vertical bars show the number of units (filters) mapped to each concept. If we look at the concepts identified using the previous approach, we see some concepts which are relevant to the target dataset, such as bed, bottle, cabinet, sofa and stove. However, we also find many identified concepts which are unlikely to appear in the images of bedrooms, kitchens or living rooms, such as airplane, bathtub, bus, car, horse and train. In Section 5.2.2, we will look at some example images attributed to these irrelevant concepts to see whether there is a valid reason for identifying them or not.

As we explained in Section 3.1, the problem is that all these concepts are identified by measuring the activations from a fixed secondary dataset of images, which may include concepts irrelevant to the target dataset and task of the CNN model under analysis. This is in contrast to the concepts identified by POEM, which are more relevant to the target place classes, because we have segmented and analyzed the target dataset related to the CNN’s task for concept identification, rather than any other dataset. Although the segmentation model we use is itself pretrained to identify a large but finite set of concepts, the fact that we use it to segment the target dataset images helps to identify concepts meaningful in the



(a) Concepts from the previous approach



(b) Concepts from POEM

Figure 5.1: Concepts identified using the previous approach (a) and POEM (b) for Use Case 1 in different categories, with number of units (filters) mapped to each concept displayed through the vertical bars

target dataset. This focus on the target dataset also leads to identifying fewer concepts overall, in comparison with the previous approach.

5.2.2 Concept Attribution

To compare the accuracy of concepts attributed to images using the previous approach and POEM, we can look at examples of different images, by highlighting the area on each image used to decide attributing a specific concept to the image. In the previous approach, these highlighted areas show the high-activation areas of the image for a specific filter mapped to the concept of interest. In POEM, they indicate the high-importance areas on the saliency map of a filter mapped to the concept, which we explained in Section 4.2. Figure 5.2 shows sample images associated with the ‘bus’ concept by the previous approach, with the corresponding ‘bus’ filter high-activation areas highlighted. As these examples show, nothing similar to bus exists in these images, and we do not have any evidence to claim that the CNN model sees something similar to a bus in these images.

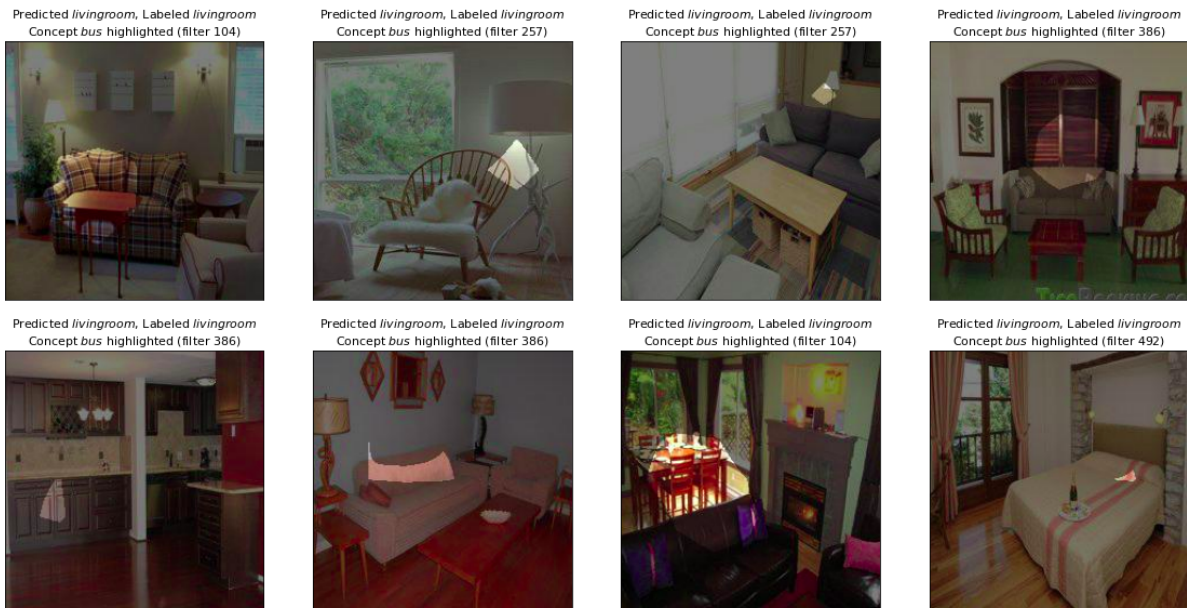


Figure 5.2: Sample images with ‘bus’ concept high-activation areas highlighted, based on the previous approach

We can also look at the high-activation areas related to more relevant concepts identified using the previous approach. Figure 5.3 shows sample images with their ‘chair’ concept high-activation areas highlighted. In some of these images, either the chair or something similar to that does not exist, or the high-activation areas do not match the location of the chair in the image. Such examples imply that the concepts attributed to images using the previous approach may not always be reliable indicators of what the CNN model pays attention to. As a result, some of the extracted patterns cannot serve as accurate explanations for the CNN model.

If we look at the high-importance areas used by POEM for concept attribution, we can clearly see that the concepts attributed to images are more accurate. For example, Figure 5.4 shows sample images attributed to the concept ‘sofa’, with the related high-importance areas highlighted. We can see that all the highlighted areas cover the actual sofa in the images either partially or completely. Furthermore, because these highlighted areas are based on the activation-gradient maps extracted from the CNN model, we are more confident that these areas have played a role toward the model’s prediction for each image.

