Towards More Explainable Image Segmentation

In Richtung einer besser erklärbaren Bildsegmentierung
Master thesis in the field of study "Computational Engineering" by Max Zimmermann
Date of submission: 13 November 2023

1. Review: Prof. Stefan Roth, Ph.D.
2. Review: Robin Hesse, M.Sc.
Darmstadt

TECHNISCHE
UNIVERSITÄT
DARMSTADT

Erklärung zur Abschlussarbeit gemäß § 22 Abs. 7 APB TU Darmstadt

Hiermit erkläre ich, Max Zimmermann, dass ich die vorliegende Arbeit gemäß § 22 Abs. 7 APB der TU Darmstadt selbstständig, ohne Hilfe Dritter und nur mit den angegebenen Quellen und Hilfsmitteln angefertigt habe. Ich habe mit Ausnahme der zitierten Literatur und anderer in der Arbeit genannter Quellen keine fremden Hilfsmittel benutzt. Die von mir bei der Anfertigung dieser wissenschaftlichen Arbeit wörtlich oder inhaltlich benutzte Literatur und alle anderen Quellen habe ich im Text deutlich gekennzeichnet und gesondert aufgeführt. Dies gilt auch für Quellen oder Hilfsmittel aus dem Internet.

Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Mir ist bekannt, dass im Falle eines Plagiats (§ 38 Abs. 2 APB) ein Täuschungsversuch vorliegt, der dazu führt, dass die Arbeit mit 5,0 bewertet und damit ein Prüfungsversuch verbraucht wird. Abschlussarbeiten dürfen nur einmal wiederholt werden.

Bei einer Thesis des Fachbereichs Architektur entspricht die eingereichte elektronische Fassung dem vorgestellten Modell und den vorgelegten Plänen.

Darmstadt, 13 November 2023

M. Zimmermann

Abstract

In the age of Artificial Intelligence (AI) there is an increasing interest in explainable AI (XAI), which deals with interpreting or explaining the decisions made by neural networks. In the field of computer vision, XAI methods are commonly applied on the task of image classification, while disciplines like image segmentation are often overlooked.

In this thesis, we follow a use case driven approach to extending current XAI approaches to image segmentation. Specifically, we firstly discuss current approaches about how explanation methods can be evaluated. In this context, we focus on human-study evaluations in AI-asssisted decision-making tasks and reiterate the evaluation metric "Agreement Task" from the HIVE framework [28]. Using our modified version of the Agreement Task, we evaluate different explanation methods on the simpler task of image classification. Our evaluations indicate that the recent prototype explanation method PIP-Net [38] outperforms comparable works in the Agreement Task.

Based on these results, we extend and adapt PIP-Net to provide explanations for image segmentation. To evaluate explanations for image segmentation, we also extend the Agreement Task accordingly and apply segmentation specific modifications on it. We apply our Agreement Task for image segmentation on our PIP-Net for image segmentation and compare it against the concurrent works ProtoSeg [45] and L-CRP [15]. In our tests, PIP-Net for image segmentation outperforms the other explanation methods in terms of interpretability.

Contents

1 Introduction

Imagine, you are outside in nature, maybe on a vacation in a very tropical country, and you find some type of flower, you've never seen before. You are curious and want to find out what kind of flower it is. Thanks to modern Artificial Intelligence (AI) technology, you are able to take a photo of that flower and let AI do the magic. Within seconds, you get information about what kind of flower you just spotted. But you wonder, how did the process of classifying the flower just work? On what basis did the AI decide that it's exactly this type of flower and not a similar looking type of flower? What is the AI's decision-making process?

Unfortunately, most modern AI is not able to answer these kinds of questions. Although AI is very good at accomplishing most tasks - often even better than us humans are - they fail to provide interpretability and do not allow for "a look inside" its reasoning process because of their "black-box" nature.

For the above example, the benefits of an interpretable AI model would just be convenience and satisfying curiosity, but there are a lot of applications where interpretability matters. Consider high-risk domains, like medical image analysis or autonomous driving, where AI could have huge benefits, but good performance alone will not be enough. In these use cases, a wrong decision can rarely be tolerated and might have significant consequences.

In situations like these, a look inside the AI's decision-making process can be important, for example to verify that the AI bases its decision on the correct reasons, to judge the correctness of its predictions and to assess the trustworthiness of the AI model in general [28].

Explainable Artificial Intelligence (XAI) is a field of research, that addresses these challenges. It allows AI models to be interpreted or let them explain themselves, i.e. making their behavior and their inner workings understandable to humans [16]. In the following, we will use explainability and interpretability interchangeably.

A classic discipline where XAI is applied is computer vision. When performing inference on images, e.g. by predicting the category/class of an object shown in the image, we can utilize XAI methods to visualize what parts of the image were the important factors in making the prediction. In the introductory example, the important parts of the flower could have been the shape of the blossoms or the type of stem.

Most methods in the realm of interpretable AI in computer vision are resided in the task of image classification, where a single label is assigned to the entire image. Other tasks, like object detection or image segmentation, where objects in the image are additionally localized, either by bounding boxes which contains the object or by pixel accurate segments, have received comparability little attention of interpretability works [45, 53, 15].

As many of the high-risk domains include image segmentation tasks, e.g. self-driving cars or medical imaging analysis, there is a need for XAI beyond the scope of image classification.

In this work, we want to contribute to making the task of image segmentation more interpretable. Specifically, we want to create an explanation method for image segmentation, that is *verifiably* useful in common use cases, where XAI is applied.

For an explanation method to have a good utility in a use case, there is a need for some kind of evaluation that asses its utility. Since there is no ground truth information about what is actually a good explanation, automated evaluation metrics are commonly used, that capture a proxy-task, which is usually disconnected from real use cases. First works have shown, that the scores from these automated evaluation metrics correlate badly with real world utility [28, 18]. As an alternative, human-based evaluation tests can be conducted. Human-based evaluations per design capture real use cases better and thus can be more representative for measuring the utility of explanations.

In the realm of image classification, recently conducted human studies show, that current explanations methods are still not reliable enough to be used in real world tasks [28, 48]. Possible reasons could be the limitation of some explanations only providing information about *where* something important in the images is but not showing *what* the important feature is. While there are explanation methods that provide both, e.g. in the form of a prototypical image that looks like the detected feature [9, 37], there are some concerns about a misleading notion of similarity within these models [39, 38, 24].

A promising new work from 2023 that utilizes prototypical explanations, called PIP-Net [38], claims to generate explanations that align better with human perceived similarity. This could potentially result in more intuitive explanations that can be verified to perform better in human-based evaluation metrics.

Our main contributions in this work are the following:

- We adapt and reinterpret an existing human-based evaluation metric for image classification [28], which we motivate based on a real life use case in the area of AI-assisted decision-making.
- We extend this evaluation metric to the task of image segmentation. To the best of our knowledge, we are the first ones to apply human-based evaluation metrics on image segmentation.
- We extend the existing XAI method PIP-Net [38] from image classification to image segmentation.
- We evaluate and compare PIP-Net both for classification and segmentation on our respective evaluation metrics. In the case of image classification, PIP-Net achieved an unmatched level of interpretability, which clearly stands out compared to other XAI methods. Our extension of PIP-Net to segmentation keeps up the high interpretability and outperforms all concurrent tested works.

This subsequent thesis is structured as follows:

In Chapter 2 we introduce background and related work regarding XAI methods, evaluation metrics and image segmentation. In Chapter 3 we discuss an evaluation metric for image classification, which we will extend to image segmentation in Chapter 4. In Chapter 5 we propose our XAI method for image segmentation, which is based on the Nauta et al.'s PIP-Net [38]. In Chapter 6 we first conduct our evaluation metric for image classification, then show the results of our new XAI method for segmentation, which we will afterwards evaluate in our evaluation metric for image segmentation. Finally, in Chapter 7, we conclude our findings and discuss future work.

2 Background & Related Work

In the last decade, Artificial Intelligence (AI) has gotten very popular, being adopted more widely year by year. Even though performance of AI solutions often times is unmatched by conventional algorithms, there is usually some concern against AI models, because of their "black-box" nature. Because of their high number of parameters, deep-learning models are so complex, that by looking at the parameters it's not possible to get an insight into the models reasoning and decision-making. This lack of transparency dampens the success of AI models in high-risk domains, where there can't be any unexpected behaviour. To mitigate these problems and allow for the usage of AI even in high-risk domains, there have been big efforts to make AI more interpretable. This research domain is called Interpretable AI or Explainable AI (XAI). XAI gives the ability to explain models (or let them explain itself) in a way that is understandable for humans and allows humans to "look-inside" the model and its decision-making process.

