Concept-based Explanations for Convolutional Neural Network Predictions

A Thesis Submitted

in Partial Fulfilment of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

by

Vidhya Kamakshi V

(2017CSZ0005)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY ROPAR

October, 2023

Vidhya Kamakshi V: *Concept-based Explanations for Convolutional Neural Network Predictions*

Dedicated to all my dear ones.

Declaration of Originality

I hereby declare that the work which is being presented in the thesis entitled **Concept-based Explanations for Convolutional Neural Network Predictions** has been solely authored by me. It presents the result of my own independent investigation/research conducted during the time period from August 2017 to April 2023 under the supervision of Dr. Narayanan C Krishnan, Associate Professor and Head, Department of Data Science, Indian Institute of Technology Palakkad. To the best of my knowledge, it is an original work, both in terms of research content and narrative, and has not been submitted or accepted elsewhere, in part or in full, for the award of any degree, diploma, fellowship, associateship, or similar title of any university or institution. Further, due credit has been attributed to the relevant state-of-the-art and collaborations (if any) with appropriate citations and acknowledgments, in line with established ethical norms and practices. I also declare that any idea/data/fact/source stated in my thesis has not been fabricated/ falsified/ misrepresented. All the principles of academic honesty and integrity have been followed. I fully understand that if the thesis is found to be unoriginal, fabricated, or plagiarized, the Institute reserves the right to withdraw the thesis from its archive and revoke the associated Degree conferred. Additionally, the Institute also reserves the right to appraise all concerned sections of society of the matter for their information and necessary action (if any). If accepted, I hereby consent for my thesis to be available online in the Institute's Open Access repository, inter-library loan, and the title & abstract to be made available to outside organizations.

Signature

Name: Vidhya Kamakshi V
Entry Number: 2017CSZ0005
Program: PhD
Department: Computer Science and Engineering
Indian Institute of Technology Ropar
Rupnagar, Punjab 140001

Date: 21 October 2023

Acknowledgement

I would first like to thank my supervisor, Dr. Narayanan C Krishnan. He always helped me whenever I had any problems. I am grateful to him for his continuous support during my Ph.D. study and related research and for his patience, motivation, and guidance. I could not have imagined having a better advisor. I owe him way more than my words could ever express. Besides my advisor, I am also grateful to my doctoral committee members: Dr. Shashi Shekhar Jha, Dr. Viswanath Gunturi, Dr. Manju Khan, and Dr. Apurva Mudgal, for sparing time from their busy schedules to attend all my presentations. I am indebted to everyone at IIT Ropar for providing me with all the necessary facilities for this research. My heartfelt gratitude to all faculty who have inspired me with their insightful research talks and excellent teaching aptitude. My heartfelt gratitude goes to everyone at IIT Palakkad who facilitated my in-person research with my supervisor Dr. Narayanan C Krishnan, when he moved to IIT Palakkad. The resources provided by the 'PARAM Shivay Facility' under the National Supercomputing Mission, Government of India at the Indian Institute of Technology, Varanasi, and under the Google TensorFlow Research award are gratefully acknowledged. Very special gratitude goes out to Dr. Sanatan Sukhija, Akanksha Paul, Aroof Aimen, Shivam Gupta, Abhishek Singh Sambyal, Nikhil Reddy, Sam Zabdiel, Namrata Lodhi, Prateek Munjal, Uday Gupta, Rajat Sharma, Ashish Kumar, Karan Sehgal, Prerna Garg, Pratham Gupta, Sahil Sidheekh and Vasanthan. My heartfelt gratitude to all seniors and guest researchers whose inspiring presentations increased my zeal to pursue the search for answers to my unknown research questions. I also place on record my sense of gratitude to one and all who, directly or indirectly, have assisted me. Last but not the least, I must express my very profound gratitude to Lord Almighty and my family for providing me with unfailing support and continuous encouragement throughout my years of study. This accomplishment would not have been possible without all of you. Thank you.

Vidhya Kamakshi V
Indian Institute of Technology Ropar.

Certificate

This is to certify that the thesis entitled **Concept-based Explanations for Convolutional Neural Network Predictions**, submitted by **Vidhya Kamakshi V (2017CSZ0005)** for the award of the degree of **Doctor of Philosophy** of Indian Institute of Technology Ropar, is a record of bonafide research work carried out under our guidance and supervision. To the best of our knowledge and belief, the work presented in this thesis is original and has not been submitted, either in part or full, for the award of any other degree, diploma, fellowship, associateship or similar title of any university or institution. In our opinion, the thesis has reached the standard fulfilling the requirements of the regulations relating to the Degree.

Signature of the Supervisor
Name: Dr. Narayanan C Krishnan
Department: Data Science
Indian Institute of Technology Palakkad
Palakkad, Kerala 678557
21/10/2023 Date:
(On lien from IIT Ropar)

Lay Summary

Image classification is the task in which a machine learning model predicts the class/category of an object contained in a given image from the set of known classes. Convolutional Neural Networks(CNN) have achieved state-of-the-art image classification results. However, how these models arrive at predictions for a given image is unclear. The research field Explainable AI (XAI), aims to unravel the working mechanism used by these accurate, opaque black boxes. If the explanations are closer to how humans interpret images, they help better understand the working mechanism of CNNs. This fact was proved by previous experiments as reviewed from existing XAI literature. Studies show that humans process images in terms of sub-regions called concepts. For instance, a peacock is identified by its characteristic concepts like green feathers, blue neck, etc. This thesis aims to automatically extract such concepts learned by CNN from the data.

Three novel frameworks are proposed to provide automatically extracted concept-based explanations for standard image classifiers. The first framework, PACE, automatically extracts class-specific concepts relevant to the prediction. While class-specific concepts unravel the blueprints of a class from CNN's perspective, concepts are often shared across classes; for instance, gorillas and chimpanzees naturally share many characteristics as they belong to the same family. The second framework, SCE, unravels the concept sharedness across related classes from CNNs perspective. The relevance of the extracted concepts towards prediction and the primitive image aspects, like color, texture, and shape encoded by the concept, are estimated after training the explainer.

The thesis identifies a void in XAI's panorama that much attention is given to classifiers trained and tested using the same data. However, allied paradigms have been shown to add to state-of-the-art successes. Despite the data hunger of deep models, domain adaptation techniques have been employed to leverage a huge amount of related data to help learn a classifier that is expected to work on scarce data of interest. The third framework XSDA-Net, builds a supervised domain-adapted classifier that can explain itself in terms of concepts extracted from the different datasets the classifier is exposed to.

Experiments demonstrate the utility of all three proposed frameworks in automatically extracting concepts from the data such that they unravel the working mechanism of the image classifiers. The thesis reviews the different types of explanations prevalent in the XAI field and enlightens the possible future research avenues for potential researchers looking to venture into XAI.

Abstract

Convolutional Neural Networks(CNN) have achieved state-of-the-art image classification results. The research sub-field, Explainable AI (XAI), aims to unravel the working mechanism used by these accurate, opaque black boxes to enhance users' trust, and detect spurious correlations, thereby enabling the pervasive adoption of AI systems. Studies show that humans process images in terms of sub-regions called concepts. For instance, a peacock is identified by its green feathers, blue neck, etc. So explanations in terms of such concepts are proven to be helpful for humans to understand the working of CNN better. Existing approaches leverage an external repository of concept examples to extract the concept representations learned by the CNNs. However, distributional differences that may exist between the external repository and the data on which the CNN is trained, the faithfulness of these explanations, i.e., if the extracted representations truly represent the learned representations, is not guaranteed. To circumvent this challenge, the thesis proposes three novel frameworks that automatically extract the concepts from the data.

The first framework, PACE, automatically extracts class-specific concepts relevant to the black-box prediction. It tightly integrates the faithfulness of the explanatory framework into the black-box model. It generates explanations for two different CNN architectures trained for classifying the AWA2 and Imagenet-Birds datasets. Extensive human subject experiments are conducted to validate the human interpretability and consistency of the extracted explanations.

While class-specific concepts unravel the blueprints of a class from CNN's perspective, concepts are often shared across classes; for instance, gorillas and chimpanzees naturally share many characteristics as they belong to the same family. The second framework, SCE, unravels the concept sharedness across related classes from CNNs perspective. The relevance of the extracted concepts towards prediction and the primitive image aspects, like color, texture, and shape encoded by the concept, are estimated after training the explainer, enabling it to shed light on the various concepts on which the different black box architectures trained on the Imagenet dataset group and distinguish related classes.

The secondary focus of the thesis is to extend the fruits of explainability to allied learning paradigms contributing to state-of-the-art image classification successes. Domain adaptation techniques that leverage knowledge from an auxiliary source domain for learning in labeled data-scarce target domain increase accuracy. However, the adaptation process remains unclear, particularly the knowledge leveraged from the source domain. The third framework XSDA-Net uses a case-based reasoning mechanism to explain the prediction of a test instance in terms of similar-looking regions in the source and target train images. The utility of the proposed framework is theoretically and empirically demonstrated by curating the domain adaptation settings on datasets popularly known to exhibit part-based explainability. Ablation analyses show the importance of each component of the learning objective.

This thesis also provides a complete description of the XAI field, summarizing the state-of-the-art contributions to the different types of explanations. The underlying principle, limitations, and improvements made to these seminal contributions have also been highlighted. Furthermore, this thesis also presents future research directions and unexplored avenues in XAI research.

Keywords: Explainable AI; Concept-based Explanations; Posthoc Explainability; Explainable by Design; Supervised Domain Adaptation

List of Publications

The content of the thesis is based on the following papers.

Journals

1. Vidhya Kamakshi and Narayanan C Krishnan. Explainable Image Classification: The Journey so Far and the Road Ahead, Accepted at AI, Special Issue on Interpretable and Explainable AI Applications, 2023.

2. Ashish Kumar, Karan Sehgal, Prerna Garg, Vidhya Kamakshi, and Narayanan C Krishnan. MACE: Model Agnostic Concept Extractor for explaining image classification networks. IEEE Transactions on Artificial Intelligence, 2021.

Conference Proceedings

1. Vidhya Kamakshi, Uday Gupta, and Narayanan C Krishnan. PACE: Posthoc Architecture-agnostic Concept Extractor for explaining CNNs. In 2021 International Joint Conference on Neural Networks (IJCNN), pages 1–8. IEEE, 2021.

2. Vidhya Kamakshi and Narayanan C Krishnan. Explainable Supervised Domain Adaptation. In 2022 International Joint Conference on Neural Networks (IJCNN), pages 1–8. IEEE, 2022.

Other publications (Not a part of the thesis)

1. Rajat Sharma, Nikhil Reddy, Vidhya Kamakshi, Narayanan C Krishnan, and Shweta Jain. MAIRE-a Model-Agnostic Interpretable Rule Extraction procedure for explaining classifiers. In International Cross-Domain Conference for Machine Learning and Knowledge Extraction, pages 329–349. Springer, 2021.

2. Sam Zabdiel Sunder Samuel, Vidhya Kamakshi, Namrata Lodhi, and Narayanan C Krishnan. Evaluation of saliency-based explainability method. In ICML Workshop on Human Interpretability in Machine Learning, 2021.

Under Review

1. Vidhya Kamakshi and Narayanan C Krishnan. SCE: Shared Concept Extractor to Explain a CNN's Classification Dynamics.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Object recognition is the task of identifying objects present in an image, for instance, a computer, an animal, or a bird. This is a task that a human can easily accomplish. However, it is challenging to get an object automatically recognized through computers. Computer Vision techniques realize the object classification task as choosing the category of the object contained in an image from a given set of object categories [1]. A typical image classification model has two major steps: feature extraction and classification. Feature extraction is the process of extracting relevant attributes called features from the image that contain traces enabling identification of the object class. The classifier combines the extracted features to predict the object class.

Traditional Computer Vision techniques focused on developing hand-crafted features like the Scale Invariant Feature Transform [2], Histogram of Oriented Gradients [3], etc., to extract features whose aggregation would yield the prediction. However, using hand-crafted features yielded limited results with the growing complexity of data [4]. The advent of deep Convolutional Neural Networks (CNN), since the AlexNet [5], brought in a paradigm shift in the community's notion of feature extraction, as these models were able to extract the discriminative features automatically from the data. With time, deeper architectures with more hidden layers[6, 7] demonstrated higher performances.

However, the increased accuracy with the increased number of parameters comes at the cost of decreased transparency. Traditional machine learning models, for instance, a decision tree, are interpretable by nature as their working mechanism can be summarized by means of if-then-else rules. Summing up the working mechanism of a CNN in a similar manner is not trivial. It is well recognized that the initial layers closer to the input detect rudimentary features like edges or contours while the latter layers closer to the output layer process complex image components like object parts [8, 9]. Gaps persist in the community's understanding of how an image is decomposed, and the extracted features are aggregated to deduce an instance's class. As this opacity of CNNs can limit their widespread use in many safety-critical paradigms like medicine [10, 11], judiciary [12], where transparency regarding the working mechanism of the deployed model is sought, it becomes important to develop mechanisms to explain the working of these deep black-boxes. Moreover, The Right to Explanation Act by the European Union(EU) [13] has made it mandatory for businesses leveraging Artificial Intelligence(AI) in their work processing pipeline to explain why a certain decision made by the AI model was carried out. This has led to a spurt in the development of Explainable AI.

1.1 What is Explainable AI (XAI)?

Explainable AI refers to the set of techniques and methodologies used to make AI systems more transparent, interpretable, and understandable to humans. These techniques can be used to help humans understand how an AI system makes a decision, what factors are considered, and how confident the system is in its decision. In traditional models like linear regressors, the coefficients reveal the importance the model gives to a certain input feature. Similarly, the working logic encoded in a decision tree can be translated into a set of if-then-else rules. XAI algorithms are developed to unravel the working mechanisms of complex, accurate models like random forests, neural networks, etc., whose working is difficult to summarize in a similar human-interpretable manner. There have been several attempts to explain the working of different types of black boxes that work with data of different modalities like tabular data [14], text [15], images [16], etc. Explainable AI methods that unravel the internals of the black boxes can have several benefits for both users and developers of AI systems. For users, XAI can help build trust in the system, as they can understand how it works and why it made a particular decision. This can be especially important in high-stake applications, such as healthcare [10, 11] or finance [17, 18], where decisions made by an AI system can have significant consequences. For developers, XAI can help debug and improve the system, as they can understand how the system makes decisions and identify areas for improvement [19]. As image classification, the task in the scope of this thesis is typically done using CNNs, the focus of this thesis will be on explaining the working of CNNs.

Firstly it is important to clarify what it means to explain a CNN. A CNN takes an image as an input, extracts features using the convolutional and pooling layers, and combines them using the fully connected layers to classify the instance into one of the several categories. Considering the example of the previously stated classical machine learning model, namely the decision tree, translating the path traversed from the root to the leaf node into if-then-else rules yields the features that led to the prediction. Similarly, an explanation that unravels a CNN's working mechanism to classify a given image is expected to highlight the significant image features to arrive at the prediction. We illustrate the benefits of an explanation through a motivating example.

Consider a CNN model that recognizes birds in images and classifies them as either an *albatross*, *hummingbird*, or *pelican*. If a test image of an *albatross* is misclassified as a *pelican*, one may be curious to know why the instance is misclassified. One may turn to XAI algorithms to analyze the feature in the *albatross* image that is misjudged as that of a *pelican*. A good explanation that can justify the misclassification may be that in the given image, the beak of the *albatross* looks similar to that of the *pelican*, resulting in the *albatross* instance being misclassified as a *pelican*.

While misclassification is one scenario where understanding the CNN's working mechanism is sought, explanations may also be needed for correct classifications. Such explanations can reveal what features the model relies on to make predictions and enlighten the

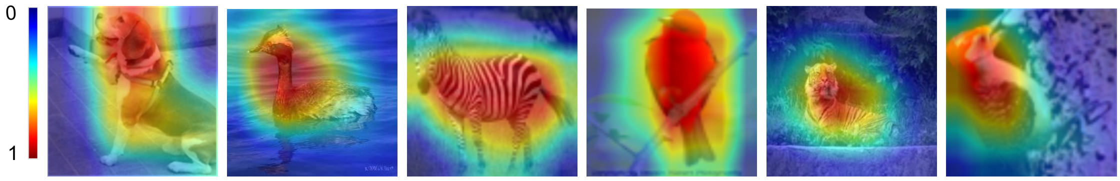


Figure 1.1: CNN explanation in the form of a saliency map localizing the image region contributing to the prediction.

correctness of the model's working. Moreover, explanations can highlight spurious biases [20, 21, 22] that the model relies on, helping to assess the model's deployability in real-world scenarios. For example, in the bird classification task, a model may rely on the presence of a *water* background to distinguish *pelicans* from other birds. This correlation may enhance accuracy in the given dataset, but the model cannot be deployed in a real-world bird recognition task, where the background need not always contain water that the model has encountered during the training time.

To improve the user's trust in the deep model and to ensure their ethical deployment for real-world tasks, the XAI research community aims to develop methods that explain the internal working mechanism of the learned CNN models, which are essentially black boxes.

1.1.1 Types of Explanation

The need for XAI has resulted in the development of various explanation mechanisms to understand a CNN. The different types of explanations have been overviewed to identify the missing gaps in the existing proposals that may motivate proposing newer viewpoints on what should make an ideal explanation.

The most common outlook for explaining a CNN is identifying the key image regions contributing to the predictions [21, 23, 24, 25]. These key regions are often displayed using a saliency map, where the image regions are color-coded based on their importance. Several examples of these saliency maps are shown in Figure 1.1. As illustrated, the image region containing the entire object is almost always said to contribute to the prediction. This type of explanation confirms whether the classifier focuses on the object or relies on any spurious associations that are not relevant to the object. However, additional fine-grained information, such as the contribution of image primitives like colors, textures, shapes, and parts towards the predictions, cannot be obtained from such an explanation. The XAI community proposed dividing the image into smaller segments to obtain fine-grained information. The explanation algorithm ranks the segments based on their importance to the prediction [20, 26, 27]. An example of how such an explanation would appear can be seen in Figure 1.2, where the segments covering the ears, muzzle, legs, and black body are highlighted to be significant to the prediction. While at the outset, it may seem that this proposal achieves extracting fine-grained explanations, it is to be noted that the image is segmented using different known techniques [28, 29]. There is no guarantee that the CNN uses a similar segmentation to process the images [30]. For instance, in the way, the *beagle* image has been segmented in Figure 1.2, the ears and muzzle are a part of the same segment. It is not necessary that CNN also considers these parts together. In other words, the faithfulness of the explanation to CNN is not guaranteed, i.e., what is revealed to be important for prediction by the explanation need not be what CNN actually deems important.

The viewpoints on explaining a CNN discussed so far are deliberative, meaning they intend to explain a given prediction. They are mostly used to justify a classifier's predictions and diagnose any spurious correlations it relies on. On the other hand, the misclassification scenario, as discussed previously, involves comparing pairs of images of different categories to justify/diagnose the misclassification. To address this concern, the counterfactual perspective on explanations was introduced, in which pairs of images are compared to determine the minimal edits to the query image that flips the prediction [31], as shown in Figure 1.2 where a change in the muzzle structure flips the prediction of a given query image from *beagle* to *basset*. This standpoint of associating instance pairs has been derived from explanations developed for models that work with tabular datasets where the range of feature values is known, and the minimal set of features whose perturbation flips the prediction can be deduced. However, the range of values that the pixels, which constitute an image's input features, can theoretically take is the entire real space. Realistic images lie within a manifold, but determining this manifold is non-trivial. Therefore, generating images within this manifold, such that the generated instance looks realistic and lies within the data distribution, is challenging.

Another line of work focused on estimating the free-form, human-interpretable, and fine-grained explanations decomposes the prediction into contributions from human-interpretable concepts, which can encode any image primitives like color, texture, or parts. Consider a *zebra* that can be thought of as a *horse* having alternate black and white *stripes* throughout the body. These explanations, termed concept-based explanations [32, 33, 34], estimate the impact of each of these concepts, such as stripes, black, white, horse-like mane on the back, etc., towards predicting a zebra. The representations of these human-interpretable concepts are extracted from positive and negative examples that denote the presence and absence of these concepts. Its importance is quantified by assessing the effects of perturbing a concept representation on the prediction. However, a key bottleneck in these approaches is the need for annotated examples that depict the presence or absence of concepts. As the aim of extracting the concept representations from the CNN is to explain the working of the classifier, the examples depicting the presence and absence of concepts must be curated from the same distribution as that of the manifold on which the classifier is trained. When the distributions differ, the representations learned may not reflect the representations used by the CNN [35]. Curating annotated examples that depict the concepts so that the explanations faithfully unravel the importance of these concepts is the main challenge associated with these techniques.

Similar to how CNNs learned to extract features automatically from the data, the XAI

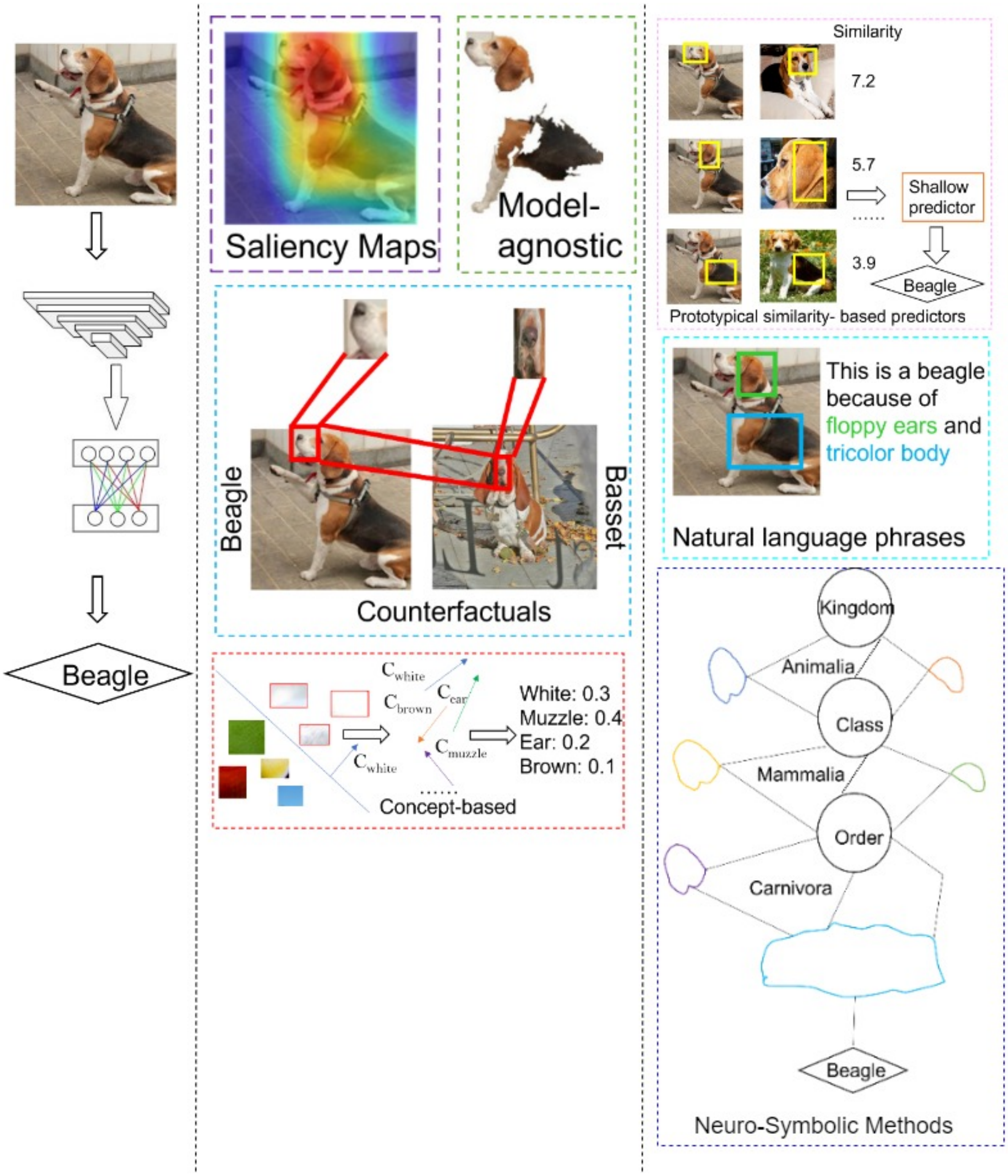


Figure 1.2: Perspectives of explanations - an illustration

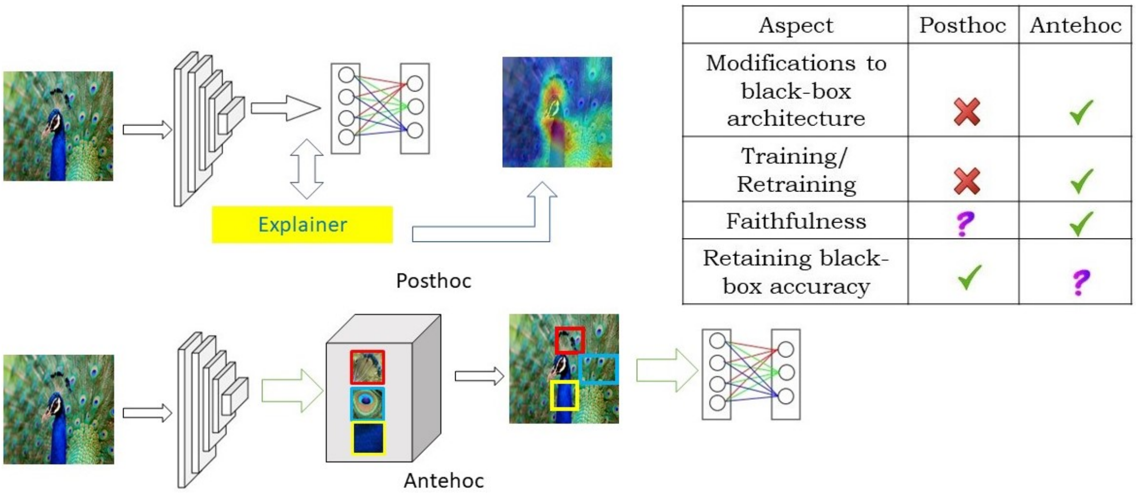
community proposed enforcing the CNNs to learn interpretable concepts automatically from the data and use them to predict the object category [36, 37, 38]. An illustration can be seen in Figure 1.2, where the characteristic regions similar to that of the muzzle, ears, body, etc., of a *beagle* guide the shallow predictor to predict the given test instance as a *beagle*. The discriminative interpretable concepts are learned automatically from the data, and the detection of these concepts in test instances guides the prediction using an inherently interpretable predictor like a linear regressor or decision tree, allowing the complete reasoning pipeline of the modified CNN to be unearthed. In such models, an explanation is incorporated in its training phase by design. As the ability to explain has been incorporated during the training phase, and the CNN is guided to use these explainable components to make predictions, the faithfulness of these explanations is guaranteed. In other words, whatever information the explanation reveals is truly what the model uses to arrive at the prediction. However, it must be retrained from scratch to incorporate such explainability into a CNN. This perspective can be leveraged when the model is yet to be deployed, and it is desirable to deploy a model that can explain itself but cannot be employed for an already deployed model.

The interpretations of the working of CNN, which have been discussed until now, rely on the visual aspects only. There are also attempts to leverage natural language expressions to obtain human-interpretable descriptive explanations [39, 40, 41, 42, 43, 44]. The key principle in this line of works is to generate natural language phrases that correspond to different regions of an image, which in turn justify the CNN's prediction. Figure 1.2 shows an example of this type of explanation, where a CNN is said to predict the given test instance as a *beagle* due to the presence of floppy ears and a tricolor body localized by color-coded bounding boxes. While this perspective has some advantages, such as providing a more human-readable explanation of CNN's prediction, it also has some limitations. For example, it requires large amounts of annotated data to train the natural language model to provide accurate justifications, which can be expensive and time-consuming. Furthermore, the language model used to generate the natural language phrases is another black box whose working mechanism is unknown and needs to be unearthed [45]. Nonetheless, this can be used as an additional tool for understanding how CNNs operate and can be a useful complement to other types of explanations [46, 47, 48].

1.1.2 Explainable AI Approaches

The discussions thus far have centered around the types of explanations. This section provides an overview of the XAI methods. A detailed review analyzing the pros and cons of individual mechanisms can be found in the next chapter.

The XAI techniques can be classified into two broad families: Posthoc and Antehoc techniques, based on how the explanations are incorporated. In Posthoc techniques, explanations are generated without modifying the underlying CNN architecture. The method may [23, 25] or may not assume access [20, 26] to the intermediate layers of the CNN. As the black box, aka the CNN, is undisturbed, there is no need to retrain the



Aspect	Posthoc	Antehoc
Modifications to black-box architecture	✗	✓
Training/ Retraining	✗	✓
Faithfulness	?	✓
Retaining black-box accuracy	✓	?

Figure 1.3: Comparison of antehoc and posthoc explainability methods

model to incorporate explainability. This makes posthoc methods preferred to generate explanations from an already deployed model. However, ensuring faithfulness of the generated explanation to the working mechanism of the CNN is a key challenge when posthoc methods are employed to explain a CNN, i.e., ensuring the consistency between the explanation's ranking of the features based on their significance to the prediction and the ranking by the black box CNN being explained is a non-trivial requirement to be fulfilled by the posthoc explanation method employed.

On the other hand, Antehoc methods incorporate the aspect of explainability and maximizing the classification accuracy into the learning pipeline. For this, they either modify the existing black-box architectures [49] or propose novel architectures where explainable artifacts are detected. These detections then guide the prediction [36, 38]. As explainability is a part of the training pipeline, the generated explanations are faithful to the CNN, i.e., the explanation reveals the true underlying mechanism used by the CNN to arrive at its prediction. However, retraining the CNN or modifying its architecture to extract faithful explanations comes at the cost of lowered accuracy, i.e., it is challenging to achieve the classification accuracy of an unrestricted CNN in the modified version with explanatory bottlenecks incorporated by its design. Thus, as it can be seen from the summary in Figure 1.3, the Explainable AI methods have an accuracy-interpretability tradeoff.

1.2 Automated Concept-based Explanations

The explanations generated by the Explainable AI approaches will be used by humans to assess the trustworthiness of the CNNs. Hence it is important for the algorithms to furnish faithful explanations in a way that is aligned with how humans process the images. Lake et al. [50] studied how humans process images and ascertained the previous theories [51, 52] supporting recognition of complex images in terms of individual concepts. For instance, a skateboard is recognized by means of individual components like wheels, handles, etc.

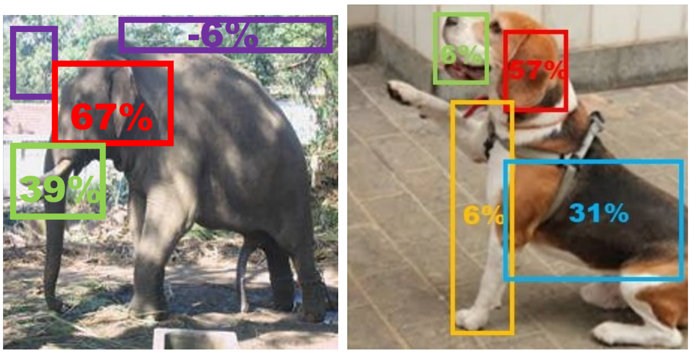


Figure 1.4: Explanation - an illustration

The study suggests that this recognition in terms of concepts (parts) aids humans in recognizing sketches that look slightly different from that of the real object. Also, a human can recognize these concepts in a novel, related class, such as a motorbike with wheels and handles. The novel concept of a seat present in the motorbike is learned, and thus humans quickly expand their knowledge by leveraging already-known concepts. Kim et al. [32] suggest that concept-based explanations help users better diagnose and trust the CNNs as the explanation in terms of individual concepts is closely aligned with how humans process images [53]. These studies motivate the thesis to propose *concept-based explanations with the intent to learn concepts automatically from the data.*

A concept-based explanation is thus defined as a set of concepts c and its corresponding relevance r, formally defined as $e = \{(c, r)\}$. The concepts c are technically abstract vectors in a latent space. These vectors encode image primitives like colors, textures, and parts present in the image sub-regions. The pipeline to unravel the working mechanism of a CNN gets completed when the significance of these extracted concepts towards predicting a given instance is estimated. This is termed the relevance r, which is a normalized score $r \in [-1, 1]$, indicating the extent to which a concept supports/inhibits a specific prediction. Figure 1.4 illustrates an example of an explanation generated by our proposed frameworks as a set of concepts and their corresponding relevances expressed in percentage. Differently colored contours represent different concepts, and the numbers beside them indicate their relevance to the prediction. For the image of an *elephant* shown in Figure 1.4, the concept of the elephant face contributes 67% to the prediction, while the concept of the elephant trunk contributes 39%. On the other hand, the green trees present in the background suppress the prediction confidence by 6%, i.e., the detection of these green trees inhibited the prediction to be steered towards the *elephant* class.

1.3 Research Problems

There are two key questions addressed by this thesis. The first question addresses a challenge associated with existing posthoc concept-based explanations. In real-world scenarios, concept annotations may not be available. Additionally, leveraging any other available concept repository may not yield faithful explanations [35] due to the possibilities

of distribution shift between the data on which the CNN was trained and the data from which concept representations are learned to be extracted by the explainer. To faithfully explain a CNN in a posthoc fashion and human-friendly manner, *can concepts learned by a CNN relevant to prediction be automatically extracted?*
While much focus of the XAI community is on explaining in-domain classifiers, which are trained and tested on data sampled from the same distribution, the second research question addressed in the thesis extends explainability methods to unearth the working of a cross-domain classifier that is trained and tested on data sampled from different distributions. Domain adaptation is one of the key drivers of deep learning success that enables its application to other fields where data availability is scarce. Although many state-of-the-art domain adaptation techniques have been developed that yield competitive performances using deep models with very limited data, the underlying process that enables the domain adaptation algorithms to leverage the aspects of an auxiliary data-rich source domain is unclear. To bridge this gap, the thesis proposes to build *an antehoc domain-adapted classifier that can explain itself.*

1.4 Notations

This section lists the key notations used across the different frameworks proposed in this thesis.

- The black box CNN being explained is a K-way classifier, i.e., the CNN categorizes the given instances into one of the K known classes.
- The CNN processing an instance x consists of a feature extractor f yielding feature maps $f(x) \in \mathbb{R}^{H \times W \times D}$, where H and W denote the height and width of the feature map and D denotes the number of channels in the convolutional layer of interest, and a classifier h yielding a probability distribution P over the K classes.
- Explanation is in terms of a set of concepts $\mathcal{C}$.
  - The C concepts can be shared across all classes denoted by $\mathcal{C} = \{c_a\}_{a=1}^{C}$
  - Otherwise, the concepts may be class-specific denoted by $\mathcal{C} = \{\mathcal{C}_k\}_{k=1}^{K}$, where each $\mathcal{C}_k = \{c_a\}_{a=1}^{C}$. Whenever the subscript has two indices, the primary index corresponds to the class, and the secondary index corresponds to the concept being considered.
  - $c_{ab} \in \mathbb{R}^Q$ denotes the b^{th} concept of the a^{th} class, with Q denoting the dimension of the low-dimensional explanatory latent space. Typically $Q \leq D$.
- In the cross-domain classification setting, a superscript $d \in \{s, t\}$ denotes the domain being considered. For instance c_{ab}^s denotes the b^{th} concept of the a^{th} class in the source domain superscribed by s.
- The parameters of the baselines considered are superscribed as BL. For instance, c_{ab}^{BL} denotes the b^{th} concept of the a^{th} class in the baseline framework.

1.5 Contributions

The thesis proposes three novel frameworks that explain the working of black-box classifiers through the lens of concepts. The concepts are human-interpretable sub-regions that are automatically learned from the data. The first two frameworks explain in-domain classifiers that are trained and tested on data sampled from the same distribution in a posthoc manner by proposing an explainer that can automatically learn to extract concepts used by the CNN to predict a given instance; the third framework proposes a mechanism to build a cross-domain classifier that leverages data from an auxiliary data-rich domain to classify instances sampled from a data-scarce domain having a different distribution that can explain itself by design.

The first framework, Posthoc Architecture-agnostic Concepts Extractor, abbreviated as PACE, explains an in-domain classifier in a posthoc manner by extracting class-specific concepts. The concept extraction process starts by learning to extract image-specific representations called embeddings, which are latent vectors in a low-dimensional space. Class-specific discriminative latent vector representations called concepts are then learned, centered around these embeddings. These concepts are propagated forward through the rest of the network to ensure that the learned concept representations faithfully represent the features used by the black box model. A measure has been proposed exclusively for assessing the goodness of concept-based explanations called agreement accuracy. It is used to validate the faithfulness of the learned explainer. The concepts are enforced to be class-specific by encouraging the creation of tight coalitions in the latent space based on class labels. The concepts extracted must be significant for the black box to predict instances of the corresponding class. A concept is significant to the prediction if its removal causes a significant drop in the prediction probability of the black box. This is enforced as a constraint in the concept learning pipeline to extract class-discriminant concepts relevant to the black box to predict instances of the given class. The extracted concepts shed light on the black box classifier's mechanism to arrive at the prediction.