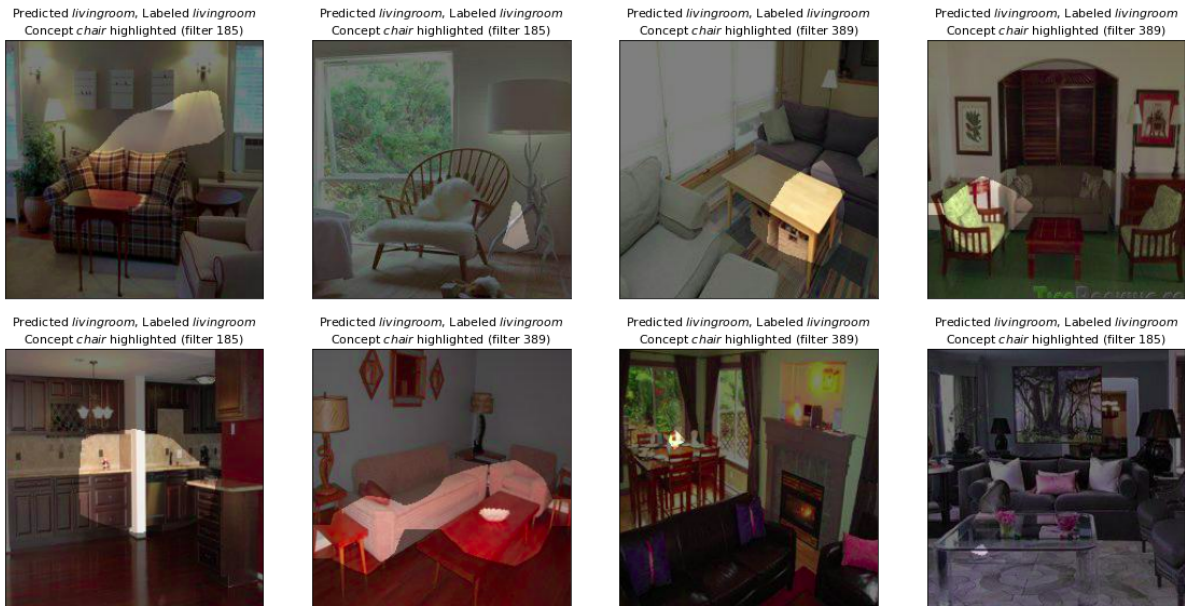


Figure 5.3: Sample images with ‘chair’ high-activation areas highlighted, based on the previous approach

5.2.3 Concept Pattern Mining and Analysis

Table 5.1 shows the top 10 patterns mined using the previous approach (top) and POEM (bottom). In the header of each table, the concept titles are shown in lowercase letters, and the properties of the patterns are displayed next in capital case. For each row which corresponds to a pattern, the included concepts are shown with “yes” or “no” values, which shows whether a specific concept is attributed to the images supporting the pattern or not. Furthermore, the CNN predicted label associated with each pattern is shown, as well as the pattern’s support, confidence, accuracy, score, and the method(s) used to find it.

When looking at the patterns from the previous approach, we can find patterns which match our common sense about characteristics of bedrooms, kitchens and living rooms. For example, pattern 2 in Table 5.1a says: if the concepts ‘bottle’, ‘cabinet’ and ‘chair’ are attributed to an image, then the CNN classifies it as kitchen in 98% of cases. We also see some less reasonable patterns with lower confidence levels, which include some of the irrelevant concepts mentioned earlier. For example, patterns 7 and 9 in Table 5.1a consider the ‘bus’ concept as one of the important concepts linked with prediction of living room and bedroom, respectively. As we looked at sample images associated with the ‘bus’



Figure 5.4: Sample images with ‘sofa’ high-importance areas highlighted, based on POEM

concept in Figure 5.2, such patterns are created based on inaccurate concept attributions.

Another observation about the patterns from the previous approach is that they all include two or more concepts, which leads to relatively narrow and less concise patterns. For example, there is no single-concept pattern, especially including concepts that are the distinguishing characteristics of certain classes, such as bed, cabinet, stove or sofa. It may seem that these concepts have not been that important for the predictions of the model, or their related patterns have not been significant enough to be chosen by the decision tree. However, if we look at the patterns from POEM in Table 5.1b, we can see multiple high-confidence single-concept patterns such as patterns 1, 2, 4 and 7. These patterns highlight the importance of the concept ‘bed’ for predicting bedrooms, ‘sofa’ for living rooms, and ‘work surface’ and ‘tile’ for kitchens, which can indicate that the CNN model is mostly looking at the right concepts in its predictions.