2.1 XAI Methods in Computer Vision

In XAI, there are many different explanation methods, that can be categorized along different dimensions:

- Implementation level: post-hoc and interpretable-by-design methods
- Explanation level: local and global methods
- Explanation type: E.g. attribution methods for vision tasks

Post-hoc explanations try to explain already trained models and usually work with different model architectures. When you want to obtain an explanation of a model, for which interpretability wasn't a concern before, post-hoc explanations are a straight-forward way. On the other hand, interpretable-by-design methods are specially designed models, that already incorporate interpretable structures. Usually, the explanation method then refers to the entire model architecture. Recent works show a trend towards interpretable-by-design methods, as they allow incorporating more flexible interpretability designs, as opposed to reverse-engineering black-box models in post-hoc methods [9, 38].

Further differentiation is done on explanation level. Global interpretability methods aim to explain a model as a whole, e.g. by inspecting their structures and parameters. Local interpretability refers to understanding a single prediction. Most methods can be categorized into either local or global methods, but some recent works combine both and are sometimes called "glocal" [1, 15, 9, 38].

Commonly used types of explanations in vision tasks include attribution maps and prototype networks. In the following, we cover both of them in further detail.

2.1.1 Attribution Maps

Attribution methods highlight which parts of the input image were relevant ("attributed") to the model's prediction. Usually this is done in the form of attribution maps (or often referred to as "heatmaps"). Attribution maps often times are gradient based, by using the gradient information to find out how much each pixel's change in value would contribute to the final output score. Intuitively, a big gradient relates to a high sensitivity of the output score to the respective pixel which in turn indicates a high contribution of that pixel to the prediction.

One of the earliest works [51] simply visualizes the magnitude of the gradient, whereas newer methods utilize more refined gradient information. For example, Grad-CAM (Gradient-weighted Class Activation Mapping) [47] computes a weighted gradient with respect to deeper layers of the network. Integrated Gradients [52] integrates the gradients along a path from a baseline input to the actual input. Layer-wise Relevance Propagation (LRP) [36] calculates attributions by propagation output scores back through the network using basic formulas. This way, the output scores are conserved and will be distributed through the layers to explain the network's decision. Concept Relevance Propagation (CRP) [1] extends LRP by also attributing output scores to hidden layers, thus revealing learned contexts by the network. Traditionally, attribution methods were predominantly post-hoc methods, but newer works like BagNet [5] or B-cos networks [3] realize attribution maps using interpretable model architectures.

2.1.2 Prototype Networks

Prototype networks are interpretable-by-design models that provide an explanation by highlighting different parts of the image and comparing them to a prototypical part (prototype), that should represent the highlighted part. Prototype networks are inspired by human concepts and aim to solve visual tasks the same way that humans do. E.g. [38] is built to mimic the recognition-by-components theory, which is how humans segment objects into individual parts to recognize them. The authors of [9] compare the method of reasoning of their model to the way that for example a radiologist compares potential tumors in medical imaging to known prototypical images for diagnosis, and with it introduced the phrase "this looks like that".

Prototype networks usually work by first finding meaningful features inside the image, which in a next step are compared to learned prototypical features. Finally, based on similarity to these prototypes and relevance for each class, the relevant features are combined to obtain a final prediction. For example, for a classification of a bird species, the prominent features present in the image are compared to different prototypes, e.g. for the beak, the tail or its feathers. If the beak *looks just like* the prototypical beak of a specific type of bird, and the feathers match the appearance of this bird's features, the prototype network will likely predict this bird species.

In the landscape of prototype networks, there are many different approaches with different emphases. In the following, we will cover the most relevant prototype networks with regard to our work.

ProtoPNet [9]

Introduced by Chen et al. [9], Prototypical Part Network (ProtoPNet) aims at providing intuitive explanations for fine-grained classification tasks. It uses a predefined number of prototypes (e.g. 10 per class), which are represented in the latent feature space, i.e. n-dimensional feature vectors, before the classification layer. In a first training step, without considering the classification weights, the encoded image patches are clustered

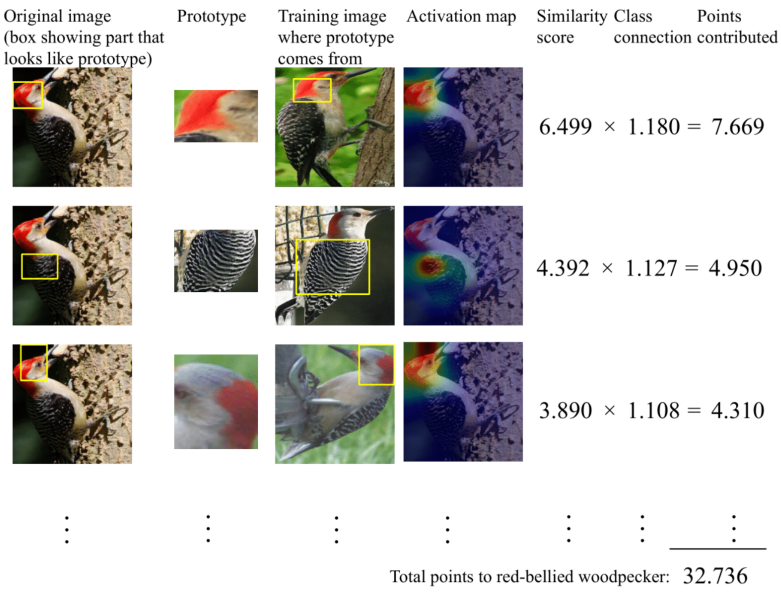


Figure 2.1: Exemplar explanation of ProtoPNet [9]. For each recognized prototype, the explanation highlights which part of the original image looks like the respective recognized prototype.

around these randomly predefined prototype vectors. The goal is to get a semantically meaningful latent representation, i.e. clusters that represent unique semantic parts of classes. Specifically, this is done by minimizing the Euclidean distance between the latent images patches and a prototype vector belonging to the same class, while maximizing the distance to all prototypes from different classes. In a later step, the classification weights are optimized, while freezing all previous layers, to obtain good classification results.

For classifying an image, ProtoPNet computes the similarity based on the distance in latent space between the image patches and prototypes. The similarity scores for each prototype are weighted using the learned classification weight and summed up to obtain the final class scores. The explanation shows, which image patch contributed to which amount to the final prediction and also visualizes their corresponding prototypes. A full ProtoPNet explanation is shown in Figure 2.1.

PIP-Net [38]

While gaining a lot of traction, ProtoPNet has been shown to have some open challenges which lead to multiple follow-up works being introduced [37, 45, 38, 56, 43, 44]. We cover open challenges and possible flaws of ProtoPNet (and other prototype networks) under Subsection 3.2.1, where we discuss why explanation methods could lead to bad result in evaluation metrics.

Here, we will cover another interesting prototype network approach called PIP-Net (Patch-based Intuitive Prototypes Network) [38], which follows a unique approach and aims to provide a few advantages over to previous related works. Specifically, PIP-Net is said to provide semantically meaningful prototypes, that are intuitive and align better with human perception of similarity. Additionally, it uses a scoring-sheet evaluation, that provides an interpretable classification process. As PIP-Net is a central part of this thesis, we will cover it in greater detail.

Compared to common approaches like ProtoPNet [9], PIP-Net clusters prototypes not by measuring distance in latent-space, but instead by using a contrastive learning approach. Two differently augmented versions of the same image are optimized to have the same latent representation. As the augmentations are selected to make augmented images look similar to the human-eye, the network learns a more human perceived notion of similarity. Instead of sampling negative examples for the contrastive loss, PIP-Net learns diverse image representations using a custom loss function, that ensures that all available prototypes are used in training. This makes the process to learn semantically similar prototypes completely self-supervised, i.e. no image labels are required.

PIP-Net improves intuitiveness further by normalizing the latent feature vectors using a softmax function, to obtain a near one-hot encoded vector. This helps interpretability, as an encoded image patch corresponds to exactly one prototype. To classify an image, the one-hot encoded feature vectors are fed through a fully connected classification layer, which is optimized to be sparse and only contains non-negative weights. By this, a prototype can only have positive attribution for a class, while ensuring compact explanations consisting of only a few prototypes. This results in the mentioned scoring-sheet, where each prototype (which is either present or not present) contributes the respective classification weight to a certain output class. The output score used for prediction is simply the sum of all prototype contributions. Irrelevant prototype-class combinations have a 0 weight.

PIP-Net's approach also results in some interesting properties. As prototypes are not pre-assigned to classes, they can contribute to multiple classes as well as no classes at all. This results in a flexible number of total prototypes, as prototypes that are not relevant for any class, will be removed after training. Furthermore, as prototypes only focus on certain image features, PIP-Net can abstain from a prediction, e.g. when out-of-distribution data, that doesn't contain learned visual features, is presented.