Although class-specific concepts help understand the notion the black box model has developed for a class, the classes are not always independent from each other. Commonalities across classes exist in nature [37], in addition to concepts that distinguish them apart, as illustrated in Figure 1.5. The second framework, Shared Concepts Extractor, abbreviated as SCE, facilitates the extraction of such shared concepts from the lens of in-domain classifiers in a posthoc manner. The concepts are extracted in the same space as the features generated from the black box. An incremental concept extraction mechanism that extracts concepts from instances processed as mini-batches facilitates concept extraction from large multi-way classifiers leveraging available memory resources. Instead of restricting concepts to be forward propagated based on their stronger presence, as done previously, forward propagation is formulated by combining the presence of concepts without capping based on the strength of their presence. The constraint regarding extracting relevant concepts has been relaxed to unravel all concepts detected

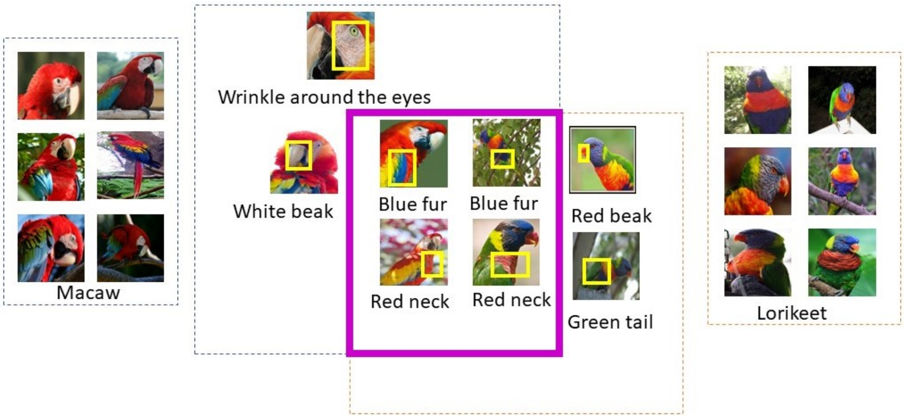


Figure 1.5: Concept sharedness - an illustration. The concepts corresponding to the class *macaw* are enclosed in a blue dotted rectangle, and those of the class *lorikeet* are enclosed in an orange dashed rectangle. The concepts shared across both classes are enclosed by a bold purple rectangle.

by the black box. The explanation pipeline is simplified by estimating the relevance of a concept after its extraction. SCE also automatically tags the image primitives like color, shape, and texture encoded by the concept by inspecting the effect of perturbing the primitive [54] on concept detection. The framework unravels the grouping of classes based on the shared concepts and highlights spurious correlations the model has picked up in line with a recent observation [22].

While the first two contributions aim to explain an in-domain classifier in a posthoc manner by varying the sharedness of the extracted concepts across different classes, the third contribution, termed the EXplainable Supervised Domain Adaptation Network, abbreviated as XSDA-Net, focuses on cross-domain classification. The goal is to propose a supervised domain-adapted classifier that can explain itself. In other words, XSDA-Net proposes an antehoc domain-adapted classifier with case-based reasoning integrated by design, which learns to extract domain-invariant concept pairs, called prototypes, that are discriminative for each class and domain. Predictions for a test instance are made by detecting concepts in it that are similar to the learned domain-invariant prototypes. As the prediction is based on the extracted interpretable concepts, the complete reasoning pipeline of the proposed case-based domain-adapted classifier, aka the XSDA-Net, can be explained.

Overall, these contributions represent a significant step forward in developing interpretable classifiers, as it allows for a deeper understanding of the reasoning process behind the predictions.

1.6 Organization of the Thesis

The thesis comprises six chapters, with Chapter 2 covering relevant literature in the field of Explainable Artificial Intelligence (XAI). This chapter discusses a wide range of XAI techniques, from those that explain the black box by observing input-output interactions to those that leverage black box parameters to generate explanations. In addition to explaining already-deployed black box models, the chapter also explores techniques for building models from scratch that incorporates human-interpretable concept detection. The chapter also covers recent posthoc techniques that map internal representations of black boxes to human-interpretable concepts, as well as discusses the gaps in the existing techniques that motivated the proposal of the three concept-based explanation frameworks.

The next three chapters (Chapters 3, 4, and 5) are dedicated to the proposed concept-based explanation frameworks. The structure of these chapters is similar. Initially, the problem statement and rationale behind the proposed framework are introduced. The framework's benefits are then presented in the context of the shortcomings of the most comparable frameworks it addresses. Subsequently, an in-depth explanation of the methodology and the training process is provided. Additionally, each chapter outlines the experimental setup, datasets, and baselines used to evaluate the performance. Finally, the results of the experiments are detailed and analyzed for each framework.

The final chapter summarizes the key insights gained from the proposed frameworks. It concludes the thesis by discussing open-ended questions in the field of XAI and possible future avenues of research. Specifically, three directions for future research are discussed - one by incorporating explainability into other learning paradigms like Few-shot Learning, Incremental Learning, etc. The second direction discusses the sufficiency of the existing metrics in assessing the goodness of the posthoc explanations when extended to other paradigms. The third direction suggests leveraging ideas from Neural Architectural Search to determine the optimal network configuration in the antehoc concept-based frameworks to achieve a minimal drop in performance compared to their non-interpretable counterparts.

Chapter 2

Literature Review

This chapter presents a condensed review of the state-of-the-art contributions to the Explainable AI (XAI) field. The underlying principle, limitations, and improvements made to these seminal contributions have also been highlighted. Based on the stage at which the explanations are incorporated, the XAI methods have been categorized into two broad families: posthoc and antehoc. While posthoc methods leave the CNN undisturbed, antehoc methods modify the training mechanism to incorporate explainability into the model. While the discussion in this chapter is based on this categorization of approaches, other perspectives exist to classify XAI frameworks.

On the basis of the scope of their explanations, i.e., whether the generated explanation unearths the whole working mechanism of the model or restricts itself to explaining how the model behaves in a limited neighborhood surrounding an instance of interest, the XAI methods are bifurcated as local or global methods. Global methods explaining the CNN in the complete instance space can be used to construct interpretable proxies mimicking the working of the CNN that can be used in safety-critical applications where explainability is essential. While this is desirable, often generating a global explanation that faithfully encodes the non-linear manifolds learned by the CNN is challenging. To manage the challenge, local explanations exploiting the local linearity of the data manifold are used to explain the CNN in a local vicinity around the instance of interest. One can obtain an approximate global explanation by aggregating local explanations over a set of instances.

Based on assumptions regarding the type of black box it queries to generate explanations, the methods are categorized into model-specific and model-agnostic methods, with model-specific methods assuming architectural constraints to generate explanations. In contrast, model-agnostic methods generate explanations by looking at the input and output interactions, assuming nothing about the black box it aims to explain. Model-agnostic explanations are useful, particularly when the black box model is not publicly available and is used through an API that supports providing inputs and accessing the corresponding outputs only. However, these methods have certain underlying principles; for instance, there exist interpretable features whose aggregation would yield the working of the black box, to facilitate working with any black box architecture or data modality. Such principles need not always be true, so it is desirable to use model-specific methods whenever the black box to be explained is completely accessible.

As per the class label on which the explanation is queried, the explanation may be categorized as deliberative if the black box prediction is justified. The other category, namely the counterfactual explanations, supports editing the given instance with the

intent to alter the predicted label. While deliberative explanations help identify biases, if any, in the learned model, counterfactual explanations are useful in *Machine Teaching* [55] where the explanations based on hypothetical counterfactual instances created shall help humans better understand the distinction between classes as the intent of generating counterfactual is to find the closest instance belonging to an alternate class of interest.
As motivated in the previous chapter, the thesis proposes concept-based explanatory frameworks to explain a CNN. The concepts encompass the global information about the model, which the users unearth through aggregating from multiple local explanations. The proposed explainers can support pseudo-counterfactual queries seeking deliberate explanations justifying a certain prediction probability for an alternate class. It is to be noted that the explanation can be generated to justify the prediction probability the black box has attributed to an alternate class through deliberate explanations from the proposed frameworks. However, the counterfactual perspective of modifying the given instance, thereby generating a hypothetical instance whose prediction is steered towards an alternate class of interest, is beyond the scope of the proposed explainers. To explain in-domain classifiers, the thesis proposes posthoc frameworks, and for cross-domain classification, the thesis proposes an antehoc supervised explainable domain adaptation framework that explains itself. This chapter reviews existing techniques that explain cross-domain and in-domain classifiers from the antehoc and posthoc perspectives, determining the stage at which interpretability is embodied.

2.1 Posthoc Methods

Posthoc XAI methods refer to techniques and methodologies used to explain the behavior of an AI system after it has been trained to make a decision. These methods do not necessarily modify the AI system itself but rather analyze the output generated by the system to provide explanations for the decision-making process. A major advantage of using these methods is that they do not require any architectural modification or black-box retraining. They probe the trained black-box model to understand its working. The posthoc methods can be subcategorized under four major heads: Saliency Map, Model-agnostic, Counterfactual, and Concept-based approaches, as discussed in the following subsections.

2.1.1 Class Activation Maps

Saliency Maps assume that the region salient towards the prediction of a class can be obtained from a weighted combination of the activation maps from the convolutional layer filters. Inspired by the observation that the latter layers encode complex parts [9], most saliency estimation approaches extract activation maps from the last convolutional layer closest to the output. Let the convolutional layer of interest have n filters. Let A_i be the activation map from the i^{th} filter. The explanation algorithms assess the salient regions that the CNN focuses on by means of a saliency map S that can be expressed as a weighted combination of the activation maps from each of the n filters, i.e., $S = \sum_{i=1}^{n} w_i A_i$. This formulation stems from the understanding that the features extracted are combined to arrive at the prediction. The low dimensional saliency map obtained through the weighted combination of the activation maps from the individual filters is then upsampled, to the full image size to generate an explanation showing the image region that the CNN focuses on to arrive at the prediction. Various mechanisms have been proposed to estimate the weights $\{w_i\}_{i=1}^{n}$ that combine the activation maps from the filters. These approaches can be bifurcated based on leveragement of gradients, as will be discussed below.

Gradients capture the direction along which the value of a function increases. Thus gradients propagated back to the convolutional layers from the output layer carry a signal indicating the features whose presence steers the model towards making a desired prediction. This signal is leveraged to estimate the weights to combine the activation maps by the gradient-based saliency approaches. Grad-CAM (Gradient-weighted Class Activation Mapping) [21] is a visualization technique for deep neural networks that helps understand where a neural network looks in an image when making a prediction. It generates a saliency map highlighting the regions of the input image that were most relevant for the neural network's prediction. It works by computing gradients of the output prediction with respect to the activations of the final convolutional layer. The activation maps are combined based on the weights obtained by averaging the gradients with respect to the corresponding filter over all the spatial locations. No additional modifications to the neural network architecture are needed to generate explanations and thus can be leveraged to explain any CNN. The following year Chattopadhyay et al. [23] observed that having the averaged gradients as weights to combine the activation maps do not localize well in images where multiple instances of the same class are present. They proposed applying different weights to gradients observed at each spatial location to uncover all regions steering the prediction; thereby, the observed limitation of Grad-CAM [21] in localizing more than one instance of the class can be overcome. The weights to these spatial locations were deduced to be obtained from higher-order derivatives whose computation could be demanding in complex architectures. Integrated gradients [56] considers a reference input and traverses the instance space across the path from a reference input to reach the given instance. The attributions with respect to the intermediate instances along the path are integrated to obtain a robust saliency map depicting the salient pixels in the given instance. Excitation backpropagation [57] utilizes a probabilistic winner take all strategy where the attribution being propagated to a downstream neuron is probabilistically determined. Guided backpropagation [58] proposes propagating attribution only to those neurons which were active during the forward pass, thereby generating finer pixel-level saliency maps compared to the vanilla backpropagation [59] that propagated gradients as attribution irrespective of the contribution of the neuron until arriving at the output layer.

Various quantitative measures [23] have been proposed to assess the faithfulness of the generated explanations. The proposed measures are based on the requirement that removing a salient region must lower the model's prediction confidence while its presence has to amplify the confidence. Viewed differently, these measures observe the effect of perturbing the regions deemed salient on the model's prediction probability. The proposal of these measures is inspired by the first principles of generating explanations that a region whose perturbation impacts the prediction is salient. Instead of going through the voluminous possibilities of all image perturbations, Chattopadhyay et al. [23] propose to use derivatives to localize salient regions and verify if the regions localized to be salient are truly salient by observing the effect of perturbing those regions on the CNN's prediction probability.

Wang et al. [60] empirically showed that the gradient-based saliency maps obtained do not vary with respect to the queried class, thereby questioning the faithfulness of these explanations. Adebayo et al. [61] proposed litmus tests that a posthoc XAI method has to pass towards its proof of faithfulness to the underlying black box model. There are two basic tests that an explanation method has to pass, namely the parameter randomization, which observes the change in explanations when the model weights are randomized, and label randomization, which observes the change in explanations when the labels are randomized and the CNN model is retrained to model the altered distribution. It has been observed that most of the gradient-based techniques fail to satisfy these proposed litmus tests. The theoretical analysis by Sixt et al. [62] attributes the invariance in the saliency map for the model parameters and query labels to the restriction of the explanatory model to the positive subspace of the activations.

Following the issues found with using gradients to determine saliency, the XAI community has proposed other methods to generate saliency maps. There have been attempts [24, 25, 63] to incorporate the effect of perturbation at the level of filters to assess the importance of the activation maps, which will, in turn, be the weights w_i combining the activation maps A_i. It is easier to manage the possible perturbations [64, 65] with n filters of the convolutional layer of interest than that of the input image of much higher dimensions. Wang et al. [24] associate the importance weight w_i to combine the activation map A_i based on the effect it has to obtain the prediction to the desired class, i.e., the prediction probability obtained when the activation map A_i is present and the other activation maps are nullified, is the weight w_i that combine the activation map A_i. Desai & Ramaswamy [25] take a complementary route by considering the drop in prediction probability when the activation map of interest A_i is ablated while forward propagating other activation maps without any modification to determine the weight w_i. A limitation of these approaches is the need for multiple forward propagations to get a single saliency map. In contrast, the previously proposed gradient-based approaches can generate the saliency map in a single backward pass. To mitigate this issue, Salama et al. [63] propose clustering similar activation maps and obtain the ablation score for a cluster from which the weights w_i for each activation map A_i can be recursively determined. There have been attempts to propagate a special signal called relevance [66, 67] from the output layer back to the input to determine the pixels salient to the prediction. However, the fact that these pixel-level saliency maps are not class-discriminative has led to the cross-pollination of ideas from these techniques to estimate the combination weights w_i of CAM [68, 69, 70, 71]. Layerwise Relevance Propagation [66] propagates the output of the neural network back through the different layers to assign relevance scores to these input features. The forward pass propagates the activation from the input layer and reaches the output layer. Relevance propagation starts in the opposite direction from the output layer, and gradually the relevance signal reaches the individual input pixels. The relevance propagation is based on the idea of conservation, i.e., the relevance signal from a neuron is distributed across all neurons that have contributed to it during the forward pass proportional to their contribution. Lee et al. [68] apply the idea of relevance propagation[66] to estimate the relevance of the filters, which can act as the weights w_i to combine the activation maps A_i. Deep-LIFT [67] is a modified form of relevance propagation where differences between activations with respect to a reference input are propagated to obtain the relevance of the different input features. Mostly the input having zero in all its dimensions is taken as the reference input. Extending the idea from Deep-LIFT [67], Jung & Oh [69] estimate the filter weights w_i to combine the activation maps A_i through the differences of the combination weights obtained with respect to a reference input. Sattarzadeh et al. [70] extend the idea of integrated gradients [56] to integrate the attribution maps obtained across the path from reference input to the given input. Wang et al. [71] generate image patches [72] and use an attention mechanism to estimate the salient regions in a given image. However, a major limitation of these saliency map approaches is that they almost always highlight the region containing the entire object to be salient [73]. While these explanations can ascertain whether the model looks at the object to arrive at its prediction or relies on any non-object spurious correlations [20, 22]. Finer explanations depicting the contributions of image primitives like colors, textures, and parts cannot be obtained from the Class Activation Maps.

2.1.2 Model-agnostic Explanations

Model-agnostic methods refer to the family of XAI methods, which explain the working of a black box model by just observing the input-output interactions. They can be applied to any machine learning model, regardless of its type or architecture, and can work to explain data of any modality like text, images, tabular data, etc., The scope of the explanations these methods provide can be local to a given instance or can globally explain the overall working of the black box. These methods aim to construct an inherently interpretable pseudo classifier that approximates the working mechanism of the black box classifier to be explained either locally around a small neighborhood of an instance for which the explanation is sought or globally, spanning the complete instance space of the classifier.

Local Interpretable Model-agnostic Explanations (LIME) [20] provides explanations for the predictions made by complex models such as neural networks. LIME generates a simpler, more interpretable model, for instance, a linear regressor or a decision tree whose complexity is optimized such that the determined approximator mimics the behavior of the original model in the local vicinity of the input space around the instance to be explained. This simpler model can then be used to provide local explanations for individual predictions. It can be observed that different explanations can be generated for the same instance depending on the sampled neighbors based on which the local neighborhood is estimated. Zafar & Khan [74] propose a deterministic approach to sampling neighbors utilizing agglomerative hierarchical clustering and sampling k-nearest neighbors using which an interpretable approximator is constructed. Collaris et al. [75] hint at the possibility of sampling fewer neighbors when sampling is performed independent of the queried instance to be explained and propose to sample from a hypersphere around the instance to obtain a robust local explanation. Anchors [26] generate explanations for individual predictions using if-then rules constructed in a bottom-up fashion such that the rule precisely covers the local neighbors of the instance to be explained. MAIRE [27] extends Anchors [26] to handle continuous-valued attributes by learning to construct an optimal orthotope automatically, unlike the prior approach [76] that requires the range of values to construct the orthotope. Local explanation methods aim to extract explanations that are faithful in a local neighborhood by means of special measures like coverage which estimates the fraction of instances that lie within the explainer's vicinity, and precision which denotes the fraction of covered instances whose prediction by the explainer matches with the prediction by the black box CNN. Constructing a MAIRE [27] explainer maximizes the coverage, ensuring faithfulness to the underlying black box by satisfying a precision level set by the user. Though these methods offer local explanations, a global understanding of the model can only be obtained by aggregating the local explanations over a set of instances.

There have also been attempts to build an explainer that approximates the global behavior of the model as a whole. SHAP [77] uses the principles from game theory (Shapley values) to assign an importance score to each input feature, indicating how much each feature contributes to the output of the system. These importance scores can be used to identify the most relevant features and understand their influence on the system's decisions. Computing Shapley values requires considering all possible subsets of the feature space and assessing the effect of perturbation of each subset on the output. This is computationally exhaustive due to the exponential time complexity, and there have been many approaches proposed based on Shapley values approximated by considering only the perturbation of one feature at a time. Permutation feature importance [78] calculates the importance of each input feature by randomly permuting its values and measuring the decrease in the model's performance. Partial dependence plots [79] visualize the relationship between an input feature and the model's prediction while holding all other features constant. Despite approximations [80] to compute Shapley values efficiently, there has been a recent observation [81] highlighting their inadequacy in faithfully capturing the global behavior of the black box being explained. Huang & Marques-Silva [81] construct a boolean dataset where a set of features relevant to determine the output are known. A global explanation is ideal if it assigns zero importance to irrelevant features and non-zero importance to features that correlate with the output. It was observed that there existed features that were truly irrelevant to the prediction but had non-zero Shapley values; Also, when pairs of features were analyzed such that one was actually a relevant feature and the other was irrelevant to the prediction, Shapley values of the irrelevant features were higher compared to that of the relevant features. Sometimes Shapley values for truly relevant features turned out to be zero, contrary to the basic requirement that a global explanation must capture the feature importances accurately. Huang & Marques-Silva [81] conclude that the Shapley values are not always correlated with the actual relevance of features for the black box predictions.

Another important observation is that the model-agnostic methods are developed to generate explanations for any black box model, and hence no assumption regarding its architecture is made. The explanation is given in terms of input features that are significant towards the prediction. In images, the pixels constitute the input features. As pixel-level explanations are not easily interpretable for humans, a workaround suggested using a collection of spatially closer pixels called the superpixels. These superpixels serve as complex input features on which the model-agnostic methods can generate explanations. For this, the existing model agnostic approaches [26, 27] use different predefined image segmentation algorithms [28, 29] to obtain segments constituting the superpixels on which model agnostic explanations are sought. On the surface, it may seem that this workaround achieves a satisfactory level of human interpretability when model-agnostic explanations are sought on images. However, it is to be noted that CNN need not process the image by segmenting it in a manner similar to the model-agnostic explainer [30]. This refutes the preliminary necessity of the proposed approximator, aka the explainer, to be faithful to the underlying black box, aka the CNN being explained.

2.1.3 Counterfactual Explanations

Counterfactual explanations involve generating alternative scenarios to explain the behavior of an AI system. For example, if an AI system for processing loan applications denies a loan application, a counterfactual explanation might involve generating a set of hypothetical inputs that would have resulted in an approved application [82]. These counterfactual explanations can help users understand the decision-making process and identify potential biases or errors in the system. They differ from the deliberative explanations in the sense that the deliberative explanations aim to justify why a certain prediction was made. Counterfactual explanations go a step further to analyze the changes to the input to get another desired prediction. This explanation can be applied to analyze a classifier that works with any data modality, be it tabular, text, or image. The methods try to perform minimal edits to the given query instance such that the prediction is steered towards an alternate desired class. This can be thought of as perturbations intending to flip the prediction. In the case of tabular data, where the efficacy of the counterfactual approaches has been mostly demonstrated, the perturbations are manageable as the range of values the tabular features can take is known, and the instance can be perturbed to generate another realistic instance that lies within the manifold on which the classifier was trained. Determining this realistic manifold is non-trivial in the case of images whose constituents, aka the pixels, can theoretically assume any real value. The objective of explaining using a perturbed instance is common in adversarial learning, except that it does not have a target class towards which the prediction has to be steered. The objective in generating an adversarial example is that prediction on the generated instance must not be the same as that of the unperturbed instance. Caution has to be observed as a random perturbation can generate an adversarial example [83], which may flip a prediction towards the target class of interest but may not be an ideal candidate to extract counterfactual explanations as the instance may be an outlier with respect to the realistic training images' manifold, thereby questioning the faithfulness of the generated counterfactual explanation to the underlying model and data. To circumvent this challenge, the existing approaches [31, 84] either maintain an image bank from which the closest counterfactual image is chosen, or a generative model [85] is used to sample the counterfactual neighbors of the query instance from the distribution on which the CNN is trained.

There have also been some deliberative explanation approaches that allow querying explanation with respect to another class of interest [21], harnessable to generate a counterfactual explanation for the alternate target class of interest. However, these approaches do not generate explanations that vary significantly with respect to the alternate queried class [73].

The preliminary approach to generating counterfactual explanations through realistic instances is by maintaining an image bank from which the closest counterfactual instance to a given test instance is chosen. Various approaches have considered different ways to estimate the closest instance. SCOUT [86] generates deliberate explanations for the given test instance and all instances in the counterfactual image bank and chooses the instance containing features supporting the counterfactual class and no information of the predicted class as the closest counterfactual instance. Goyal et al. [31] simulate permuting feature maps to obtain features closer to that of the counterfactual instances that steer prediction towards the desired class. A main limitation of these approaches is the necessity to skim through the image bank for every test instance to be explained. Additionally, the image bank must be sampled from the same distribution as the data on which the CNN is trained.

To maintain the distribution, an alternate set of approaches employed variants of Generative Adversarial Networks (GAN) [87] to learn the underlying distribution. Singla & Pollack [85] sample instances that vary the prediction probability to navigate through the manifold of the counterfactuals. Zhao [88] proposes using a Star-GAN [89] to generate robust counterfactuals faster. However, it is to be noted that the generative models employed to learn the underlying distribution are, again, black boxes whose working is unknown. This complicates the problem at hand as techniques to interpret GAN [90] need to be employed on top of the existing counterfactual explainers.

2.1.4 Concept-based Explanations

Humans process images through the lens of concepts [50], which can be abstract textures, colors, parts, etc. Concept-based explanations have been proposed to align the explanation algorithms closer to human-like thinking, i.e., the explanations are generated in terms of abstract vector representations that can be mapped to human-interpretable concepts. Typically, a set of examples where the concept is present (termed positive examples) and absent (termed negative examples) are provided, from which the abstract vector representations are learned. Koh et al. [91] proposed a family of classifiers called the concept bottleneck models, which forces the classification to be done through the set of known concepts, which act as a bottleneck through which the processing pipeline has to pass. The basic idea behind the concept bottleneck models is to insert a bottleneck layer between the feature extractor and the classifier of the original model and then train the bottleneck layer to capture the most important concepts from the features of the input data. This approach allows for extracting the salient concepts from the original model, which can be used to create a more interpretable approximator. The training of the concept bottleneck models can be done in three modes. In the sequential mode, a bottleneck layer is designated to detect concepts, enabling the classifier to use the detected concepts to arrive at its prediction. The joint mode of training enforces a weighted optimization of the concept detection and classification objectives. While the third mode of training, namely the independent mode, treats the training of concept detectors and the classifier independently by utilizing the available ground truth. At the test time, the model mimics the pipeline of a sequentially trained model. While the model proposed by Koh et al. [91] may require retraining, Yuksekgonul et al. [34] suggest the usage of a dimensionality reducer as the bottleneck layer that can faithfully map the space of the CNN features to an interpretable low-dimensional concept space, keeping the CNN untouched. Kim et al. [32] leverage the given positive and negative examples to extract representations from the CNN layer of interest. The boundary that separates the positive examples containing a concept from the rest is learned using these representations. The vector in the direction of the positive examples and orthogonal to the learned decision boundary is chosen to be the representative vector denoting the concept. This is illustrated in Figure **??**, where the vector color-coded in red color orthogonal to the linear decision boundary separating the striped instances from others is chosen to denote the CNN's representation of the concept-stripes. Once the concept representation is extracted, its relevance is estimated by inducing perturbation of the concept captured by the directional derivatives. As directional derivatives approximate the inherent non-linearity in the CNN being explained, Pfau et al. [92] propose propagating the perturbed concept through the rest of the CNN and observing the impact of the perturbation on the probability as this could be a more faithful measure due to accounting of the non-linearity of the CNN. However, a key challenge associated with generating such concept-based explanations is the need for annotated examples denoting the presence and absence of concepts. Ramaswamy et al. [35] observed that the curated examples have to be sampled from the same distribution as that of the data on which the CNN is trained so that the extracted concept representations faithfully capture the internals learned by the CNN. Ghorbani et al. [33] propose to use segmentation to subdivide the images at different granularities and curate them to extract examples depicting the presence and absence of concepts automatically. This reintroduces the issue associated with model-agnostic approaches for explaining a CNN regarding the questionable guarantee of the CNN processing images in terms of segments [30], thereby raising a question on the faithfulness of the generated explanation. Arendsen et al. [93] propose leveraging natural language word vectors to learn additional concepts automatically. However, this approach leverages another black box whose working needs to be unearthed [45].

2.2 Antehoc Explanations

Antehoc explainability, or explainability by design as it is popularly called, refers to the practice of building AI systems with explainability and interpretability in mind from the outset rather than as an afterthought. By incorporating explainability into the design process, these methods aim to create AI systems that are inherently transparent, interpretable, and trustworthy. Despite the advantages like inherent interpretability and trustworthiness that antehoc explanations can offer, designing such models can be challenging and may require domain-specific knowledge and expertise. Additionally, some interpretability methods may come at the cost of model performance, limiting their usefulness in certain applications. To incorporate explainability, the architecture of existing CNN architectures may be modified [49], or novel components may be devised that are interpretable by design. The explanation may be highlighting visual artifacts leading to the prediction or providing textual descriptions justifying the predictions. Alternately one may look up to existing knowledge bases to learn models whose working reflects the real-world application requirements.

2.2.1 Visual Explanations

The earliest visual explanatory approaches used attention [94, 95, 96, 97], which is a selective retainment of features to classify the test instance. Attention can be hard or soft in the sense that the selection of regions from the features may be deterministic or probabilistic. The regions attended would be turned in as an explanation. However, there have been observations [98, 99] that an attention map visualized need not be an ideal explanation. Extending the analyses of Jain & Wallace [98] unearthing the limitations of attention-based approaches to explain natural language models, Akula & Zhu [99] conduct extensive human subject experiments, which reveal the usefulness of non-attention based approaches [31, 32] compared to attention-based approaches [20, 21, 66, 56] that explain an image classifier. The authors conduct quantitative tests, which reveal the supremacy of non-attention-based explanations in facilitating the user to think like the CNN as well as qualitative analyses where the users are asked to rate the quality of explanations on various parameters like satisfaction, completeness, etc., as defined by Hoffman et al. [100] on a 10-point Likert scale show that attention-based approaches are not suitable explanations Zhang et al. [101] propose to use mutual information to explicitly enforce the CNN filters to encode distinct parts so that the filters can be visualized to understand the impact of each part of the image. To facilitate the explanation generation, Zhou et al. [49] propose to change the architecture of the CNN to replace the series of fully connected layers incorporating non-linearity by means of a single linear layer which accumulates the average pooled features to get a prediction. The weights that combine these average features are used to combine the feature maps and visualize the salient regions contributing to the prediction. Li et al. [102] propose an autoencoder-based case-based reasoning architecture that looks at characteristic prototypical examples learned from the distribution of instances whose proximity determines the class the test instance belongs to. Chen et al. [36] extend this architecture to learn class-specific concepts called prototypes automatically from data such that the learned concepts are class-discriminant and guide the interpretable classifier that follows it to do the prediction. Many extensions to this approach have been proposed. Hase et al. [103] propose to perform interpretable hierarchical classification by applying the explainable ProtoPNet [36] at every level of the hierarchy. Wang et al. [104] propose modeling instances as a member of class-specific orthogonal subspaces in the feature space. Hoffman et al. [105] and Huang et al. [106] analyze the prospective shortcomings of the ProtoPNet variants. The assumption of class discriminativeness need not be completely true, as concepts may be shared across classes. This idea of sharedness is exploited after training by encouraging sharing of connections to different classes [107]. Nauta et al. [38] construct a decision tree based on learned concepts that implement sharedness by design. However, using decision trees induces negative reasoning, which is overcome by Protopool [37], which enforces a Gumbel-Softmax distribution across prototypes to enforce sharedness closer to real-world sharedness.

2.2.2 Natural Language Explanations

Natural language explanation approaches [39, 40, 41, 42, 43] aim to generate textual descriptions that provide insight into how an image classifier makes its predictions. The key idea behind this approach is to leverage the vast amounts of linguistic knowledge that has been accumulated over centuries of language use and incorporate it into the model. This can help the model generate more coherent and natural-sounding explanations that humans can understand and interpret.

This approach assumes the availability of natural language description for the classes under consideration and for individual instances from which the mapping between visual aspects and natural language phrases can be learned. A trained language model is incorporated to act as an explainer into the classification pipeline to construct a CNN that can justify it's working through natural language phrases. The visual features extracted from the feature extractor of the CNN are fed into the language model, which is trained to generate captions describing the image's content. A critic module then assesses the correctness of the generated caption to the image content. To train the critic module, the ground truth (image, caption) pairs are randomized, and the model is trained to provide a low score for a randomized instance where the image and caption don't agree and a high score on true instances where image and captions agree. The visual features and generated captions from the test image are fed to the critic module, which outputs a score denoting the goodness of the generated caption. To avoid multiple back-and-forth passes through the CNN and caption generator based on the feedback from the critic module, the top-k captions from the caption generator are considered, and the top-ranked caption from the critic is passed into a localization module to localize the corresponding image region contributing to the generation of the caption. This can be seen in Figure 1.2, where a given test instance classified as beagle is justified by localizing the characteristic floppy ears and tricolor body through similarly color-coded bounding boxes.

The approach is mostly used to justify the predictions made in related computer vision tasks, specifically vision-language tasks like image captioning [108], visual question answering [47, 48], etc. where the task involves understanding both visual and linguistic aspects and can be preferably explained when the explanation mechanism also incorporates both vision and language features. Wickramanayake et al. [44] incorporate the textual embedding of the language model to guide the detection of characteristic concepts that drive predictions. This is an explainable-by-design model that leverages both the vision and language aspects.

However, designing effective natural language explanation approaches can be challenging and may require domain-specific knowledge and expertise. Additionally, the quality and effectiveness of the generated explanations can vary depending on the complexity and accuracy of the underlying image classifier and the quality of the available linguistic annotations. Another key challenge to be addressed when incorporating natural language explanations is that the language model which facilitates justifying the prediction is another black box whose working mechanism needs to be unearthed [45].

2.2.3 Neuro-symbolic methods

An alternative family of approaches, known as neuro-symbolic approaches [109], leverages existing knowledge bases or ontologies to acquire the necessary concepts for predicting a given instance, akin to utilizing domain knowledge curated by experts. This phenomenon was initiated with the proposal by Maillot & Thonnat [110], who advocated for collecting knowledge from domain experts and using it to train machine learning models that can base their predictions on the domain experts' knowledge. Marino et al. [111] propose a few-shot classification task by harnessing knowledge encoded in a graphical format. The classifier is trained to traverse different nodes of the knowledge graph and search for image features that match the descriptions associated with the investigated node. As the model navigates through the knowledge graph, the explanation is generated by identifying the localized image regions with the highest degree of match. Alirezaie et al. [112] aim to alleviate the problem of uninterpretable misclassifications by leveraging symbolic knowledge. Daniels et al. [113] propose the design of a bottleneck model [91], which compels the classifier to explore the available knowledge repository and base its predictions on the acquired knowledge. The authors hypothesize that such a design, which enforces the prediction to pass through the knowledge repository bottleneck, enhances the robustness of the learned model. Liao & Poggio [114] investigate the reasons why machine learning models lack the generalizability exhibited by humans. They hypothesize that models adopt a feature-oriented perspective, processing images as a sequence of tensor operations, which leads to variations in representation as objects manifest differently. In contrast, human knowledge processes images in terms of objects and concepts [51, 52, 50], exhibiting invariance to modifications in image manifestations. The authors propose mechanisms to transform the operations performed by feature-oriented models into an object-centric view, aiming to incorporate human-like processing. Ordonez et al. [115] propose a multimodal neuro-symbolic model that combines textual and visual knowledge to predict the entry-level categories to which an image belongs. For example, a neuro-symbolic classifier may have learned encyclopedic categories like *trachypithecus johnii* from the knowledge base, which refers to a species of monkey commonly known as a langur among wildlife enthusiasts. Ordonez et al. [115] address the challenge of mapping from encyclopedic categories to common categories, initially approaching it as an instance of hypernym search in a textual knowledge graph. Acknowledging the potential errors associated with visual cues in the knowledge base due to images of different categories appearing visually similar to humans, the authors propose a learning objective that combines cues from the visual and textual knowledge base to predict the appropriate entry-level category for an image. Icarte et al. [116] demonstrate the utility of a general-purpose ontology in retrieving realistic images that are closest to a given natural language query.

2.3 Causal Explanations

For the sake of completeness, this section discusses the various attempts of the XAI community to generate causal explanations. In real-world data, the features are rarely independent, which can be observed by a corresponding change in another feature when a feature is perturbed. This relationship may be a mere correlation or causal, i.e., the features have a cause-effect relationship. For instance, if the sales of pen increase with an increase in the temperature of the city, this relationship is just a correlation, as there is no known relationship between a pen and temperature. However, an increase in sales of an umbrella with an increase in temperature has a causal relationship, as it is well-known that people tend to look for umbrellas with increasing temperatures. Viewed differently, an increase in temperature causes an increase in sales of umbrellas, where the increase in temperature is a cause, and the higher sales of umbrellas as an aftermath is a result. Many such cause-effect relationships exist in nature. It is of interest to the research community to see if the machine learning models capture such causal relationships [117, 118, 119] and design models which work based on causal relationships so that the spurious correlations [22] are not picked up to arrive at the prediction [120, 121, 122].

Frye et al. [123] leverage a causal graph depicting the causal relationship between features to assign Shapley values respecting the causal order where source variables are attributed more than the effects. While relationships may be intuitive in simpler tabular datasets, such causal relationships are unclear to humans in images [124]. For instance, the proposal by Kancheti et al. [121] to build models whose reasoning is aligned with the prior knowledge of the underlying causal structure obtained from the domain experts based on a specialized regularization scheme could not be demonstrated in any image dataset due to non-availability of causal knowledge on image pixels. In the absence of a complete causal structure existing between the pixels, which are the input features of images, Watson et al. [125] suggest using eye-gaze data as a proxy for ground truth causal structure, which can guide the model training to avoid picking up spurious correlations. Though the inter-dependencies between image pixels are less intuitive to humans, inter-dependencies at the level of concepts are known. For example, the presence of a car can be ascertained only when it has wheels. The detection of a concept car causes an increase in confidence in the detection of the concept of wheels [126]. Qin et al. [127] propose a causal interventional training to incorporate such causal concept relationships. Bahadori & Heckerman [120] propose using instrumental variables to debias concept representations learned by Concept Bottleneck Models [91]; thereby, the effect of confounding or correlational concepts on the prediction is mitigated. Dash et al. [122] propose leveraging the causal structure to uncover biases learned by a CNN by generating suitable counterfactuals, which can then be used to retrain the CNN in a regularized manner to debias the CNN. Singla et al. [128] leverage vision-language models to associate concept descriptions to image regions and estimate the causal relationships captured by the trained model by observing the effect of intervening the concept. Yang et al. [117] and Goyal et al. [129] propose a specialized variational autoencoder to facilitate concept-level intervention. Panda et al. [118] hypothesized that the most sparse and class discriminant features are causal and leverage a neural network to determine those causal superpixels that maximize the mutual information. However, it is to be noted that these architectures are, again, black boxes whose working needs to be explained, adding up to the problem at hand of explaining the CNN of interest. To eliminate the introduction of another black box to provide a causal explanation, Causal CAM [119] echoes the hypothesis of Panda et al. [118] that the class discriminant features are causal by eliminating the context features that are salient for other classes from the saliency maps generated by Grad-CAM [21], thereby yielding a saliency map highlighting the causal features. However, as noted in the paper, this approach cannot be scaled to a multi-class classification scenario as it involves enumerating all possible subsets of the set of all class labels except the class of interest to estimate the context features, whose computation grows exponentially.