In general, it seems the patterns found by POEM are more sparse and concise, and thus more interpretable than the previous approach. Part of this may be related to the more accurate concept identification and attribution we are applying in our pipeline. Another reason for this observation can be related to the structure of a decision tree as explained in Section 3.3, which generally tends to produce narrow non-overlapping patterns with higher

	bathtub	bed	bottle	bus	cabinet	car	chair	screen	sofa	stove	train	PREDICTION	SUPPORT	CONFIDENCE	ACCURACY	SCORE	METHOD
1			yes		no							kitchen	0.06	0.68	0.97	0.01	CART
2			yes		yes		yes					kitchen	0.06	0.98	0.99	0.01	CART
3		no	yes		yes		no					kitchen	0.06	0.99	0.99	0.00	CART
4			no		yes				no	yes		kitchen	0.05	0.91	0.99	0.00	CART
5		yes	yes		yes		no					kitchen	0.03	0.96	1.00	0.00	CART
6			no		yes				no	no		kitchen	0.06	0.60	0.96	0.00	CART
7			no	yes			yes		yes			livingroom	0.05	0.64	0.91	0.00	CART
8			no	no		no	yes		yes			livingroom	0.04	0.89	0.95	0.00	CART
9			no	yes			no	yes	yes		no	bedroom	0.06	0.72	0.95	0.00	CART
10		no	no		no				no			livingroom	0.06	0.47	0.84	0.00	CART

(a) Patterns from previous approach

	bed	curtain	green	orange	person	red	sofa	tile	window	work	surface	PREDICTION	SUPPORT	CONFIDENCE	ACCURACY	SCORE	METHOD
1	yes											bedroom	0.22	0.99	0.99	0.22	Exp, IDS, CART
2							yes					livingroom	0.05	0.89	0.96	0.04	IDS
3	yes		no	no								bedroom	0.22	0.99	0.99	0.02	Exp
4											yes	kitchen	0.03	0.99	1.00	0.02	IDS
5	no		no								no	livingroom	0.75	0.44	0.89	0.02	Exp
6	no						yes					livingroom	0.05	0.90	0.96	0.01	CART
7								yes				kitchen	0.03	0.96	0.96	0.01	IDS
8	no				no		no		no			kitchen	0.64	0.48	0.97	0.01	CART
9	no		no			no	no	no				kitchen	0.70	0.46	0.97	0.01	Exp
10	yes	no		no		no	no		no			bedroom	0.20	0.99	0.99	0.01	Exp

(b) Patterns from POEM

Table 5.1: Top patterns for Use Case 1 extracted using the previous approach (a) and POEM (b)

rule size. This is why patterns based on decision trees may not be adequate for exploring different significant subspaces of data.

Using POEM, we can also look at other data categories, such as non-matching and wrongly-predicted images related to each pattern, to obtain further insights about the CNN model and the related properties of target data. Figure 5.5 shows sample images matching pattern 2 in Table 5.1b, but predicted incorrectly by the model as living room instead of bedroom or kitchen. Some of these examples which are labeled as bedroom, especially those in the top row, seem to include either a sofa in the bedroom, or a bed similar to a sofa. Such examples can somehow reveal the characteristics of those bedroom images which are predicted incorrectly by the model. Including more of these examples in

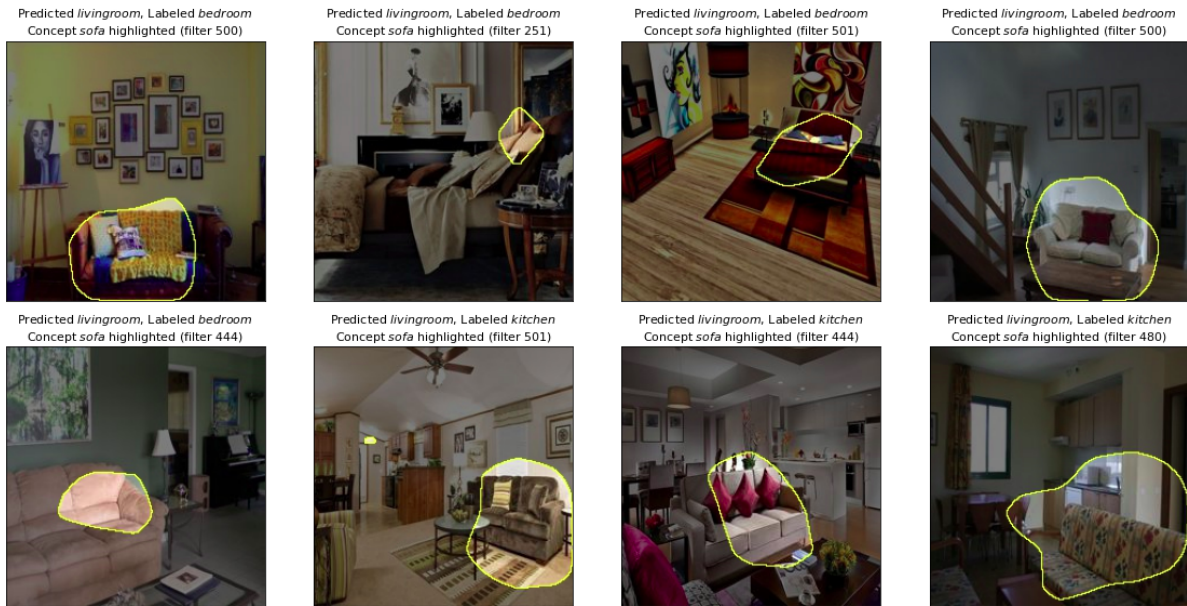


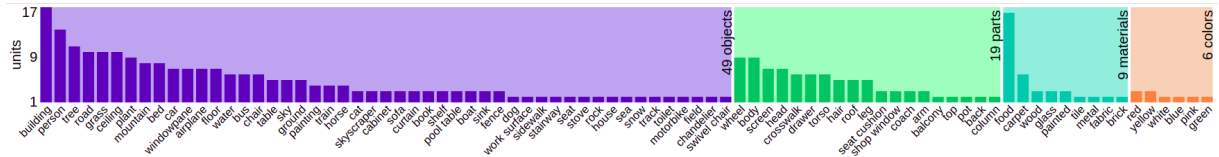
Figure 5.5: Sample images matching pattern 2 in Table 5.1b, but wrongly-predicted by the model

the training data of the model may be one way to improve the model’s prediction accuracy for such complex examples.