More implementation-specific details of PIP-Net can be found in Section 5.2, where we adapt PIP-Net to the task of image segmentation.

2.2 Evaluation of Explanation Methods

With a big variety of explanation methods available, there is the need for methods to evaluate the utility of these explanations. Unfortunately, there is a lack of standardized evaluation for XAI methods [28]. A reason for this is the unavailability of ground-truth data about what actually is a good explanation. Current evaluation approaches can be categorized into 3 different types:

1. Automated Evaluation Metrics
2. Qualitative Examples
3. Human-based Studies

2.2.1 Automated Evaluation Metrics

Automated evaluation metrics are commonly used to evaluate attribution methods. To determine the utility of an explanation, artificial proxy-tasks are used, which are usually disconnected from real use cases and define a broad objective for an explanation to be good or correct. A typically used proxy-task is the "Pointing Game", where the attribution map is compared to a predefined region (segmentation or bounding-box). One variant

deems an attribution map as correct, when its highest intensity pixel lies within that region [58], while other variants calculate how much energy falls into the region [55], or measure the ratio of overlap between a bounding box and the attribution [59]. A different method, which doesn't require additional annotation, uses a grid of multiple images and measures how much attribution is assigned to the correct image in the grid [4].

Contrary to Pointing Games, Pixel Deletion Protocols [4, 19, 8, 22] measure how much of an image can be removed while still preserving the original classification performance. Pixels are removed in increasing order of determined attribution. Intuitively, if all the pixels with low attribution scores can be removed without a change in performance, the attribution is considered good.

Interestingly, there exist perturbation based explanation methods [19], that generate attribution maps utilizing the same principle. This means, that an XAI method could essentially be evaluated by itself, allowing for theoretically perfect results, which highlights a central flaw of automated evaluation metrics.

The main critic point about automated evaluation metrics is that they are disconnected from real-world use cases. Automated evaluation metrics usually follow some loosely-defined heuristics by developers, that might not relate to utility in real world scenarios. [28] additionally shows, that the scores of automated metrics do not correlate with human-based evaluation scores. Finally, almost all automated evaluation metrics are exclusively defined for attribution maps, preventing the usage on and comparison to other types of explanations like prototype networks.

2.2.2 Qualitative Examples

Many works, especially interpretable-by-design methods, demonstrate the utility of their methods using qualitative examples. These qualitative evaluations often do not allow for cross-method comparison, as they are specific to the respective explanation method. Furthermore, the utility of these methods could be overrated by hand-selecting exclusively good examples. Recent works have shown, that interpretability often times is not as good as initially claimed [2, 25].

2.2.3 Human-based Evaluations

In recent years, there has been a trend of conducting human based evaluations in vision tasks [28, 48, 18]. [48] investigates, whether showing an explanation increases the ability of humans to detect incorrect predictions. The authors found out, that instead of increasing the guessing accuracy, explanations actually decreased the ability to identify incorrect predictions. [18] and [28] conduct studies about users predicting the model outputs.

Most similar to our work is HIVE by Kim et al. [28], which we will describe in more detail in the following.

HIVE [28]

HIVE (Human Interpretability of Visual Explanations) [28] is a recently introduced human-based evaluation framework for XAI methods in vision tasks. HIVE is developed according to the principles of falsifiable hypothesis testing, human-centered evaluation and cross-method comparison. The 2 evaluation tasks introduced in HIVE are used to find out the utility of explanation methods for the application area of AI-assisted decision-making. The first task, "Agreement Task", shows a prediction explanation pair, on which users should

assess their confidence in the prediction. In the second "Distinction Task" users should identify the correct prediction out of 4 given explanations. Both tasks are carried out on 2 different datasets (CUB birds [54] and ImageNet [42]), using 4 different explanations of different types (Grad-CAM [47], BagNets,[5] ProtoPNet [9] and ProtoTree [37]).

Using the Agreement Task, the authors discovered a confirmation bias. While users could confidently identify correct prediction as correct based on the provided explanations, they tend to also believe that incorrect predictions were correct. We will comment on the Agreement Task and the issue of confirmation bias extensively in Chapter 3. The Distinction Task showed, that users overall struggle to identify the correct prediction, when given multiple choices. For correct model predictions, users were more likely to guess the correct ground-truth class, compared to incorrect model predictions.

In general, the results of HIVE show, that current evaluation are not reliable enough to be useful for AI-asssisted decision-making tasks. Additionally, a poor correlation between several automated evaluation metrics and human studies could be observed, indicating that automated evaluation metrics are less suited for measuring the utility of explanations in these tasks.

2.3 Semantic Segmentation

Next to the task of image classification, which is extensively covered in XAI, there exist more sophisticated tasks in computer vision, like image segmentation.

Image segmentation roughly describes the segmentation of an image into multiple parts that belong together. While it is hard to characterize and define what "belongs together", historically there have been many classic computer vision approaches at segmenting images.

A more well-defined sub-task of image segmentation is semantic segmentation. It describes the segmentation of an image in combination with recognizing each object, i.e. assigning each segment to a category. Semantic segmentation tasks are usually solved with the approach of classifying each pixel individually. While a first approach uses solely Conditional Random Fields (CRFs) [50], more recent approaches are deep-learning based using CNNs [35, 41, 40, 20, 10, 11].

Over the course of this thesis, we will be discussing the task of semantic segmentation and we will use it interchangeably with the terms image segmentation or segmentation.

Fully Convolutional Networks (FCNs) [35] set the basis for future work by employing an encoder-decoder structure. The encoder consists of a CNN backbone, whose output is the so-called "bottleneck", i.e. the layer with the lowest spatial resolution. This layer is then upscaled by the decoder to the size of the original image. In its simplest implementation, the decoder consists of a bilinear interpolation or a parameterless deconvolutional network. To refine the segmentation, more advanced methods were introduced, like deconvolutional layers with learnable parameters and skip connections. Skip connections allow passing more granular information from intermediate encoder layers directly to the intermediate layers with same spatial dimensions in the decoder.

One noteworthy segmentation model called U-Net by Ronneberger et al. [41] uses skip-connections to obtain high-resolution features for the task of segmenting medical imaging. Other works improve the model's performance by combining CNNs with other previously employed non-deep learning based mechanisms, like CRFs [7, 32] or MRFs [34]. In the series of the DeepLab networks [10, 12, 11] Chen et al. utilized dilated convolutions and spatial pyramidal pooling to set a new milestone of segmentation networks.

2.3.1 XAI Methods for Image Segmentation

While most of the literature on XAI in computer vision focuses on the task of image classification, only a few works explore image segmentation. In this section, we give an overview over all to our knowledge existing XAI methods for image segmentation.

Seg-Grad-CAM [53]

Seg-Grad-CAM is an extension of the Grad-CAM [47] attribution method for image classification, which we already mentioned in Section 2.1.1. The standard Grad-CAM method describes the attribution of the last convolutional layer for a class of interest. As segmentation produces output scores for each pixel, Seg-Grad-CAM works by summing up all output scores in a region of interest, e.g. all pixels belonging to a segment of a predicted class or a single pixel.

The authors tested their method on the U-Net segmentation model [41] on the Cityscapes dataset [13]. They discovered, that the layer at the bottleneck tend to produce more informative explanations than the final convolutional layer. In their work, Vinogradova et al. show simple qualitative examples that produce plausible results, e.g. that the attribution for the class "Sky" highlights a tree, which could be informative context for its prediction.

As Seg-Grad-CAM is at the time of writing the most cited work for interpretable image segmentation, there have been a few follow up works [27, 46, 21]. Humer et al. [27] generalized the notion of the region of interest in Seg-Grad-CAM to produce classification-like outputs for segmentation models. Schorr et al. [46] integrated Seg-Grad-CAM, as well as their extension of Guided Grad-CAM [47] to segmentation into the explanation toolbox "Neuroscope". Hasany et al. [21] extend HiResCAM [14] for classification to the segmentation explanation method Seg-XRes-CAM in a similar fashion as Seg-Grad-CAM.

Grid Saliency

Grid-Saliency [26], introduced by Hoyer et al., generates attribution maps using perturbations. Specifically, they find an optimal attribution map by replacing every pixel outside of the attribution by uninformative values. As many pixels as possible are perturbed, while the confidence of the original prediction must still be matched. For segmentation, only the predictions in a region of interest are considered, e.g. all pixels belonging to a specific class segment. Grid-Saliency also produces context explanations that highlights the attribution to a class segment outside the segment itself, e.g. when a rider of a bike is classified as rider instead of person because of the context given by the bike. Additionally, the authors create a synthetic dataset that can be used to evaluate the amount of context used by a segmentation network. Unfortunately, the resources of this work are not publicly available and thus could not be utilized in this thesis.