2.4 Explaining Cross-domain Classification

Much effort of the XAI research community is towards explaining classifiers trained and tested on the data sampled from the same underlying distribution, called the in-domain classifiers. Cross-domain classification also plays an important role in extending the fruits of the data-hungry deep models to be reaped for data-scarce applications by adapting the models trained using large amounts of other related data to work on the scarce data sampled from a different distribution. Specifically, domain adaptation refers to the process of adapting a model trained on a data-rich source domain to a data-scarce target domain where the distributions of the data may be different [130]. In this context, explainability can help understand how the model adapts to the differences in the source and target domains.

Zunino et al. [131] propose to leverage explainability approaches [21] to identify common features across both domains. Once the domain-invariant features are identified, the CNN is enforced to focus on these features to classify the instances. This, by design, forces the CNN to pay attention to discriminative domain-invariant features; thereby, the model would be accurate on any domain, and hence a domain-generalized classifier is built.

Szabó et al. [132] explores the temporal process of transfer learning. An Imagenet [133] trained model is adapted to perform a face recognition task, and the features encoded by the different filters of the CNN are analyzed using Activation Maximization (AM) [134] that performs gradient ascent in the input image so that the activation of a desired neuron of interest gets maximized. It was observed that the initial layers only adjust trivial features like color-space etc., to adapt to the target domain, while the latter layer filters undergo significant transformation. However, interpreting the results of AM requires expertise and may not be suited to explain to people with good domain expertise but limited deep learning expertise, as the optimization process of AM may generate perturbed pixels from which abstracting the underlying concept as similar to how humans process images [50] is challenging. Neyshabur et al. [135] perform a detailed analysis to unearth the role of feature reuse and pretrained weights during the process of fine-tuning.

Zhang et al. [136] extend the idea of Li et al. [102] to learn characteristic source domain prototypes whose similarity would determine the class of the given test instance. They propose building an unsupervised domain-adapted classifier with case-based reasoning ability incorporated by design. As no labeled target domain instances exist in unsupervised domain adaptation, the classifier is trained using the source domain instances sampled from the same distribution from where prototypes are learned. To instill domain invariance, GAN-based domain adaptation mechanisms [137, 138] are employed to generate domain-invariant features so that the target domain test instances may be classified using the same classifier, which was trained to classify the labeled source domain instances based on proximity to prototypes. A main drawback of this approach is that the prototypes are complete images, unlike recent antehoc approaches [36, 38] that offer part-level explanations. Hence, this framework needs to use the framework proposed by Nauta

et al. [54] as an add-on to obtain finer information regarding the prototypes.
Xiao et al. [139] attempt to build a posthoc approximator for an unsupervised domain adapted classifier based on ProtoPNet [36] whose prototypes are learned using the labelled source domain instances which when visualized through the unlabelled target domain instances reveals the mapping between the source and target domain instances leveraged to classify the unlabelled target domain instances. However, this approach has challenges regarding the fidelity of the explanation as there is no consensus regarding assessing the correctness of how the features are aligned across the source and target domains. Furthermore, other frameworks [54] have to be applied to get additional information on what is encoded by the class-specific prototypes learned from the source domain instances. Overall, concept-based explanations which are closely aligned with the human-friendly manner of processing images [32, 50] have to be extended to support automatic extraction of concepts learned by an already deployed CNN to circumvent the possible loss of faithfulness due to distribution shift [35]. Similarly, the fruits of concept-based explanatory approaches need to be extended to other learning paradigms which have contributed to deep learning. These are the critical loopholes one can identify from the survey of related literature discussed in this chapter. The thesis shall propose three novel frameworks to address them in the next three chapters. Specifically, in line with extending the explainability techniques to the allied learning paradigms, domain adaptation is of interest. Domain adaptation techniques aim to leverage a classifier trained on huge volumes of data on related data, which is scarce, by bridging the distribution differences between them. The process of adaptation, specifically how different features of the source domain are adapted to be reused on the data of interest, is unclear and needs to be unearthed. To achieve this, the thesis proposes an antehoc domain adapted classifier that uses a case-based reasoning pipeline to predict its instances which has been detailed in Chapter 5. While this is a baby step taken to unravel the working of allied learning paradigms, there can be many possible extensions that are out of the scope of this thesis and shall be suggested as a future avenue for researchers interested in exploring the field of explainability.

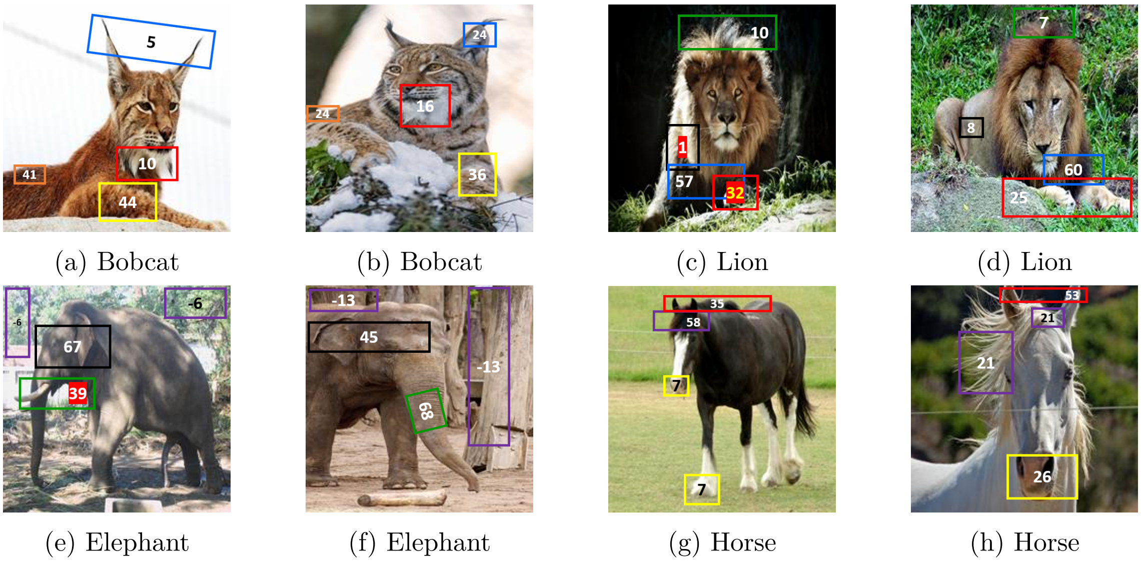
Chapter 3

Posthoc Class-specific Automatic Concept Extractors

The previous chapters motivated the need for explaining a CNN and discussed various attempts by the XAI research community to achieve this goal. This chapter proposes Posthoc Architecture-agnostic Concept Extractor, abbreviated as PACE, a posthoc explanatory framework that can automatically extract concepts from the data to explain a prediction. The extracted concepts are enforced to be class discriminative and relevant to the prediction. The concept extraction process occurs in four steps, namely embedding extraction, concept mining, prediction approximation, and relevance estimation. In the embedding extraction step, the features obtained from the individual instances are projected onto a low-dimensional latent space. The concept mining step clusters the projected embeddings subject to certain constraints to obtain class-specific concept vectors, which are abstract vectors in the low dimensional latent space expected to encode the salient aspects that identify a class. These concepts form the explanatory backbone of the proposed framework. It is important that the learned explanatory backbone faithfully captures the working of the CNN being explained. To achieve this, the feature embeddings at spatial locations where concepts are strongly observed are replaced by the dominant concepts and then reprojected back to the classifier. These reprojected features are enforced to yield a prediction probability distribution as close as possible to that of the original distribution obtained by the CNN. Thus faithfulness of the posthoc explanation is embedded in the explainer learning pipeline by design as its prediction approximation involves querying the black box, thereby guaranteeing faithfulness. In the final step, the relevance of the concepts is estimated by mimicking the effect of its perturbation on prediction probability. The PACE framework has been used to generate explanations for two different CNN architectures trained for classifying the AWA2 and Imagenet-Birds datasets. Extensive human subject experiments are conducted to validate the human interpretability and consistency of the explanations extracted by PACE. The results from these experiments suggest that over 71% of the concepts extracted by PACE are human-interpretable.

3.1 Introduction

Humans recognize an object through its different salient features [36, 103]. For instance, an elephant is identified based on the presence of characteristic features like face, ears, trunk,



(a) Bobcat (b) Bobcat (c) Lion (d) Lion

(e) Elephant (f) Elephant (g) Horse (h) Horse

Figure 3.1: Class-specific concepts extracted by the model from test images of different classes from the AwA2 dataset with their percentage contribution in the box

tusk, etc that define it. PACE aims to mimic this style of reasoning for explaining the behavior of a black box image classification model by extracting smaller salient regions in the given image called concepts, which a black box classifier deems relevant for the prediction. Ideally, a concept can be any human interpretable feature/image region, say, legs of a lion, stripes of a tiger, body texture of a leopard, background information such as the presence of water, grass, etc. Few concepts extracted from some of the test images are shown in Figure 3.1. As can be seen, the concepts represent salient parts of the different animals such as ears of the bobcat, mane of the lion, trunk of the elephant, mouth of the horse, etc.

The PACE framework assumes that every class can be explained by the presence (or absence) of certain characteristics - the concepts. The concepts, represented as vectors in a latent space, are global in the sense that they cater to the explanation of a class as a whole. Simultaneously, every input image has different manifestations of the concept vectors - named embedding vectors. The embedding vectors are extracted through an encoder that works on the feature maps obtained from the black box. The similarity between the embedding and concept vectors determines the presence of a concept. Image regions with a strong presence of the concept aid in its visualization. The embeddings are learned such that the output (classification probabilities) of the black-box model for each of the classes is preserved on passing the reconstructed feature map. The relevance of the embedding (and thereby the concept) is obtained by mimicking its removal and observing the drop in the classification probability. This definition of relevance incorporates the faithfulness of the explainer to the black box by design.

To explain how a test image has been classified, PACE highlights the salient concepts and provides relevance, denoting the concepts' contribution toward the prediction. The relevance values lie in the range $[-1, 1]$. A positive relevance indicates that the concept supports the prediction, and a negative relevance denotes that the concept's presence inhibits the prediction. The relevances are normalized, and the percentage contribution of the different concepts towards the prediction of various test images has also been shown in Figure 3.1. For instance, consider the elephant's image shown in Figure 3.1e. The concept face has a contribution of 67%; the trunk has a contribution of 39%. These concepts support the prediction of the image as an elephant. At the same time, the concept of trees has a negative contribution (-6%). This can be understood as trees may be present in the background of different animals. Hence, the presence of trees may not support the prediction of the animal. Due to the presence of trunk and face that strongly supports the animal being predicted as an elephant, the given test image was predicted as an elephant.

3.2 Related Work

While a detailed discussion of the various attempts of the research community in explaining the working of a CNN can be found in the previous chapter, this section draws attention to the most relevant attempts whose issues motivated the proposal of this PACE framework. Concept-based explanation approaches aim to explain the working of the black box by means of human interpretable concepts, which are vectors in the latent activation space. TCAV [32] requires the users to provide examples of concepts, while ACE [33] uses segmentation to automatically extracts concepts. A limitation of these approaches is that they formulate finding the relevance of a concept using directional derivatives, which is a weaker (linear) approximation, given the non-linearity in the network. Without the image segment assumption of concepts, Concept SHAP [53] extracts concepts in an unsupervised manner whose relevance is quantified by means of Shapley values [77]. However, there is a two-layer network involved in Concept SHAP to learn the concept embeddings, which leads to using another black box to explain the given black box. The antehoc paradigm of This looks like That [36] proposes using a convolutional encoder to learn class-specific prototypes automatically from the data, which are then linearly combined to perform classification such that the complete reasoning pipeline can be completely unearthed. However, to incorporate such explainability, the CNN has to be retrained, making this method unsuitable for explaining an already deployed classifier. A key difference of the proposed approach compared to the other previous concept-based approaches is that it learns class-specific concepts [36] in a posthoc manner. In contrast, the previous works aim to learn generic concepts for the whole dataset [102].

3.3 Contributions

The major contributions of the proposed work are:

- This is the first work that extracts relevant and discriminative class-specific concepts to explain the behavior of any black-box CNN.

- The approach tightly integrates the relevant concept extraction into the explanation learning process instead of leaving it as a post-training step.

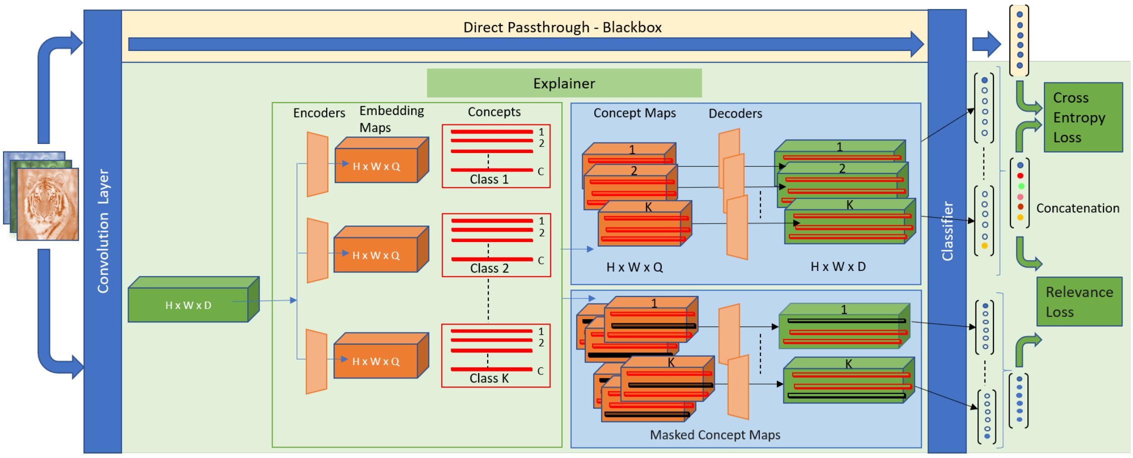


Figure 3.2: Various modules in the proposed PACE framework

- Extensive human-subject experiments are conducted to validate the consistency and interpretability of the concepts.

3.4 Methodology

PACE is capable of explaining the working of any convolutional layer. This is a grey box explainability approach in the sense that just the feature map from the layer of interest and the probability distribution, which is the output of the black box, are needed to generate explanations. The aim is to learn class-specific concepts that are integral to black box predictions.

PACE dissects the convolutional layer of a black-box model to uncover latent representations of class discriminative image regions. Figure 3.2 presents the schematic diagram of the framework, consisting of two primary components, namely, an autoencoder and the global concept representations (concepts). The encoder part of the autoencoder transforms the convolutional feature map of an input image into a representation in the space of concepts. In contrast, the decoder part of the autoencoder projects the vectors in the latent concept space back to the space of the convolutional feature map. The search for the presence of the concepts happens in the latent concept space.

The encoder is designed as a $1D$ convolutional layer that aims to project the feature map representation onto a low dimensional (Q) embedding concept space. The encoder aims to coalesce the information pertaining to different concepts spread across different feature maps into a more compact representation. The encoder is linear and thus retains interpretability - the concept representations may be interpreted as weighted combinations of the input features. Other approaches like Concept-SHAP [53] use non-linear activations, thereby reducing the explanation framework's interpretability. The decoder in the PACE framework is also designed to be a linear transpose convolutional layer transforming the vectors in the latent space to the space of feature maps. The working of the PACE explainer's modules can be interpreted in terms of weighted combinations of input features that are passed to those modules because of the use of simple 1 × 1 convolution and transposed convolution layers.

Each class k is represented by a set of C concepts that lie in the latent space of the auto-encoder, denoted by $\mathcal{C}_k = \{c_{kj}\}_{j=1}^{C}$ such that the latent representation of the same(different) concept of a class are similar(dissimilar) across different instances of that class. To explain a K-way classifier, PACE leverages K independent autoencoders, each dedicated for a class. The feature maps $F \in \mathbb{R}^{H \times W \times D}$ from the convolutional layer of interest are passed through each of the K autoencoders. H and W denote the feature maps' height and width, and D denotes the number of channels. The k^{th} encoder (parameterized by θ_k) is trained independently to learn concepts related only to the k^{th} class. The latent space for every autoencoder is different, though the dimensionality is the same.

The encoder's output for an input image $\mathbf{x}$ is an embedding map ($E_k \in \mathbb{R}^{H \times W \times Q}$). The embedding map E_k denotes the concepts' manifestation at each of the $H \times W$ locations in the feature map. Once the latent concept vectors are learned (the learning procedure will be explained later), the similarity of the Q-dimensional embedding vector at each of the $H \times W$ locations with respect to the concept vectors for the k^{th} class i.e., $\mathcal{C}_k$ can be determined. This results in C similarity matrices denoted by $S_k = \{S_{kj}\}_{j=1}^{C}$, each of dimension $H \times W$. The inverse of the Euclidean distance between the embedding vector and the concept vector is used as the similarity measure. A concept c_{kj} is present in the feature map at the spatial location (l, m) if $S_{kj}[l, m]$ exceeds a threshold τ determined relative to the maximum value. The similarity matrices can be treated as masks to visualize the concepts after suitable resizing.

The decoder (parameterized by ϕ_k) of the autoencoder for the k^{th} class works on the embedding map E_k to reconstruct the original feature map F. The concept vector should lie in the embedding manifold. Then replacing the embedding vector in E_k with the most similar concept vector at locations with the strong presence of the concept should not alter the decoder's output. This idea is used to enforce alignment between the concept vectors and the embedding manifolds, thus assisting in learning the concept vectors. Specifically, the embedding vector at a spatial location (l, m) is replaced with the most similar concept vector $c_{kj} \in \mathbb{R}^Q$ if the j^{th} concept of the k^{th} class, i.e., c_{kj} is strongly present at that spatial location. This gives the Concept Map $\tilde{E_k} \in \mathbb{R}^{H \times W \times Q}$, which is then passed through the decoder to obtain the reconstructed feature map $\hat{F_k}$ corresponding to the the k^{th} class module.

The reconstructed feature map, $\hat{F_k}$, is then passed through the rest of the black box h to get the prediction probabilities. Let p_k represent the prediction probability obtained for the k^{th} class using $\hat{F_k}$, and P the concatenation of the corresponding class probabilities obtained from all the K reconstructed feature maps. Each autoencoder (θ_k, ϕ_k) learns to detect concepts that are integral only for the k^{th} class, therefore is only reliable in explaining the output of the black-box model for the k^{th} class. By aggregating information from each pair, the class k, whose concepts are predominantly present, can be estimated. Hence the concatenation helps the explainer predict the class label based on the aggregated information about the detected class-specific concepts. According to the PACE explainer, the class label with the highest probability in P is the predicted label.

As discussed before, if the embedding and concept vectors are close, then the probability distribution P obtained via the reconstructed feature maps $\hat{F_k}$ should be similar to the classification probabilities obtained from the original feature map F. This is enforced by using a Cross-Entropy loss between P and the black-box prediction $h(\mathbf{x})$ defined as

$$\mathcal{L}_C = CrsEnt(P, h(\mathbf{x}))$$

As a result, even if the manifold of the randomly initialized concept vectors is not aligned with the embedding manifold, minimizing the above loss will eventually bring them closer. Further, to ensure that the concept vectors are different from each other, the pairwise Euclidean distance between these vectors of a single class is maximized as given below

$$\mathcal{L}_D = \sum_{k=1}^{K}\sum_{a=1}^{C}\sum_{b=1}^{C} ||c_{ka} - c_{kb}||_2^2$$

The process of extracting distinct concept vectors is reinforced by applying the triplet loss on the corresponding most similar embedding vectors. Specifically, for the instance, $\mathbf{x}_i$ in a batch of B images, the embedding vector $e_{kj}(i)$ that is most similar to the concept c_{kj} is obtained. The embedding vectors most similar to the concept c_{kj} obtained from the other images in the batch belonging to the k^{th} class form the set of anchor positives $\mathcal{P}_{kj}(i)$. Similarly, the embedding vectors most similar to the other concept vectors $\mathcal{C}_k \setminus c_{kj}$ from the images in the batch belonging to the k^{th} class form the set of anchor negatives $\mathcal{N}_{kj}(i)$. As suggested by Schroff et al. [140], all anchor-positive pairs are used, while semi-hard negatives are selected for anchor-negative pairs. The margin α is set to 1 so as to encourage orthogonal embeddings. The triplet loss is thus defined as

$$\mathcal{L}_T(i,j,k) = \sum_{e_p \in \mathcal{P}_{kj}(i)} \sum_{e_n \in \mathcal{N}_{kj}(i)} ||e_{kj}(i) - e_p||_2^2 - ||e_{kj}(i) - e_n||_2^2 + \alpha$$

The triplet loss requires a sufficient number of anchor positives to learn a good separation [140]. To ensure this, the training strategy uses a mix of pure and mixed batch instances. A batch is pure if all batch instances are predicted to be of the same class by the black box; otherwise, it is a mixed batch. It is to be noted that pure batches' formation is based on the predicted label (output from the black box CNN) and not the ground truth. This is done so that the explainer learns the functioning of the black box. A single iteration succeeds every ρ number of training iterations involving pure batches over a mixed batch. This helps to learn the interplay of concepts across different classes.

A concept's relevance is estimated by mimicking its removal and observing the drop in prediction probability. Specifically, the relevance $r_{kj} \in [-1, 1]$ for concept c_{kj} is obtained in the following manner. At all spatial locations (l, m) where c_{kj} is present, $\tilde{E_k}[l, m]$ is forced to be $= \mathbf{0}$, resulting in a masked concept map $M_{kj} \in \mathbb{R}^{H \times W \times Q}$. M_{kj} is passed through the decoder ϕ_k to get the reconstructed feature map (where the j^{th} concept is removed), and on passing that through the rest of the black box h, the final classification probability for the k^{th} class, p_{kj} is obtained. Relevance is then computed as the difference in the probabilities. i.e. $r_{kj} = p_k - p_{kj}$. A positive relevance value denotes that the concept supports the prediction of the k^{th} class. In contrast, a negative relevance value denotes that the concept inhibits the prediction of the k^{th} class. Concepts relevant to the prediction are learned by applying the Squared Error loss between the relevance and the explainer probability, defined as

$$\mathcal{L}_R = \sum_{k=1}^{K} \sum_{j=1}^{C} ||r_{kj} - p_k||_2^2$$

Thus, the overall loss for training the PACE framework is the weighted combination of these four losses defined as

$$\mathcal{L} = \beta\mathcal{L}_C + \gamma\mathcal{L}_R - \delta\mathcal{L}_D + \omega \sum_{i=1}^{B} \sum_{k=1}^{K} \sum_{j=1}^{C} \mathcal{L}_T(i, j, k)$$

This results in an end-to-end training of the PACE framework for learning $\{\theta_k, \phi_k, \mathcal{C}_k\}_{k=1}^{K}$

3.5 Experiments

The PACE framework is used to explain image classifiers trained on two different datasets - Imagenet-Birds [133] and Animals With Attributes 2 (AWA2) [141]. Imagennet [133] is a large-scale image database comprising 1000 categories of objects organized according to the Wordnet [142] hierarchy. A subset of 10 bird classes was taken from the 1000-way Imagenet dataset [133] to build the Imagenet-Birds dataset. In a similar manner, a subset of 20 classes was taken from the AWA2 dataset [141] consisting of 50 categories of animals to demonstrate the scalability of the proposed framework with the increase in the number of classes K. The classes are chosen such that each class has at least 500 images.

The PACE framework was used to explain the behavior of two different CNN architectures, namely, VGG16 and VGG19. These models were pretrained on the ImageNet dataset [133] and fine-tuned on the corresponding datasets of interest with a train, validation, and test split of 80%, 10%, and 10%, respectively. Adam optimizer [143] is used to optimize the objective in all experiments. In all the classifier fine-tuning setups, the batch size was 64; the number of train epochs was 100; the learning rate is 10^{-3}, and the regularization weight decay parameter is 5×10^{-5}. The test accuracy of the different black boxes is tabulated in Table 3.1.

The PACE explainers for the three models are trained for 100 epochs with a batch size of 32, a learning rate of 10^{-4}, and the regularization weight decay parameter is 0.1. The values of the other hyper-parameters for PACE are $C = 10$, $Q = 32$, $\tau = 95\%$, $\rho = 5$, $\beta = 100$, $\gamma = 1000$, $\delta = \omega = 1$ obtained via cross validation.

Black-box	Dataset	Test Accuracy
VGG16	Imagenet (Birds)	96.6%
VGG19	Imagenet (Birds)	97.1%
VGG16	AWA2	92.9%

Table 3.1: Classifier performance

Black-box	Dataset	PACE	Baseline
VGG16	Imagenet (Birds)	**94.7%**	67.3%
VGG19	Imagenet(Birds)	**94.1%**	70%
VGG16	AWA2	**88.2%**	51.4%

Table 3.2: Explainer agreement accuracies

3.5.1 Comparison with Principal Component Analysis (PCA) and Clustering Baseline

The proposed approach is the first to automatically extract class-specific concepts, and hence a baseline has been curated to assess its efficacy. A strong baseline to compare the proposed approach would be to cluster the representations obtained after applying Principal Component Analysis (PCA) on the feature maps. PCA replicates the linearity of the autoencoder learned by our model, and the clustering (K-means) represents the application of the triplet loss used to learn distinct concepts by the PACE framework. However, this baseline cannot automatically learn class-wise concepts, unlike PACE. This is overcome by explicitly learning the cluster centroids for each class independently using pure batches. Specifically, given a pure batch containing the images for class k, a low dimensional embedding map E_k^{BL} is obtained via PCA from the feature map F. These embeddings are clustered to get C clusters representing concept vectors for that class k denoted by $\mathcal{C}_k^{BL}$. The low dimensional embedding map E_k^{BL} (with the embedding vector replaced by the most similar cluster centroid) can be transformed to obtain the approximation to the feature map F, which in turn can be used to obtain the classification probabilities.

The % of test instances where the label as predicted by the explainer ($arg\max_k p_k(x)$) and the black-box ($arg\max_k h_k(x)$) agree is termed the agreement accuracy. These scores for the three CNN models with respect to the PACE explainer and the baseline explainer are presented in Table 3.2. It can be seen that PACE significantly outperforms the baseline in all the cases. The baseline is not included for the human subject experiments due to the low agreement accuracy.

3.5.2 Human Subject Experiments

To assess the interpretability, consistency, and relevance of the concepts extracted by the PACE model from a human point of view, human-subject experiments were conducted. PACE extracted class-specific concepts which matched with certain unique features using which one can identify animals in those classes. The objective of the survey was to get

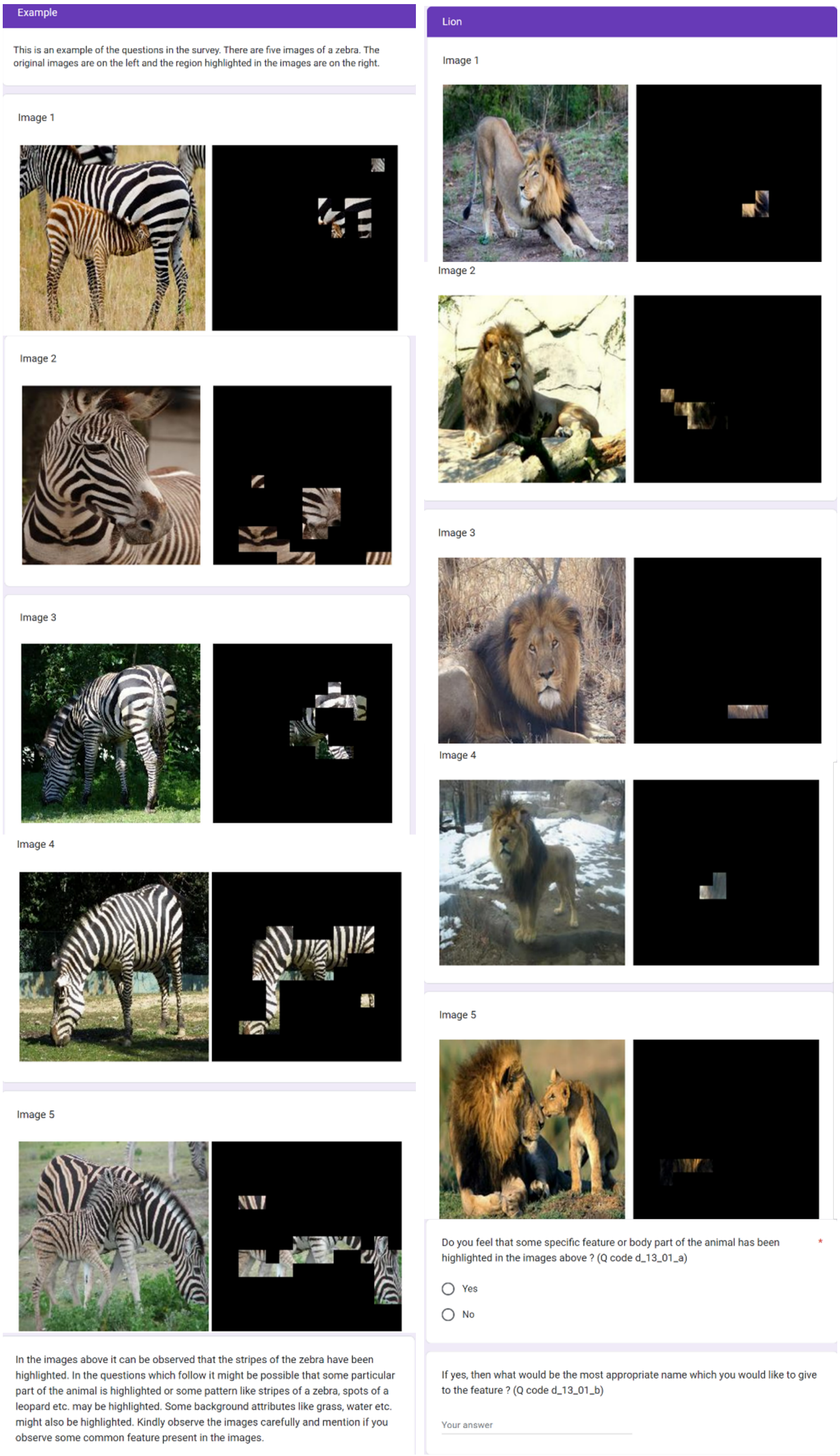
Class	**Interpretable Concept Tags**
Bobcat	Legs, Ears, Body hair, Back, Ear hair, Grass, Ear tips, Beard
Chihuahua	Ears
Elephant	Head, Trees, Ground, Eyes, Ears, Face, Trunk, Grass, Water
Gorilla	Limbs, Forehead, Grass, Wood, Trees, Head
Hippopotamus	Legs, Feet, Water, Back, Background, Sand
Horse	Mouth, Nose, Nostrils, Mane, Ears, Grass, Hair, Neck, Back
Leopard	Mouth, Grass, Trees, Spots
Lion	Lower mane, Trees, Mouth, Back skin, Head, Upper mane, Grass, Paws
Tiger	Paws, Ears, Legs, Background, White skin
Zebra	Stripes, Grass, Feet, Ground, Ears, Mouth

Table 3.3: Key concepts tagged by participants for each class

validation from a wider group about the concepts which could indeed be interpreted as distinguishing visual features of those animals and to find out the strength and consistency of the labels allocated to these concepts by the participants.

A concept-tagging experiment involving 100 subjects was conducted using the concepts extracted by the PACE framework on the VGG16 model trained for the AWA2 dataset. Participants were asked twenty unique questions. In each question pertaining to a single concept, the participant was presented with five different images from the same class having the visualization of the concept. The participants were asked if they could observe any common pattern across the five visualizations and, if so, were also asked to tag the concept. A screenshot of the interface is shown in Figure 3.3. An illustration of the task is provided to the participants before they get into the actual tagging task. Two randomly selected questions were duplicated to validate the consistency of the responses of the individual participants.

The consistency of the concepts is measured as the percentage of the participants who agreed with the presence of a common pattern across the five visualizations. The overall consistency from the experiments was observed to be 71% as seen from the pie chart in Figure 3.4a. Figure 3.4b presents the class-wise consistency of the concepts. All classes except *chihuahua* demonstrate high consistency. The concept visualizations of some concepts for this class are shown in Figure 3.6b. It throws light on a possible reason for the low consistency of the chihuahua class, as it is a domestic animal among all other animals in the chosen subset of classes. Hence the model uses the presence of household objects and dog ties, which are not found in other classes as discriminative concepts, to identify a chihuahua. This being different from how humans perceive a chihuahua might have led to rating the concepts of the chihuahua class as uninterpretable. Figure 3.5 presents the visualizations of the concepts and the tags given by the human subjects. It can be observed that the concepts are human interpretable, as the tags are meaningful. The extracted concepts indeed are some of the features of the animals and their natural surroundings according to which humans identify these animals. The tags for the different concepts across the classes labeled by the participants are shown in Table 3.3.



Example

This is an example of the questions in the survey. There are five images of a zebra. The original images are on the left and the region highlighted in the images are on the right.

In the images above it can be observed that the stripes of the zebra have been highlighted. In the questions which follow it might be possible that some particular part of the animal is highlighted or some pattern like stripes of a zebra, spots of a leopard etc. may be highlighted. Some background attributes like grass, water etc. might also be highlighted. Kindly observe the images carefully and mention if you observe some common feature present in the images.

(a) Task Illustration

Lion

Do you feel that some specific feature or body part of the animal has been highlighted in the images above ? (Q code d_13_01_a) *

○ Yes

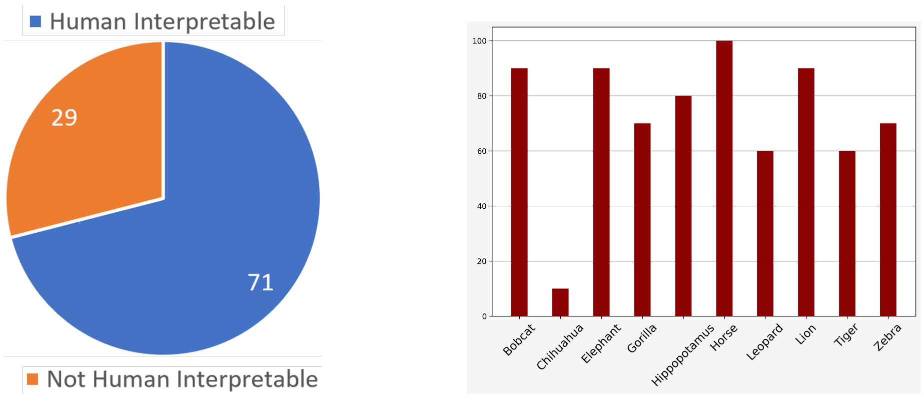
○ No

If yes, then what would be the most appropriate name which you would like to give to the feature ? (Q code d_13_01_b)

Your answer

(b) Posed question for tagging

Figure 3.3: User interface used for human subject experiments. First, the task of concept tagging is illustrated using an example. This illustration is followed by the actual task where the visualizations corresponding to a concept are shown, followed by an assessment of its stability and interpretability from the user.



(a) (b)

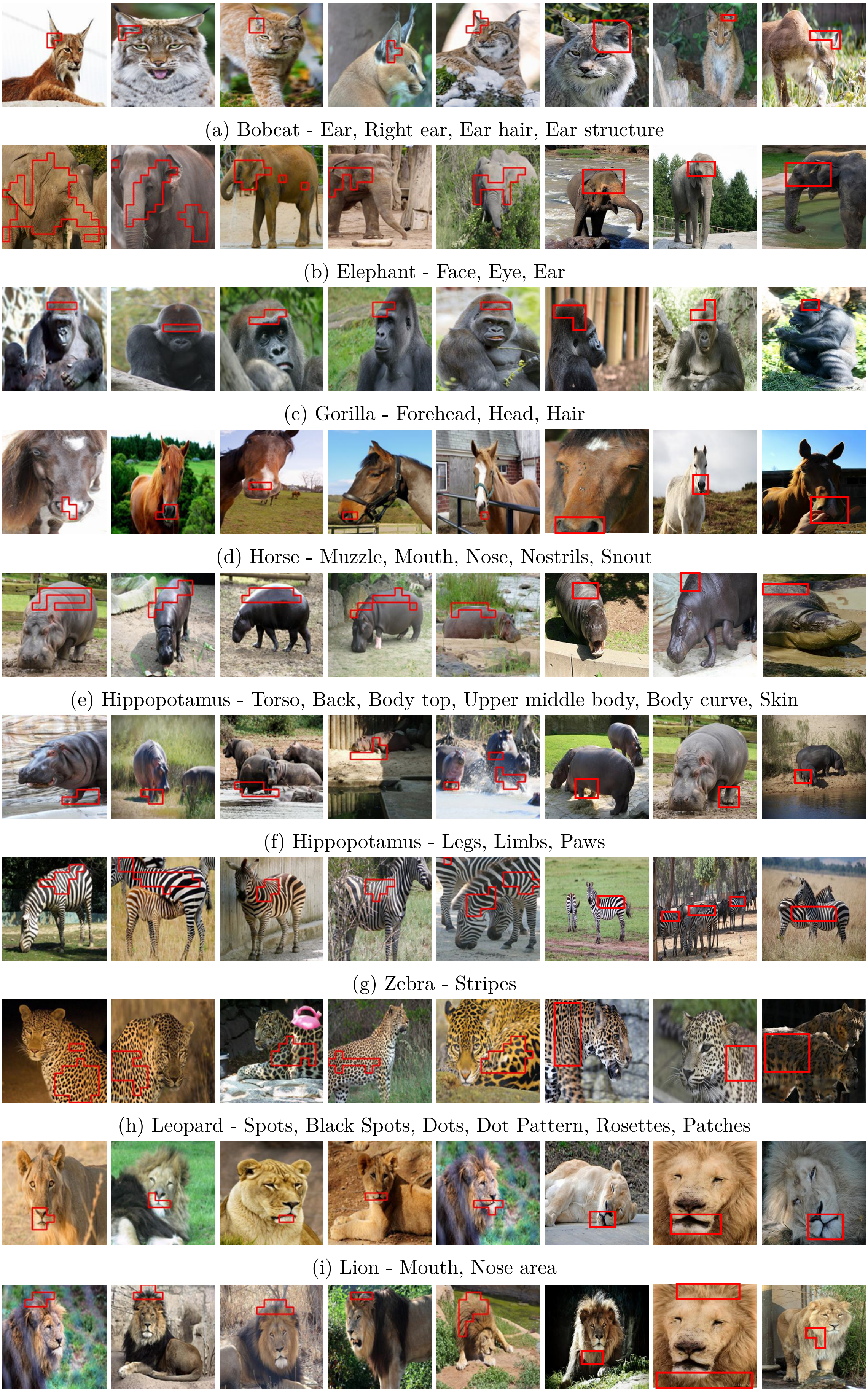
Figure 3.4: Percentage of human interpretable concepts. (a) denotes the overall distribution, and (b) denotes the classwise distribution

3.5.3 Qualitative Concept Analysis

In Figure 3.5, it can be seen that various concepts like ears of bobcat in Figure 3.5a, face of an elephant in Figure 3.5b, etc. being extracted by the PACE framework. A good visual consistency backed up by human subject votes is observed in the extracted concepts. Figure 3.5g tagged as stripes of the zebra seem to consistently highlight the stripes present in the torso region of the animal. A similar observation can be made in Figure 3.5h tagged as spot patterns, the concept highlighted consistently shows the torso of the animal. This qualitatively shows that consistent concept embeddings have been learned as expected.