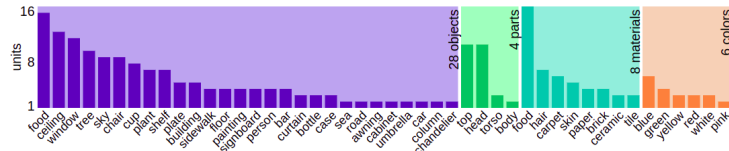
If we look at those examples in the bottom row of Figure 5.5 which are labeled as kitchen, we see they are complex examples of images including the view of both the living room and the kitchen. We can even claim that some of these examples may be mislabeled as kitchen, because the dominant and closer view shows the living room, as detected correctly by the CNN model. By analyzing such examples, we can identify both the strengths of the model and the potential data quality issues.

5.3 Use Case 2: VGG for Classifying Coffee Shops and Restaurants

In the second use case, we analyze the VGG-16 CNN model, this time using the coffee shop and restaurant classes of the Places dataset, which are outdoor places with more complex details than the previous use case. The prediction accuracy of the model on the target



(a) Concepts from the previous approach



(b) Concepts from POEM

Figure 5.6: Concepts identified using the previous approach (a) and POEM (b) for Use Case 2 in different categories

classes is 88.1%.

5.3.1 Concept Identification

Figure 5.6 shows the concepts identified using the previous approach (top) and POEM (bottom). Similar to Use Case 1, we can see many concepts which are unlikely to appear in the images of coffee shops and restaurants, but are present in the secondary dataset used to identify the concepts, such as airplane, bed, car, motorbike, mountain, toilet, etc. For example, Figure 5.7 shows the high-activation areas for some images attributed to the ‘toilet’ concept. None of them seem to include anything close to a toilet. Some of these images are showing a coffee cup, which in some sense may be partly similar to the body of a toilet and the water inside it. In fact, the same filters which are activated on cups may have also activated when images of toilets from the secondary dataset were passed through them. However, we do not see a consistent pattern about such observations in the related image examples, and we do not have enough evidence to make such claims. More relevant concepts which are related to the target task of the CNN model (e.g. ‘cup’ concept in this case) can provide more consistent and reliable evidence about the behavior of the model.

If we examine the concepts identified using POEM in Figure 5.6, we mostly see concepts which are more consistent with our perception of coffee shops and restaurants, such as bottle, chair, cup, food, plate, shelf, etc. Furthermore, fewer concepts are identified compared with the previous approach, similar to Use Case 1. We can also find some con-

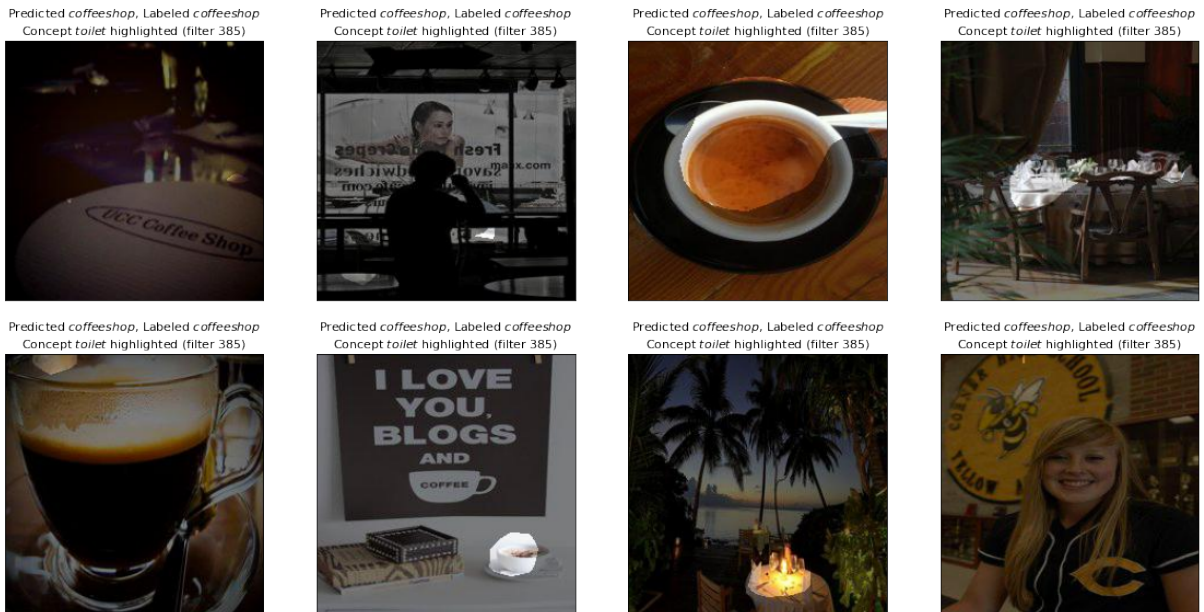


Figure 5.7: Sample images with ‘toilet’ concept high-activation areas highlighted, based on the previous approach

cepts which may not seem relevant at first sight, such as sea, sky, and tree. To check this further, in Figure 5.8 we can see sample images from the target dataset with the ‘sea’ concept highlighted. All of these images seem to show outdoor coffee shop or restaurant views where the sea is part of the landscape and has led to activations of specific filters in the CNN model.