U-Noise [29]

U-Noise is a perturbation based, post-hoc explanation method, that uses 2 models to generate explanations. The model to interpret is called "Utility Model" and the 2nd "Interpretability Model" is used to generate the perturbation noise. Its output is the standard deviation of the noise for each pixel, that is tolerated on the input image. The overall optimization goal is the maximization of the noise for each pixel from the Interpretability

Model, while maintaining a good classification loss on the Utility Model. The chosen architecture for both models is U-Net [41].

ProtoSeg [45]

ProtoSeg [45] by Sacha et al. is an extension of the prototype network ProtoPNet [9] to image segmentation. To our knowledge, ProtoSeg is the only other prototype network for image segmentation, and thus comes closest to our work. ProtoSeg builds on top of the semantic segmentation networks U-Net [41] and DeepLab [10] and implements their prototype architecture after the final layers. While most of the approaches of ProtoPNet are kept, the originally used cluster and separation loss are removed. Instead, a new prototype diversity loss is added, which enforces that prototypes of the same class are activated in different parts of the image and is based on KL divergence. ProtoSeg is applied on the Cityscapes [13] and Pascal VOC 2012 [17] datasets and produces slightly lower mean intersection over union (mIoU) scores than their non-interpretable baselines. Regarding evaluation of interpretability, only a limited number of qualitative examples are presented.

L-CRP

L-CRP [15] is an extension of the explanation method for image classification CRP [1], which we mentioned in Subsection [**subsec:AttributionMaps**]. L-CRP is designed to work both on object detection and image segmentation tasks. As an extension of CRP, L-CRP takes similar approaches to Seg-Grad-CAM [53] and similar models, by integrating the attributions for output pixels of the prediction. Pixels are taken from a region-of-interest that can e.g. be a specific class output.

3 Discussing an Evaluation Metric for Image Classification

In this chapter, we describe on a general level, what our desired evaluation metric should look like. Firstly, we use the literature to identify important use cases of XAI. Afterwards, we decide on a use case that we want to inspect in our evaluation metric. Our chosen use case is already captured in the HIVE's Agreement Task [28], but interestingly it's interpreted differently. We discuss possible problems with the interpretation of HIVE's authors and propose a new way to interpret the results.

3.1 Use Cases of Explanations

To develop an XAI method that works well in practice, it's crucial to know and define precise objectives. For this, we first need to identify possible use cases of XAI, finding out where and to whom it can be useful.

Zednik [57] covers XAI from a theoretical point of view and introduces multiple stakeholders, each requiring different kinds of explanations that relate to different questions asked about the model. Stakeholders that use, operate or are affected by AI systems usually ask more high-level questions like "*What* does the model do?" and "*Why* does the model do what it does?". Creators of AI systems (developers) typically ask questions on an algorithmic level relating to *How* the system does what it does and *Where* on the algorithmic level certain concepts are realized.

[16] distinguishes between the 2 stakeholders: *ML system developers* and *end-users*. For developers, XAI can be used to better understand the underlying problem the model tries to solve, give insights to why a model failed or might fail and even could help to obtain new insights about the model's decision-making. For typical end-user tasks like AI-assisted decision-making, XAI methods can provide details about *why* a model made the decision it made. This increases user's trust in the model and can generally encourage the adoption of AI systems in high-stakes domains.

We follow the proposed division between developers and end-users, and will describe a typical use case respectively in more detail. For developer use cases we cover "Model-debugging/understanding", for end-users the typical use case of "AI-assisted decision-making" is described.

Model-Debugging/Understanding

A popular use case for developers is to identify defects in the model that might lead to incorrect behavior or model predictions. In that regard, commonly mentioned are *context-biases*. Context-biases describe the case, where an object is recognized by its context. As described in [49], context-bias can be advantageous, e.g. when ambiguity of similar appearing classes can be solved using context, but also can be lead to errors

when objects appear in previously unseen context. An often times mentioned story describes a neural network for detecting tanks, which due to bad data collection learned to discriminate the images based on irrelevant context, e.g. the background scene, instead of actually focusing on the tank - the object of interest. Using common XAI methods like attribution maps context-biases can be identified, which leads to less biased and generally more robust models.

AI-Assisted Decision-Making

In critical/high-stakes tasks where AI is applied, it is important that the decision made by the AI is trustworthy and based on the correct reasons. Here, explanations provide additional information to the prediction, which can allow for better evaluation of the decision made. According to [28], providing explanations generally increases user's trust in the prediction made by AI models. Explanations are particularly relevant in difficult tasks, like medical imaging analysis or autonomous driving, where the AI's decision-making capabilities go beyond those of humans. Consider the example of detecting a tumor on an MRI or CT scan. As this is a naturally difficult task, AI could be leveraged to help detect such malformations. To reduce the risk of misclassification, a surgeon should always double-check the predictions by the AI. Here, an additional explanation to the prediction itself could be useful. With the explanation, the operator can take a look inside the model's reasoning process and can identify potential errors. This allows a better evaluation of the prediction and ultimately leads to a more profound decision-making.

With now different use cases of XAI identified, we select a use case that we want to model in our evaluation metric. We follow approaches from [28, 48, 18] and choose to evaluate the performance of XAI methods in the use case of an AI-assisted decision-making scenario, like the example described above.

3.2 Revisiting HIVE's Agreement Task [28]

With the decision that we will model a use case of an AI-assisted decision-making scenario, we need to find a representative evaluation metric for this use case. As already described in Section 2.2, a good choice for this is a human-based evaluation metric, as these capture real-life use cases better than other types of evaluation metrics.

The in HIVE [28] introduced Agreement Task captures our outlined AI-assisted decision-making use case, which we described in the previous section, quite well. Because of that, we decide to move forward and base our evaluation metric on the Agreement Task.

Described in Subsection 2.2.3, in the Agreement Task the participants are presented an image in combination with an explanation for a certain prediction. The participants are asked to assess how confident they are in the model's prediction being correct, based on the provided explanation. An example of HIVE's Agreement Task is shown in Figure 3.1.

Interestingly, in the original work, the Agreement Task was only used to measure the amount of confirmation bias and was not employed as a dedicated evaluation metric. In the following, we will take a closer look at the Agreement Task and inspect the author's thesis of confirmation bias.

Kim et al. [28] conducted the Agreement Task on 4 different methods [9, 37, 47, 5] consisting of attribution maps and prototype networks, on 2 different datasets [54, 42]. Over all combinations, the users tend to believe the explanations and deemed the prediction to be correct, independent of it actually being correct

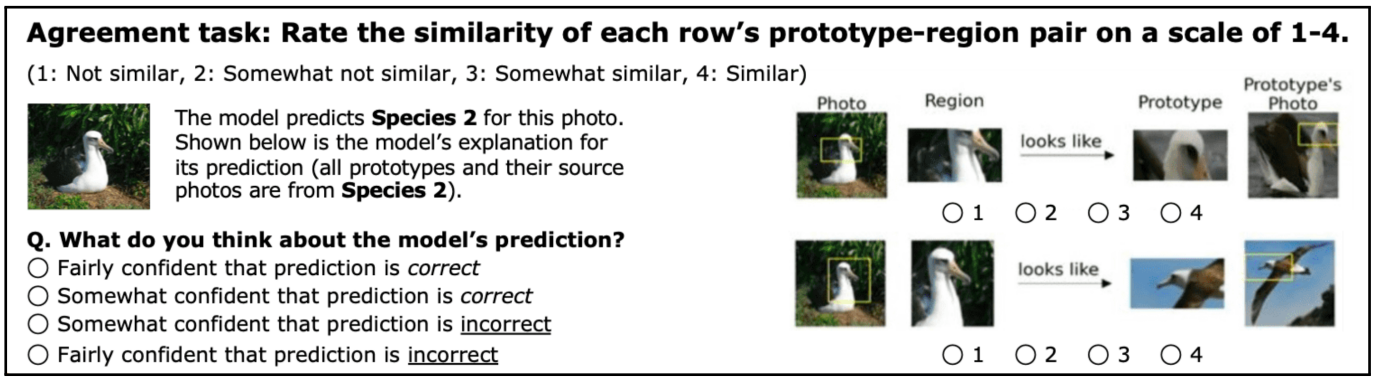


Figure 3.1: User Interface of HIVE's [28] Agreement Task using explanations from ProtoPNet [9].

or incorrect. More precisely, for correct predictions, about 70 % of the users believed the prediction to be correct. However, for incorrect predictions, only roughly 40 % of the users identified the incorrect prediction as incorrect. This means, that for incorrect predictions, about 60% of the time users also believed the predictions to be correct based on the explanation. Precise Agreement Task results from [28] are shown in Table 3.1. We report mean accuracy and standard deviation.