A few concepts that were marked uninterpretable by the human subjects are presented in Figure 3.6. Figure 3.6a highlights sand dirt around the legs of the lion, and Figure 3.6c highlights grass around the tiger. As can be seen that the area highlighted to depict the concept itself is very small. Only participants with greater attention to detail were able to tag such concepts. The majority of the participants deemed it to be uninterpretable. The detection of uninterpretable concepts from the feature maps can be associated with the residuals extracted from the feature map during matrix factorization based explanation techniques [144]. This also proves the effectiveness of PACE that a good approximation of the internals in the feature map has been extracted through interpretable (conceptually analogous to factors in matrix factorization [144]) and uninterpretable concepts (conceptually analogous to residuals in matrix factorization [144]).

Figure 3.7 shows the concepts extracted by PACE for the VGG19 black-box model trained on the Imagenet-Birds dataset. Salient parts of the birds like the eyes of the Great Grey Owl in Figure 3.7b, its beak in Figure 3.7c, feathers of a peacock in Figure 3.7e, blue neck in Figure 3.7f, toucan's characteristic colorful bill in Figure 3.7h, the crest of Sulphur Crested Cockatoo in Figure 3.7i, etc. seems to be detected by PACE. These parts are indeed discriminatory features that help distinguish the particular bird species from other bird species. Good visual consistency can also be found in the concepts visualized across different images.



(a) Bobcat - Ear, Right ear, Ear hair, Ear structure

(b) Elephant - Face, Eye, Ear

(c) Gorilla - Forehead, Head, Hair

(d) Horse - Muzzle, Mouth, Nose, Nostrils, Snout

(e) Hippopotamus - Torso, Back, Body top, Upper middle body, Body curve, Skin

(f) Hippopotamus - Legs, Limbs, Paws

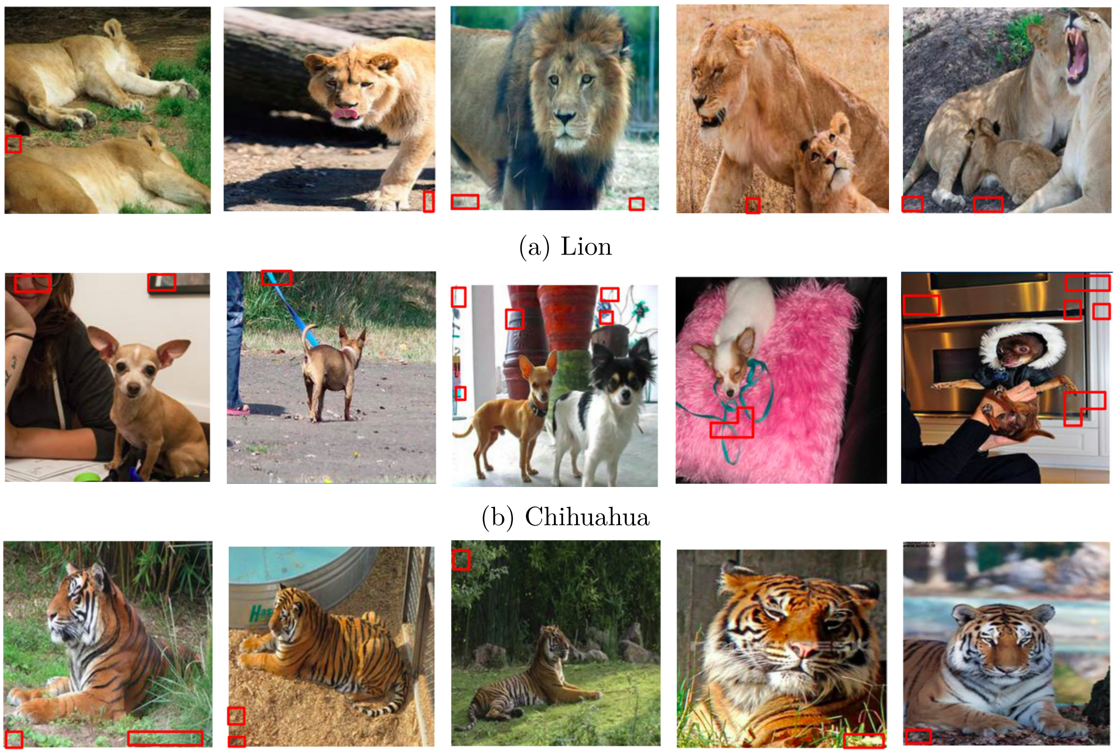
(g) Zebra - Stripes

(h) Leopard - Spots, Black Spots, Dots, Dot Pattern, Rosettes, Patches

(i) Lion - Mouth, Nose area

(j) Lion - Mane, Top of head, Forehead hair, Upper hair on face

Figure 3.5: Concept visualizations and tags given to them by the survey participants.



(a) Lion

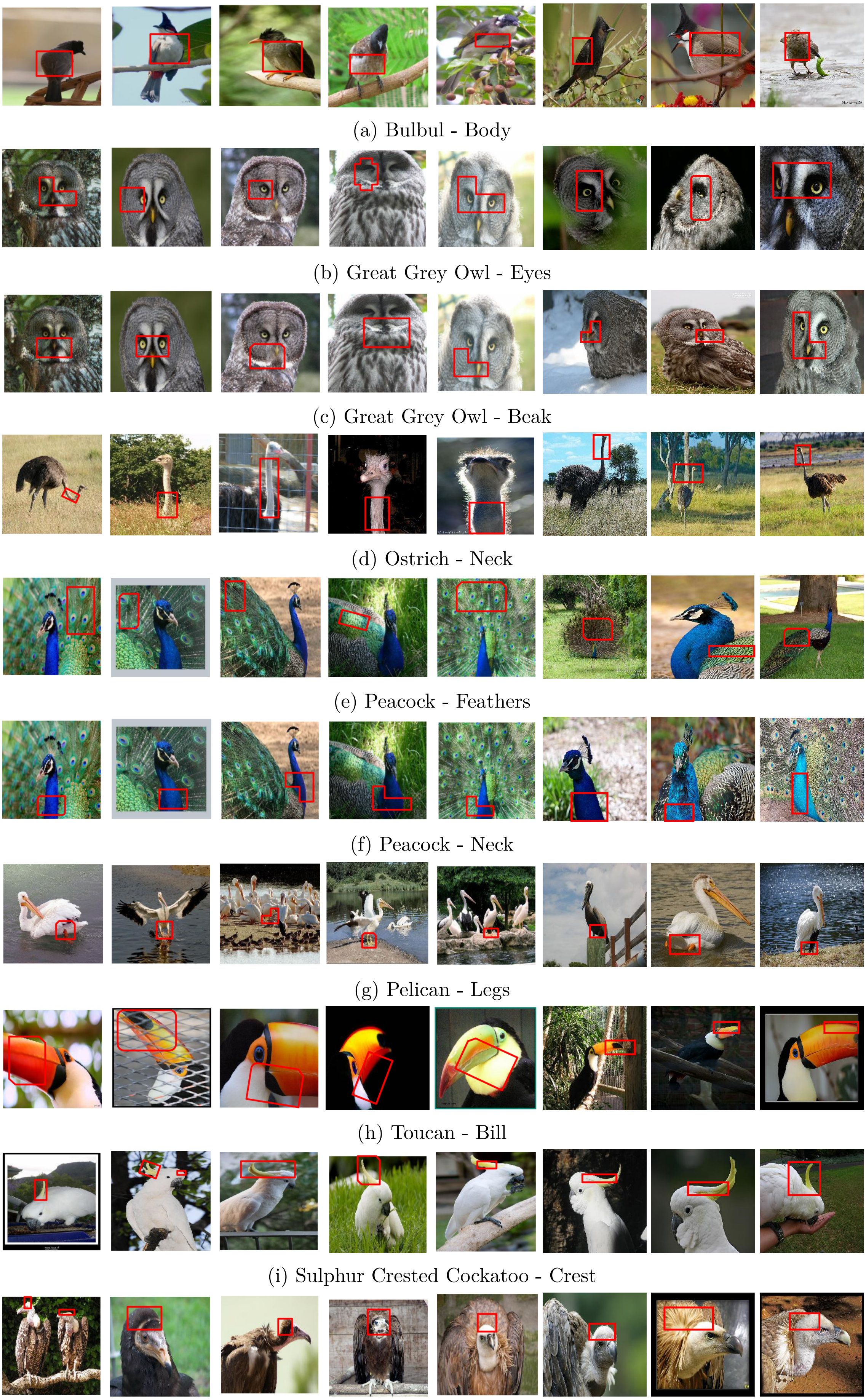
(b) Chihuahua

(c) Tiger

Figure 3.6: Examples of uninterpretable concepts

3.5.4 Explaining Misclassifications

The class-discriminative concepts learned by PACE can be used to explain black-box model misclassifications. Figure 3.8 presents a few examples of misclassified images and their topmost salient concepts extracted by PACE. Figure 3.8a shows an image of a *german shepherd* misclassified as a *hippopotamus*. Understandably, the model uses the concept of water specific to the *hippopotamus* class for this prediction. Also, it was seen that a swimming *german shepherd* as shown in the test instance in Figure 3.8a, was not found in 99% of the train instances. However, in train images corresponding to the *hippopotamus* class, around 67% of the images had a water background. This could have probably led the CNN to erroneously pick up the spurious correlation of detecting water to predict a *hippopotamus*. Similarly, Figure 3.8b shows a *collie* being misclassified as a *horse* due to high support from the concept corresponding to the mane of the horse. An observation of the train set reveals that almost 98% of the *collie* images had their facial features like the muzzle, eyes, etc., visible, unlike the test instance in Figure 3.8b where the face is covered by fur, while 99% of the train images belonging to the *horse* class had hair on its mane region visible, probably making the model associate hair to predict the given instance as a *horse*. Figure 3.8c shows a *hippopotamus* misclassified as an *elephant* due to high support from the concept corresponding to the head of the elephant. As examined already, just around 33% of the *hippopotamus*, train instances do not have a water background, probably misleading the model to associate its body features to its related class *elephant*, resulting in the misclassification.Figure 3.8d shows a *cow* misclassified as an *ox* due to high support



(a) Bulbul - Body

(b) Great Grey Owl - Eyes

(c) Great Grey Owl - Beak

(d) Ostrich - Neck

(e) Peacock - Feathers

(f) Peacock - Neck

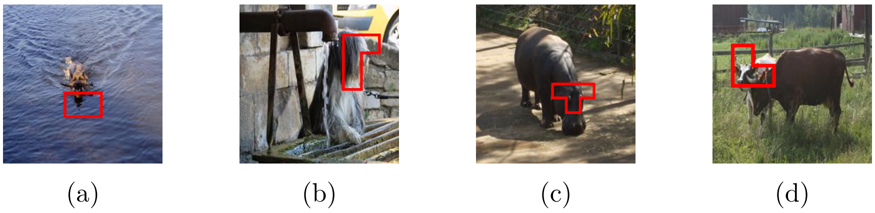
(g) Pelican - Legs

(h) Toucan - Bill

(i) Sulphur Crested Cockatoo - Crest

(j) Vulture - Head

Figure 3.7: Visualization of the concepts extracted from VGG19 model trained on Imagenet-Birds dataset



(a) (b) (c) (d)

Figure 3.8: Misclassified images - (a) German Shepherd misclassified as Hippopotamus, (b) Collie misclassified as Horse, (c) Hippopotamus misclassified as Elephant, (d) Cow misclassified as Ox

from the presence of characteristic horns of *ox*. Around 85% of the train images of *cow* did not have horns, while 98% of the *ox* images in the train set had horns. This could have made the model associate the presence of horns in the given test instance in Figure 3.8d to steer the prediction towards the *ox* class, thereby causing a misclassification. The explanations show that the model is wrong for the right reasons.

3.6 Summary

The PACE framework that learns to extract class-specific concepts relevant to the black-box prediction is proposed. The relevance is formulated such that the explanations are faithful to the black-box prediction by design. The explainer's applicability on datasets like AWA2 and Imagenet-Birds and black-box architectures like VGG16 and VGG19 has been experimented with. Qualitative and quantitative analyses show that PACE extracts concepts that are consistent and relevant. Extensive human subject experiments show that the proposed framework provides interpretable concepts.

Chapter 4

Shared Concepts Extractor

The previous chapter proposed a concept-based explainer that extracts class-specific discriminant concepts automatically from the data. While class-specific concepts can help extract the blueprints that define a class from the classifier's perspective, concepts need not be mutually exclusive in nature always. One may observe shared concepts in nature. For instance, different breeds of dogs may have common features like a muzzle, four legs, etc. Training an explainer to learn different representations for such concepts which are shared in nature is an overkill. To model this type of sharedness in the proposed framework, different concepts are encouraged to be shared across multiple classes. By doing so, one can gain insight into how these models view the concept sharedness across related classes, as often observed in the real world. With this in mind, the proposed work aims to leverage an incremental Non-negative Matrix Factorization technique to extract shared concepts in a memory-efficient manner, which reflects the sharedness of concepts across classes. Post-training, the relevance of the extracted concepts towards prediction, as well as the primitive image aspects such as color, texture, and shape encoded by the concept, is also estimated. This approach reduces training overhead and simplifies the explanation pipeline, thereby shedding light on the various concepts - some genuine, some spurious - on which the different black box architectures trained on the Imagenet dataset group and distinguish related classes.

4.1 Introduction

Concept-based explanations offer explanations close to how humans process images. The framework proposed in the previous chapter learns class-specific concepts. These concepts encompass the blueprint of a class from CNN's perspective. However, it has been observed in nature that the concepts of different classes need not always be mutually exclusive. For instance, the animals *gorilla* and *chimpanzee* share many common features in nature as they belong to the same family. The framework proposed in this chapter aims to unravel such sharedness from the lens of CNN in a posthoc manner.

Analyzing the mechanism the existing posthoc concept-based explainers leverage to learn concepts, one can find that the concept learning backbone comprises clustering and dimensionality reduction techniques. These techniques can be generalized by a mathematical framework called Matrix Factorization. As CNNs have non-linear thresholding functions that restrict the activations to the positive subspace, Non-negative Matrix Factorization (NMF) can be employed. It restricts itself to the positive subspace and can extract concepts from non-negative activations processed forward by the Convolutional Neural Networks [145].

NMF [146, 147] decomposes a non-negative matrix into two non-negative matrices such that the product is as close as possible to the original matrix. The decomposition identifies latent factors (concepts) that, when combined, result in the matrix elements or features. The feature vectors generated by the black box are assumed to be linear combinations of these concepts.

Incremental NMF [148, 149, 150, 151] is employed to extract concepts from large datasets like Imagenet [133], which is challenging with existing concept-based explanation techniques. The concepts are encouraged to be shared across classes to model commonalities. The contribution of the concept to a class is quantified through a relevance metric, assessed after concept extraction, and makes the training pipeline simple compared to existing concept-based approaches [53]. The primitive aspect based on which a concept is shared across different classes like color, shape, or texture can be assessed in terms of the effect of its perturbation [54] on concept detection. The proposed SCE framework helps determine the impact of certain spurious features [22] on prediction. Concept sharedness helps uncover the reasons for the higher performance of certain architectures.

4.2 Related Work

This section traces the most related literature that motivated the proposal of the Shared Concepts Extractor framework. Antehoc approaches incorporate explainability into the classifier from the training stage. As explainability is incorporated from scratch, it would be good if the explainable components closely reflect the real-world relationships between the classes. Looking at the gap in the existing proposals [36, 104], recent antehoc approaches [37, 38, 107] suggest the need for concepts to be shared across related classes to closely reflect the real-world concept sharedness. Nauta et al. [38] enable sharedness by design by proposing the use of a decision tree to process the detected characteristic concepts and arrive at the prediction. Rymarczyk et al. [37] highlight the possibility of negative reasoning in that framework by proposing class-specific slots that share a set of concepts. Each slot learns to capture a distribution of concepts that discriminately identify that class. While sharedness is incorporated by design in these frameworks, the SCE framework aims to incorporate such sharedness in a posthoc, human-interpretable manner.

A parallel framework proposed by Zhang et al. [144] to learn class-specific concepts like the PACE framework in the previous chapter, was based on Non-negative Matrix Factorization (NMF) taking insights from the previous studies [145, 148] that enlighten its capability to extract semantically meaningful concepts from activations. SCE intends to extract concepts that can be shared across classes, revealing the relatedness of classes from the black box's perspective. Incremental NMF [148] is employed to comfortably manage the available memory to learn the concepts shared across multiple classes, thereby preventing the need to preprocess the features to fit in the available memory [144]. Unlike previous approaches [53] that estimate relevance as part of the learning pipeline, SCE computes relevance after learning the concept extractor. Nauta et al. [54] propose a mechanism to add another layer of interpretability by unraveling the primitive aspect like color, texture, or shape that is being encoded by a learned concept representation. This helps understand if different concepts whose visualizations appear similar [36] capture the same or different aspect of that image region. This may be used as a guiding tool for the prototype pruning process, which the antehoc approaches employ to minimize the accuracy-interpretability tradeoff. Inspired by this proposal [54], SCE assesses the primitive aspects (shape, color, texture) encoded by a concept by perturbing that aspect and observing its impact on concept detection.

4.3 Contributions

The key contributions of the proposed work are:

- A matrix factorization-based approach has been proposed to extract the concepts shared across different classes from a learned CNN.
- One can faithfully estimate the relevance of these concepts towards predicting a given class after the concept extraction process using the proposed framework.
- The SCE framework is flexible to estimate the primitive image aspect encoded by a learned concept vector by observing the effect of the perturbation on concept detection.

4.4 Methodology

A Convolutional Neural Network (CNN) based object recognition model can be divided into two parts: the feature extractor and the classifier. The feature extractor, represented by f, comprises convolutional layers that extract features from the input image. The classifier, represented by h, comprises fully connected layers and uses the extracted features to predict the class label for an instance x.

In most CNN architectures, positive activations are fed forward using ReLU activation functions. Thus, this chapter proposes to use Non-negative Matrix Factorization (NMF) to identify a set of latent vectors, referred to as "concepts," represented by $\mathcal{C}$. These concepts, when linearly combined with weights $\mathcal{W}$, produce the features F. The concepts are latent vectors that can be visualized in terms of image regions to aid human understanding.

The importance of each concept in predicting the class label of an instance is determined by the impact that perturbing the combination weights has on the prediction probability. By examining the effect of perturbing a particular aspect of the image, such as color, shape, or texture, one can automatically tag the concept with information about what it represents.

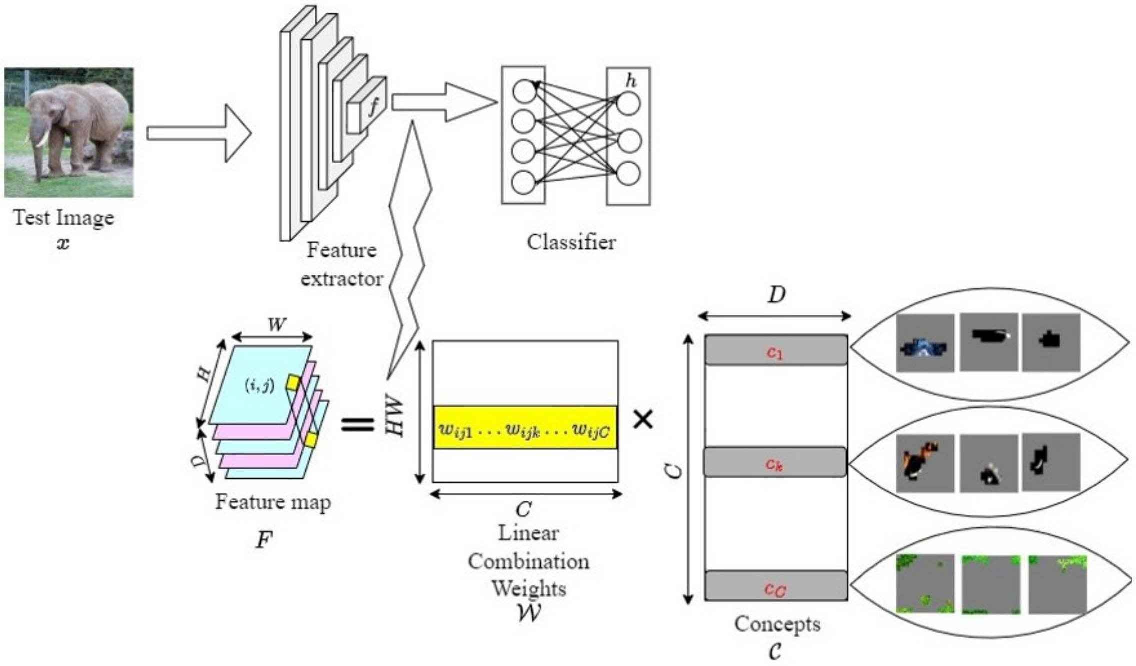


Figure 4.1: Architectural diagram illustrating concept extraction from the black box. The feature maps from the feature extractor are decomposed into a set of concepts that can be mapped to image sub-regions whose linear combination yields the features.

The subsequent sections will delve into each step of the explanation generation process in more detail.

4.4.1 Mini-batch NMF

The pipeline for our proposed algorithm is shown in Figure 4.1. The feature extractor encodes features at each spatial location as a D-dimensional vector, which is assumed to be expressed as a linear combination of C non-negative basis vectors, known as concepts. This restriction on the concepts and their weights to be non-negative enhances interpretability, as per the findings in [144].

These constraints can be mathematically satisfied using Non-negative Matrix Factorization (NMF). Let $F \in \mathbb{R}^{T \times D}$ represent the set of feature vectors. Based on the assumption, $F \approx \mathcal{W}\mathcal{C}$, where $\mathcal{C} \in \mathbb{R}^{C \times D}$ are the C basis vectors (the concepts) and $\mathcal{W} \in \mathbb{R}^{T \times C}$ are the weights of the linear combination. NMF solves the optimization problem $\mathcal{L}(\mathcal{W}, \mathcal{C}) = \min_{\mathcal{W},\mathcal{C}} ||F - \mathcal{W}\mathcal{C}||_2^2$ subject to $\mathcal{W} \geq 0$ and $\mathcal{C} \geq 0$ using the majorization-minorization principle.

The optimization problem is solved using the Multiplicative Update solver [152]. Despite the optimization being non-convex in both $\mathcal{W}$ and $\mathcal{C}$, it is convex separately for each. A block coordinate descent scheme is used, alternating between solving for $\mathcal{W}$ and $\mathcal{C}$ while keeping the other fixed.

The gradients of the objective function $\mathcal{L}$ with respect to the parameters $\theta = \{\mathcal{W}, \mathcal{C}\}$ can be deduced to be $\nabla_{\mathcal{W}}\mathcal{L}(\mathcal{W}, \mathcal{C}) = 2\mathcal{W}\mathcal{C}\mathcal{C}^T - 2F\mathcal{C}^T$ and $\nabla_{\mathcal{C}}\mathcal{L}(\mathcal{W}, \mathcal{C}) = 2\mathcal{W}^T\mathcal{W}\mathcal{C} - 2\mathcal{W}^T F$. As it can be seen that the gradients contain both positive and negative terms. Application of Vanilla Gradient descent may introduce negative elements into the decomposed matrices, violating the non-negative constraint on the parameters $\mathcal{W}$ and $\mathcal{C}$. Lee & Seung [152] suggest separating positive and negative terms from the gradient so that negative entries do not get introduced in the decomposed matrices. Let $\theta = \{\mathcal{W}, \mathcal{C}\}$ be the parameters to be learned. The update rule is given as

$$\theta \leftarrow \theta * \frac{\nabla_\theta^- \mathcal{L}(\theta)}{\nabla_\theta^+ \mathcal{L}(\theta)} \tag{4.1}$$

where, $\nabla_\theta^- \mathcal{L}(\theta)$ denotes the negative terms in the gradient and $\nabla_\theta^+ \mathcal{L}(\theta)$ denotes its positive terms.

The update rule in equation 4.1 was proposed by Lee & Seung [152], and these updates are proven to ensure that the Euclidean distance-based divergence corresponding to the optimization objective remains non-increasing, thereby converging to an optimal set of parameters.

Applying the update rule in equation 4.1 to the parameters $\theta = \{\mathcal{W}, \mathcal{C}\}$,

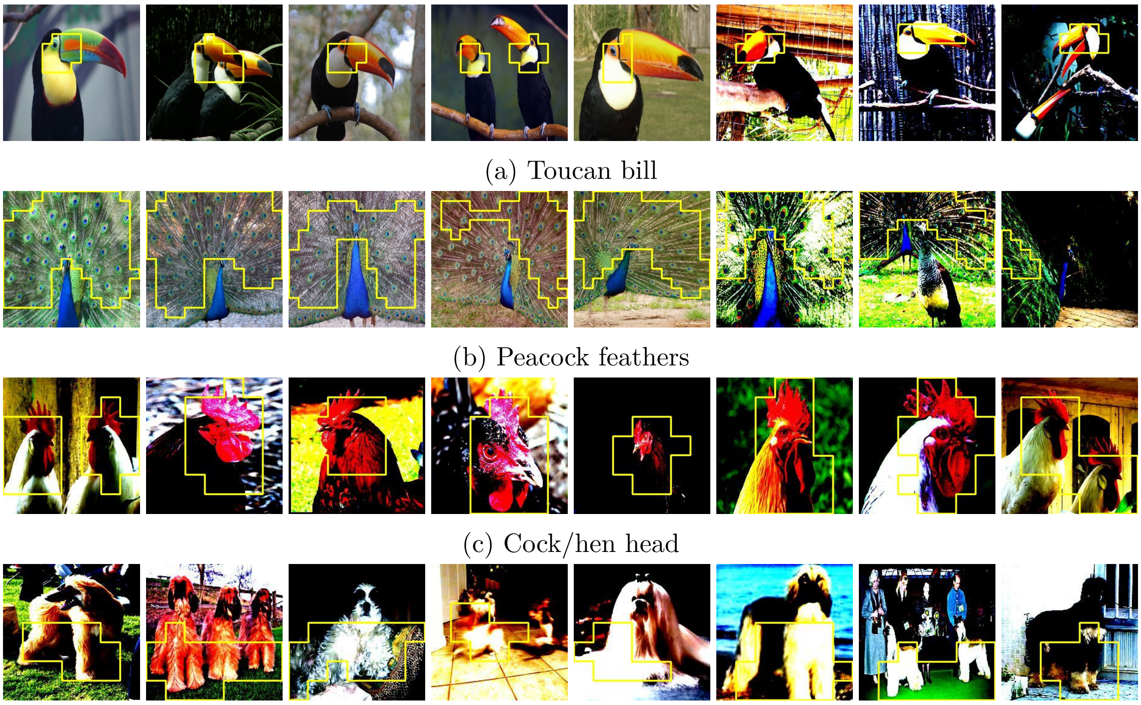
$$\mathcal{W} \leftarrow \mathcal{W} * \frac{F\mathcal{C}^T}{\mathcal{W}\mathcal{C}\mathcal{C}^T} \ \texttt{and} \ \mathcal{C} \leftarrow \mathcal{C} * \frac{\mathcal{W}^T F}{\mathcal{W}^T \mathcal{W}\mathcal{C}}$$

Traditional NMF algorithms require all instances to be presented simultaneously to learn the concepts $\mathcal{C}$. But, when the dataset is large, processing all of it at once can be challenging due to memory constraints. Hence, features are processed in mini-batches containing B instances. This does not affect updating the linear combination weights, but it does impact learning the concepts, which must be updated based on the current and past batches. This is addressed by using an online NMF algorithm [149, 150, 151]. This algorithm incrementally learns the concepts, retaining the necessary information from past instances. The modified multiplicative update equation to learn the concepts incrementally is given in [149, 150, 151] and is as follows:

$$\mathcal{C}_{t+1} = \frac{\rho\mathcal{C}_t + \mathcal{W}^T F}{\rho\mathbf{1} + \mathcal{W}^T \mathcal{W}\mathcal{C}_t}$$

where, $\mathcal{C}_t$ represents the concepts learned from the t^{th} mini-batch. ρ is a scalar called the memory parameter, which determines the extent to which concepts from previous mini-batches should be retained while updating the concepts from the current mini-batch. $\mathbf{1}$ is a unit matrix of dimensions $C \times D$. Unrolling the equation shows that the memory from the first mini-batch is carried forward, allowing the concepts $\mathcal{C}$ to abstract the information learned from the entire dataset efficiently within the memory constraints.

SCE uses the online NMF algorithm for scalability and to interpret large multi-way classifiers. The concepts are made sharable across classes by inputting mini-batches with different classes to the online NMF routine, instead of learning class-specific concepts as in previous works [144].



(a) Toucan bill

(b) Peacock feathers

(c) Cock/hen head

(d) Dog body

Figure 4.2: Visualization of a few concepts with the tags describing them collected from our human subject experiments.

4.4.2 Concept Visualizations

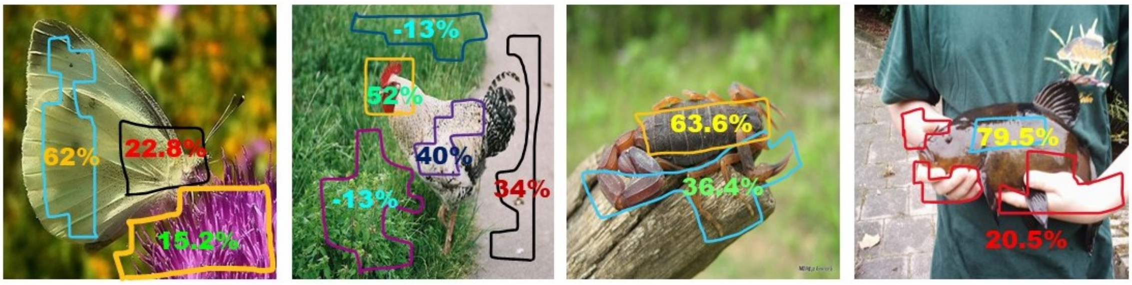
The concepts extracted in the previous step leveraging incremental NMF are abstract latent vectors. A human interpretable connotation to them is given by mapping them to image regions. Given a test image x, the first step is to extract the feature map $f(x)$ of height H and width W. The learned set of concepts $\mathcal{C}$ is kept fixed, and the weights $\mathcal{W}$ to combine the concepts to obtain $f(x)$ is estimated. Let w_{ijk} be the weight to combine the k^{th} concept c_k to obtain the feature at spatial location (i, j). To assess the extent of the presence of a concept in the image, the weights are added across all spatial locations, i.e., the presence of concept c_k in image x is given by $\zeta_k(x) = \sum_{i=1}^{H} \sum_{j=1}^{W} w_{ijk}$. The images in the held-out set where their presence is strong are visualized to understand the underlying aspect encoded in the concept. While visualizing the concept in a given image, the spatial locations with higher weights (i.e., weight exceeds a relative threshold τ) are thresholded to localize the visualization to only those regions of the image where the concept is dominant. A few example visualizations across different black box architectures are shown in Figure 4.2. Visually, it can be seen that the highlighted region depicts similar parts across different example instances, and hence the concepts are stable.

4.4.3 Concept Importance

Estimating the importance of the extracted concepts towards the prediction helps completely explain the working of a deep model using these concepts. A concept is important to predicting a given class if its perturbation significantly impacts the prediction probability. The weighted combination $\mathcal{WC}$ yields the features at every spatial location. By perturbing the weights $\mathcal{W}$, one can perturb the combination, thereby impacting the generated features. The weights that combine the concepts to generate features at those spatial locations (i, j) where the presence of a concept is dominant are scaled, thereby altering its presence. Mathematically, the perturbed linear combination weights are obtained as shown below.

$$\hat{\mathcal{W}} = \begin{cases} w_{ijk} + \delta, & \texttt{if } \; w_{ijk} \geq \tau \\ w_{ijk}, & \texttt{otherwise} \end{cases}$$

where δ is the additive factor, and τ is the relative threshold to determine the concept dominance. The features from the perturbed combination $\hat{F} = \hat{\mathcal{W}}\mathcal{C}$ are passed through the classifier h to get the prediction probability $h(\hat{x})$. Let $h(\tilde{x})$ denote the probability obtained by forward propagating the reconstructed feature map $\tilde{F} = \mathcal{WC}$ through the classifier h. The difference in these probabilities $h(\hat{x}) - h(\tilde{x})$ yields the impact of the given concept towards the prediction of different classes, which is termed relevance. A concept whose stronger presence increases the prediction probability is a supporting concept (positive relevance), and that which decreases the prediction probability is an inhibiting concept (negative relevance). The strength of inhibition and support is known by the magnitude of change in probability due to the perturbation. Figure 4.3 shows a few explanations where the different concepts that are dominantly present in the image are highlighted. The numbers within the contours show the corresponding concept relevances. Every concept $\{c_k \in \mathcal{C}\}_{k=1}^{C}$ has a presence ζ_k in the image x. Only those concepts whose presence exceeds the relative threshold τ are considered to be present in the image. It can be seen that in the *cock* image 4.3(b), the plants in the background seem to be inhibiting the prediction as class *cock*. Similarly, the supporting concepts, namely, flowers in the *cabbage butterfly* image 4.3(a), human hands in the *tench* image 4.3(d), highlight the spurious features picked up by the deep models[22].



(a) (b) (c) (d)

Figure 4.3: Sample explanations depicting concepts dominant in an image. Image (a) is a Cabbage Butterfly, (b) is a Cock, (c) is a Scorpion, and (d) is a Tench. The numbers within the contours denote the corresponding normalized concept relevance. The concepts with positive(negative) relevance support(inhibit) the prediction.

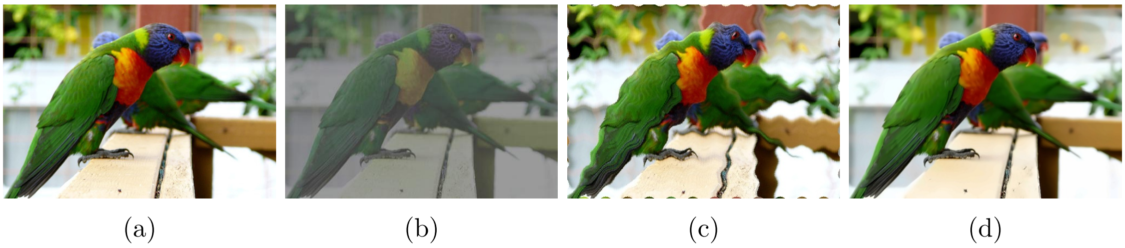


Figure 4.4: An example of image aspect perturbations. (a) shows the original image, (b) color perturbation, (c) shape perturbation and (d) texture perturbation.