5.3.2 Concept Attribution

Figure 5.9 shows sample images associated with the ‘swivel chair’ concept by the previous approach. While some of the examples seem to correctly highlight swivel chairs, some other examples either do not include any concept similar to a chair, or the high-activation areas do not cover any part of the chair.

On the other hand, as the experiments for Use Case 1 also demonstrated, concepts attributed to images using POEM and the related high-importance areas highly correspond to the locations of related concepts in the images. Figure 5.10 shows this correspondence for sample images attributed to the ‘red’ color using POEM.

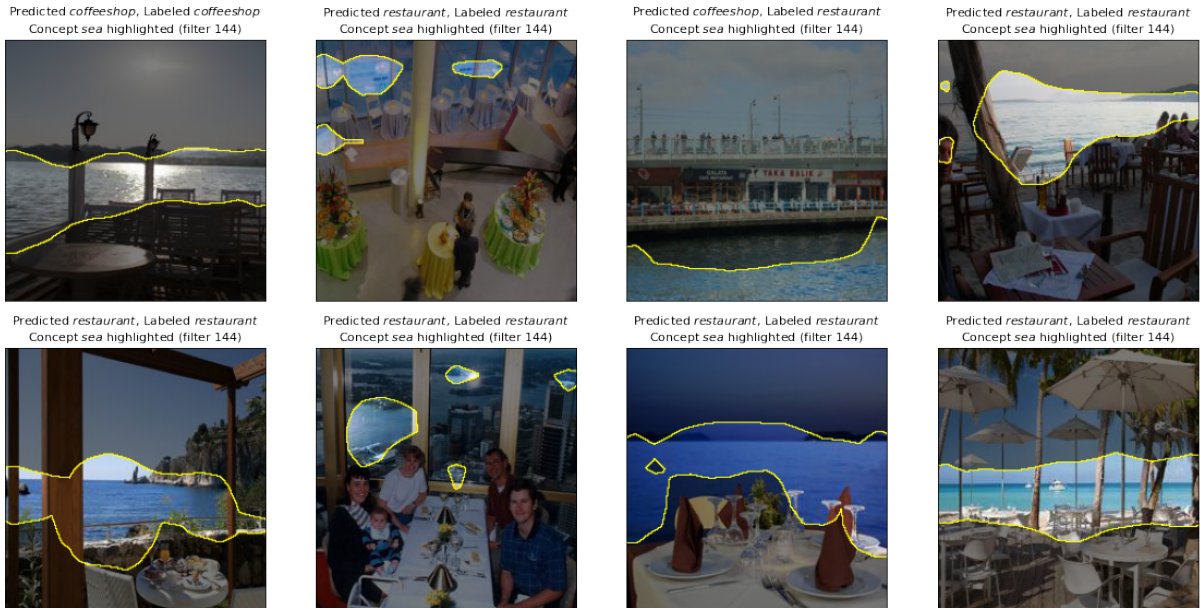


Figure 5.8: Sample images with ‘sea’ high-importance areas highlighted, based on POEM

5.3.3 Concept Pattern Mining and Analysis

Table 5.2 shows the top 10 patterns mined using the previous approach (top) and POEM (bottom). These results confirm our observations in Use Case 1 that patterns extracted using the previous approach are not much concise and mostly include more than a few related concepts. Furthermore, the issues with concept identification and attribution have led to patterns which may not all explain the CNN model predictions accurately.

In contrast, patterns found by POEM generally include fewer concepts, and highlight the importance of concepts such as ‘shelf’ for detecting coffee shops and ‘top’ (i.e. table top) for detecting restaurants. Patterns 1 and 5 in Table 5.2b also link ‘ceiling’ and ‘red’ concepts with restaurant predictions, but with a lower confidence. Clearly, most of the mentioned concepts are not intrinsic characteristics of the coffee shops or restaurants, but probably indicate that the CNN has seen shelves more in coffee shop training images and table tops, ceilings and red colors in restaurant images. Such observations can help us identify the potential vulnerabilities of the CNN model.