Table 3.1: Results of HIVE's Agreement Task [28]. For each score, the mean accuracy and standard deviation is shown. **Bold** denotes the highest performing method in each group.

Dataset	Explanation Method			
CUB [54]	Grad-CAM [47]	BagNet [5]	ProtoPNet [9]	ProtoTree [37]
Correct	72.4 ± 21.5 %	**75.6 ± 23.4** %	73.2 ± 24.9 %	66.0 ± 33.8 %
Incorrect	32.8 ± 24.3 %	42.4 ± 28.7 %	**46.4 ± 35.9** %	37.2 ± 34.4 %
ImageNet [42]	Grad-CAM [47]	BagNet [5]	-	-
Correct	**70.8 ± 26.6** %	66.0 ± 27.2 %	-	-
Incorrect	**44.8 ± 31.6** %	35.6 ± 26.9 %	-	-

The authors of HIVE claimed, that these results show an issue of confirmation bias. According to its literal definition, this would suggest, that there is a preexisting belief of explanation-complemented predictions being correct, which users then affirm by deeming the prediction correct without regarding the explanation itself. Apart from small modifications that restrict the usage of human prior knowledge, the Agreement Task closely corresponds to our earlier defined use case. This would imply, that this kind of use case could hardly be employed in real-life, as operators or users would also be susceptible to this confirmation bias. We find this conclusion rather unsatisfying, and suspect, the reason for the bad human accuracy on incorrect predictions might be more complex than simply a confirmation bias. Instead, we believe, that the bad accuracies might relate to flaws of the explanation methods tested themselves, which we will describe in more detail below.

Afterwards, in Section 3.3, we highlight why considering human accuracy on incorrect prediction should be an important requirement for a useful explanation method.

3.2.1 On the Topic of Confirmation Bias

While not contesting the issue of confirmation bias itself, we suspect that the occurrence of confirmation bias is not a problem of explanations in general. Instead, we think, that the tested XAI methods are not able to

highlight incorrect predictions clear enough. As explanations for incorrect predictions don't look distinctly *incorrect enough*, a possibility for confirmation bias is created, which finally results in poor performance on the Agreement Task.

We later claim, that an ideal explanation method should diminish the effect of confirmation bias by clearly pointing out to the user, when a prediction is incorrect.

Attribution Maps in HIVE

To support our claim, that the tested explanation methods are unable to highlight incorrect predictions well enough, we provide a simple example for attribution map explanations. In our example, we use the well-established attribution methods Integrated Gradients [52] and Grad-CAM [47], but the same observations can be made for other attribution methods as well.

In Figure 3.2, we provide the input image and attribution maps for both correct and incorrect predictions. It is visible, that both the explanations for correct and the incorrect predictions, look similar to each other. All of them highlight the main subject or parts of its individual features. HIVE's Distinction Task [28] showed, that in direct comparison (at least for correct predictions) users are generally able to select the correct prediction, based on the shown attribution map. While this suggests, that some attribution maps look more correct than others, we think, that when looking at a single explanation there is too little visible indication of an attribution map that suggests a prediction is incorrect. If we imagine, that in Figure 3.2, only the attribution maps for incorrect predictions are shown, it would be very difficult to identify them as incorrect. This could lead to many false positives guesses (incorrect predictions are believed to be correct), which we think is the main reason for the observed low accuracy on incorrect predictions in the Agreement Task in the case of the tested attribution methods.

This assumption is also in line with the Distinction Task scores for incorrect predictions. Here, users were not significantly better than chance in identifying the correct prediction, using the provided attribution maps.

It should however be noted, that this concern might not be a systematic problem of attribution maps in general. We only observe, that current attribution methods provide limited interpretability in that sense.

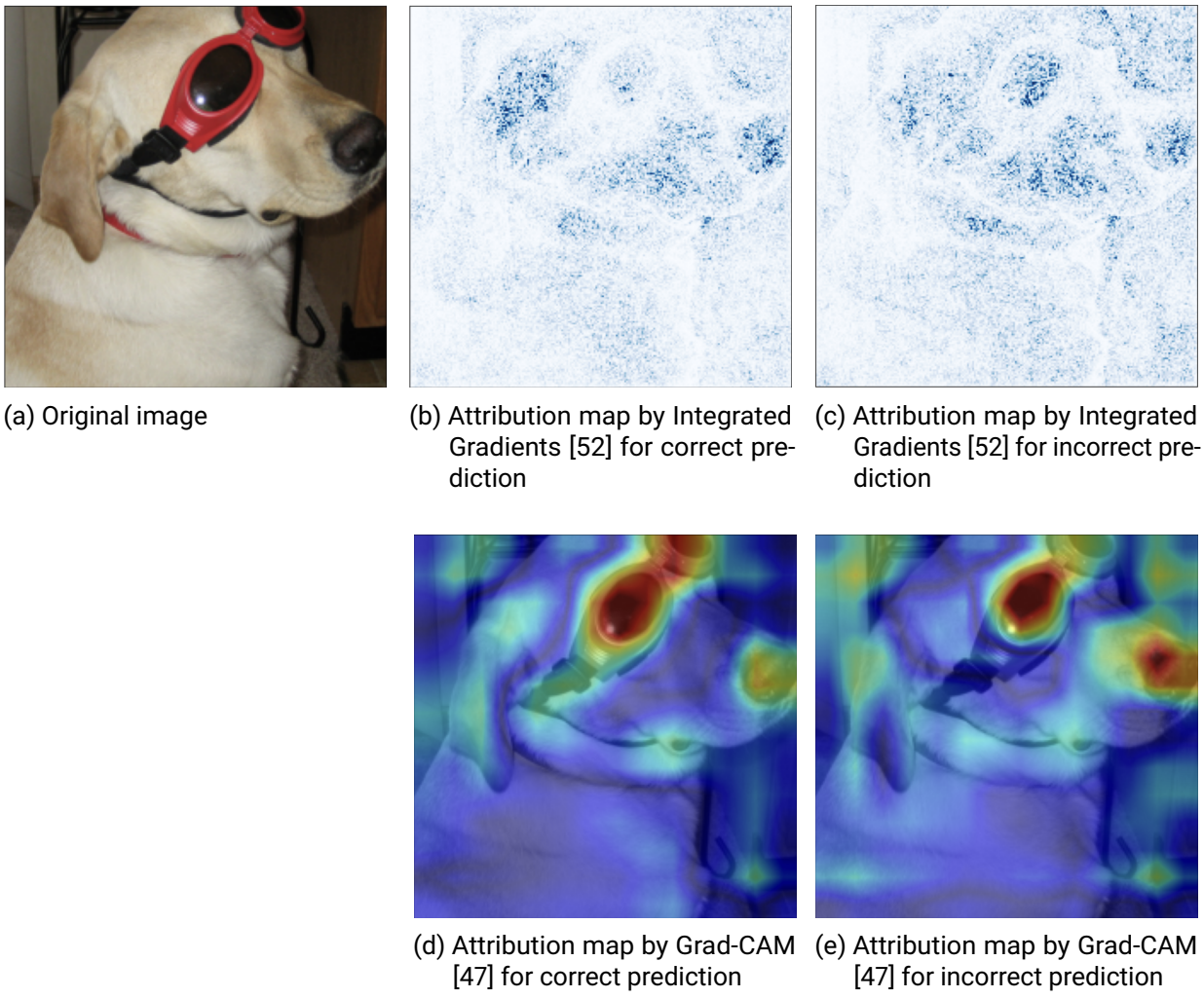
Prototype Networks in HIVE

When looking at Prototype explanations, they generally provide more detail than attribution maps per design. As other works have pointed out, they not only show *where* something important in the image is, but also show *what* the model detected [9, 15]. Specifically, they show a matching prototype image for a relevant region in the original image. In theory, when there is more information in an explanation, there is also more information that could indicate an incorrect prediction.

In practice, the prototype networks evaluated in HIVE [28] on average only perform marginally better than attribution methods. As can be seen in Table 3.1, the accuracy on incorrect predictions is best for ProtoPNet [9], whereas ProtoTree [37] is already worse than the attribution method BagNet [5].

Possible reasons for the worse than expected performance could be related to the details of ProtoPNet and ProtoTree. As ProtoTree is based on ProtoPNet, arguments for ProtoPNet apply for both prototype networks.