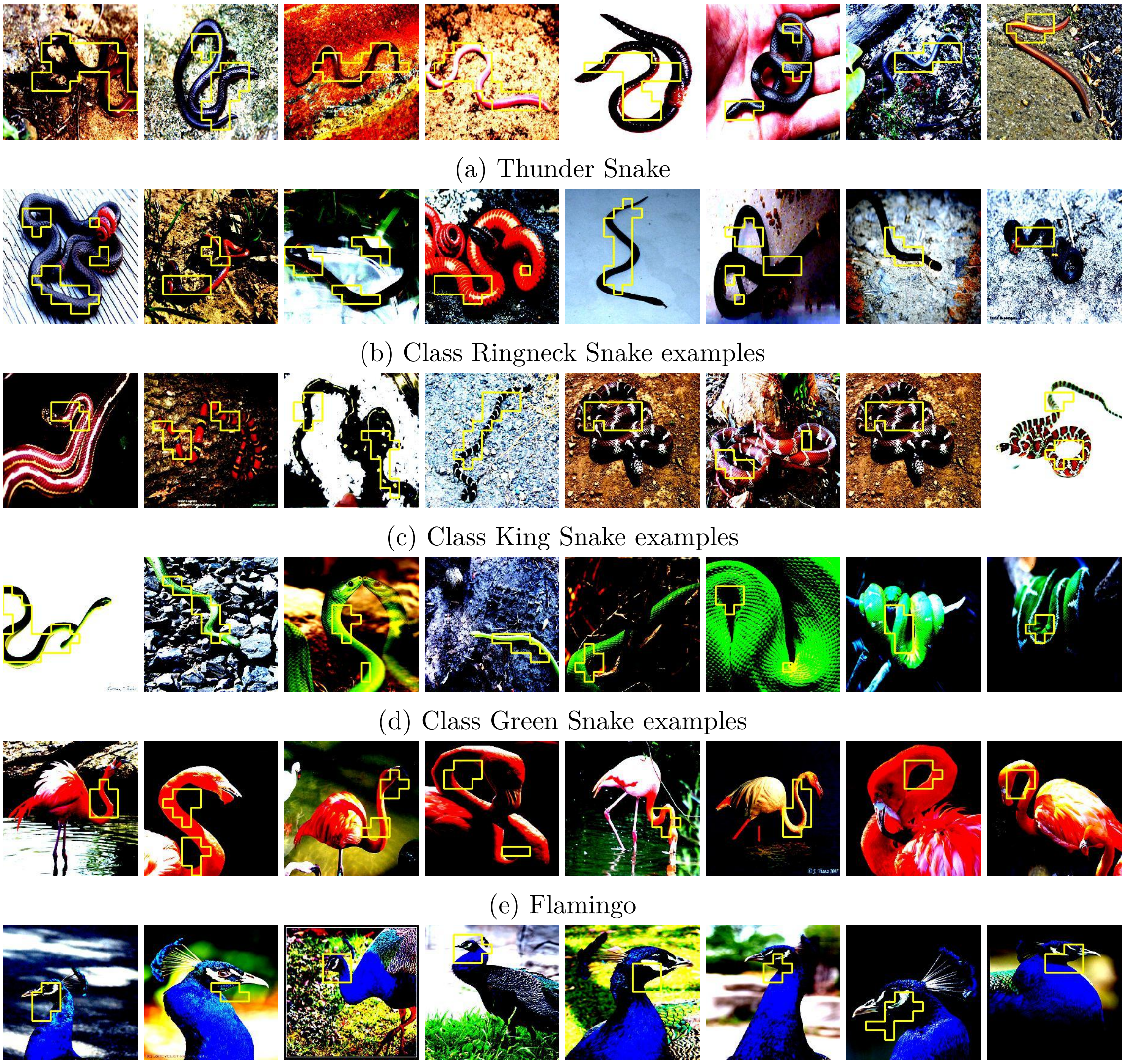
4.4.4 Associating Concepts to Image Aspects

While concept visualization may be one step to associating a human interpretable connotation to the extracted latent concept vectors, finer information regarding encoding of color, shape, or texture may not be identified from concept visualization alone. If different concept indices map onto the same image regions, previous approaches [36] suggest aggregating relevance. However, if the relevances have higher variances, this suggests additional information that could not be unraveled by concept visualization to be present in the latent concept vectors. The primitive aspect that is encoded by a latent concept vector is determined by performing image perturbations and observing the impact of these perturbations on concept detection. The aspect that causes the maximum drop in the presence of the concept is said to be encoded by the concept. For instance, if the concept c_k has a presence ζ_k in an image and the presence drops the most after color perturbation, then it can be said that the concept c_k encodes color information. This process is repeated for each concept, and the aspects encoded by different concepts are determined in this way.

The shape of the image is modified by warping it through a sine wave and shuffling the pixels. To suppress the texture, a denoising algorithm [153] is applied. Color being determined by properties like brightness, contrast, saturation, and hue, is altered by a convex combination of the image with its greyscale counterpart as suggested by Nauta et al. [54]. An example perturbation is shown in Figure 4.4. A few examples of the curved shape encoded in a concept can be seen in Figure 4.5. A concept encoding the wrinkled skin texture and another encoding the red color can be seen in Figures 4.6 and 4.7 respectively.

4.5 Experiments

The SCE framework is employed to explain two well-known CNN architectures of varying depths, namely, VGG and ResNet using the SCE framework. Due to computational resource limitations, a subset of 50 classes out of the 1000 Imagenet classes has been chosen. The hyperparameters used are the number of concepts, $C = 100$, relative concept dominance threshold $\tau = 0.5$, memory parameter $\rho = 0.7$, and the batch size $B = 128$ determined by cross-validation. The additive factor δ is scaled relative to the



(a) Thunder Snake

(b) Class Ringneck Snake examples

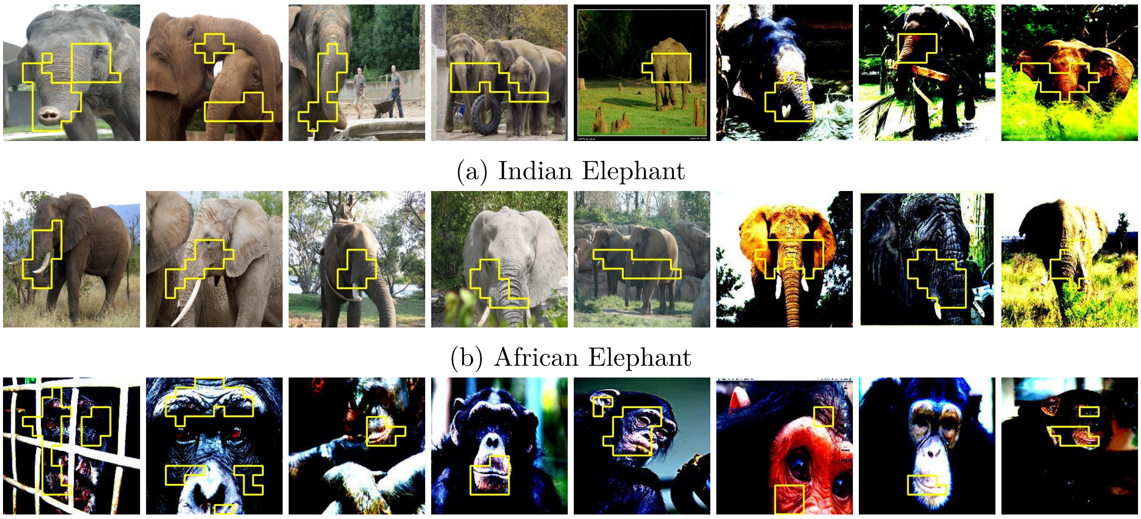
(c) Class King Snake examples

(d) Class Green Snake examples

(e) Flamingo

(f) Peacock

Figure 4.5: A concept encoding the primitive aspect - Shape. The top tags human subjects provided for this concept are - curved shape, body shape, and curved body.

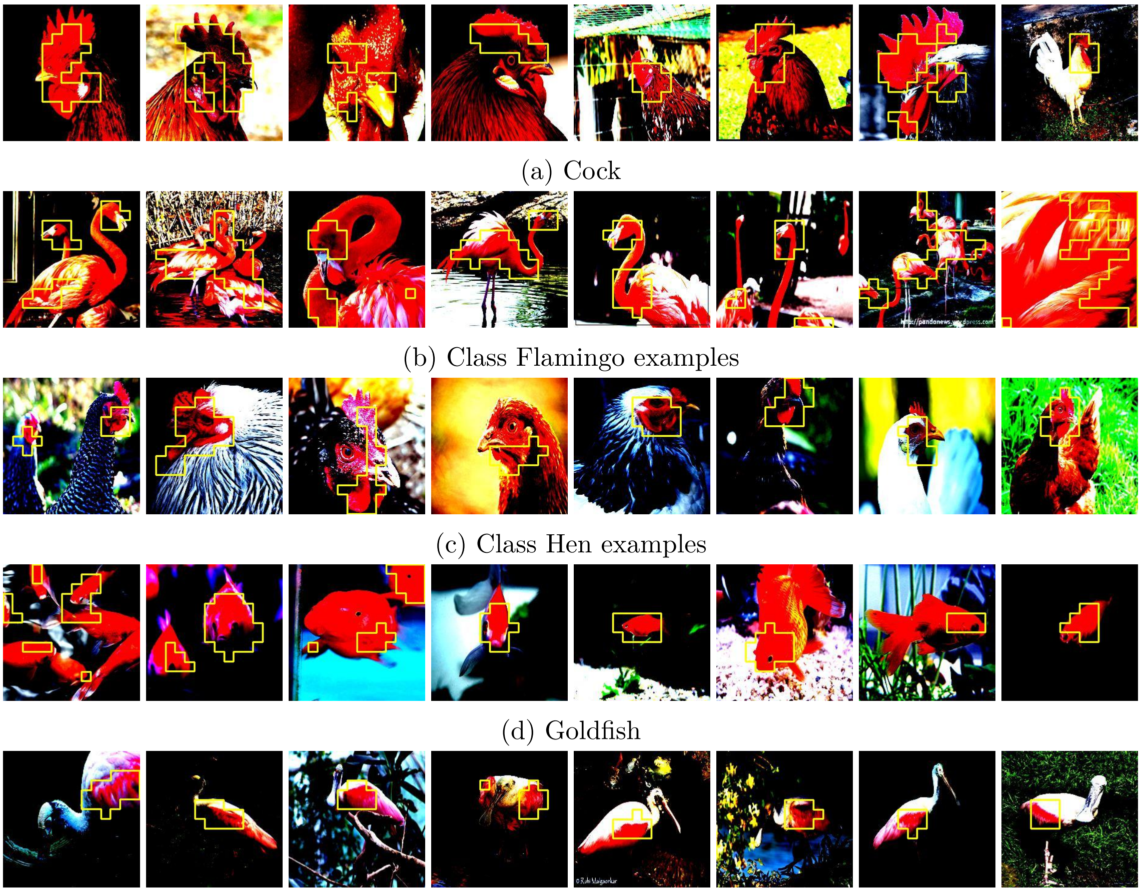


(a) Indian Elephant

(b) African Elephant

(c) Chimpanzee

Figure 4.6: A concept encoding the primitive aspect - Texture. The top tags human subjects provided for this concept are - wrinkled skin, scaled skin, and animal skin.



(a) Cock

(b) Class Flamingo examples

(c) Class Hen examples

(d) Goldfish

(e) Spoonbill

Figure 4.7: A concept encoding the primitive aspect - Color. The top tags human subjects provided for this concept are - red color, red shade, and red skin.

image-specific weights w_{ijk} as these weights can have any value, and a constant additive factor may not necessarily enhance the detection of the concept. All analyses are done on the subset of images whose explainer prediction distribution is close enough to the black box distribution. SCE is compared with PCA, a well-known dimensionality reduction technique, and a special case of matrix factorization. The principal components would yield the concepts $\mathcal{C}^{BL}$, and the projected low dimensional vectors would be the weights $\mathcal{W}^{BL}$ whose linear combination with the concepts would yield back the feature maps F.

4.5.1 Faithfulness

The explainer learns concepts that, when combined linearly, estimate the features of a given instance. The predicted distribution, $arg \max_y h(\tilde{x})$, is obtained when these estimated features are passed through the rest of the black box. The explainer is considered faithful to its underlying black box if it can regenerate the features such that the predictions on the regenerated features match the predictions of the black box, $arg \max_y h(x)$. The faithfulness is measured as the agreement accuracy, which is the percentage of instances where $arg \max_y h(\tilde{x})$ equals $arg \max_y h(x)$. This agreement accuracy is presented in Table 4.1. It can be seen that the agreement accuracies of our shared concept extractors are at par with accuracies reported on existing class-specific concept extractors [144].

Black-box	SCE Agreement Accuracy
VGG16	78.1%
VGG19	80.5%
ResNet18	83.3%
ResNet50	86.8%

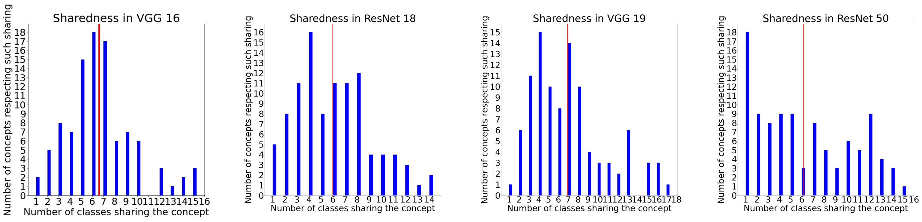
Table 4.1: Explainer agreement accuracies

The agreement accuracy seems to be affected by the model depth; however, similar concepts are observed within the same model family, while variations are seen across architectures. Therefore, for simplicity, the results from the shallowest and deepest models are discussed.

4.5.2 Concept Sharedness Across Classes in Different CNN Architectures

As SCE does not cap the number of classes that can share a concept, it is necessary to determine the classes where a concept is dominant. The average presence of concept k on images x of class y i.e., $a_{yk} = Mean(\zeta_k(x))$, is computed for all classes y. The dominant classes are included till 50% of the total dominance is achieved. The concepts are grouped based on the number of classes they share, and the distribution is plotted in Figure 4.8. The vertical red line shows the average number of classes a concept shares.

The top 8 instances ranked by concept presence are shown for different concepts in Figure 4.9.



(a) VGG16 (b) ResNet18 (c) VGG19 (d) ResNet50

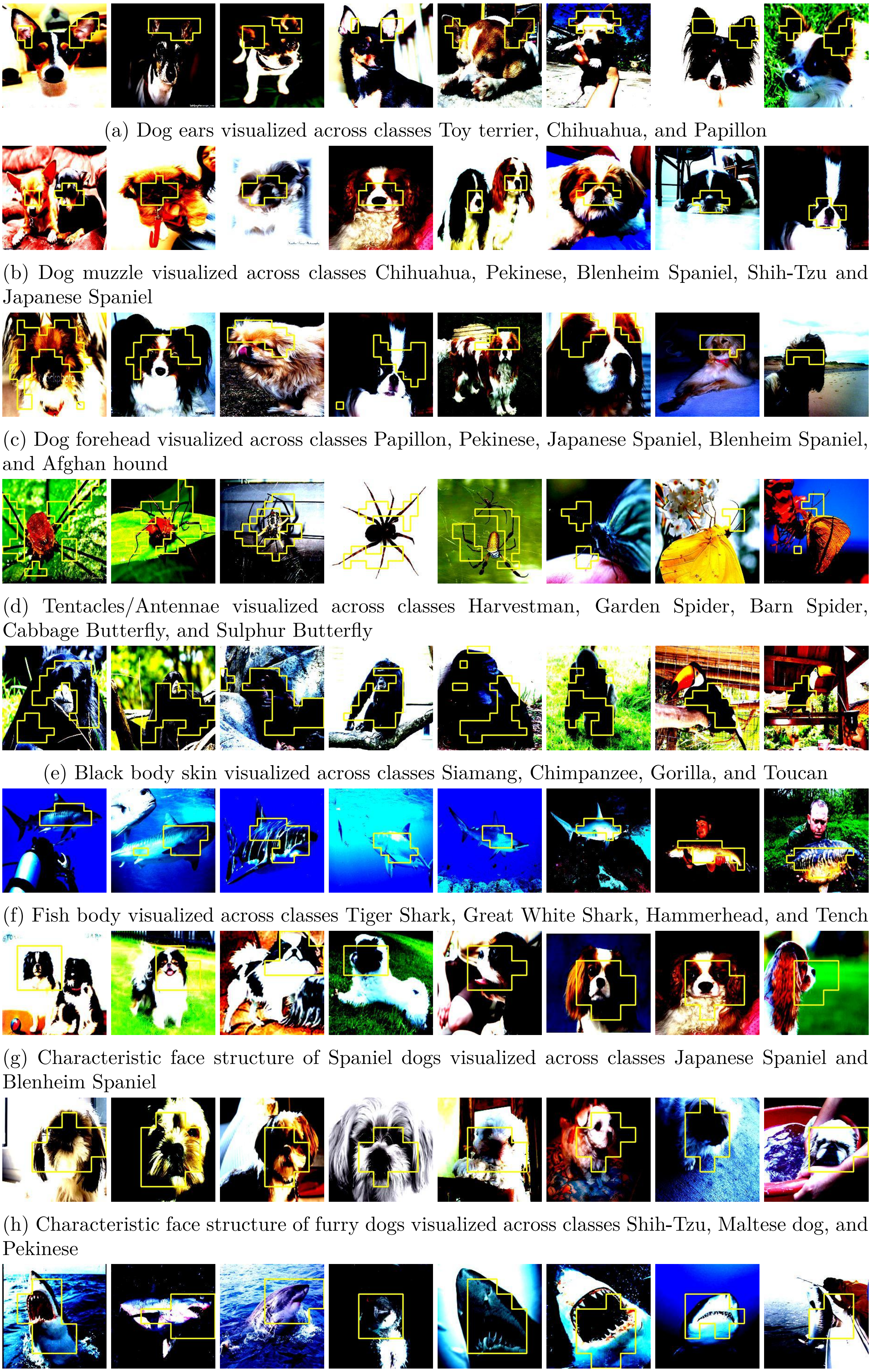
Figure 4.8: Concept sharedness exhibited by different CNN architectures shown in increasing order of depths. The vertical red line shows the average number of classes a concept shares.

In general, concepts extracted using the VGG model as the black box have smaller receptive fields, covering smaller parts such as dog ears and muzzle, as shown in Figures 4.9a and 4.9b respectively. On the other hand, the ResNet model with a larger receptive field encompasses larger image regions, such as entire dog faces. The ResNet model, which considers larger regions together, distinguishes between Spaniels (Figure 4.9g) and other furry dogs (Figure 4.9h), which may contribute to its higher performance compared to the VGG model. Unlike the VGG model, which processes all fishes similarly (Figure 4.9f), the ResNet model identifies shark teeth (Figure 4.9i) to distinguish sharks from other fishes.

Different from human intuition, SCE unravels the grouping of spiders and butterflies based on the structure of their antennae or tentacles, as seen in Figure 4.9d. Similarly, monkeys and toucan birds are grouped based on their black-colored bodies, as shown in Figure 4.9e.

Although not constrained, it can be seen that certain concepts are exclusively dominant in a single class as can be seen to be grouped under the leftmost bars from Figure 4.8. The higher performance of the ResNet model might be attributed to the detection of many distinguishing exclusive concepts, for instance, the bill structure of a spoonbill, the elongated face of an Afghan hound, etc., as shown in Figure 4.10.

While exclusive concepts are a special case, the other extreme could be a concept being dominant in many classes. The average number of classes dominant for all concepts serves as a yardstick to quantify the subjective term 'many'. Any concept dominant in more than this average number of classes is generic. This is shown by a red vertical line in Figure 4.8. The concepts which fall to the right side of this red line are termed generic concepts. These concepts mostly encode the spurious characteristic backgrounds like leaves, iron rods of a cage, human hands, flowers, water, grass, etc., as seen from Figure 4.11. In line with the observation of Neuhaus et al. [22], the SCE framework helps unearth the spurious correlations picked up by the black box, which is further ascertained by an instance of a cabbage butterfly in Figure 4.3(a) where the presence of flowers and another instance of tench in Figure 4.3(d) with human hands enhance the prediction probability contrary to the human intuition.



(a) Dog ears visualized across classes Toy terrier, Chihuahua, and Papillon

(b) Dog muzzle visualized across classes Chihuahua, Pekinese, Blenheim Spaniel, Shih-Tzu and Japanese Spaniel

(c) Dog forehead visualized across classes Papillon, Pekinese, Japanese Spaniel, Blenheim Spaniel, and Afghan hound

(d) Tentacles/Antennae visualized across classes Harvestman, Garden Spider, Barn Spider, Cabbage Butterfly, and Sulphur Butterfly

(e) Black body skin visualized across classes Siamang, Chimpanzee, Gorilla, and Toucan

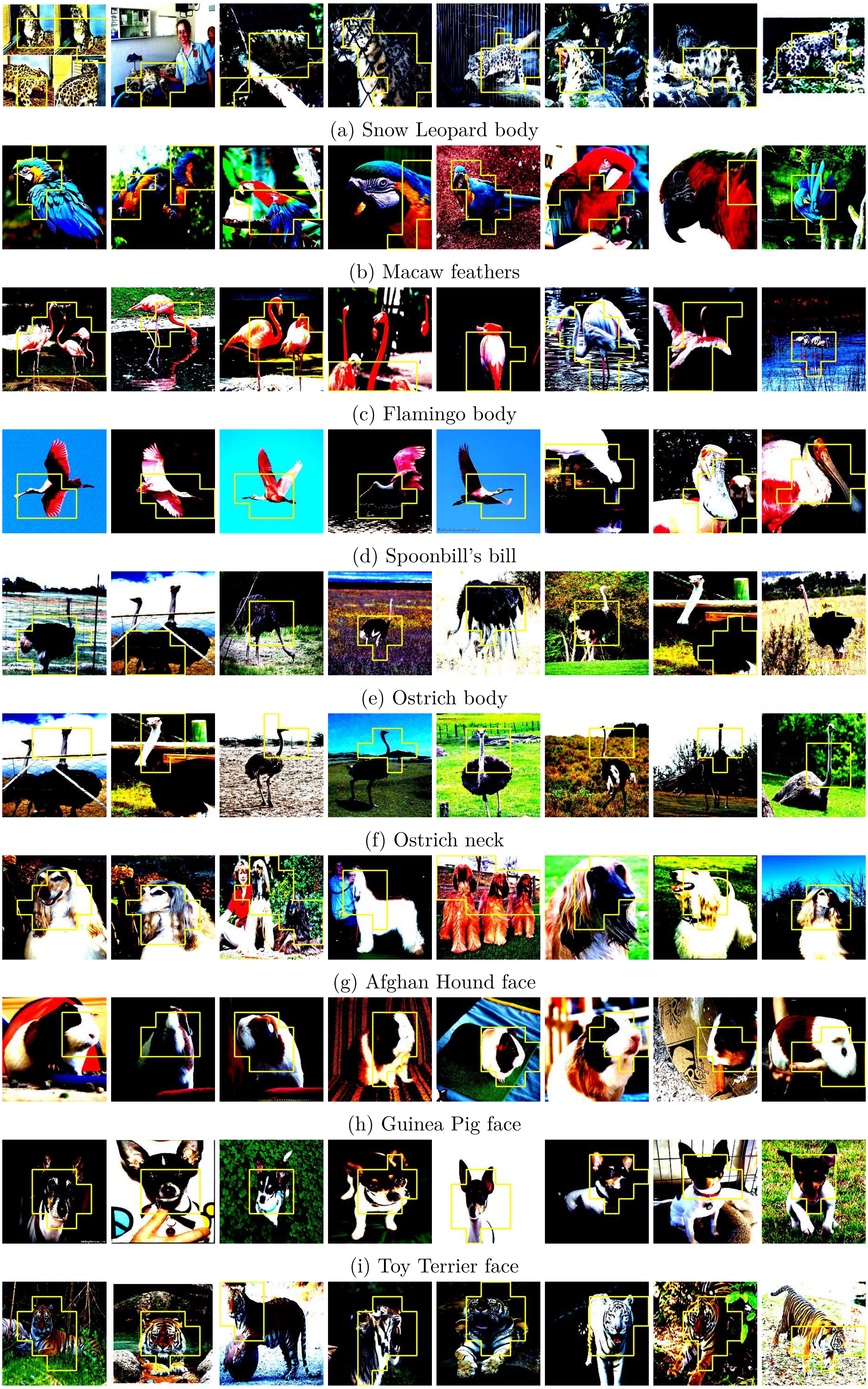
(f) Fish body visualized across classes Tiger Shark, Great White Shark, Hammerhead, and Tench

(g) Characteristic face structure of Spaniel dogs visualized across classes Japanese Spaniel and Blenheim Spaniel

(h) Characteristic face structure of furry dogs visualized across classes Shih-Tzu, Maltese dog, and Pekinese

(i) Pointed tooth structure of Sharks visualized across classes Great White Shark, and Tiger Shark

Figure 4.9: Concepts shared across different classes are shown with human-provided tags. The last three rows show concepts extracted from ResNet50, and the other rows contain concepts extracted from VGG16.



(a) Snow Leopard body

(b) Macaw feathers

(c) Flamingo body

(d) Spoonbill's bill

(e) Ostrich body

(f) Ostrich neck

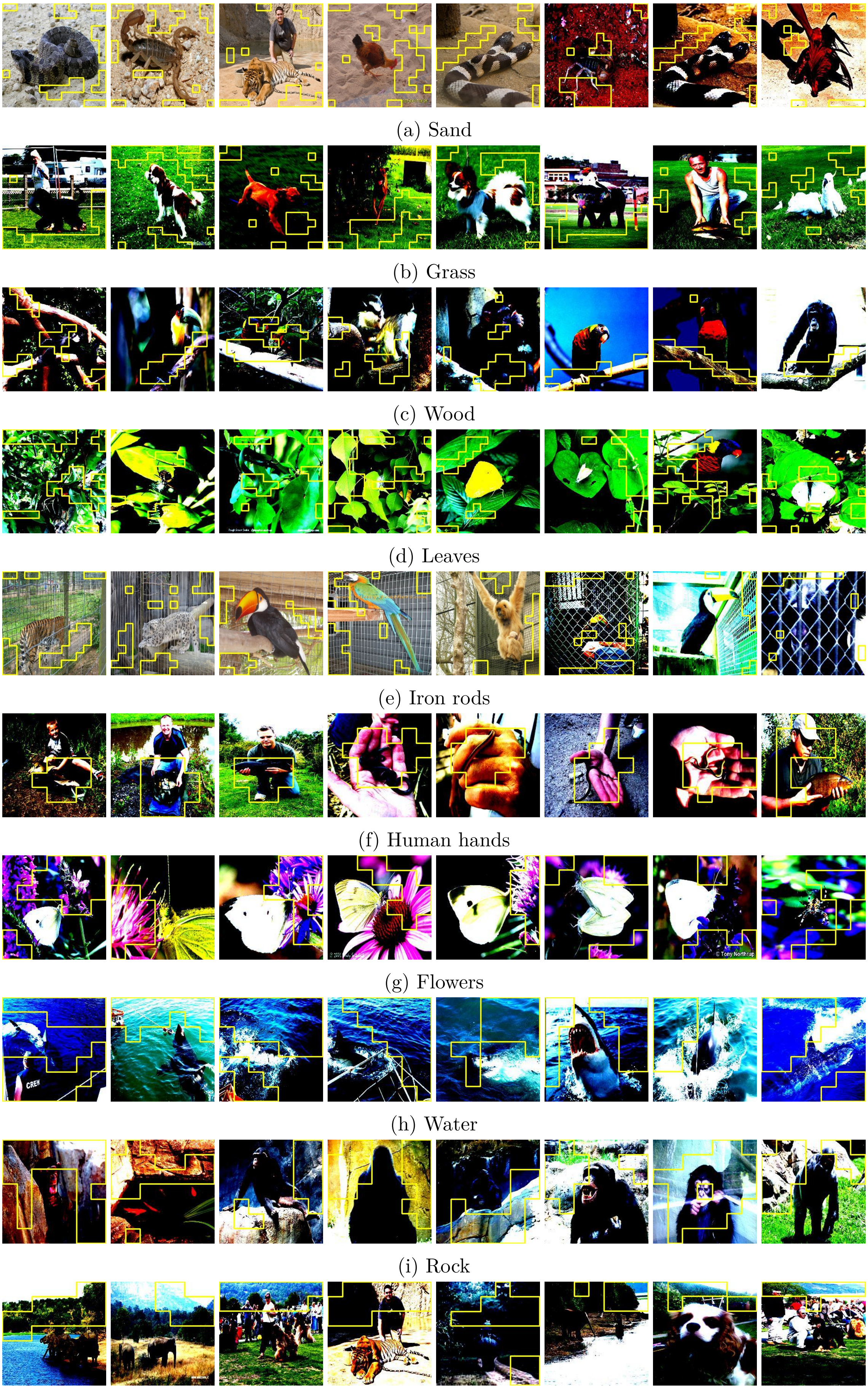
(g) Afghan Hound face

(h) Guinea Pig face

(i) Toy Terrier face

(j) Tiger face

Figure 4.10: Concepts extracted from ResNet50 that are exclusively dominant in a single class. Tags describing the concepts are collected from human subject experiments.



(a) Sand

(b) Grass

(c) Wood

(d) Leaves

(e) Iron rods

(f) Human hands

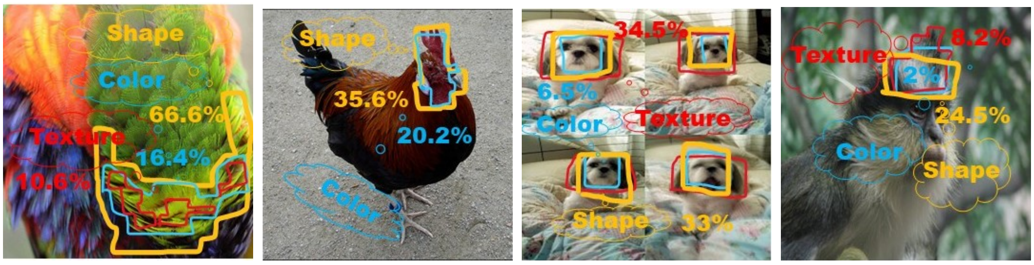
(g) Flowers

(h) Water

(i) Rock

(j) Mountain

Figure 4.11: Generic concepts shared across many classes and their descriptive tags obtained from human subject experiments. The first five rows show generic concepts extracted from VGG16, and the last five rows show generic concepts from ResNet50.



(a) (b) (c) (d)

Figure 4.12: Sample explanations where multiple concepts highlight almost the same image region. Image (a) is a Lorikeet, (b) is a Cock, (c) is a Shih-Tzu, and (d) is a Guenon. Blue, yellow, and red contours mark the concepts encoding color, shape, and texture respectively. The numbers color-coded similar to the contours denote the corresponding concept importance. For brevity, only those concepts which share the same image region are shown.

4.5.3 Analysing Concept Associations to Image Aspects

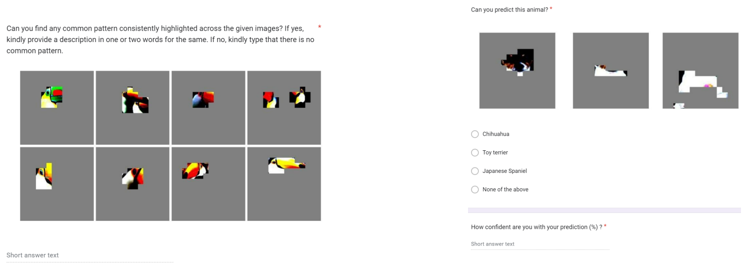
Figure 4.12 displays test instances where multiple concept representations highlight similar image regions. However, the computed importance of each concept representation, determined through perturbations, differs. This may cause confusion for users, as it does not provide insight into the inner workings of the model. To clarify the situation, the approach proposed by Nauta et al. [54] can be used to understand the aspect of the image region that is encoded by the corresponding concept representation. For example, in Figure 4.12c, the face of a Shih-Tzu is encoded by three different concept vector representations, each contributing differently to the prediction. However, investigating the concept associations with different image primitives, it becomes clear that the shape of the face region has the highest contribution to the prediction, followed by texture and color.

4.5.4 Human Subject Experiments

It is essential to evaluate the effectiveness of explanations for human understanding in practical applications. Therefore, thorough human subject experiments are conducted to compare the quality of explanations generated by our approach and a PCA baseline on two CNN architectures (VGG16 and ResNet50). Our experiments consisted of four sets of explanations, as there are two CNN architectures and two explanation methods. 100 human subjects participated in these experiments to assess the explanations' quality. The subjects were shown explanations for ten classes.

Stability & Interpretability

Ideally, regions that primarily contain a given concept should appear visually similar when they are visualized across different images having a significant presence of the concept. This property is known as stability. The user is presented with visualizations of the concepts on images from the validation set, sorted based on their presence. The user is



Can you find any common pattern consistently highlighted across the given images? If yes, kindly provide a description in one or two words for the same. If no, kindly type that there is no common pattern. *

Short answer text

Can you predict this animal? *

○ Chihuahua

○ Toy terrier

○ Japanese Spaniel

○ None of the above

How confident are you with your prediction (%) ? *

Short answer text

(a) (b)

Figure 4.13: Human subject experiments - User interface. (a) denotes the concept tagging interface, and (b) denotes the quality assessment interface

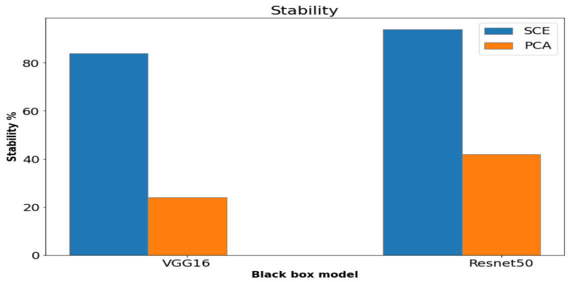
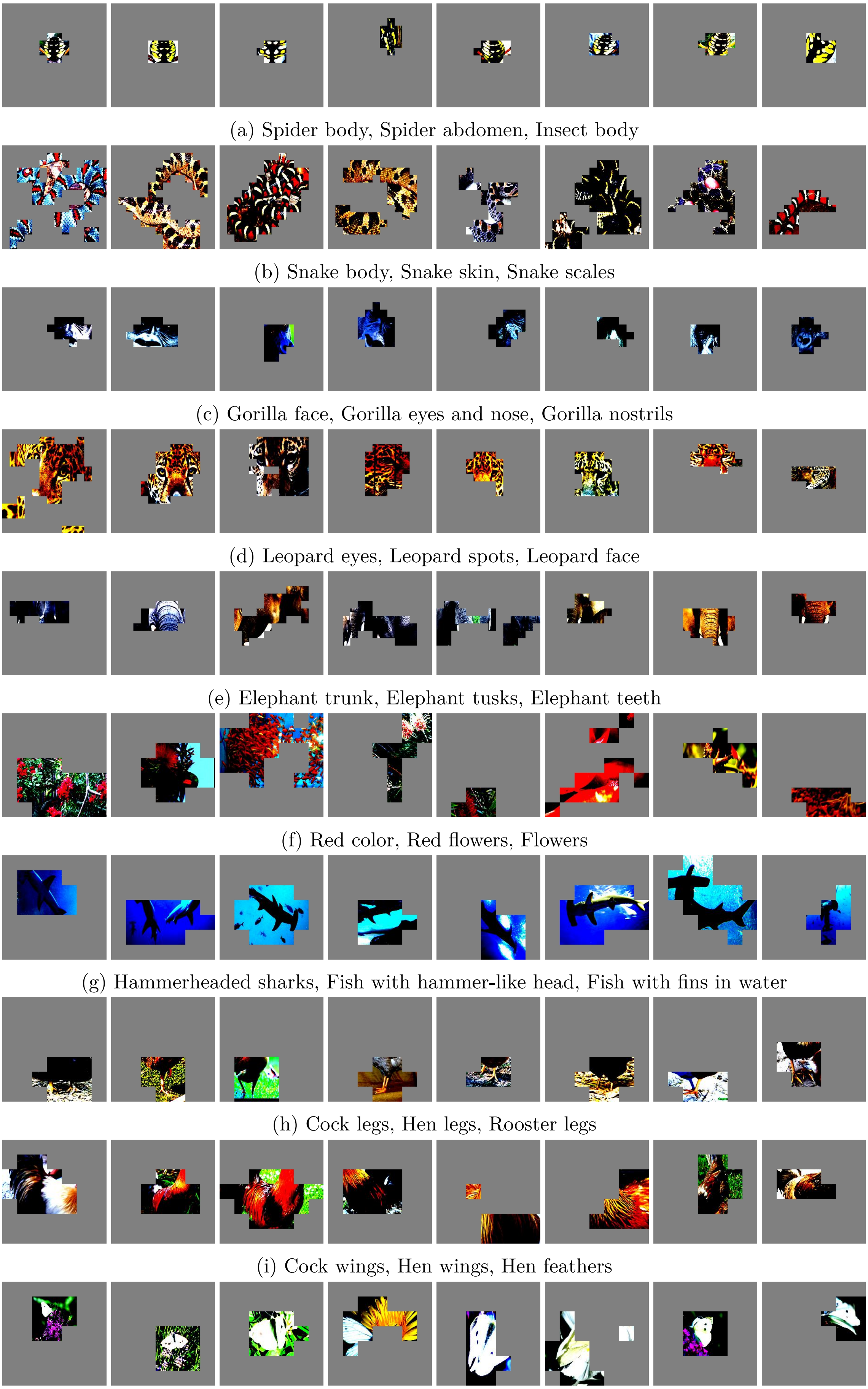


Figure 4.14: A plot of stability of concepts extracted from different architectures using different explanation algorithms

then asked to examine the visualizations and determine if a common pattern is highlighted across the images. If a common pattern is observed, the user is asked to provide a tag that describes the concept. A screenshot of the user interface for the task is shown in Figure 4.13a.

To ensure that only genuine user responses are considered while assessing stability, random insertions of repetitions of the same concept visualization were carried out, and the participants were asked to provide their answers. Participants who consistently answered at least 50% of the time across repetitions are considered genuine. Responses from other participants are disregarded. The next step is then to count the number of genuine users considering a concept unstable, meaning they cannot observe a visually common pattern. If more than 50% of the genuine participants label a concept as unstable, it is deemed unstable.

The stability of concepts across each architecture and explanation method is shown in Figure 4.14. It is observed that the concepts extracted by SCE are consistently considered stable by the participants, with a significant difference compared to the PCA baseline. For the stable concepts, the user-provided tags are analyzed to assess their interpretability. Examples of these tags can be seen in Figure 4.15. It is evident that the tags accurately describe the highlighted regions. The concepts extracted by the proposed framework that the human subjects deemed uninterpretable are shown in Figure 4.16. Similar visualizations corresponding to the PCA explainer are shown in Figures 4.17 and 4.18.



(a) Spider body, Spider abdomen, Insect body

(b) Snake body, Snake skin, Snake scales

(c) Gorilla face, Gorilla eyes and nose, Gorilla nostrils

(d) Leopard eyes, Leopard spots, Leopard face

(e) Elephant trunk, Elephant tusks, Elephant teeth

(f) Red color, Red flowers, Flowers

(g) Hammerheaded sharks, Fish with hammer-like head, Fish with fins in water

(h) Cock legs, Hen legs, Rooster legs

(i) Cock wings, Hen wings, Hen feathers

(j) Butterfly wings, Butterfly spotted wings, White Butterfly

Figure 4.15: Top tags provided by human subjects to different concepts. The first five rows show the concepts extracted from VGG16, and the last five rows show the concepts from ResNet50.