In order to analyze some of the POEM patterns further, we can examine sample images from the related data categories. For example, we may want to look at non-matching images of pattern 4 in Table 5.2b, which are those attributed to ‘shelf’, but predicted by

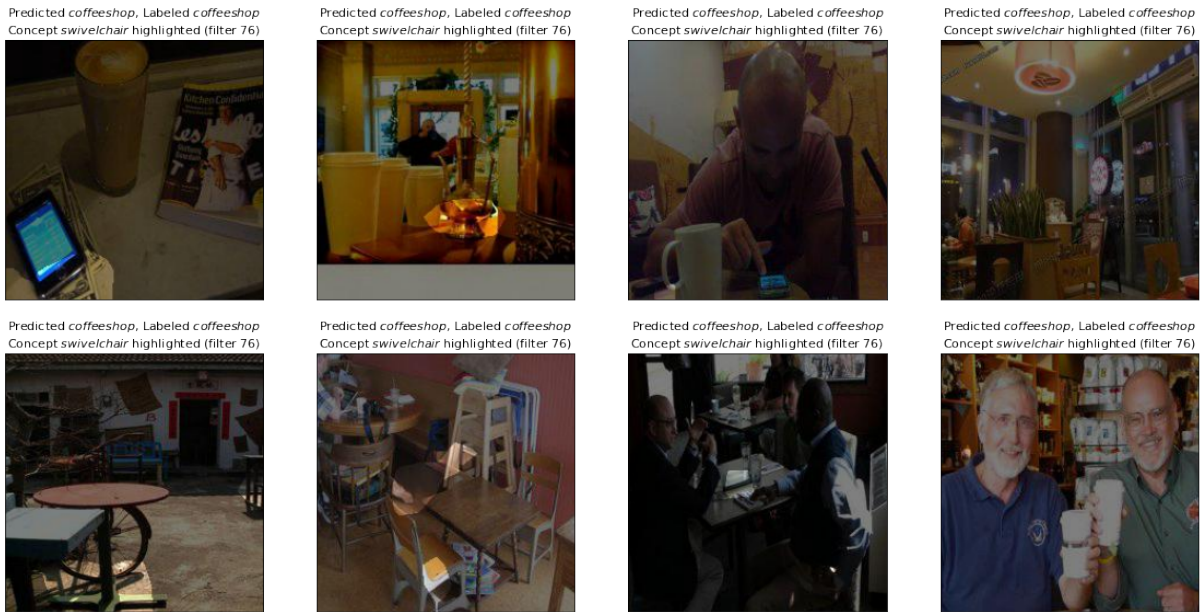


Figure 5.9: Sample images with ‘swivel chair’ high-activation areas highlighted, based on the previous approach

the model as restaurant rather than coffee shop. Figure 5.11 shows such images. We can see an example image (leftmost) which is predicted incorrectly by the model as restaurant. However, the other three examples are more tricky, and they may even be mislabeled as restaurant. By looking at such examples, we find that distinguishing between coffee shop and restaurant images is complex not only for the CNN model, but also for humans. This is especially true when we look at incomplete views of these places. It may be because of the fact that we cannot easily identify the main concepts that define these two types of places. This lack of distinguishing characteristics also affects the predictive performance of the CNN model when applied to complex examples.

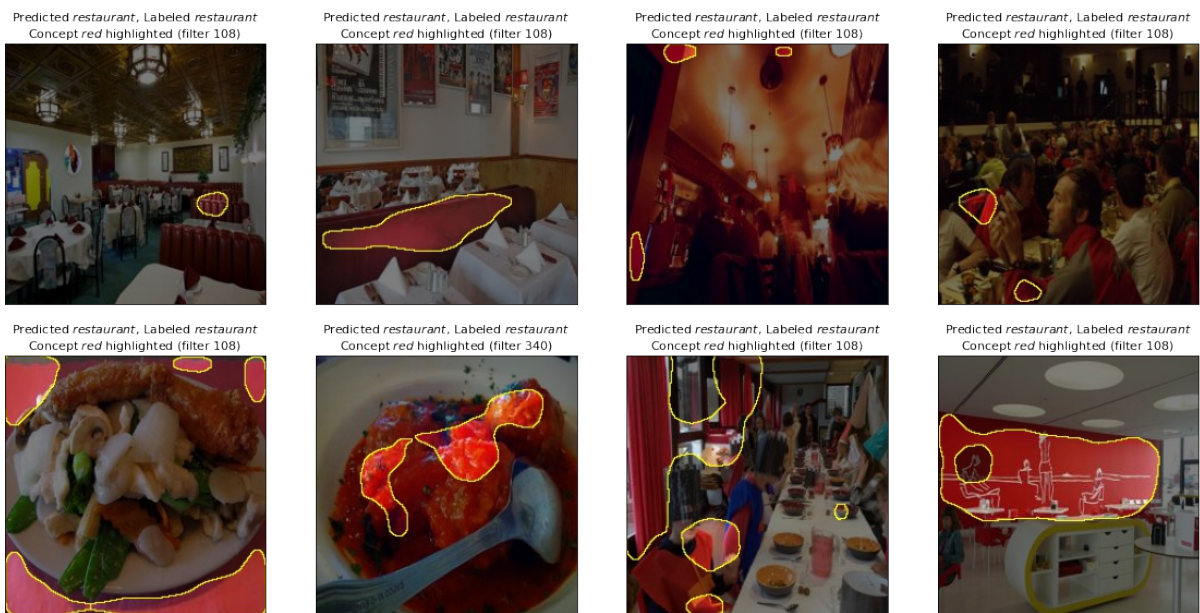


Figure 5.10: Sample images with ‘red’ high-importance areas highlighted, based on POEM