One of the most significant problems of ProtoPNet is, that many of its learned prototypes are not aligned with human notion of similarity [24, 38, 39], i.e. image patches deemed similar by ProtoPNet do look similar to



(a) Original image

(b) Attribution map by Integrated Gradients [52] for correct prediction

(c) Attribution map by Integrated Gradients [52] for incorrect prediction

(d) Attribution map by Grad-CAM [47] for correct prediction

(e) Attribution map by Grad-CAM [47] for incorrect prediction

Figure 3.2: Original image from Imagenet [42] with Integrated Gradients [52] and Grad-CAM [47] attribution maps for both correct and incorrect predictions.

the human eye. This could be, because similarity in ProtoPNet is measured in latent space, i.e. on class-level, which can be disconnected from the visual appearance of the respective image parts. For example, the beak and the tail of the same bird species would be considered similar in ProtoPNet, even though they don't share any visual similarity. These findings align with those of [31, 28, 25], which point out that there exists a "semantic gap" between the learned prototypes and human concepts of prototypes and their similarity.

Nauta et al. identify further points in ProtoPNet that could hurt human interpretability. They show, that ProtoPNet's explanations contain redundant prototypes [39], i.e. multiple prototypes that capture the same visual feature and also argue that ProtoPNet's explanation size is too big. Furthermore, the authors of [6] point out, that prototypes are not localized well and report issues regarding pixel grounding and heat map visualizations.

Conclusively, prototype networks based on ProtoPNet still face open challenges, that could be the reason for their worse than expected performance in HIVE's Agreement Task.

3.3 A good XAI Method should make it clear when a Prediction is incorrect

With possible points identified, why explanation methods tested in HIVE's Agreement Task [28] performed poorly regarding incorrect predictions, we now have plausible reason to assume that the underlying problem is not confirmation bias of explanations in general. Instead, we think it should be possible for a *good* explanation method to clearly highlight an incorrect prediction, i.e. provide enough evidence for users to make it clear when a prediction is incorrect.

Based on this hypothesis, we repurpose HIVE's Agreement Task as a stand-alone evaluation metric without the limitation of only measuring confirmation bias.

While keeping its general methodology unchanged, we introduce the additional requirement for explanations to be able to make it clear, when a prediction is incorrect, which is measured using the human accuracy on incorrect predictions. In this context, it's important to also consider the accuracy for correct predictions. An explanation method, which always produces bad explanations, would often times lead users to think the prediction is wrong and ultimately would result in a good accuracy for incorrect predictions, which is not desirable.

The repurposing of HIVE's Agreement Task obviously raises the question of how explanation methods can meet the proposed requirements. Below, in Section 3.5, we propose to evaluate a novel prototype network called PIP-Net [38], and explain why it could potentially provide better accuracies in the Agreement Task. In the experiments (Section 6.1), we evaluate PIP-Net on the Agreement Task and discuss the results.

After reinterpreting HIVE's Agreement Task, we additionally spot some potential design improvements to reduce the influence of human prior knowledge.

3.4 Preventing the Usage of Human Prior Knowledge

To ensure, that users actually use the provided explanations, evaluation tasks must be designed carefully to prevent that users are unable to use prior knowledge to solve the tasks.

In the following, we first present the measures taken in HIVE [28] to reduce the usage of prior knowledge. Afterwards, we discuss a possible weak point in HIVE's Agreement Task, that could potentially lead to humans using prior knowledge and which we will further investigate using ablations of the Agreement Task.

3.4.1 Measures taken in HIVE [28]

To ensure, that users actually use the provided explanations to solve the evaluation tasks, and not rely on prior knowledge, [28] introduced 2 different measures:

1. Usage of fine-grained classification tasks
2. Omitting semantic class labels

Fine-grained classification tasks like the CUB dataset [54], which consists of 200 different bird species, are usually hard to solve for humans - assuming most of the people are non-bird experts. Here, the main source of information to solve the task is the explanation. Even ImageNet, which is the other dataset used in HIVE, traditionally consists of 1000 different classes. With many semantically similar classes, e.g. different types of animals, the usage of human prior knowledge is also limited.

Additionally, the authors of HIVE decided to remove semantic-class labels. Without the removal, users would be able to compare the class they recognized in the original image and compare it with the semantic class label of the prediction. For fine-grained datasets like CUB [54], this is less relevant, as people still might not be familiar with the specific bird species' name. In the case of coarser-grained datasets, this is more of a concern.

3.4.2 Original Images of Prototypes

Even with the measures taken in HIVE, we observed a potential weak point in the case of prototype explanations, where users could take advantage of their prior knowledge. Taking a look at the Agreement Task in Figure 3.1, on the right we can see that additionally to the prototype, the prototypes' original image is shown.

This allows users to compare the input image with the prototype's original image, instead of only comparing the image region and its respective prototype. While technically being part of the explanation of prototype networks, this kind of information can be provided for all kinds of models without the need for an interpretable model or a sophisticated post-hoc explanation. For example, one could save the class output scores of all images in the training data and in the explanation show the training image that relates to the highest output score of the class predicted.

As this could potentially influence the results on the Agreement Task, we propose 2 ablations of the Agreement Task to evaluate the influence of showing the prototype's original image. For the first ablation, we only show the prototype and omit the prototype's original image. In the second ablation, we omit the prototype and only show the source image of the predicted class. We selected the source image based on highest output score over the training data, like mentioned above.

Finally, we simplified the confidence assessment compared to HIVE [28]. As can be seen in Figure 3.1, in HIVE's Agreement Task, users had 4 options when evaluation the prediction: Fairly confident/somewhat confident that the prediction is correct/incorrect. Instead, we simply provide a binary choice: The prediction is correct/incorrect. This keeps the task simple, while, for us, providing sufficient information for evaluation.

3.5 PIP-Net as promising Explanation Method

A promising new prototype network, called PIP-Net [38], is able to learn prototypes that correlate better with human notions of similarity, compared to works like ProtoPNet [9] or ProtoTree [37]. PIP-Net does this by leveraging a self-supervised representation learning, that aligns two views of an augmented image patch to the same prototype. This way the prototypes are aligned based on the same appearance in the image, rather than on class level, as it is done previous works [9, 37]. More information on PIP-Net can be found in Related Work in Section 2.1.2.

We believe, that PIP-Net could achieve better scores than previously tested methods and will thus evaluate it using the Agreement Task. Experiments and results are described in Chapter 6.

4 Designing an Evaluation Metric for Image Segmentation

In Chapter 3, we set the theoretical foundations for our human-based, use-cased-centered evaluation task and reinterpreted previous work to get an evaluation metric for image classification. In this chapter, we want to create an evaluation metric specifically for the task of image segmentation, for which we extend our adapted version of HIVE's Agreement Task [28].

4.1 Adapting the Agreement Task to Segmentation

When adapting the Agreement Task to segmentation, we first need to find a binary criterion, i.e. something that can be evaluated as correct/incorrect. In classification, a single prediction could be either correct or incorrect, whereas for segmentation, the evaluation of a prediction is more complex. Usually, a segmentation (referring to the prediction in a segmentation task) is not about being correct or incorrect, but instead, about its quality or how precise it is. Its quality is usually measured in metrics like mean Intersection Over Union (mIoU), referring to the overlap of a predicted segment with the ground truth segment, or the average classification accuracy over all pixel over the image, i.e. how many pixels were assigned to the correct class.

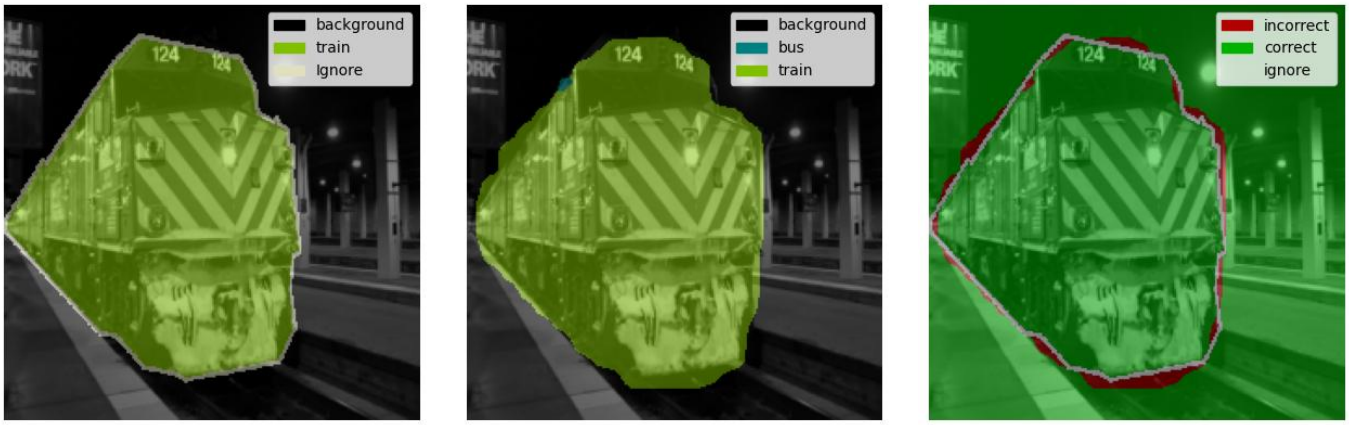
To get a binary choice like in classification tasks, our approach is to select a small segment or a single pixel, that has been assigned either to the correct or incorrect class. By this, we can ask users if the prediction *at the specific location* is correct or incorrect.