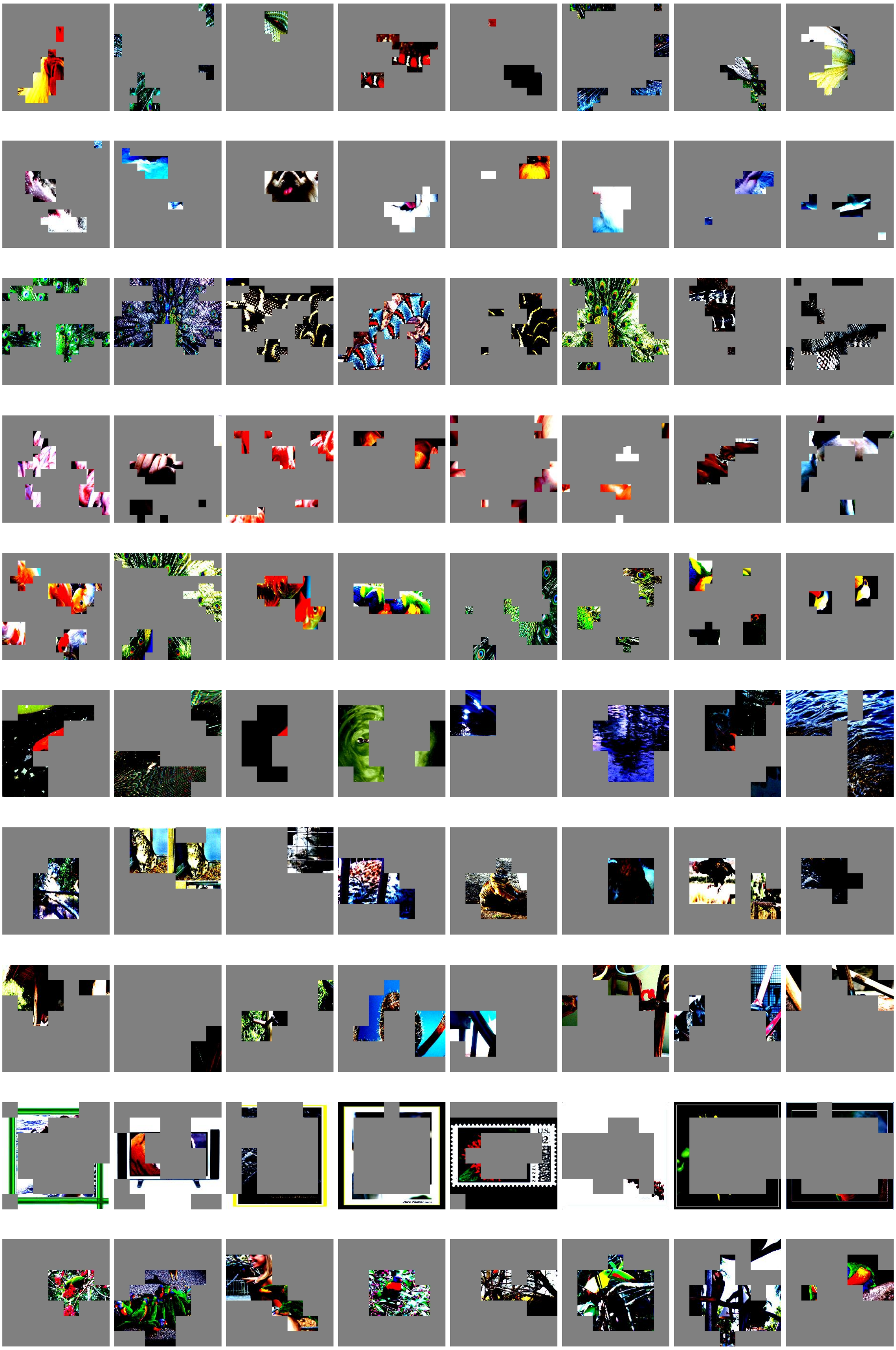
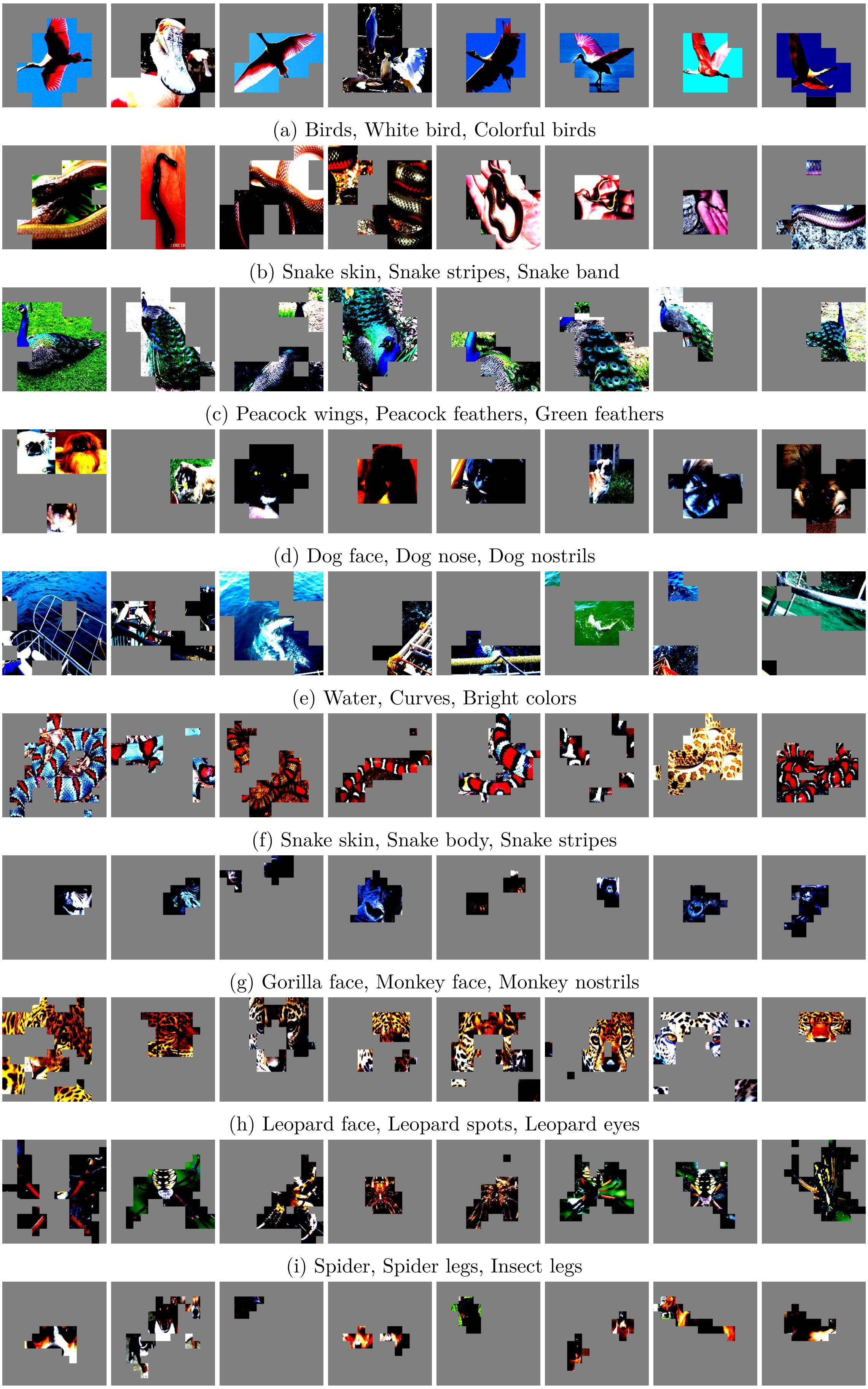


Figure 4.16: SCE concepts deemed uninterpretable by human subjects. Each row corresponds to a single concept. The first five rows show the concepts extracted from VGG16, and the last five rows show the concepts from ResNet50.



(a) Birds, White bird, Colorful birds

(b) Snake skin, Snake stripes, Snake band

(c) Peacock wings, Peacock feathers, Green feathers

(d) Dog face, Dog nose, Dog nostrils

(e) Water, Curves, Bright colors

(f) Snake skin, Snake body, Snake stripes

(g) Gorilla face, Monkey face, Monkey nostrils

(h) Leopard face, Leopard spots, Leopard eyes

(i) Spider, Spider legs, Insect legs

(j) Dog nose, Dog muzzle, Dog face

Figure 4.17: Top tags provided by human subjects to different concepts extracted by PCA. The first five rows show the concepts extracted from VGG16, and the last five rows show the concepts from ResNet50.

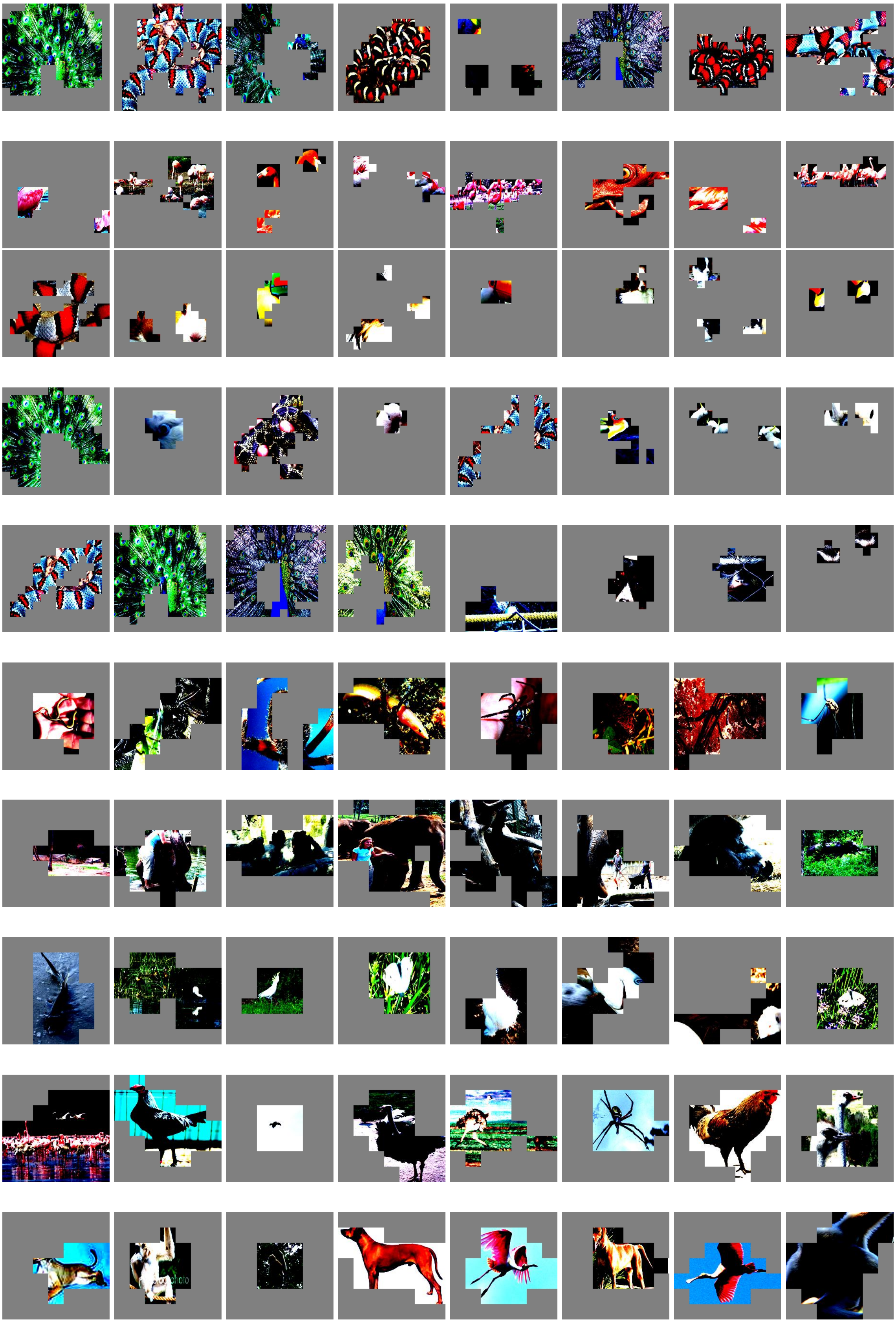


Figure 4.18: PCA concepts deemed uninterpretable by human subjects. Each row corresponds to a single concept. The first five rows show the concepts extracted from VGG16, and the last five rows show the concepts from ResNet50.

Quality

Metric	Architecture	Explainer	Rating	(t,p)
Understandability	VGG16	SCE	3.5 ± 0.9	(0.61,0.54)
		PCA	3.3 ± 0.9	
	ResNet50	SCE	3.8 ± 0.9	(0.09,0.92)
		PCA	3.8 ± 0.9	
Satisfaction	VGG16	SCE	3.7 ± 0.7	(1.39,0.16)
		PCA	3.3 ± 0.9	
	ResNet50	SCE	4 ± 0.9	(0.79,0.43)
		PCA	3.8 ± 0.9	
Sufficiency	VGG16	SCE	3.9 ± 0.9	(1.15,0.29)
		PCA	3.2 ± 0.9	
	ResNet50	SCE	3.6 ± 1.1	(0.36,0.72)
		PCA	3.5 ± 1.1	
Prediction Accuracy	VGG16	SCE	$57.8 \pm 12.5\%$	(0.16,0.87)
		PCA	$56.8 \pm 14.2\%$	
	ResNet50	SCE	$69.5 \pm 8.7\%$	**(2.29,0.02)**
		PCA	$64.7 \pm 11.3\%$	
Prediction Confidence	VGG16	SCE	$76.8 \pm 14.9\%$	(1.68,0.13)
		PCA	$70.2 \pm 15.9\%$	
	ResNet50	SCE	$84.8 \pm 9.6\%$	(1.59,0.11)
		PCA	$80.9 \pm 13.2\%$	
Completeness	VGG16	SCE	3.5 ± 0.9	(0.19,0.78)
		PCA	3.4 ± 0.9	
	ResNet50	SCE	3.9 ± 0.9	(0.12,0.91)
		PCA	3.8 ± 0.9	

Table 4.2: Quality rating - statistical significance

The quality of explanations is assessed using metrics proposed by Hoffman et al. [100]. The evaluation is performed by dividing the experiment into two phases, learning and test phases. During the learning phase, participants are presented with images of ten animal classes and are asked to learn to differentiate between them. In the test phase, participants are shown an animal image and are asked to predict its class. This process whose interface is shown in Figure 4.13b, helps us identify participants familiar with image classification and assess the quality of the explanations.

The participants are then presented with explanations highlighting the top 3 relevant concepts used by the model to make its predictions. The participants' prediction for the class label, along with their confidence in the prediction, would be recorded. If the explanation is informative, the participant should be able to predict the correct class confidently. After several such explanations, participants are asked to rate the explanation method based on various quality parameters such as understandability, sufficiency, completeness, etc., suggested by Hoffman et al. [100]. Understandability measures the user's understanding of the explanation. A good explanation method should consistently highlight informative concepts that help users understand how the prediction was made. Satisfaction assesses the explanation quality from a psychological perspective, measuring how the user feels regarding the helpfulness of the explanations in unearthing the working of the deep neural networks. Sufficiency measures if the top concepts displayed are sufficient for making the prediction. Finally, participants are asked to rate the extent to which the explanation provides a complete picture of the black box's workings.

It was observed that the prediction accuracy of participants based on SCE explanations for the ResNet50 model is 69.5 ± 8.7%, which is higher and statistically significant (p-value was 0.02 and calculated t-statistic was 2.29) compared to PCA explanations on which the recorded prediction accuracy was 64.7 ± 11.3%. To prove a claim that the proposed approach is better than the baseline, it has to pass the statistical significance test. If the p-value is within 0.05, the claim that the proposed approach is better than the baseline can be proved statistically. As can be seen from Table 4.2, on other metrics, the p-value exceeds the threshold to pass the claim. Hence no statistical significance could be established for the other metrics. This is in accordance with the findings reported in a recent work proposed by Zhang et al. [144], which leveraged vanilla NMF to extract class-specific concepts in a posthoc manner.

4.5.5 Ablations

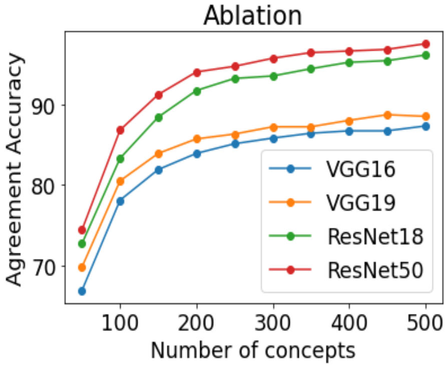


Figure 4.19: Agreement accuracy Vs Number of concepts

The key hyper-parameter the user can determine in SCE is the number of concepts C. This hyper-parameter can be varied, and the corresponding change in agreement accuracy due to this variation has been plotted in Figure 4.19. The agreement accuracies seem to be proportional to the model depth. Also, the agreement accuracy of the ResNet model family is higher than that of the VGG model family. Zooming into the architecture, one can observe that the classifier h of the ResNet model is a single linear layer, while the VGG classifier h is a multi-layer network. The residual error when propagated across a single layer could be less impactful compared to the case of a multi-layer classifier. However, on plotting the sharedness distribution as in Figure 4.8, an invariance in the distribution with a change in the number of concepts was observed. This may indicate that SCE explains concepts that reflect how the black box model has learned to generate features that discriminate classification. To optimize accuracy-simplicity tradeoff [35], $K = 100$ has been chosen for other analyses.

4.6 Summary

SCE framework has been proposed for extracting human-interpretable concepts using incremental NMF in a shared manner, reflecting shared concepts in nature. SCE's accuracy is evaluated, and qualitative visualizations offer unique insights into animal species. SCE hints at the reasons for superior classification performance in certain architectures and reveals the impact of spurious patterns on model predictions. Ablation analyses highlight that the model architecture has a significant effect on the concepts' nature. The SCE framework shall offer newer insights when applied to applications where an already deployed model needs to be explained.

Chapter 5

Explainable Supervised Domain Adaptation Network

Domain adaptation techniques have contributed to the success of deep learning. Leveraging knowledge from an auxiliary source domain for learning in labeled data-scarce target domain is fundamental to domain adaptation. While these techniques result in increasing accuracy, the adaptation process, particularly the knowledge leveraged from the source domain, remains unclear. This chapter proposes an explainable by design supervised domain adaptation framework - XSDA-Net. A case-based reasoning mechanism has been integrated into the XSDA-Net to explain the prediction of a test instance in terms of similar-looking regions in the source and target train images. The utility of the proposed framework is empirically demonstrated by curating the domain adaptation settings on datasets popularly known to exhibit part-based explainability.

5.1 Introduction

Deep learning has seen great successes in the recent past [5, 6, 7] with the availability of large datasets [133]. However, acquiring labeled data is an expensive and time-consuming process. Harnessing a deep classifier trained on related large labeled datasets does not generalize well on the dataset of interest where limited labeled data is available due to the changes in data distribution, often called domain shift [154]. This shift may be due to differences in the marginal distribution of features or class label-based conditional distribution. Domain Adaptation encompasses techniques that help bridge the domain shift between the source and target domains. Domain adaptation has helped to learn accurate models in many critical situations where limited data is available in various tasks like image classification, activity recognition, sentiment analysis, indoor localization.

Despite such state-of-the-art accuracies, the deep models are not readily adopted in all application domains. The opaqueness of a deep network's internal mechanism contributes to its hesitancy in adoption [10]. Moreover, the right to explanation act by EU has made it mandatory to provide explanations to the users involved in the decisions made by the AI systems, leading to the development of mechanisms for explaining deep networks. Recent work is on explaining a general-purpose classifier [21, 25, 36]. However, little attention is given to explaining a domain-adapted classifier where knowledge from two domains is leveraged [132, 155, 156].

This chapter proposes a framework that incorporates explainability by design into the domain-adapted classifier. The underlying assumption is that a set of prototypical features describes a given class's instances. The framework aims to learn these prototypical features in a latent space where domain-invariance is achieved through supervised domain adaptation. XSDA-Net looks for similar features in a test image to predict the class label using the learned domain-invariant prototypical features. This prediction can be explained in a case-based reasoning fashion. A sample explanation expected from the proposed model is shown in Figure 5.1. The prototypical features detected in the test image are shown in various colored rectangles. The top and bottom rows show the source and target domain concepts, respectively, that are most similar to the concepts detected in the test image. The contribution to a label's output is computed as a linear combination of the similarity scores, where the linear coefficients are learned during the training procedure. Contribution to each class is calculated, and a softmax operator will be applied to obtain the corresponding class probabilities. This helps build a model that transparently unearths the whole reasoning pipeline.

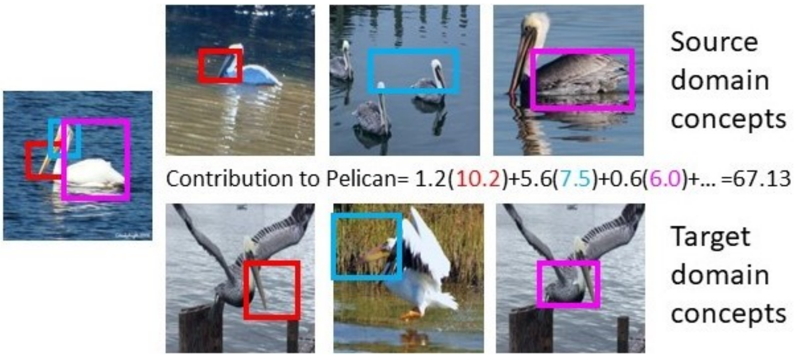

Source
domain
concepts
Contribution to Pelican= 1.2(10.2)+5.6(7.5)+0.6(6.0)+... =67.13
Target
domain
concepts

Figure 5.1: Sample explanation

5.2 Domain Adaptation

Supervised domain adaptation refers to the umbrella of techniques that utilize a source domain $\mathbb{D}^s = \{x_i, y_i\}_{i=1}^{N^s}$ with abundant labelled examples to learn a classifier for the target domain denoted by $\mathbb{D}^t = \{x_i, y_i\}_{i=1}^{N^t}$ with limited labelled examples. The source and target domains differ in the underlying marginal and conditional distributions. Most supervised domain adaptation approaches [157, 158] perform class-wise alignment such that the instances are clustered based on class labels ignoring domain differences which aid a classifier to learn a decision boundary that separates them. The supervised domain adaptation approaches can be categorized into discrepancy-based and adversarial techniques. In Discrepancy-based techniques [159, 160, 161], a discrepancy measure indicative of the domain gap is minimized, leading to the domains getting aligned closer. The same classifier trained on the source domain may then be reused to classify the target domain instances, or a new classifier can be trained using the aligned source and target labeled instances. The adversarial techniques utilize the GAN principle to align the domains. The feature extractor part of the network acts as a generator. A domain discriminator that aims to distinguish source and target domains provides feedback to the generator to generate domain invariant features [137, 138]. These domain invariant features generated for the target domain instances can then be passed through the classifier learned using the labeled source domain instances to perform classification. However, all the state-of-the-art supervised domain adaptation techniques are not interpretable. The aspects of the source and target domains focused by the classifier remain a mystery. The proposed work aims to demystify this process by integrating explainability into the design of the domain adaptation framework.

Despite advancements in XAI for explaining in-domain classifiers, less focus is given to explaining the working of domain-adapted classifiers. Szabó et al. [132] uses Activation Maximization (AM) [134] to visualize the filters during the transfer learning process. However, the use of AM makes the explanation less useful for non-experts. Hao & Zheng [155] use a GAN to understand features that help achieve domain invariance. However, using a black box to explain a black box makes the explanation less faithful. Neyshabur et al. [135] perform a detailed analysis to unearth the role of feature reuse and pretrained weights during the process of fine-tuning. In contrast, this framework explains the domain-adapted classifier using class-specific prototypical parts. The concept discovery process is tightly integrated into the domain adaptation module, thus realizing explainability by design to leverage the model's knowledge gained from the data to generate the explanations.

5.3 Contributions

The main contributions of this framework are:

- A method that integrates explainability by design into a domain adaptation framework has been proposed.
- A case-based reasoning style to explain a prediction based on class-specific characteristic prototypical features identified in the given test image has been adopted.
- Domain adaptation settings have been curated using datasets commonly used in explainability literature to validate the proposed framework's utility
- A theoretical framework that analyzes the impact of projecting the abstract concepts of XSDA-Net to the nearest train image patches on the model's accuracy during the training process has been developed.
- The key assumptions of the theoretical framework are empirically verified.
- Ablation experiments investigate the impact of the different learning objectives on the classification performance of XSDA-Net.

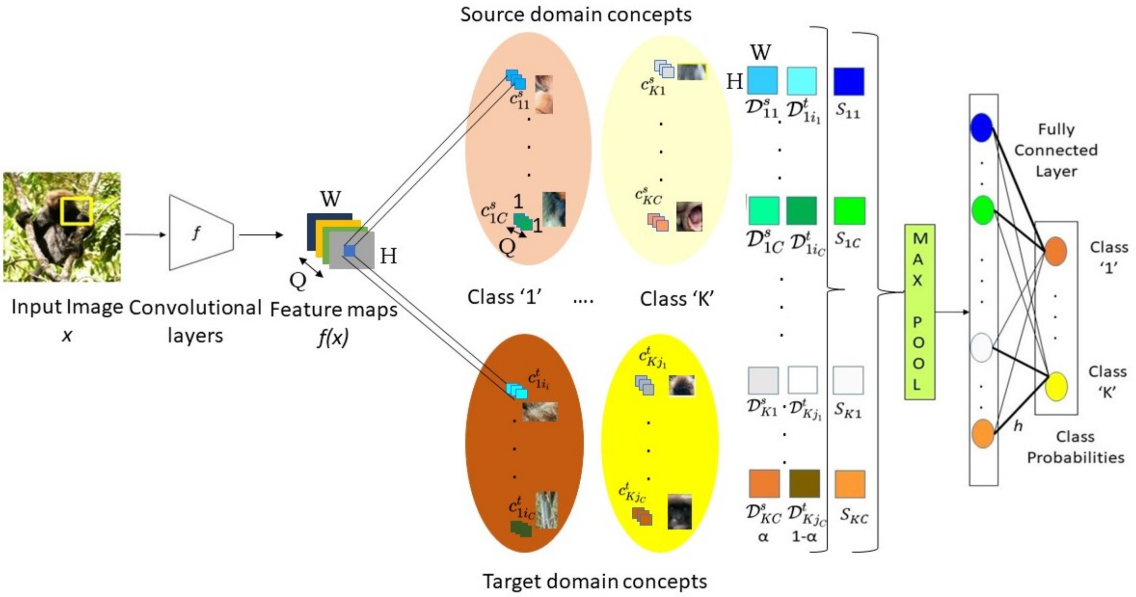


Figure 5.2: XSDA-Net architecture

5.4 Methodology

The architecture of the proposed explainable supervised domain adaptation network (XSDA-Net) is illustrated in Figure 5.2. An input image x is passed through a convolutional backbone f. The feature map $f(x)$ obtained from the convolutional backbone has dimensions $H \times W \times Q$.

The feature maps $f(x)$ are passed through the explanatory backbone g consisting of the concept layers corresponding to the source and target domains g^s and g^t, respectively. Every class has C concepts of dimensions $1 \times 1 \times Q$ per domain. c^s_{kl} denotes the l^{th} source concept of the k^{th} class and c^t_{kl} denotes the l^{th} target concept of the k^{th} class. The procedure to classify a target test image using the trained XSDA-Net is described first. Learning the XSDA-Net is described later.

Let $\mathcal{C}^s_k = \{c^s_{kl}\}_{l=1}^{C}$ and $\mathcal{C}^t_k = \{c^t_{kl}\}_{l=1}^{C}$ denote the set of source and target concepts for k^{th} class respectively. For each source concept c^s_{kl}, there is a paired target concept c^t_{kj} such that $||c^s_{kl} - c^t_{kj}||_2^2$ is the least among the set of target concepts, $\{c^t_{ki}\}_{i=1}^{C}$. Given the feature map $f(x)$, the Euclidean distance between each $1 \times 1 \times Q$ patch in $f(x)$ and all the source and target concepts are computed. Let $\mathcal{D}^s_{kl}$ and $\mathcal{D}^t_{kl}$ denote the $H \times W$ matrices representing the distance of each of the $H \times W$ patches in $f(x)$ from c^s_{kl} and c^t_{kl} concepts of the source and target respectively. The convex combination of the distance matrices $\mathcal{W}_{kl} = \alpha\mathcal{D}^s_{kl} + (1-\alpha)\mathcal{D}^t_{kj}$ that covers information from both source and target domains is converted into a similarity score by means of a monotonically increasing function given by $S_{kl} = \log\left(\frac{\mathcal{W}_{kl}+1}{\mathcal{W}_{kl}+\epsilon}\right)$ (where ϵ is a small non-zero value used to avoid numerical instability during the element-wise division operation). Each element in S_{kl} denotes the similarity of each patch in $f(x)$ with respect to the learned concept. The maximum similarity value obtained through max-pool layer is passed to a fully connected linear layer h that outputs the classification probabilities. Max pool is used because the dominant presence of a pattern similar to that of the learned concepts has to impact classification irrespective of the location of the pattern. Furthermore, S_{kl} can be upsampled to visualize the region in the test image with the maximum similarity. The regions in the test image with the maximum similarity to the source and target concepts form part of the explanation. Thus, the XSDA-Net can be represented as a composition $h \circ g \circ f$, where f is the feature extractor, g is the concept-based explanation backbone and h is the aggregator that performs prediction based on the interpretable components extracted by f and g.

5.4.1 Training Procedure

A three-phase training cycle is adopted to learn the concept layer g comprising the source and target domain concept set g^s and g^t, respectively. In the first phase, the aim is to learn a latent space where class discriminativeness is achieved by employing different loss functions and bridging the domain gap. The other two phases are initiated concurrently at regular intervals, followed by the initial learning phase. The main aim of the second phase is to reinforce the explainability of the framework, where the learned representations are mapped onto a training image patch that will be visualized to understand the learned aspect. The third phase trains the dense layer's weights connecting the similarity vector to the output.

5.4.2 Learning explanatory latent space

The first phase aims to learn a meaningful latent representation of the concepts. The concepts are to be class-specific discriminative image regions that aid the classification of any test instance. To instill class-specificity, the given class k concepts must be clustered closer in the latent space. This is achieved through the clustering loss [36] applied to each domain independently $d \in \{s, t\}$, defined as

$$\mathcal{L}_C^d = \frac{1}{N^d} \sum_{i=1}^{N^d} \min_{j: c_j \in \mathcal{C}_{y_i}^d} \; \min_{z \in patches(f(x_i))} ||z - c_j||_2^2$$

The clustering loss makes sure that the learned concept representation is closer to at least one training image patch of the corresponding domain having the same ground truth as that of the concept.

The overall clustering loss is given as a weighted combination of clustering loss at each domain, as given below.

$$\mathcal{L}_C = \sum_{d \in \{s,t\}} \beta^d \mathcal{L}_C^d$$

The concepts of a given class k have to be far apart from the concepts of other classes $k^{'} \neq k$. This is enforced by means of a separation loss on the concepts of both the source and target domains $d \in \{s, t\}$, defined as

$$\mathcal{L}_S^d = -\frac{1}{N^d} \sum_{i=1}^{N^d} \min_{j:c_j \notin \mathcal{C}_{y_i}^d} \min_{z \in patches(f(x_i))} ||z - c_j||_2^2$$

The separation loss ensures that the learned concept representation is farther from all training image patches of the corresponding domain having ground truth class labels other than that of the concept.

The overall separation loss is given as a weighted combination of the separation loss for each domain, given as

$$\mathcal{L}_S = \sum_{d \in \{s,t\}} \gamma^d \mathcal{L}_S^d$$

Unique concepts are learned by enforcing that the representation corresponding to a given concept is distinct and far apart from that of other concepts. This is enforced through a distinction loss, as given below.

$$\mathcal{L}_D^d = -\frac{1}{N^d} \sum_{i=1}^{N^d} \sum_{j,j':j \neq j';c_j,c_{j'} \in \mathcal{C}_{y_i}^d} \min_{z_j \in patches(f(x_i))} ||z_j - c_{j'}||_2^2$$

The distinction loss ensures that all concepts of a given class are not clustered around the same image patch.

The overall distinction loss is given as a weighted combination of distinction loss at each domain.

$$\mathcal{L}_D = \sum_{d \in \{s,t\}} \delta^d \mathcal{L}_D^d$$

Cross-entropy loss is minimized to improve the classification output. The fully connected layer weights are initialized such that the weights connecting the concept to its corresponding class are kept at 1, and the rest are kept at -0.5. This facilitates the model to learn that the stronger presence of the concepts should enhance the prediction probability for its corresponding class. The domain-specific cross-entropy loss and the overall loss are defined as

$$\mathcal{L}_{CE}^d = \frac{1}{N^d} \sum_{i=1}^{N^d} CrsEnt(h \circ g \circ f(x_i), y_i)$$

Thus the overall cross-entropy loss is given by,

$$\mathcal{L}_{CE} = \sum_{d \in \{s,t\}} \omega^d \mathcal{L}_{CE}^d$$

Minimizing domain adaptation loss $\mathcal{L}_{DA}$ aligns the concept representations of source and target domains in the latent space. In this framework, the d-SNE technique [162] is leveraged to perform supervised domain alignment. The loss is applied to the concepts of each domain. This loss $\mathcal{L}_{DA}$ is given in equations 5.1 and 5.2. The main idea is to separate the classes in the latent space of concepts by minimizing the maximum distance among the concepts belonging to the same class (i.e., minimizing the maximum intra-class concept distance) and maximizing the minimum distance among concepts of different classes (i.e., maximizing the minimum inter-class concept distance) across the domains.

$$\mathcal{L}_{DA} = \frac{1}{K \times C} \sum_{k=1}^{K} \sum_{l=1}^{C} \mathcal{L}_{DA}(c^{d}_{kl}) \tag{5.1}$$

where

$$\mathcal{L}_{DA}(c^{d}_{kl}) = arg \max_{j} ||c^{d}_{kl} - c^{d'}_{kj}||^2_2 - \forall_{y \neq k} arg \min_{i} ||c^{d}_{kl} - c^{d'}_{yi}||^2_2 \tag{5.2}$$

$d, d' \in \{s, t\}$ denotes source and target domains and $d' \neq d$.
The overall objective function comprising of all the loss terms discussed above as given in equation 5.3 is minimized using an Adam optimizer [143]

$$\mathcal{L} = \mathcal{L}_C + \mathcal{L}_S + \mathcal{L}_D + \mathcal{L}_{CE} + \kappa \mathcal{L}_{DA} \tag{5.3}$$

5.4.3 Projecting the Concepts

The main aim of this phase is to map the learned concept vectors to humanly understandable train image patches. The prototypical representations learned are assigned to the nearest patch among the train images of the corresponding class of the domain under consideration. This can be mathematically represented as $c^{s}_{kl} \leftarrow arg \min_{z \in patches(f(x)); x \in \mathbb{D}^{s}_{k}} ||z - c^{s}_{kl}||_2$ and $c^{t}_{kl} \leftarrow arg \min_{z \in patches(f(x)); x \in \mathbb{D}^{t}_{k}} ||z - c^{t}_{kl}||_2$. A rectangular box covering the maximally activated image region yields the visualization of the concept.
To determine the impact of projection on classification, how the classifier's logits are affected by the projection operation needs to be examined. Misclassification occurs when the logit corresponding to the ground truth class decreases and the logits of other classes increase. Therefore, the maximum possible decrease in the logit of the correct class and the corresponding increase in the logits of other classes would be calculated to assess the potential for misclassification. If the difference between the logits of the top two predictions falls within the bounds of the change in logits caused by the projection operation, then the impact of projection on the classifier's accuracy is likely to be insignificant.

Theorem 1. *Let, c^{d}_{kl} denote the l^{th} concept of the k^{th} class corresponding to the domain $d \in \{s, t\}$, b^{d}_{kl} denote value of c^{d}_{kl} before projection and a^{d}_{kl} denote value after projection. Let q be the ground truth class label of x and $z^{d}_{kl} = arg \min_{z \in patches(f(x))} ||z - b^{d}_{kl}||_2$ be nearest training image patch among images of the domain d before projection. If $\exists \delta \in (0, 1)$ that satisfies the following axioms:*

- *For all incorrect class concepts $k \neq q; \forall l \in \{1, 2, \ldots, C\}$, $d \in \{s, t\}$,*

$$||a^{d}_{kl} - b^{d}_{kl}||_2 \leq \theta ||z^{d}_{kl} - b^{d}_{kl}||_2 \tag{5.4}$$

where, $\theta = min(\sqrt{1+\delta} - 1, 1 - \frac{1}{\sqrt{2-\delta}})$.

- *For all correct class q concepts;* $\forall l \in \{1, 2, \ldots, C\}$, $d \in \{s, t\}$,

$$\|a^d_{ql} - b^d_{ql}\|_2 \leq (\sqrt{1+\delta} - 1)\|z^d_{ql} - b^d_{ql}\|_2 \tag{5.5}$$

and,

$$\|z^d_{ql} - b^d_{ql}\|_2 \leq \sqrt{1-\delta} \tag{5.6}$$

then the projection operation does not impact the classification, provided the difference between the highest and the second-highest logit before projection is at most $2\Delta L_{max}$*, where* $\Delta L_{max} = C\log((2-\delta)(1+\delta))$.

It is to be noted that a common δ is assumed to hold the bounds in both source and target domains as the domains are aligned by the Supervised Domain Adaptation loss.