Figure 5.11: Sample non-matching images of pattern 4 in Table 5.2b

	balcony	blue	book	chandelier	coach	green	motorbike	painting	pool table	seat	shelf	swivel chair	toilet	work surface	PREDICTION	SUPPORT	CONFIDENCE	ACCURACY	SCORE	METHOD	
1									yes	no				no	yes	coffeeshop	0.04	0.92	0.91	0.0	CART
2	no			no		no				yes						coffeeshop	0.06	0.70	0.78	0.0	CART
3			yes	yes				yes		yes	no					restaurant	0.06	0.84	0.95	0.0	CART
4									no	no				no		coffeeshop	0.04	0.60	0.86	0.0	CART
5		yes							yes	no				no	no	coffeeshop	0.04	0.87	0.92	0.0	CART
6				no	no	yes					yes					restaurant	0.04	0.69	0.90	0.0	CART
7		yes					yes			no		yes	yes			coffeeshop	0.04	0.85	0.88	0.0	CART
8		no		no		no				no				yes		coffeeshop	0.04	0.80	0.94	0.0	CART
9			no	yes				yes	yes	no						restaurant	0.04	0.73	0.92	0.0	CART
10				yes				no	yes	no						restaurant	0.03	0.64	0.98	0.0	CART

(a) Patterns from previous approach

	building	ceiling	head	red	shelf	top	PREDICTION	SUPPORT	CONFIDENCE	ACCURACY	SCORE	METHOD
1			yes				restaurant	0.12	0.73	0.92	0.06	IDS
2						yes	restaurant	0.07	0.90	0.94	0.06	IDS, CART
3			no	no	no		coffeeshop	0.80	0.60	0.85	0.03	IDS
4					yes		coffeeshop	0.03	0.96	0.94	0.03	Exp
5				yes			restaurant	0.03	0.78	0.90	0.02	Exp, IDS
6				no	yes		restaurant	0.07	0.90	0.94	0.01	Exp
7		no				yes	restaurant	0.07	0.90	0.94	0.01	Exp
8		no	no	no		no	coffeeshop	0.69	0.56	0.86	0.01	CART
9			yes			no	restaurant	0.10	0.70	0.92	0.01	CART
10		no			no	yes	restaurant	0.07	0.91	0.95	0.01	Exp

(b) Patterns from POEM

Table 5.2: Top patterns for Use Case 2 extracted using the previous approach (a) and POEM (b)

Chapter 6

Conclusion and Future Work

In this work, we introduced POEM as a framework for explaining image classification CNN models using patterns of semantic concepts. POEM improves over the previous rule-based approaches for explaining CNNs in multiple ways. In POEM, we use the latest version of Network Dissection for identifying the main concepts learned by the last layer filters of a CNN model, which helps to identify concepts which are relevant to the target dataset and task of the CNN model under analysis. For accurate attribution of the identified concepts to each of the images in the target dataset, we use a semantic segmentation model to check that a concept is present in an image. Furthermore, we ensure that the CNN model pays attention to the concept in the image by examining the corresponding filter activations and gradients when the image passes through the model. Finally, we use an ensemble of rule mining methods for finding varied, informative and concise patterns which explain how the concepts in images are related to the CNN model predictions. We also created a web interface for POEM which enables interactive visual analysis of the patterns and their related image examples. This type of analysis helps us to extract further insights about the strengths and weaknesses of the CNN model and potential data quality issues.

POEM has its own limitations, which can also indicate directions for future work. The Network Dissection method we use for concept identification assumes that each CNN filter focuses at most on a single main concept rather than multiple concepts. Several related work have questioned this assumption [11, 38, 9], explaining that some filters may be learning a mix of multiple concepts, or may jointly learn a single concept together. Research on identifying the concepts learned by CNNs will continue to progress, and more effective methods introduced in future may be adapted to be used in place of Network Dissection in the framework we proposed.

For concept attribution, we mainly use a gradient-based approach similar to Grad-CAM, which uses filter activations and gradients as indicators of what the CNN model is paying attention to in its predictions. While we can use these indicators to explain the relation between concepts and model predictions, we cannot claim that the concepts attributed to images in this way are actually the main causes for the predictions of the CNN. A better way to perform such a causal analysis is attributing concepts to images based on the counterfactual approach, which means identifying those concepts whose removal from an image changes the model prediction. We aim to use this approach in a future work in the concept attribution step of our framework.

POEM provides global explanations about the CNN model mainly through the patterns of concepts, which are extracted from the target dataset of images related to the CNN model. In order for the patterns to serve as generalized explanations for the CNN model, the target dataset requires to be large and representative enough. This is another limitation of such rule-based explanations based on the surrogate approach. Moreover, while we qualitatively compared the patterns found by POEM with the related work, it can be helpful to conduct a user study to evaluate the accuracy and interpretability of the explanations provided by POEM in real-world applications and scenarios.

Finally, we think the general framework we introduced has the potential to be adapted for explaining other types of deep neural architectures such as *Recurrent Neural Networks (RNNs)* and *Transformers*. The *attention* mechanism used in Transformer models already allows measuring the importance of words in each input sentence given to the model [3, 33]. However, generalizing such local explanations to global ones may require identifying concepts more general than just single words. This is another direction for future research that can be explored.