To get adequate pixel locations for human evaluations, we have to sample pixels in a specific way. In the following, we will describe problems, that would arise in a completely random sampling setup, and propose our solution.

Pixel Sampling Strategy

For our Agreement Task, we want to sample both correctly and incorrectly classified pixels. When we take a look at current segmentation models on common datasets like Cityscapes [13] or Pascal VOC [17], we see that these models achieve very good pixel accuracies of up to 90 % and more.

For a standard segmentation prediction, shown in Figure 4.1, we can see, that pixels that were misclassified are usually located in boundary regions between two segments of different classes. In this case, the misclassifications on pixel level happen because the segment boundaries aren't predicted precisely enough, while the class segments themselves are predicted correctly in most of the cases. These kinds of incorrect predictions are different from those in classification tasks, where misclassifications always relate to an entire image being classified incorrectly.



(a) True label (b) Prediction (c) Incorrect/correct predicted

Figure 4.1: Example of a typical segmentation prediction compared to its true label, on the Pascal VOC dataset [17].

Preliminary tests showed, that using these kinds of pixels as incorrect samples in our human evaluation task is disadvantageous for 2 reasons.

Firstly, pixels close to the segment boundaries are often times ambiguous and hard to guess right - even for humans. Using these pixels would introduce another level of complexity to our task, which could distort the evaluation results. In the same context, we think, that small errors on the boundaries are tolerable and are not too meaningful when regarding incorrect predictions. A segmentation that matches its ground truth almost perfectly is still considered good enough. We solve this by only sampling pixels (both correct and incorrect) that have at least a certain distance to the closest incorrect/correct classified pixel. In that case, the sampled pixel is far enough away from the ground truth segment boundary which should make it more distinctive to humans, what the pixel actually represents.

Secondly we observed, that incorrectly classified pixels due to wrong segment boundaries, even when the distance threshold is met, don't produce any meaningful explanations. Let's consider the example of a segment of the class "plane" that is predicted too big and is next to a segment of class "background". We would now observe pixels that were predicted as plane but are actually background pixels. Our preliminary tests showed, that both prototype explanations and attribution explanations displayed features regarding the plane class. This is expected, but on the other hand, there is no distinctive explanation why that specific pixel or patch outside the true plane segment has been classified as plane.

For this reason, we want to limit the sampling of incorrect pixels to "true" misclassifications, like in the case of image classification tasks. We do this by sampling only incorrectly predicted pixels from classes in the prediction, that don't appear in the true label, This is for example the case, when a cow segment is (partly) misclassified as a horse while the horse class is not present in the ground truth segmentation.

In conclusion, for both correct and incorrect predictions, we sample pixels that are far enough away from the respective other type. Additionally, for incorrect predictions, we only sample actual misclassifications by only considering pixels that have been assigned to classes not appearing in the true label.

4.2 Prior Knowledge in Segmentation Tasks

As already mentioned in Section 3.4, the Agreement Task must be carefully designed to prevent users taking advantage of their prior knowledge. E.g. the authors of HIVE [28] removed semantic class labels, because otherwise users might identify the correct prediction based on the name of the class, instead of using the explanation. Additionally, we identified, that for prototype explanations, showing the prototype's original image might also lead to usage of prior knowledge (see Section 6.1.2 for results). Showing the prototype's original image allows users to compare it against the image to classify, instead of comparing the highlighted region with the prototype.

For the task of image segmentation, this measure might be even more relevant. To understand, we have to take a look at common tasks used in image segmentation. Two of the most commonly used segmentation datasets are the Cityscapes dataset [13] and the Pascal VOC dataset [17], which both consist of only a handful of classes and are very coarse-grained, i.e. classes are semantically very distinct from each other. For example, the Pascal VOC dataset contains images, that in a broad sense contain the same scenes as the ImageNet dataset [42], e.g. everyday life, nature and urban environments. While the traditional ImageNet dataset contains 1000 different classes, the Pascal VOC dataset only contains 20 different classes.

With this coarse granularity of the mentioned segmentation tasks in mind, showing the prototype's original image would make it trivial for users to guess the prediction correctly. However, this draws the user's attention away from using the core of the explanation, which is the comparison of the image patch and its prototype. To ensure that users assess the model's prediction using the actual information provided by the prototype, we decide to omit the prototype's original image in our Agreement Task for segmentation.

Additionally, we decide to show a maximum of 3 different prototypes for each prediction. We keep in mind Kim et al.'s argument [28] that showing too many prototypes could become confusing for users, but think that a maximum of 3 prototypes is still manageable. On the other hand, showing multiple prototypes adds more value to the explanation, while not increasing the risk of humans utilizing prior knowledge. To allow users to weigh the significance of the different prototypes shown, we include the similarity values for each prototype.

In preliminary tests we noticed, that the labeling in segmentation tasks, especially for the Pascal VOC dataset, some classes are hard to distinguish without the knowledge of the semantic classes. For example, the classes "chair" and "sofa" could potentially be thought to be the same class, when not made aware that they are distinct classes. As we can't show semantic labels for neither true label nor prediction, we decide to visualize all semantic class labels of the dataset during the evaluation task on the side.

Finally, we adopt all measures, that were already applied in previous Agreement Tasks on classification (see Section 3.4 for details).

An example of our Agreement Task for image segmentation is shown in Figure 4.2. For visualization purposes, we don't show the mentioned list of the dataset's class labels.

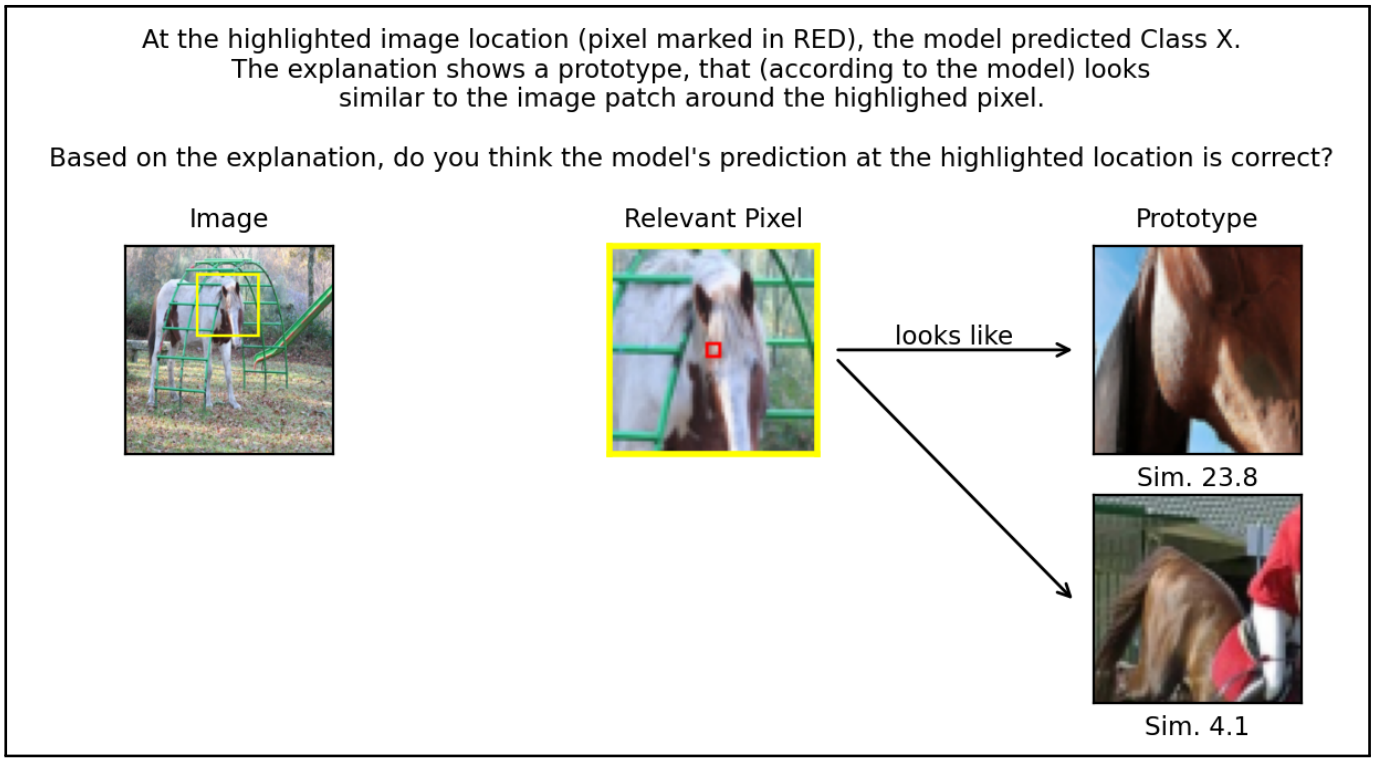


Figure 4.2: Our Agreement Task for segmentation using an explanation from PIP-Net [38] (adapted to segmentation) on the Pascal VOC dataset [17].