Proof. The output logits for a given class k- L_k can be expressed as the linear combination of its concept similarities, i.e. $L_k = W_h g_{\mathbf{P}_k}$. But as the fully connected layer weights W_h are sparse, the logit computation expression reduced to:

$$L_k = \sum_{l=1}^{C} \log(\frac{\alpha\|z^s_{kl} - c^s_{kl}\|_2^2 + (1-\alpha)\|z^t_{kj} - c^t_{kj}\|_2^2 + 1}{\alpha\|z^s_{kl} - c^s_{kl}\|_2^2 + (1-\alpha)\|z^t_{kj} - c^t_{kj}\|_2^2 + \epsilon})$$

And thus, the change in logits due to projection can be expressed as:

$$\Delta L_k = \sum_{l=1}^{C} \log(\frac{\alpha\|z^s_{kl} - a^s_{kl}\|_2^2 + (1-\alpha)\|z^t_{kj} - a^t_{kj}\|_2^2 + 1}{\alpha\|z^s_{kl} - b^s_{kl}\|_2^2 + (1-\alpha)\|z^t_{kj} - b^t_{kj}\|_2^2 + 1}$$

$$.\frac{\alpha\|z^s_{kl} - b^s_{kl}\|_2^2 + (1-\alpha)\|z^t_{kj} - b^t_{kj}\|_2^2 + \epsilon}{\alpha\|z^s_{kl} - a^s_{kl}\|_2^2 + (1-\alpha)\|z^t_{kj} - a^t_{kj}\|_2^2 + \epsilon})$$

If,

$$\nu_{kl} = \frac{\alpha\|z^s_{kl} - a^s_{kl}\|_2^2 + (1-\alpha)\|z^t_{kj} - a^t_{kj}\|_2^2 + 1}{\alpha\|z^s_{kl} - b^s_{kl}\|_2^2 + (1-\alpha)\|z^t_{kj} - b^t_{kj}\|_2^2 + 1} \tag{5.7}$$

$$\phi_{kl} = \frac{\alpha\|z^s_{kl} - b^s_{kl}\|_2^2 + (1-\alpha)\|z^t_{kj} - b^t_{kj}\|_2^2 + \epsilon}{\alpha\|z^s_{kl} - a^s_{kl}\|_2^2 + (1-\alpha)\|z^t_{kj} - a^t_{kj}\|_2^2 + \epsilon} \tag{5.8}$$

$$\Psi_{kl} = \nu_{kl}.\phi_{kl} \tag{5.9}$$

ΔL_k can be rewritten as, $\Delta L_k = \sum_{l=1}^{C} \log \Psi_{kl}$
One possible cause of misclassification may be a decrease in the logits of the correct class q. A bound on the maximum decrease in the logits of the correct class L_q to quantify the impact on the classifier's output would be derived now.

Note that Equation 5.7 has the lower bound,

$$\nu_{ql} \geq \frac{1}{\alpha\|z^s_{ql} - b^s_{ql}\|^2_2 + (1-\alpha)\|z^t_{qj} - b^t_{qj}\|^2_2 + 1}$$

From Equation 5.6,

$$\|z^d_{ql} - b^d_{ql}\|^2_2 \leq (1-\delta)$$

Hence,

$$\nu_{ql} \geq \frac{1}{2-\delta}$$

By applying triangle inequality in Equation 5.5,

$$\|z^d_{qj} - a^d_{qj}\|^2_2 \leq (1+\delta)\|z^d_{qj} - b^d_{qj}\|^2_2$$

Substituting these deductions in Equation 5.8 for the correct class q, the bound, $\phi_{ql} \geq \frac{1}{1+\delta}$. Thus, the lower bound for Ψ_{kl} for the correct class q is,

$$\Psi_{ql} \geq \frac{1}{(2-\delta)(1+\delta)} \tag{5.10}$$

The change in logits for the class q can then be bounded as,

$$-\Delta L_q \leq C\log((2-\delta)(1+\delta))$$

The worst case decrease in correct class q logits is $\Delta L_{max} = C\log((2-\delta)(1+\delta))$.
Another cause of misclassification may be an increase in the logits of any class k other than that of the correct class q. Now the bound of the maximum increase in the logits of any incorrect class to assess the impact of projection on the classifier's output has to be obtained.
Substituting $\theta = \sqrt{1+\delta} - 1$ in Equation 5.4,

$$\|z^d_{kl} - a^d_{kl}\|^2_2 \leq (1+\delta)\|z^d_{kl} - b^d_{kl}\|^2_2$$

that when applied in equation 5.7, the upper bound, $\nu^k_l \leq 1+\delta$. When $\theta = 1 - \frac{1}{\sqrt{2-\delta}}$,

$$\|z^d_{kl} - a^d_{kl}\|^2_2 \geq \frac{1}{2-\delta}\|z^d_{kl} - b^d_{kl}\|^2_2$$

resulting in the lower bound $\nu_{kl} \geq \frac{1}{2-\delta}$. Thus, the overall bounds for ν_{kl} are,

$$\frac{1}{2-\delta} \leq \nu_{kl} \leq (1+\delta)$$

Similarly, substituting $\theta = 1 - \frac{1}{\sqrt{2-\delta}}$ in equation 5.4,

$$\|z^d_{kl} - b^d_{kl}\|^2_2 \leq (2-\delta)\|z^d_{kl} - a^d_{kl}\|^2_2$$

that results in the upper bound $\phi_{kl} \leq (2-\delta)$. When, $\theta = \sqrt{1+\delta} - 1$ we get,

$$\|z_{kl}^d - a_{kl}^d\|_2^2 \leq (1+\delta)\|z_{kl}^d - b_{kl}^d\|_2^2$$

resulting in the lower bound $\phi_{kl} \geq \frac{1}{1+\delta}$. Thus, the bounds for ϕ_{kl} are,

$$\frac{1}{1+\delta} \leq \phi_{kl} \leq (2-\delta)$$

The bounds on ν_{kl} and ϕ_{kl} indicate that

$$\frac{1}{(2-\delta)(1+\delta)} \leq \Psi_{kl} \leq (2-\delta)(1+\delta) \tag{5.11}$$

Thus, the change in logits for an incorrect class, is bounded as follows:

$$\Delta L_k \leq C \log((2-\delta)(1+\delta))$$

Hence the worst case increase in incorrect class logits is $\Delta L_{max} = C \log((2-\delta)(1+\delta))$.

□

The axioms and basic inequalities deduce that the worst-case difference between logits of the highest and second highest ranked class is at most $2\Delta L_{max}$. Thus the projection operation does not impact the performance of the domain-adapted classifier.

5.4.4 Learning the classifier

The fully connected layer will use the prototypical representations modified in the projection phase to perform classification. Thus the main aim of the third phase, post the projection phase, is to finetune these fully connected layer weights. The feature extractor f and the explanatory backbone g are frozen in this phase. The fully connected layer weights are finetuned to accommodate the changes due to the projection phase. Sparse connection weights are encouraged employing a L_1 regularizer. As the contribution to a class is a weighted combination of the similarity scores, sparsity in weights results in fewer concepts contributing more to the final output.

5.4.5 Gradual training

A warm-start training strategy is used, where the feature extraction backbone f is initialized and frozen for a few epochs, initially using a standard pretrained network (VGG, ResNet). The explanatory backbone g and the fully connected layer h of XSDA-Net are trained, utilizing the knowledge encoded in the pretrained weights. Further, learning the concepts is like picking a vector blindfolded in the latent space. However, in the end, the concepts have to be mapped back to the training images for explainability. Theoretically, the entire latent space can be searched, the search is restricted to only the subspace containing the training image features by executing the projection phase every ρ epochs.

5.5 Experiments

5.5.1 Datasets

The explanations generated by XSDA-Net are demonstrated using two datasets (Birds and Monkeys) commonly used to study XAI algorithms. The datasets contain animal images characterized by distinct concepts corresponding to different image regions, serving as ideal candidates to validate the XSDA-Net. The domain differences in the prevalent domain adaptation datasets are on finer features like color and textures [155]. Domain expertise is required to understand such finer feature-based explanations. In contrast, our framework generates readily interpretable explanations based on parts or regions of the image.

5.5.2 Birds Dataset

Eight bird classes present in both Imagenet [133] and CUB [163] datasets are used. A few sample instances from the two domains are shown in Figure 5.3. The bird images from the Imagenet and CUB are considered as source and target domains, respectively. The Imagenet dataset has more than 1000 images per class, while the CUB dataset has at most 180 images per class. The target domain train and validation splits each has ten images per class, and the rest are part of the test split.

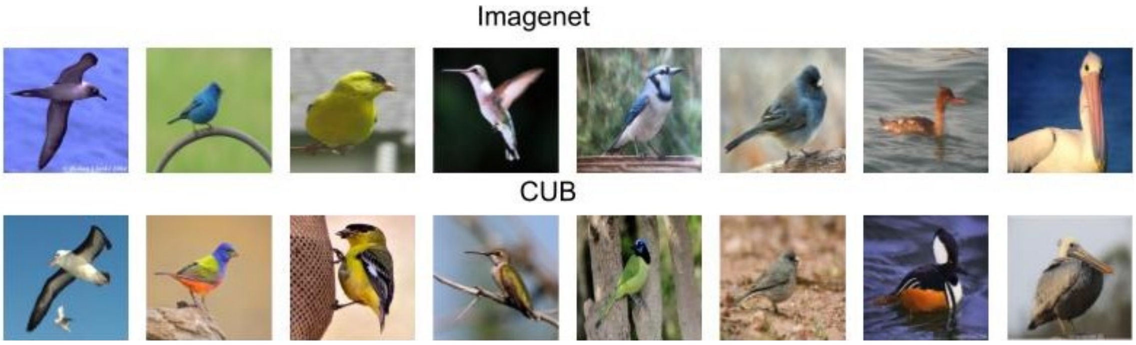


Figure 5.3: Birds dataset visualization. Each column depicts a class.

5.5.3 Monkeys Dataset

Seven classes that are present in both Imagenet and monkey species classification datasets obtained from Kaggle is used. A few sample instances from the different domains are shown in Figure 5.4. Like the previous dataset, Imagenet [133] with more than 1000 images per class serves as the source domain. The Kaggle dataset, with at most 270 images per class is the target domain. The train and validation splits of the target domain each have ten images per class, and the rest are part of the test split.

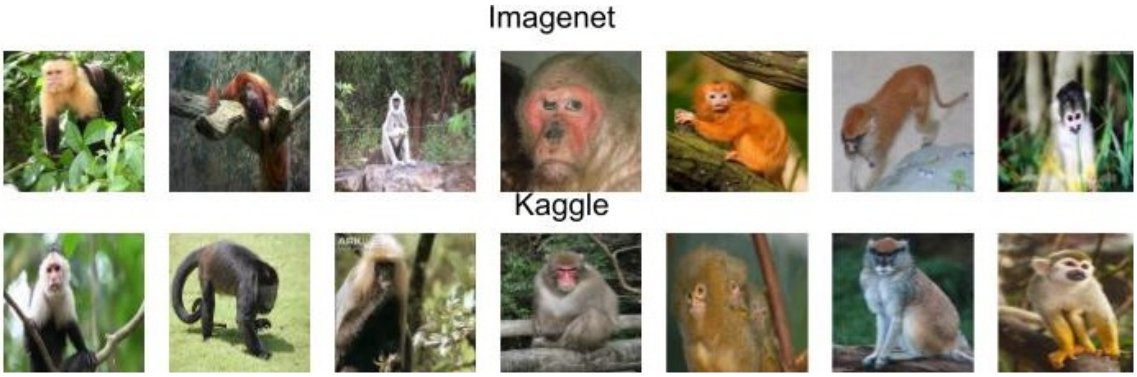


Figure 5.4: Monkeys dataset visualization. Each column depicts a class.

The hyper-parameters used to train the explainer were $C = 10$, $\alpha = 0.5$, $\beta^s = \beta^t = 10$, $\gamma^s = \gamma^t = -0.08$, $\delta^s = \delta^t = -0.01$, $\omega^s = \omega^t = 50$, $\kappa = 20$, $\rho = 10$. The optimal values were obtained through extensive cross-validation experiments involving only the train and validation splits.

The test accuracy of the black-box domain adapted VGG16 classifier without the explanation module is 96.2% and 98.3% for the birds and monkeys dataset, respectively. In contrast, the explainable by design domain adaptation framework, XSDA-Net's accuracy is 92.4% and 96.8%, respectively. The marginal drop in the performance due to architectural modifications is tolerable considering the significant benefits of extracting interpretable explanations.

5.5.4 Correct Classification

This subsection illustrates and discusses explanations generated by the XSDA-Net for a few correctly classified instances from the target domain. Figure 5.5 presents explanations for eight test instances along with the reasoning pipeline for the classification. For brevity, the illustrated concept pairs from the source and target train set (marked by the colored rectangles) are restricted to the top 3 concepts ranked by the similarity scores. A salient observation is the close resemblance of the concept pairs from the source and target domains. The contribution score of the source and target concept pair (represented using the same colored rectangle) to a particular class computed as a weighted combination of the calculated similarity scores is displayed using the same color. The weights of this combination (numbers displayed in black) are learned during the training phase. In these examples, as it can be seen, despite visual differences, the corresponding body parts are aligned between the source and target domains due to the explicit training scheme. The test images are correctly classified due to the high similarity with the aligned concepts from the source and target train sets. For example, body parts such as beaks and wings are used by the model to correctly classify *bunting* and other bird species. Similarly, the model considers face, body, and limbs among the top concepts for correctly classifying *patas* and other monkey species. It is also interesting to note that a concept for the *merganser* bird species is the background water. While the background may not be a characteristic for

Objective	Birds Accuracy	Monkeys Accuracy
$\mathcal{L} \setminus \mathcal{L}_C$	90.3%	94.4%
$\mathcal{L} \setminus \mathcal{L}_S$	90.2%	93.8%
$\mathcal{L} \setminus \mathcal{L}_D$	91.3%	94.4%
$\mathcal{L} \setminus \mathcal{L}_{DA}$	91.5%	96.1%
$\mathcal{L}$	92.4%	96.8%

Table 5.1: Effect of the different components of the learning objective on accuracy

this class, the explanations highlight the bias in the dataset (all *merganser* images have water in the background). Despite lowered performance due to architectural modifications, our explainable by design framework discovers the underlying learning and case-based reasoning process that is impossible from a black-box pipeline.

5.5.5 Misclassification

Figure 5.6 illustrates the explanations for a few misclassified examples. A justifiable visual similarity is seen between the detected test image regions and the learned concepts of both predicted and ground truth classes. Due to incorrect assessment of contribution scores, misclassification has occurred. Especially for the *marmoset* image that is misclassified as *capuchin* and *bunting* image that is misclassified as *hummingbird*, it can be seen that the model assessed a higher similarity with the background, considering it a part of test instance, leading to the misclassification. Also, looking at the paired concepts, one can see that despite visual differences between species in both domains, due to our explicit training scheme enforcing part-based alignment, the corresponding body parts are aligned. This empirically shows the effectiveness of XSDA-Net as an explainable domain adaptation network.

5.6 Ablation studies

Table 5.1 summarizes the effect of each component of the objective function $\mathcal{L}$ thereby quantifying its importance. It is to be noted that as the cross-entropy loss $\mathcal{L}_{CE}$ establishes the connection between the different modules f, g and h, it cannot be removed from the learning objective $\mathcal{L}$.

5.6.1 Cluster Loss

The cluster loss $\mathcal{L}_C$ is designed to bring together the concepts of the same class in the latent space. To evaluate its effectiveness, the average intra-class concept distance is calculated. For a given concept c^d_{kl}, the distance from all other concepts c^d_{ki} such that $i \neq l$ is calculated. The average of these distances μ^d_{kl} is then determined. This is done for all c^d_{kl} where $k \in 1, 2, \ldots, K$, $l \in 1, 2, \ldots, C$ and $d \in \{s, t\}$ and the average of all resulting μ^d_{kl} gives the desired metric, namely the average intra-class concept distance. A lower value for this metric indicates a better-learned latent space.

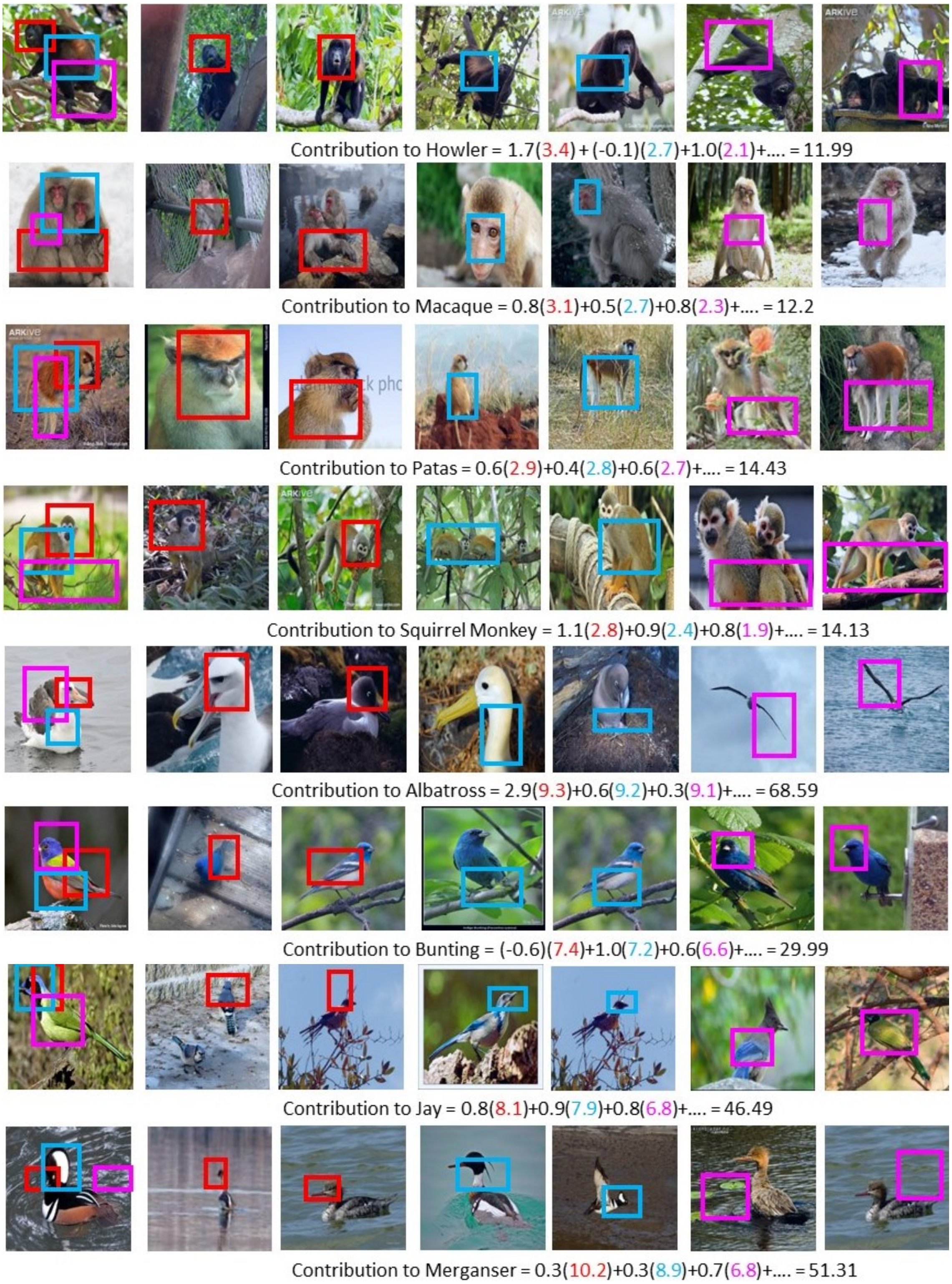

Contribution to Howler = 1.7(3.4) + (-0.1)(2.7)+1.0(2.1)+.... = 11.99
Contribution to Macaque = 0.8(3.1)+0.5(2.7)+0.8(2.3)+.... = 12.2
Contribution to Patas = 0.6(2.9)+0.4(2.8)+0.6(2.7)+.... = 14.43
Contribution to Squirrel Monkey = 1.1(2.8)+0.9(2.4)+0.8(1.9)+.... = 14.13
Contribution to Albatross = 2.9(9.3)+0.6(9.2)+0.3(9.1)+.... = 68.59
Contribution to Bunting = (-0.6)(7.4)+1.0(7.2)+0.6(6.6)+.... = 29.99
Contribution to Jay = 0.8(8.1)+0.9(7.9)+0.8(6.8)+.... = 46.49
Contribution to Merganser = 0.3(10.2)+0.3(8.9)+0.7(6.8)+.... = 51.31

Figure 5.5: Explanations for a few correctly classified test instances. The test image regions (colored rectangles in the image in the first column) map to the regions in the source and target image regions (successive image pairs with rectangles of the same color). The source and target image concept pairs are sorted based on the similarity to the test image region. The fully connected layer weight connecting the concept to the corresponding class is in black. The contribution to the corresponding class is computed via this weighted combination.

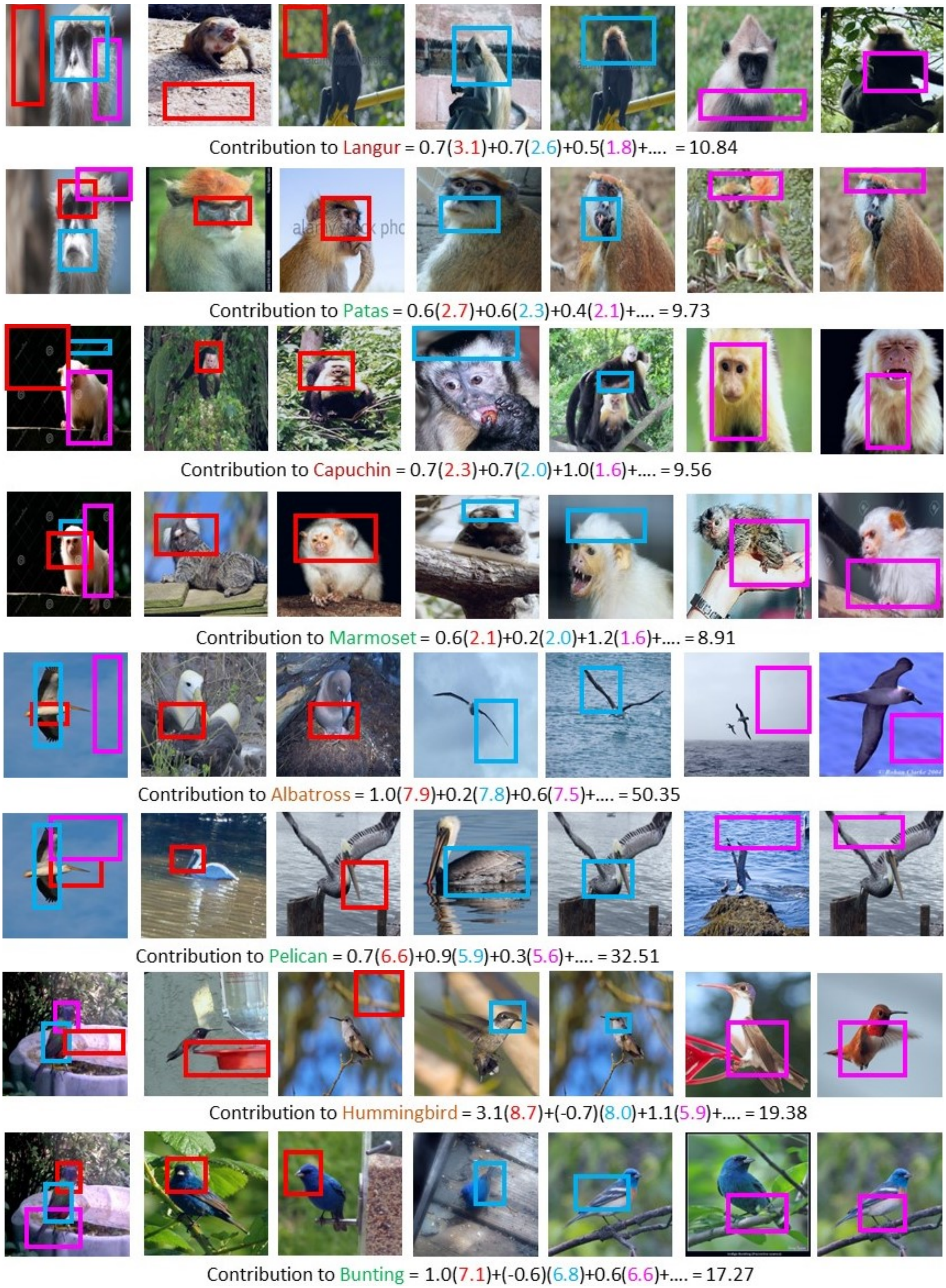

Contribution to Langur = 0.7(3.1)+0.7(2.6)+0.5(1.8)+.... = 10.84
Contribution to Patas = 0.6(2.7)+0.6(2.3)+0.4(2.1)+.... = 9.73
Contribution to Capuchin = 0.7(2.3)+0.7(2.0)+1.0(1.6)+.... = 9.56
Contribution to Marmoset = 0.6(2.1)+0.2(2.0)+1.2(1.6)+.... = 8.91
Contribution to Albatross = 1.0(7.9)+0.2(7.8)+0.6(7.5)+.... = 50.35
Contribution to Pelican = 0.7(6.6)+0.9(5.9)+0.3(5.6)+.... = 32.51
Contribution to Hummingbird = 3.1(8.7)+(-0.7)(8.0)+1.1(5.9)+.... = 19.38
Contribution to Bunting = 1.0(7.1)+(-0.6)(6.8)+0.6(6.6)+.... = 17.27

Figure 5.6: Misclassified images. The explanation for each misclassified instance spans across two rows. The first row shows the explanation corresponding to the class incorrectly predicted by the model whose label is given in brown color. The second row shows the explanation corresponding to the ground truth class whose label is given in green color. The test image regions (colored rectangles in the image in the first column) map to the regions in the source and target image regions (successive image pairs with rectangles of the same color). The source and target image concept pairs are sorted based on the similarity to the test image region. The fully connected layer weight connecting the concept to the corresponding class is in black. The contribution to the corresponding class is computed via this weighted combination.

This metric is calculated when the cluster loss $\mathcal{L}_C$ was included and excluded from the learning objective $\mathcal{L}$. When $\mathcal{L}_C$ was excluded, the average intra-class concept distance was 2.254 for the birds dataset and 1.354 for the monkeys dataset. In contrast, when $\mathcal{L}_C$ was included, the average intra-class prototypical distance was 0.039 for the birds dataset and 1.268 for the monkeys dataset. In both cases, the accuracy of the explainable domain-adapted classifier dropped by around 2% when $\mathcal{L}_C$ was excluded from the learning objective $\mathcal{L}$.

5.6.2 Separation Loss

The separation loss $\mathcal{L}_S$ is a measure of how well the concepts of different classes are separated in the latent space. To evaluate the efficacy of this loss function, the average inter-class concept distance is calculated. For a given concept c^d_{kl}, its distance from all other concepts c^d_{yi}, where $y \neq k$ is calculated. The average of these distances is called λ^d_{kl}. This calculation was performed for all c^d_{kl}, $k \in \{1, 2, \dots, K\}$, $l, i \in \{1, 2, \dots, C\}$ and $d \in \{s, t\}$. The average of all λ^d_{kl} values gives us the average inter-class concept distance, which is the metric of interest. Higher values of this metric indicate better separation in the latent space.

When $\mathcal{L}_S$ was included, an increase in the average inter-class concept distance (from 2.84 to 5.00 on the birds dataset) was observed, which indicates that the concepts of different classes were more separated in the latent space. The accuracy dropped to 90.2 % when $\mathcal{L}_S$ was excluded. In the case of the monkeys dataset, the exclusion of $\mathcal{L}_S$ reduced the average inter-class concept distance modestly by 0.154, a significant drop (by 3%) was observed. Overall, these results suggest that separation loss is beneficial to learning.

5.6.3 Distinction Loss

The distinction loss $\mathcal{L}_D$ aims to map the concepts to different representations to the best possible extent. For a given class k, in the domain $d \in \{s, t\}$ consider the $C \times C$ matrix ζ^d_k whose elements $\zeta_{ij} = distance(c^d_{ki}, c^d_{kj})$. It can be observed that ζ^d_k is a symmetric matrix whose diagonal elements are 0. The triangular matrix below the diagonal contains $\frac{C(C-1)}{2}$ values. Let χ^d_k denote the number of non-zero values among these $\frac{C(C-1)}{2}$ values. $\chi = \sum_{d \in \{s,t\}} \sum_{k=1}^{K} \chi^d_k$ is calculated when $\mathcal{L}_D$ is included and excluded from the learning objective $\mathcal{L}$. It can be observed that $\chi \leq KC(C-1)$. In the explainable domain adapted classifier trained on the birds dataset ($K = 8$), excluding $\mathcal{L}_D$ in its learning objective $\mathcal{L}$ yields $\chi = 719$. In other words, a pair of learned concepts were repeating. In a similar scenario with the monkeys dataset($K = 7$), χ was observed to be 628. In other words, two pairs of learned concepts were repeating. When $\mathcal{L}_D$ was included in the learning objective $\mathcal{L}$, $\chi = KC(C-1)$, the maximum possible value was achieved in both datasets. In other words, including distinction loss $\mathcal{L}_D$, makes all concepts distinct. Empirically it was seen that distinction loss $\mathcal{L}_D$, which is the novel aspect of the proposed work, mitigates the problem of repeating concepts observed in prior explainable by design approaches [36].

The accuracy dropped to 91.3% in the birds dataset and 94.4% in the monkeys dataset when $\mathcal{L}_D$ was excluded.

5.6.4 Domain Adaptation Loss

$\mathcal{L}_{DA}$ aligns the concepts of the different domains closer, bridging the domain gap. The average inter-domain concept distance i.e. the distance between the source domain concept c^s_{kl} and the target domain concept c^t_{mn} for $k, m \in \{1, 2, \ldots, K\}$ and $l, n \in \{1, 2, \ldots, C\}$ is calculated for all K^2C^2 possible combinations. The average of these values gives the necessary metric. Lower metric values indicate the closer alignment of the domains. In the explainable domain adapted classifier trained on the birds dataset excluding $\mathcal{L}_{DA}$ from the learning objective $\mathcal{L}$, the average inter-domain concept distance turned out to be 2.646, which fell to 0.056 on including $\mathcal{L}_{DA}$. The accuracy dropped to 91.5% due to this exclusion. In the case of the monkeys dataset, the average inter-domain concept distance when $\mathcal{L}_{DA}$ was excluded was 1.899, and the value fell to 1.661 when $\mathcal{L}_{DA}$ was included to $\mathcal{L}$. Due to this exclusion, the accuracy dropped to 96.1%.

5.7 Summary

Thus the XSDA-Net that can unearth the reasoning pipeline in a classifier aligned via domain adaptation has been proposed. XSDA-Net uses case-based reasoning to explain the output of the domain adapter classifier. Specifically, it explains the model's output for a test image in terms of highly similar prototypical regions from source and target train image pairs, along with the contribution of the similarity to the final output. Experiments on curated domain adaptation datasets illustrate the XSDA-Net's effectiveness in explaining correct and incorrect classifications despite a marginal decrease in the accuracy compared to its non-explainable counterparts.

Chapter 6

Conclusion

The thesis traces the significant improvements in the object recognition task, hinting at the opaqueness that creeps in as a side effect of the increasing complexity of the accurate classifiers. The opaqueness that hinders the opening up of the working mechanism of the accurate deep CNNs, like that of traditional shallow models, has been discussed. The need to explain the working of the CNN has been motivated, and thereby the attempts of the XAI research community to generate explanations for a CNN have been overviewed. The supremacy of concept-based explanations [32, 33] due to their close resemblance to how humans process images [50] has been illustrated.

Despite the supremacy of concept-based explanations and attempts to generate such explanations, the dependence of these mechanisms on the annotated concept examples based on which concept representations are learned acts as a bottleneck for the widespread adoption of concept-based explainers. When the annotated concept examples are from a different distribution than the distribution from which the training examples are sampled, the representations need not truly reflect the learned representations of the CNN [35]. To overcome the need for annotated concept examples, Yeh et al. [53] propose to extract concept representations automatically from the data and estimate concept relevances using Shapley values [77]. However, that framework uses a two-layer network which is another black box, thereby complicating the problem at hand of explaining the CNN of interest.

6.1 Summary of the Proposed Frameworks

The thesis identifies a void in the space of concept-based explanations that there needs to be a mechanism that can automatically extract concepts learned by the CNN in a fully interpretable manner. Two frameworks are proposed in this direction by varying the sharedness of the concepts across different classes. Chapter 3 proposed PACE, which learned to extract class-specific, relevant concepts automatically from the data. Faithfulness is embedded by design by enabling approximation from the explanation to be based on queries from the black box. The relevance estimation is tied to the learning pipeline by enforcing the drop in prediction probability of a class when the corresponding concept is ablated. The framework's efficacy has been shown using different CNN architectures and datasets. This being the first approach that automatically extracts class-specific concepts in a posthoc manner, was compared with a curated baseline mimicking the properties of the modules of the framework. Quantitative analyses reveal that the proposed framework has a superior faithfulness measured by the agreement accuracy metric proposed exclusively for assessing the faithfulness of concept-based explanations. Thus qualitative analyses have been carried out using results obtained from the proposed framework only. Human subject experiments show that 71% of the extracted concepts are humanly interpretable, that humans can tag the concept depicted by the consistent visualizations. The tags provided by the humans are analyzed and found to be relevant to the visually depicted aspects. The salient parts discriminative of the category of the animals were extracted. Misclassifications were justified due to reliance on a concept that is visually similar to the concept that is generally a characteristic of the class incorrectly predicted.

While the PACE framework proposed in Chapter 3 encouraged learning class-specific concepts to unravel the discriminant blueprints used by CNN to predict the given test instances, the concepts tend to be shared, especially if relatedness between classes exists. To reflect this perspective, several antehoc approaches which enforced real-world concept sharedness by design were proposed. However, a void existed in the posthoc explanation space to model such sharedness constraints. Although the proposal by Yeh et al. [53] learns shared concepts in a posthoc manner, the framework leveraged black boxes in its concept learning pipeline, hindering its application to faithfully unravel the working mechanism of the black box of interest, aka the CNN. Thus a mechanism whose working is fully interpretable and can unravel the sharedness of concepts from the lens of the CNN being explained needs to be developed. This was the aim of the Shared Concepts Extractor (SCE) framework proposed in Chapter 4. The framework was based on Non-negative Matrix Factorization (NMF) technique which was known to extract semantically meaningful concepts from the activations [144, 145, 148]. The incremental NMF technique [148] was leveraged to learn the shared concepts utilizing the available memory resources. The key assumption behind the explainer was that the features could be expressed as a linear combination of a set of basis vectors called the concepts. As most architectures used ReLU activations to process forward only the positive activations, NMF, which enforces a non-negative constraint on the linear combination weights and the corresponding basis vectors, aka the concepts, was employed. The relevance estimation was carried out after concept learning, unlike the PACE framework proposed in Chapter 3, where relevance is a part of the learning pipeline. This shall help unravel all concepts learned by CNN, irrespective of their relevance to predicting a specific class. The framework is flexible to unravel the primitive aspects like color, shape, or texture encoded by the concepts by estimating the impact of perturbing the corresponding aspect on concept detection [54]. The incorporation of shared concepts unraveled sharedness across classes as learned by the CNN, some in line with human intuition, some giving newer perspectives on how classes may be grouped. The exclusive concepts seen in ResNet architectures seem to contribute to its higher accuracy. In other words, although ResNet recognizes shared concepts, it also detects discriminant concepts like the characteristic bill structure of the *spoonbill*, face structure of a *guinea pig*, etc., that aids in minimizing the misclassifications compared to the VGG models whose feature extractors detect shared concepts and the reliance to classify the given instance is probably higher on the fully connected layers. This suspicion stems from the observation in the antehoc approaches [49] where removing the multiple fully connected layers with non-linear activation function hurts the classification accuracy. The explainer unravels the spurious correlations learned by the CNNs in line with a recent observation [22].

While much of the XAI community's efforts are on explaining in-domain classifiers, the mechanism employed in cross-domain classification, which helps extend the benefits of data-hungry deep models to data-scarce scenarios, is not explored. Particularly the domain adaptation process, which bridges the distributional differences between the data-rich source domain and the data-scarce target domain of interest so that a classifier trained on the source domain can be leveraged to classify the instances sampled from the target domain of interest. Chapter 5 proposes a supervised domain-adapted classifier that can explain itself. Specifically, the framework learns class-specific prototypical concepts and uses a case-based reasoning strategy to predict the class a test instance belongs to. The concepts are enforced to be distinct and clustered based on the class whose blueprint it encodes. The domain differences are aligned by means of maximizing the least inter-class concept distance as well as minimizing the highest intra-class concept distance [162]. These losses enforce the creation of tighter coalitions of concepts in the latent space such that they are domain-invariant and class-discriminant and guide the prediction of given test instances based on the detection of these concepts. Domain adaptation settings have been curated on datasets that exhibit part-based explainability. The effect of incorporating interpretability into a domain-adapted classifier has been verified theoretically and empirically. The importance of each component of the learning objective has been reinforced through ablation studies.

6.2 Limitations of the Proposed Frameworks

Existing concept-based explanatory frameworks depended on the concept examples to be sampled from the same distribution from which the training data is sampled for the explanations to be faithful. The thesis has proposed three novel frameworks that automatically extract concepts from the data to circumvent the need for external concept examples. However, there are a few limitations to the proposed frameworks.

An inherent challenge in any posthoc explanatory framework is establishing faithfulness to the explained black box. This is a difficult goal to achieve because if there is a mechanism to know the ground truth of the working of the classifier, then the need for an explanation algorithm would become obsolete. With no gold standard to aim for, the existing approaches propose proxy metrics to assess the faithfulness of the generated explanation based on the perturbation effect. However, those metrics are unsuitable for concept-based explanations due to the possibility of amalgamation of concepts covering the whole image, thereby nullifying the perturbation process. The thesis proposes a proxy metric to estimate the faithfulness of concept-based explanatory frameworks. The basic

assumption of the metric is that the explainer perfectly mimics the working of the black box. A mechanism to circumvent this over-reliance on the posthoc explainer is needed.
The antehoc explainer proposed in Chapter 5 also suffers from a limitation inherent to explainable by design frameworks. There is a drop in accuracy due to the incorporation of explainability compared to the black box counterpart. The proposed framework builds a domain-adapted classifier that explains itself by design. Several frameworks were proposed later to incorporate explainability into the allied learning paradigms [164, 165] also suffer from this tradeoff. To circumvent this, a mechanism to explain classifiers employing allied learning paradigms faithfully in a posthoc manner needs to be developed.