References

- [1] Arjun R. Akula, Keze Wang, Changsong Liu, Sari Saba-Sadiya, Hongjing Lu, Sinisa Todorovic, Joyce Chai, and Song-Chun Zhu. Cx-tom: Counterfactual explanations with theory-of-mind for enhancing human trust in image recognition models. *iScience*, 25(1):103581, 2022.
- [2] Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes, 2016.
- [3] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate, 2014.
- [4] D. Bau, B. Zhou, A. Khosla, A. Oliva, and A. Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *Computer Vision and Pattern Recognition*, 2017.
- [5] D. Bau, J.-Y. Zhu, H. Strobelt, A. Lapedriza, B. Zhou, and A. Torralba. Understanding the role of individual units in a deep neural network. *Proceedings of the National Academy of Sciences*, 2020.
- [6] Guido Bologna and Silvio Fossati. A two-step rule-extraction technique for a cnn. *Electronics*, 9(6), 2020.
- [7] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Wadsworth and Brooks, Monterey, CA, 1984.
- [8] Sophie Burkhardt, Jannis Brugger, Nicolas Wagner, Zahra Ahmadi, Kristian Kersting, and Stefan Kramer. Rule extraction from binary neural networks with convolutional rules for model validation. *Frontiers in Artificial Intelligence*, 4, 2021.
- [9] Z. Chen, Y. Bei, and C. Rudin. Concept whitening for interpretable image recognition. *Nature Machine Intelligence*, 2:1–11, 12 2020.

- [10] R. El Shawi, Y. Sherif, and S. Sakr. Towards automated concept-based decision tree explanations for cnns. In *EDBT 2021 24th International Conference on Extending Database Technology*, 04 2021.
- [11] R. Fong and A. Vedaldi. Net2vec: Quantifying and explaining how concepts are encoded by filters in deep neural networks. *arXiv preprint arXiv:1801.03454*, 2018.
- [12] N. Frosst and G. E. Hinton. Distilling a neural network into a soft decision tree. *ArXiv*, abs/1711.09784, 2017.
- [13] K. El Gebaly, G. Feng, L. Golab, F. Korn, and D. Srivastava. Explanation tables. *IEEE Data Engineering Bulletin*, 41:43–51, 2018.
- [14] A. Ghorbani, J. Wexler, J. Zou, and B. Kim. Towards automatic concept-based explanations. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.
- [15] Yash Goyal, Ziyang Wu, Jan Ernst, Dhruv Batra, Devi Parikh, and Stefan Lee. Counterfactual visual explanations. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2376–2384. PMLR, 09–15 Jun 2019.
- [16] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Gian-notti, and Dino Pedreschi. A survey of methods for explaining black box models. 51(5), aug 2018.
- [17] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition, 2015.
- [18] S. Jia, P. Lin, Z. Li, J. Zhang, and S. Liu. Visualizing surrogate decision trees of convolutional neural networks. *J. Vis.*, 23(1):141–156, feb 2020.
- [19] B. Kim, M. Wattenberg, J. Gilmer, C. Cai, J. Wexler, F. Viégas, and R. Sayres. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *ICML*, 2018.
- [20] Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 5338–5348. PMLR, 13–18 Jul 2020.

- [21] H. Lakkaraju, S. Bach, and J. Leskovec. Interpretable decision sets: A joint framework for description and prediction. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 1675–1684, New York, NY, USA, 2016. Association for Computing Machinery.
- [22] Fei-Fei Li, Justin Johnson, and Serena Yeung. Training neural networks. http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture6.pdf last accessed on 22/07/2022.
- [23] A. Mahendran and A. Vedaldi. Visualizing deep convolutional neural networks using natural pre-images. *International Journal of Computer Vision*, 120, 12 2016.
- [24] Aravindh Mahendran and Andrea Vedaldi. Understanding deep image representations by inverting them. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5188–5196, 2015.
- [25] Yao Ming, Huamin Qu, and Enrico Bertini. Rulematrix: Visualizing and understanding classifiers with rules. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):342–352, 2019.
- [26] Anh Nguyen, Alexey Dosovitskiy, Jason Yosinski, Thomas Brox, and Jeff Clune. Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.
- [27] Anh Nguyen, Jason Yosinski, and Jeff Clune. Multifaceted feature visualization: Uncovering the different types of features learned by each neuron in deep neural networks, 2016.
- [28] Zakaria Rguibi, Abdelmajid Hajami, Dya Zitouni, Amine Elqaraoui, and Anas Bedraoui. Cxai: Explaining convolutional neural networks for medical imaging diagnostic. *Electronics*, 11(11), 2022.
- [29] R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 618–626, 2017.
- [30] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps, 2013.

- [31] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.
- [32] J. Townsend, T. Kasioumis, and H. Inakoshi. Eric: Extracting relations inferred from convolutions. In *Computer Vision – ACCV 2020*, pages 206–222. Springer International Publishing, 2021.
- [33] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.
- [34] Sandra Wachter, Brent Mittelstadt, and Chris Russell. Counterfactual explanations without opening the black box: Automated decisions and the gdpr. 2017.
- [35] T. Xiao, Y. Liu, B. Zhou, Y. Jiang, and J. Sun. Unified perceptual parsing for scene understanding. In *European Conference on Computer Vision*. Springer, 2018.
- [36] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *Computer Vision – ECCV 2014*, pages 818–833, Cham, 2014. Springer International Publishing.
- [37] Q. Zhang, Y. Yang, H. Ma, and Y. N. Wu. Interpreting cnns via decision trees, 2018.
- [38] Quanshi Zhang, Ying Nian Wu, and Song-Chun Zhu. Interpretable convolutional neural networks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8827–8836, 2018.
- [39] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.
- [40] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization, 2015.