6.3 Future Work

While the thesis proposes novel frameworks that advance the field of XAI, several open problems are available to be solved collectively by the community. Mainly, three possible research directions are discussed.
The preliminary direction shall be to extend the fruits of explainability to allied learning paradigms. Traditional deep learning methods were data-hungry as they leveraged voluminous chunks of data. However, various allied learning paradigms have been introduced to reap the fruits of deep learning to data-scarce scenarios. Transfer Learning aims to extend a classifier trained on a related data set containing many instances to work on the scarce data of interest by aligning the feature and label spaces. The thesis proposes a framework that explains a supervised domain-adapted classifier by design. In a similar fashion, there have been parallel works [136, 139] that explain an unsupervised domain-adapted classifier. Extensions to explainable classifiers using heterogeneous transfer learning and open-set domain adaptation paradigms can be a possible future avenue to explore. Few Shot Learning aims to learn classifiers from fewer examples by leveraging features learned from related classes having a larger number of instances. For instance, a zebra can be considered as a horse-like body and tiger-like stripes. A motivating example from the medical domain would follow to distinguish it from Transfer Learning. Few Shot Learning aims to leverage features learned by a pneumonia detector to detect a related disease, say COVID-19, from fewer examples. Transfer Learning may leverage COVID-19 data collected from another country where more examples are available to learn a robust classifier that can be adapted to classify instances sampled from a country having fewer positive cases. Wang et al. [164] propose an explainable by design few-shot classifier which classifies an unseen novel test instance by matching the features detected against characteristic patterns learned in the seen categories. Incremental Learning mimics how humans learn. For instance, a computer scientist does not learn to build an application in a day. First, the programming principles are learned, then he learns to implement the different data structures needed to manage the various modules and finally learns to assemble the modules to get the end product. While learning an intermediate skill, humans do not forget the preliminary skills acquired.

However, this is not the case in AI systems. When new classes are expected to be learned by a classifier trained to classify an instance into a set of classes, they tend to forget the distinctions across older classes already learned [166]. However, the reason for such behavior is unknown. The thesis envisions the application of explainability to help unravel the mechanism behind the incrementally learned classifiers, thereby guiding the research community toward building classifiers that can mimic human-level incremental knowledge expansion. Rymarczyk et al. [165] propose building an antehoc model whose explainable components are learned such that catastrophic forgetting is managed by design. This model is an extension of ProtoPNet architecture [36] where the prototypes corresponding to the novel classes are enforced to be closer to the seen classes so that catastrophic forgetting is minimized. An extension using antehoc frameworks that encourage learning shared concepts similar to how concepts are shared in nature may enable minimizing catastrophic forgetting as, despite sharedness in nature, humans expand their knowledge base without forgetting.

The secondary direction shall be to develop quantitative metrics to assess the goodness of the learned concepts. In saliency map based methods, the goodness of the explanations is quantitatively assessed by simulating the effect on perturbation of the regions deemed salient. Union of regions comprising the concepts may be unfair to assess the goodness of the concepts as the union may cover up the entire image, nullifying the assessment. The thesis proposes a new metric called agreement accuracy which assesses how well the concept-based explainer approximates the working of the CNN to be explained. Leemann et al. [167] propose using natural language models to assess the goodness of the concepts. However, interpreting the language models [45] is needed on top of the evaluation process to make it transparent. Zarlenga et al. [168] proposed metrics to assess if the learned concept representations are pure with respect to a known oracle and suggests using inter-concept disentanglement to measure if the learned representations capture dissimilar concepts. When explainability is reaped to allied learning paradigms, metrics have to be developed to assess the correctness of the peculiar aspects of those paradigms as encoded by the explainer. For instance, if a posthoc explanatory approach is developed to unravel the mapping of features across different domains, an evaluation is needed to assess if what is being unraveled is true.

The tertiary direction suggests the cross-pollination of ideas from Neural Architecture Search, which aims to identify the best architecture and parameters to model the distribution from which the dataset of interest is sampled and XAI. There have been recent proposals in this direction. Liu et al. [169] suggest using intrinsically explainable components like regressors to search for optimal configurations to achieve black-box level accuracy. Hosseini & Xie [170] propose updating the search for a suitable neural architecture based on feedback from posthoc saliency maps [21]. The thesis envisions applying the principles of neural architecture search to identify the optimal number of concepts so that the accuracy interpretability tradeoff inherent to antehoc frameworks can be minimized eventually. This, when possible, shall have a greater impact on recent classifiers employing allied learning paradigms [164, 165] where explainability is incorporated by design.

While these are the possible future avenues with potential impact on the XAI field, an alternate route that has been started and an active area of research currently [171, 172, 173] is using the feedback from the explanation algorithm and introducing humans in the loop to edit the erroneous classifier. This can be an interesting direction one can focus on, especially when working in safety-critical applications; where necessary to adhere to the working mechanism laid by experts is essential.

6.4 Implications of the Thesis

The motivations of this thesis and the problems addressed are very relevant to the current scenario in the domain of machine learning, where explainability is becoming increasingly sought after to enable the pervasive application of artificial intelligence, especially in safety-critical applications like healthcare, finance, judiciary systems, etc., A human-friendly way of explanations is preferred as ultimately the AI systems would be used by humans. Studies suggest that humans process images in terms of individual constituent concepts. Hence, concept-based explanations can help users better understand the working of AI systems. This thesis proposes three novel concept-based explanatory frameworks in successive chapters that deal with progressive levels of complexity and challenges faced under different image-classification scenarios. One major outcome of this thesis is to lay the foundations for developing a general framework for automatically extracting concepts that explain the working of a CNN. The proposed explainable cross-domain classifying framework, which marks the beginning of extending the fruits of explainability to classifiers learned using allied learning paradigms, can be easily extended to other learning paradigms whose possibilities are discussed in the previous section.

Besides, this thesis provides a complete description of the XAI research area. It also summarizes the state-of-the-art contributions to the different types of explanations of a CNN that performs image classification. The underlying principle, limitations, and improvements made to these seminal contributions have also been highlighted. Furthermore, this thesis also presents some future research directions that can be taken based on the work done in this thesis, along with some unexplored avenues in the XAI field.

References

[1] N. K. Sarkar, M. M. Singh, and U. Nandi, "Recent researches on image classification using deep learning approach," *International Journal of Computing and Digital Systems*, vol. 12, no. 1, pp. 1357–1374, 2022.

[2] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the seventh IEEE International Conference on Computer Vision*, vol. 2, pp. 1150–1157, IEEE, 1999.

[3] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 886–893, 2005.

[4] S. S. A. Zaidi, M. S. Ansari, A. Aslam, N. Kanwal, M. Asghar, and B. Lee, "A survey of modern deep learning based object detection models," *Digital Signal Processing*, p. 103514, 2022.

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[6] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *International Conference on Learning Representations*, vol. 7, 2015.

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

[8] A. Gonzalez-Garcia, *Image context for object detection, object context for part detection.* PhD thesis, The University of Edinburgh, March 2018.

[9] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Object detectors emerge from training cnns for scene recognition," in *The 3rd International Conference on Learning Representations, San Diego, CA, USA*, pp. 1–12, 2015.

[10] Z. C. Lipton, "The doctor just won't accept that! interpretable ML symposium," in *Neural Information Processing Systems*, 2017.

[11] Z. Salahuddin, H. C. Woodruff, A. Chatterjee, and P. Lambin, "Transparency of deep neural networks for medical image analysis: A review of interpretability methods," *Computers in Biology and Medicine*, vol. 140, p. 105111, 2022.

[12] S. Bonicalzi, “A matter of justice. the opacity of algorithmic decision-making and the trade-off between uniformity and discretion in legal applications of artificial intelligence,” *Teoria. Rivista di filosofia*, vol. 42, no. 2, pp. 131–147, 2022.

[13] Council of European Union, “2018 reform of eu data protection rules,” 2018. https://ec.europa.eu/commission/sites/beta-political/files/data-protection-factsheet-changes_en.pdf.

[14] F. D. Martino and F. Delmastro, “Explainable AI for clinical and remote health applications: a survey on tabular and time series data,” *Artificial Intelligence Review*, pp. 1–55, 2022.

[15] L. Zhu, Z. Zhu, C. Zhang, Y. Xu, and X. Kong, “Multimodal sentiment analysis based on fusion methods: A survey,” *Information Fusion*, 2023.

[16] W. Saeed and C. Omlin, “Explainable AI (XAI): A systematic meta-survey of current challenges and future opportunities,” *Knowledge-Based Systems*, p. 110273, 2023.

[17] P. Weber, K. V. Carl, and O. Hinz, “Applications of explainable artificial intelligence in finance—a systematic review of finance, information systems, and computer science literature,” *Management Review Quarterly*, pp. 1–41, 2023.

[18] E. Owens, B. Sheehan, M. Mullins, M. Cunneen, J. Ressel, and G. Castignani, “Explainable artificial intelligence (XAI) in insurance,” *Risks*, vol. 10, no. 12, p. 230, 2022.

[19] A. J. Barnett, F. R. Schwartz, C. Tao, C. Chen, Y. Ren, J. Y. Lo, and C. Rudin, “A case-based interpretable deep learning model for classification of mass lesions in digital mammography,” *Nature Machine Intelligence*, vol. 3, no. 12, pp. 1061–1070, 2021.

[20] M. T. Ribeiro, S. Singh, and C. Guestrin, “Why should I trust you?: Explaining the predictions of any classifier,” in *Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining*, pp. 1135–1144, 2016.

[21] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, “Grad-CAM: Visual explanations from deep networks via gradient-based localization,” in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 618–626, 2017.

[22] Y. Neuhaus, M. Augustin, V. Boreiko, and M. Hein, “Spurious features everywhere–large-scale detection of harmful spurious features in ImageNet,” *arXiv preprint arXiv:2212.04871*, 2022.

[23] A. Chattopadhyay, A. Sarkar, P. Howlader, and V. N. Balasubramanian, “Grad-CAM++: Generalized gradient-based visual explanations for deep

convolutional networks," in *IEEE Winter Conference on Applications of Computer Vision*, pp. 839–847, IEEE, 2018.

[24] H. Wang, Z. Wang, M. Du, F. Yang, Z. Zhang, S. Ding, P. Mardziel, and X. Hu, "Score-CAM: Score-weighted visual explanations for convolutional neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 24–25, 2020.

[25] S. Desai and H. G. Ramaswamy, "Ablation-CAM: Visual explanations for deep convolutional network via gradient-free localization," in *The IEEE Winter Conference on Applications of Computer Vision*, pp. 983–991, 2020.

[26] M. T. Ribeiro, S. Singh, and C. Guestrin, "Anchors: High-precision model-agnostic explanations," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.

[27] R. Sharma, N. Reddy, V. Kamakshi, N. C. Krishnan, and S. Jain, "MAIRE-a model-agnostic interpretable rule extraction procedure for explaining classifiers," in *International Cross-Domain Conference for Machine Learning and Knowledge Extraction*, pp. 329–349, Springer, 2021.

[28] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *International Journal of Computer Vision*, vol. 59, pp. 167–181, 2004.

[29] A. Vedaldi and S. Soatto, "Quick shift and kernel methods for mode seeking," in *European Conference on Computer Vision*, pp. 705–718, Springer, 2008.

[30] T. Hartley, K. Sidorov, C. Willis, and D. Marshall, "SWAG: Superpixels weighted by average gradients for explanations of CNNs," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 423–432, 2021.

[31] Y. Goyal, Z. Wu, J. Ernst, D. Batra, D. Parikh, and S. Lee, "Counterfactual visual explanations," in *International Conference on Machine Learning*, pp. 2376–2384, PMLR, 2019.

[32] B. Kim, M. Wattenberg, J. Gilmer, C. Cai, J. Wexler, F. Viegas, *et al.*, "Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (TCAV)," in *International Conference on Machine Learning*, pp. 2668–2677, PMLR, 2018.

[33] A. Ghorbani, J. Wexler, J. Y. Zou, and B. Kim, "Towards automatic concept-based explanations," in *Advances in Neural Information Processing Systems*, pp. 9277–9286, 2019.

[34] M. Yuksekgonul, M. Wang, and J. Zou, "Post-hoc concept bottleneck models," in *ICLR Workshop on PAIR2Struct: Privacy, Accountability, Interpretability, Robustness, Reasoning on Structured Data*, 2022.

[35] V. V. Ramaswamy, S. S. Kim, R. Fong, and O. Russakovsky, “Overlooked factors in concept-based explanations: Dataset choice, concept salience, and human capability,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[36] C. Chen, O. Li, D. Tao, A. Barnett, C. Rudin, and J. K. Su, “This looks like that: deep learning for interpretable image recognition,” in *Advances in Neural Information Processing Systems*, pp. 8928–8939, 2019.

[37] D. Rymarczyk, Ł. Struski, M. Górszczak, K. Lewandowska, J. Tabor, and B. Zieliński, “Interpretable image classification with differentiable prototypes assignment,” in *European Conference on Computer Vision*, pp. 351–368, Springer, 2022.

[38] M. Nauta, R. van Bree, and C. Seifert, “Neural prototype trees for interpretable fine-grained image recognition,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14933–14943, 2021.

[39] L. A. Hendricks, Z. Akata, M. Rohrbach, J. Donahue, B. Schiele, and T. Darrell, “Generating visual explanations,” in *European Conference on Computer Vision*, pp. 3–19, Springer, 2016.

[40] L. A. Hendricks, R. Hu, T. Darrell, and Z. Akata, “Grounding visual explanations,” in *Proceedings of the European Conference on Computer Vision*, pp. 264–279, 2018.

[41] L. A. Hendricks, A. Rohrbach, B. Schiele, T. Darrell, and Z. Akata, “Generating visual explanations with natural language,” *Applied AI Letters*, vol. 2, no. 4, p. e55, 2021.

[42] Y. Yang, S. Kim, and J. Joo, “Explaining deep convolutional neural networks via latent visual-semantic filter attention,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8333–8343, 2022.

[43] Y. Kim, S. Mo, M. Kim, K. Lee, J. Lee, and J. Shin, “Explaining visual biases as words by generating captions,” *arXiv preprint arXiv:2301.11104*, 2023.

[44] S. Wickramanayake, W. Hsu, and M. L. Lee, “Comprehensible convolutional neural networks via guided concept learning,” in *International Joint Conference on Neural Networks*, pp. 1–8, IEEE, 2021.

[45] S. Gurrapu, A. Kulkarni, L. Huang, I. Lourentzou, L. Freeman, and F. A. Batarseh, “Rationalization for explainable NLP: A survey,” *arXiv preprint arXiv:2301.08912*, 2023.

[46] L. A. Hendricks, R. Hu, T. Darrell, and Z. Akata, “Generating counterfactual explanations with natural language,” in *ICML Workshop on Human Interpretability in Machine Learning*, pp. 95–98, 2018.

[47] D. H. Park, L. A. Hendricks, Z. Akata, A. Rohrbach, B. Schiele, T. Darrell, and M. Rohrbach, "Multimodal explanations: Justifying decisions and pointing to the evidence," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8779–8788, 2018.

[48] J. Wu and R. Mooney, "Faithful multimodal explanation for visual question answering," in *Proceedings of the ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pp. 103–112, 2019.

[49] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2921–2929, 2016.

[50] B. M. Lake, T. D. Ullman, J. B. Tenenbaum, and S. J. Gershman, "Building machines that learn and think like people," *Behavioral and Brain Sciences*, vol. 40, p. e253, 2017.

[51] S. L. Armstrong, L. R. Gleitman, and H. Gleitman, "What some concepts might not be," *Cognition*, vol. 13, no. 3, pp. 263–308, 1983.

[52] I. Biederman, "Recognition-by-components: a theory of human image understanding.," *Psychological Review*, vol. 94, no. 2, p. 115, 1987.

[53] C.-K. Yeh, B. Kim, S. Arik, C.-L. Li, T. Pfister, and P. Ravikumar, "On completeness-aware concept-based explanations in deep neural networks," in *Advances in Neural Information Processing Systems*, pp. 20554–20565, 2020.

[54] M. Nauta, A. Jutte, J. Provoost, and C. Seifert, "This looks like that, because... explaining prototypes for interpretable image recognition," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 441–456, Springer, 2021.

[55] X. Zhu, "Machine teaching: An inverse problem to machine learning and an approach toward optimal education," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 29, 2015.

[56] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," in *International Conference on Machine Learning*, pp. 3319–3328, PMLR, 2017.

[57] J. Zhang, S. A. Bargal, Z. Lin, J. Brandt, X. Shen, and S. Sclaroff, "Top-down neural attention by excitation backprop," *International Journal of Computer Vision*, vol. 126, no. 10, pp. 1084–1102, 2018.

[58] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for simplicity: The all convolutional net," *Workshop at International Conference on Learning Representations*, 2015.

[59] K. Simonyan, A. Vedaldi, and A. Zisserman, “Deep inside convolutional networks: Visualising image classification models and saliency maps,” in *Workshop at International Conference on Learning Representations*, 2014.

[60] Y. Wang, H. Su, B. Zhang, and X. Hu, “Learning reliable visual saliency for model explanations,” *IEEE Transactions on Multimedia*, 2019.

[61] J. Adebayo, J. Gilmer, M. Muelly, I. Goodfellow, M. Hardt, and B. Kim, “Sanity checks for saliency maps,” in *Advances in Neural Information Processing Systems*, pp. 9505–9515, 2018.

[62] L. Sixt, M. Granz, and T. Landgraf, “When explanations lie: Why modified BP attribution fails,” *International Conference on Machine Learning*, 2020.

[63] A. Salama, N. Adly, and M. Torki, “Ablation-CAM++: Grouped recursive visual explanations for deep convolutional networks,” in *IEEE International Conference on Image Processing*, pp. 2011–2015, 2022.

[64] R. Fong, M. Patrick, and A. Vedaldi, “Understanding deep networks via extremal perturbations and smooth masks,” in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2950–2958, 2019.

[65] R. C. Fong and A. Vedaldi, “Interpretable explanations of black boxes by meaningful perturbation,” in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3429–3437, 2017.

[66] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek, “On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation,” *PloS One*, vol. 10, no. 7, p. e0130140, 2015.

[67] A. Shrikumar, P. Greenside, and A. Kundaje, “Learning important features through propagating activation differences,” in *International Conference on Machine Learning*, pp. 3145–3153, PMLR, 2017.

[68] J. R. Lee, S. Kim, I. Park, T. Eo, and D. Hwang, “Relevance-CAM: Your model already knows where to look,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14944–14953, 2021.

[69] H. Jung and Y. Oh, “Towards better explanations of class activation mapping,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1336–1344, 2021.

[70] S. Sattarzadeh, M. Sudhakar, K. N. Plataniotis, J. Jang, Y. Jeong, and H. Kim, “Integrated grad-CAM: Sensitivity-aware visual explanation of deep convolutional networks via integrated gradient-based scoring,” in *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 1775–1779, IEEE, 2021.

[71] P. Wang, X. Kong, W. Guo, and X. Zhang, "Exclusive feature constrained class activation mapping for better visual explanation," *IEEE Access*, vol. 9, pp. 61417–61428, 2021.

[72] P. Dabkowski and Y. Gal, "Real time image saliency for black box classifiers," in *Advances in Neural Information Processing Systems*, pp. 6967–6976, 2017.

[73] C. Rudin, "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead," *Nature Machine Intelligence*, vol. 1, no. 5, pp. 206–215, 2019.

[74] M. R. Zafar and N. Khan, "Deterministic local interpretable model-agnostic explanations for stable explainability," *Machine Learning and Knowledge Extraction*, vol. 3, no. 3, pp. 525–541, 2021.

[75] D. Collaris, P. Gajane, J. Jorritsma, J. J. van Wijk, and M. Pechenizkiy, "LEMON: Alternative sampling for more faithful explanation through local surrogate models," in *Advances in Intelligent Data Analysis XXI*, (Cham), pp. 77–90, Springer Nature Switzerland, 2023.

[76] H. Lakkaraju, E. Kamar, R. Caruana, and J. Leskovec, "Faithful and customizable explanations of black box models," in *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pp. 131–138, 2019.

[77] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems*, pp. 4765–4774, 2017.

[78] S. M. Lundberg, G. G. Erion, and S.-I. Lee, "Consistent individualized feature attribution for tree ensembles," in *ICML Workshop on Human Interpretability in Machine Learning*, 2017.

[79] J. H. Friedman and B. E. Popescu, "Predictive learning via rule ensembles," *The annals of applied statistics*, pp. 916–954, 2008.

[80] C. Harris, R. Pymar, and C. Rowat, "Joint shapley values: a measure of joint feature importance," in *International Conference on Learning Representations*, 2022.

[81] X. Huang and J. Marques-Silva, "The inadequacy of shapley values for explainability," *arXiv preprint arXiv:2302.08160*, 2023.

[82] P. Rasouli and I. Chieh Yu, "CARE: Coherent actionable recourse based on sound counterfactual explanations," *International Journal of Data Science and Analytics*, pp. 1–26, 2022.

[83] M. Pawelczyk, C. Agarwal, S. Joshi, S. Upadhyay, and H. Lakkaraju, "Exploring counterfactual explanations through the lens of adversarial examples: A theoretical

and empirical analysis," in *International Conference on Artificial Intelligence and Statistics*, pp. 4574–4594, PMLR, 2022.

[84] A. Abid, M. Yuksekgonul, and J. Zou, "Meaningfully debugging model mistakes using conceptual counterfactual explanations," in *International Conference on Machine Learning*, pp. 66–88, PMLR, 2022.

[85] S. Singla and B. Pollack, "Explanation by progressive exaggeration," in *International Conference on Learning Representations*, 2020.

[86] P. Wang and N. Vasconcelos, "SCOUT: Self-aware discriminant counterfactual explanations," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8981–8990, 2020.

[87] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in Neural Information Processing Systems*, vol. 27, 2014.

[88] Y. Zhao, "Fast real-time counterfactual explanations," 2020.

[89] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, "StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8789–8797, 2018.

[90] O. Lang, Y. Gandelsman, M. Yarom, Y. Wald, G. Elidan, A. Hassidim, W. T. Freeman, P. Isola, A. Globerson, M. Irani, *et al.*, "Explaining in style: Training a GAN to explain a classifier in stylespace," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 693–702, 2021.

[91] P. W. Koh, T. Nguyen, Y. S. Tang, S. Mussmann, E. Pierson, B. Kim, and P. Liang, "Concept bottleneck models," in *International Conference on Machine Learning*, pp. 5338–5348, PMLR, 2020.

[92] J. Pfau, A. T. Young, J. Wei, M. L. Wei, and M. J. Keiser, "Robust semantic interpretability: Revisiting concept activation vectors," in *ICML Workshop on Human Interpretability in Machine Learning*, 2020.

[93] P. Arendsen, D. Marcos, and D. Tuia, "Concept discovery for the interpretation of landscape scenicness," *Machine Learning and Knowledge Extraction*, vol. 2, no. 4, p. 22, 2020.

[94] M. Hassanin, S. Anwar, I. Radwan, F. S. Khan, and A. Mian, "Visual attention methods in deep learning: An in-depth survey," *arXiv preprint arXiv:2204.07756*, 2022.

[95] A. K. Mohankumar, P. Nema, S. Narasimhan, M. M. Khapra, B. V. Srinivasan, and B. Ravindran, "Towards transparent and explainable attention models," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 4206–4216, 2020.

[96] W. Xu, J. Wang, Y. Wang, G. Xu, D. Lin, W. Dai, and Y. Wu, "Where is the model looking at –concentrate and explain the network attention," *IEEE Journal of Selected Topics in Signal Processing*, pp. 1–1, 2020.

[97] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *International Conference on Machine Learning*, pp. 2048–2057, 2015.

[98] S. Jain and B. C. Wallace, "Attention is not explanation," in *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 3543–3556, 2019.

[99] A. R. Akula and S.-C. Zhu, "Attention cannot be an explanation," *arXiv preprint arXiv:2201.11194*, 2022.

[100] R. R. Hoffman, S. T. Mueller, G. Klein, and J. Litman, "Metrics for explainable AI: Challenges and prospects," *arXiv preprint arXiv:1812.04608*, 2018.

[101] Q. Zhang, Y. Nian Wu, and S.-C. Zhu, "Interpretable convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8827–8836, 2018.

[102] O. Li, H. Liu, C. Chen, and C. Rudin, "Deep learning for case-based reasoning through prototypes: A neural network that explains its predictions," in *AAAI Conference on Artificial Intelligence*, 2018.

[103] P. Hase, C. Chen, O. Li, and C. Rudin, "Interpretable image recognition with hierarchical prototypes," in *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, vol. 7, pp. 32–40, 2019.

[104] J. Wang, H. Liu, X. Wang, and L. Jing, "Interpretable image recognition by constructing transparent embedding space," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 895–904, 2021.

[105] A. Hoffmann, C. Fanconi, R. Rade, and J. Kohler, "This looks like that... does it? shortcomings of latent space prototype interpretability in deep networks," *arXiv preprint arXiv:2105.02968*, 2021.

[106] Q. Huang, M. Xue, H. Zhang, J. Song, and M. Song, "Is ProtoPNet really explainable? evaluating and improving the interpretability of prototypes," *arXiv preprint arXiv:2212.05946*, 2022.

[107] D. Rymarczyk, Ł. Struski, J. Tabor, and B. Zieliński, “ProtoPShare: Prototypical parts sharing for similarity discovery in interpretable image classification,” in *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 1420–1430, 2021.

[108] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, “Show and tell: A neural image caption generator,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3156–3164, 2015.

[109] I. Tiddi and S. Schlobach, “Knowledge graphs as tools for explainable machine learning: A survey,” *Artificial Intelligence*, vol. 302, p. 103627, 2022.

[110] N. E. Maillot and M. Thonnat, “Ontology based complex object recognition,” *Image and Vision Computing*, vol. 26, no. 1, pp. 102–113, 2008.

[111] K. Marino, R. Salakhutdinov, and A. Gupta, “The more you know: Using knowledge graphs for image classification,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2673–2681, 2017.

[112] M. Alirezaie, M. Längkvist, M. Sioutis, and A. Loutfi, “A symbolic approach for explaining errors in image classification tasks,” in *International Joint Conference on Artificial Intelligence*, 2018.

[113] Z. A. Daniels, L. D. Frank, C. J. Menart, M. Raymer, and P. Hitzler, “A framework for explainable deep neural models using external knowledge graphs,” in *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications II*, vol. 11413, pp. 480–499, SPIE, 2020.

[114] Q. Liao and T. Poggio, “Object-oriented deep learning,” tech. rep., Center for Brains, Minds and Machines (CBMM), 2017.

[115] V. Ordonez, W. Liu, J. Deng, Y. Choi, A. C. Berg, and T. L. Berg, “Predicting entry-level categories,” *International Journal of Computer Vision*, vol. 115, pp. 29–43, 2015.

[116] R. T. Icarte, J. A. Baier, C. Ruz, and A. Soto, “How a general-purpose commonsense ontology can improve performance of learning-based image retrieval,” in *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 1283–1289, 2017.

[117] C.-H. H. Yang, Y.-C. Liu, P.-Y. Chen, X. Ma, and Y.-C. J. Tsai, “When causal intervention meets adversarial examples and image masking for deep neural networks,” in *IEEE International Conference on Image Processing*, pp. 3811–3815, IEEE, 2019.

[118] P. Panda, S. S. Kancheti, and V. N. Balasubramanian, “Instance-wise causal feature selection for model interpretation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1756–1759, 2021.

[119] M. Prabhushankar and G. AlRegib, “Extracting causal visual features for limited label classification,” in *IEEE International Conference on Image Processing*, pp. 3697–3701, IEEE, 2021.

[120] M. T. Bahadori and D. Heckerman, “Debiasing concept-based explanations with causal analysis,” in *International Conference on Learning Representations*, 2021.

[121] S. S. Kancheti, A. G. Reddy, V. N. Balasubramanian, and A. Sharma, “Matching learned causal effects of neural networks with domain priors,” in *International Conference on Machine Learning*, pp. 10676–10696, PMLR, 2022.

[122] S. Dash, V. N. Balasubramanian, and A. Sharma, “Evaluating and mitigating bias in image classifiers: A causal perspective using counterfactuals,” in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 915–924, 2022.

[123] C. Frye, C. Rowat, and I. Feige, “Asymmetric shapley values: incorporating causal knowledge into model-agnostic explainability,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 1229–1239, 2020.

[124] C. Reimers, J. Runge, and J. Denzler, “Determining the relevance of features for deep neural networks,” in *European Conference on Computer Vision*, (Cham), pp. 330–346, Springer International Publishing, 2020.

[125] M. Watson, B. A. S. Hasan, and N. Al Moubayed, “Learning how to mimic: Using model explanations to guide deep learning training,” in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1461–1470, 2023.

[126] D. Lopez-Paz, R. Nishihara, S. Chintala, B. Scholkopf, and L. Bottou, “Discovering causal signals in images,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6979–6987, 2017.

[127] W. Qin, H. Zhang, R. Hong, E.-P. Lim, and Q. Sun, “Causal interventional training for image recognition,” *IEEE Transactions on Multimedia*, 2021.

[128] S. Singla, S. Wallace, S. Triantafillou, and K. Batmanghelich, “Using causal analysis for conceptual deep learning explanation,” in *Medical Image Computing and Computer Assisted Intervention*, pp. 519–528, Springer, 2021.

[129] Y. Goyal, A. Feder, U. Shalit, and B. Kim, “Explaining classifiers with causal concept effect (CaCE),” *arXiv preprint arXiv:1907.07165*, 2019.

[130] P. Singhal, R. Walambe, S. Ramanna, and K. Kotecha, “Domain adaptation: Challenges, methods, datasets, and applications,” *IEEE Access*, vol. 11, pp. 6973–7020, 2023.

[131] A. Zunino, S. A. Bargal, R. Volpi, M. Sameki, J. Zhang, S. Sclaroff, V. Murino, and K. Saenko, “Explainable deep classification models for domain generalization,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3233–3242, 2021.

[132] R. Szabó, D. Katona, M. Csillag, A. Csiszárik, and D. Varga, “Visualizing transfer learning,” *ICML Workshop on Human Interpretability in Machine Learning*, 2020.

[133] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, *et al.*, “ImageNet large scale visual recognition challenge,” *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[134] A. Nguyen, A. Dosovitskiy, J. Yosinski, T. Brox, and J. Clune, “Synthesizing the preferred inputs for neurons in neural networks via deep generator networks,” in *Advances in Neural Information Processing Systems*, pp. 3387–3395, 2016.

[135] B. Neyshabur, H. Sedghi, and C. Zhang, “What is being transferred in transfer learning?,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 512–523, 2020.

[136] Y. Zhang, T. Yao, Z. Qiu, and T. Mei, “Explaining cross-domain recognition with interpretable deep classifier,” *arXiv preprint arXiv:2211.08249*, 2022.

[137] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, “Adversarial discriminative domain adaptation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7167–7176, 2017.

[138] Z. Pei, Z. Cao, M. Long, and J. Wang, “Multi-adversarial domain adaptation,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, 2018.

[139] W. Xiao, Z. Ding, and H. Liu, “Visualizing transferred knowledge: An interpretive model of unsupervised domain adaptation,” *arXiv preprint arXiv:2303.02302*, 2023.

[140] F. Schroff, D. Kalenichenko, and J. Philbin, “FaceNet: A unified embedding for face recognition and clustering,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 815–823, 2015.

[141] Y. Xian, C. H. Lampert, B. Schiele, and Z. Akata, “Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 9, pp. 2251–2265, 2018.

[142] C. Fellbaum, *WordNet: An Electronic Lexical Database.* Bradford Books, 1998.

[143] D. P. Kingma and J. L. B. Adam, “A method for stochastic optimization,” *International Conference on Learning Representations*, vol. 7, 2015.

[144] R. Zhang, P. Madumal, T. Miller, K. Ehinger, and B. Rubinstein, "Improving interpretability of CNN models using non-negative concept activation vectors," in *AAAI Conference on Artificial Intelligence*, 2021.

[145] C. Olah, A. Satyanarayan, I. Johnson, S. Carter, L. Schubert, K. Ye, and A. Mordvintsev, "The building blocks of interpretability," *Distill*, 2018.

[146] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.

[147] N. Li, S. Wang, H. Li, and Z. Li, "SAC-NMF-Driven graphical feature analysis and applications," *Machine Learning and Knowledge Extraction*, vol. 2, no. 4, pp. 630–646, 2020.

[148] J. Sun, Z. Wang, H. Li, and F. Sun, "Incremental nonnegative matrix factorization with sparseness constraint for image representation," in *Pacific Rim Conference on Multimedia*, pp. 351–360, Springer, 2018.

[149] A. Lefevre, F. Bach, and C. Févotte, "Online algorithms for nonnegative matrix factorization with the Itakura-Saito divergence," in *2011 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pp. 313–316, IEEE, 2011.

[150] A. Cichocki and A.-H. Phan, "Fast local algorithms for large scale nonnegative matrix and tensor factorizations," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 92, no. 3, pp. 708–721, 2009.

[151] C. Févotte and J. Idier, "Algorithms for nonnegative matrix factorization with the β-divergence," *Neural Computation*, vol. 23, no. 9, pp. 2421–2456, 2011.

[152] D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," *Advances in Neural Information Processing Systems*, vol. 13, 2000.

[153] A. Buades, B. Coll, and J.-M. Morel, "Non-local means denoising," *Image Processing On Line*, vol. 1, pp. 208–212, 2011.

[154] S. Kumar, V. K. Kurmi, P. Singh, and V. P. Namboodiri, "Mitigating uncertainty of classifier for unsupervised domain adaptation," *arXiv preprint arXiv:2107.00727*, 2021.

[155] Y. Hou and L. Zheng, "Visualizing adapted knowledge in domain transfer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13824–13833, 2021.

[156] V. K. Kurmi, S. Kumar, and V. P. Namboodiri, "Attending to discriminative certainty for domain adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 491–500, 2019.

[157] B. Kulis, K. Saenko, and T. Darrell, "What you saw is not what you get: Domain adaptation using asymmetric kernel transforms," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1785–1792, IEEE, 2011.

[158] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, "Adapting visual category models to new domains," in *European Conference on Computer Vision*, pp. 213–226, Springer, 2010.

[159] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, "A kernel two-sample test," *The Journal of Machine Learning Research*, vol. 13, no. 1, pp. 723–773, 2012.

[160] B. Sun and K. Saenko, "Deep coral: Correlation alignment for deep domain adaptation," in *European Conference on Computer Vision*, pp. 443–450, Springer, 2016.

[161] W. Zellinger, B. A. Moser, T. Grubinger, E. Lughofer, T. Natschläger, and S. Saminger-Platz, "Robust unsupervised domain adaptation for neural networks via moment alignment," *Information Sciences*, vol. 483, pp. 174–191, 2019.

[162] X. Zhou, X. Xu, R. Venkatesan, G. Swaminathan, and O. Majumder, "d-SNE: Domain adaptation using stochastic neighborhood embedding," in *Domain Adaptation in Computer Vision with Deep Learning*, pp. 43–56, Springer, 2020.

[163] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona, "Caltech-UCSD Birds 200," Tech. Rep. CNS-TR-2010-001, California Institute of Technology, 2010.

[164] B. Wang, L. Li, M. Verma, Y. Nakashima, R. Kawasaki, and H. Nagahara, "Match them up: visually explainable few-shot image classification," *Applied Intelligence*, pp. 1–22, 2022.

[165] D. Rymarczyk, J. van de Weijer, B. Zieliński, and B. Twardowski, "ICICLE: Interpretable class incremental continual learning," *arXiv preprint arXiv:2303.07811*, 2023.

[166] M. Masana, X. Liu, B. Twardowski, M. Menta, A. D. Bagdanov, and J. van de Weijer, "Class-incremental learning: survey and performance evaluation on image classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[167] T. Leemann, Y. Rong, S. Kraft, E. Kasneci, and G. Kasneci, "Coherence evaluation of visual concepts with objects and language," in *ICLR2022 Workshop on the Elements of Reasoning: Objects, Structure and Causality*, 2022.

[168] M. E. Zarlenga, P. Barbiero, Z. Shams, D. Kazhdan, U. Bhatt, A. Weller, and M. Jamnik, "Towards robust metrics for concept representation evaluation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2023.

[169] C.-H. Liu, Y.-S. Han, Y.-Y. Sung, Y. Lee, H.-Y. Chiang, and K.-C. Wu, "FOX-NAS: Fast, on-device and explainable neural architecture search," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 789–797, 2021.

[170] R. Hosseini and P. Xie, "Saliency-aware neural architecture search," *Advances in Neural Information Processing Systems*, vol. 35, pp. 14743–14757, 2022.

[171] S. Santurkar, D. Tsipras, M. Elango, D. Bau, A. Torralba, and A. Madry, "Editing a classifier by rewriting its prediction rules," *Advances in Neural Information Processing Systems*, vol. 34, pp. 23359–23373, 2021.

[172] R. Tanno, M. F Pradier, A. Nori, and Y. Li, "Repairing neural networks by leaving the right past behind," *Advances in Neural Information Processing Systems*, vol. 35, pp. 13132–13145, 2022.

[173] J. Wang, R. Hu, C. Jiang, R. Hu, and J. Sang, "Counterexample contrastive learning for spurious correlation elimination," in *Proceedings of the 30th ACM International Conference on Multimedia*, pp. 4930–4938, 2022.