5 Explanation Method for Image Segmentation

Given our use-case-centered design, we want to create an explanation method for image segmentation, that performs well in our previously defined use case, which resides in the area of AI-assisted decision-making. As our evaluation metric is designed to specifically capture that use case, we want our explanation method to achieve good scores in that task. The theoretical foundations of our evaluation metric are covered in detail in Chapter 3. Its extension to image segmentation is described in Chapter 4.

In this chapter, we first cover different options of explanation types, before deciding for one explanation method. Subsequently we describe the details of our explanation method for image segmentation.

5.1 Types of Explanations for Image Segmentation

5.1.1 Attribution Methods

As already discussed in Chapter 3, current attribution methods are not well suited for solving the Agreement Task. We suggested, that current methods produce attribution maps, that don't highlight enough when a prediction is incorrect.

Moving towards our main objective of image segmentation, the limitations of attribution maps get more severe. In this case, the segmentation maps of the prediction already provide dense localization information, i.e. pixel wise segmentation masks. An explanation in the form of an attribution map, which only provides localization information, wouldn't add a lot of value. As the authors of [15] point out, the attribution maps often times only resemble the segmentation mask itself. Only in the case of e.g. context-biases, the attribution map can be expected to differ considerably from the segmentation mask.

It becomes clear, that for segmentation tasks, explanations that provide more means of information than attribution maps, would be of advantage.

5.1.2 Prototype Networks

Interpretable-by-design models generally allow for a more flexible design of explanations, and can be, as opposed to post-hoc explanations, more faithful by design [30, 39]. Specifically prototype networks not only explain *where* something important is located in the image, but also show a prototype to visualize *what* the network recognized.

Unfortunately, the prototype networks ProtoPNet [9] and ProtoTree [37], which were tested in HIVE's Agreement Task [28], did not perform significantly better than attribution methods. However, in our tests a recent prototype network by Nauta et al., called PIP-Net [38], yielded a significant improvement in the

Agreement Task, compared to previously tested prototype networks. The full experiment details can be found in Chapter 6.

5.2 PIP-Net for Segmentation

Because of the promising results of Nauta et al.'s PIP-Net [38] in our Agreement Task on image classification, we will move forward with it and extend PIP-Net to the task of image segmentation. To not lose interpretability capabilities, we follow the original PIP-Net's design choices as much as possible and only apply changes when necessary.

In the following, we will describe our architecture of our PIP-Net for segmentation in detail. If not mentioned otherwise, we applied the same principles from the original PIP-Net architecture [38].

5.2.1 Architecture

Segmentation Architecture

We decide to use a Fully Convolutional Network (FCN) as base architecture for segmentation. This allows us to reuse the CNN backbone of the classification model, while introducing only minimal new architectures features, that could potentially hurt interpretability.

In comparison, ProtoSeg by Sacha et al. [45] extended the prototype network ProtoPNet [9] to the segmentation architectures U-Net [41] and DeepLab [10] models. While these baseline architectures have a better prediction accuracy than FCNs, we worry that these model structures introduce new architectural features that could potentially be harder to interpret. For example, the upsampling part of U-Net is relatively unexplored in terms to prototype features and could potentially harm overall interpretability.

In this context, we want to mention again the findings of Seg-Grad-CAM [53] from Section 2.3.1, where the authors discovered that evaluating features in the "bottleneck" were more informative than features at the end. Similarly, when using FCNs, we are generating prototype features at the "bottleneck" of the segmentation, in contrast to ProtoSeg, that generates prototype features at the end of the network.

CNN Backbone & Alignment Loss

Figure 5.1 illustrates the architecture of our PIP-Net extension to segmentation. The network takes an image pair (x', x'') as input, where both images are differently augmented versions of the same original RGB image $x \in \mathbb{R}^{H_i \times W_i \times 3}$.

An image batch is processed in a CNN backbone f, which yields the latent representation $z = f(x) \in \mathbb{R}^{H \times W \times D}$. Where H, W represent the spatial dimensions of the latent feature space respectively, and D as the number of features. A Softmax activation is applied on all feature vectors to get a one-hot like encoding, so that each feature vector ideally corresponds to a single prototype.

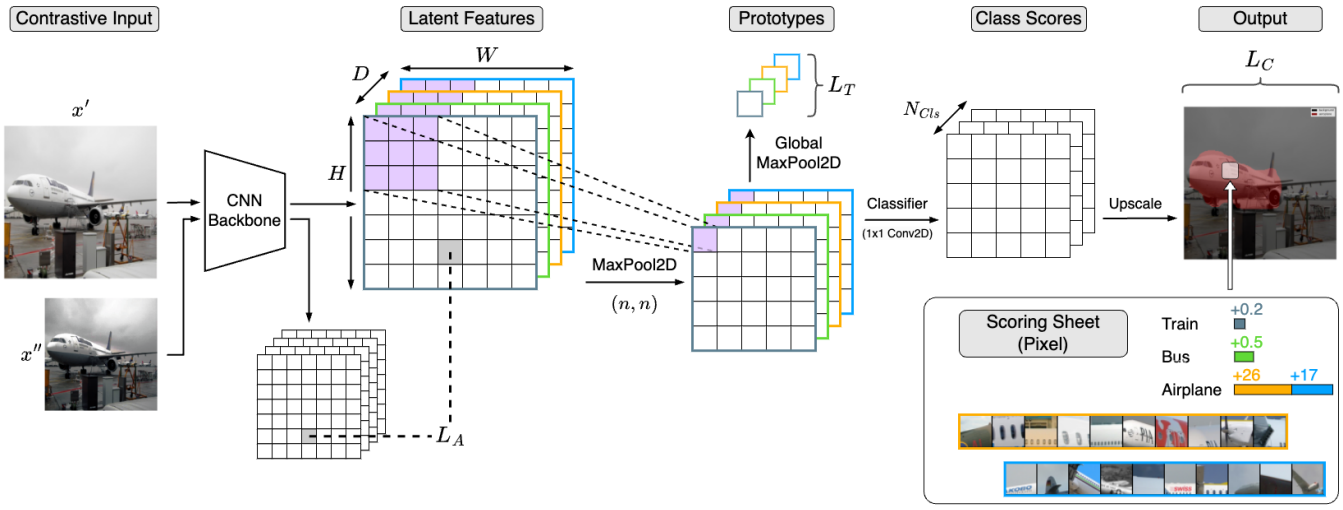


Figure 5.1: Architectural overview of our extension of PIP-Net [38] to the task of segmentation.

The alignment loss L_A enforces each feature vector pair at the same spatial location $(z'_{i,j}, z''_{i,j})$ to have the same one-hot encoding:

$$L_A = -\frac{1}{HW} \sum_{(i,j) \in H \times W} \log \left(z'_{i,j} \cdot z''_{i,j} \right) \tag{5.1}$$

Local Pooling Layer & Scoring Sheet Reasoning

In a standard FCN architecture, the next step would be the classification of each feature vector $z_{i,j}$, followed by an upscaling to the original image's size to obtain a classification score for each pixel. However, an important functionality of PIP-Net is its scoring sheet reasoning, where the classification output can be split into the prototype's individual contributions.

In the original PIP-Net architecture for image classification, a global max pooling operation would reduce the one-hot encoded latent feature vectors into a single feature vector, that aggregates all prototypes present in the latent space. In the general case of segmentation, the max pooling layer from image classification is removed, so that each feature vector is classified individually and a dense output segmentation is obtained. Since in our case, for each feature vector per design only a single prototype is active, the scoring sheet for a classification would only consist of a single prototype. Not only could this hurt human interpretability, but it could also limit classification performance, as only very limited information is available for classification.

To allow for multiple active prototypes for each classification, we introduce a local pooling layer, consisting of a small, e.g. 3x3, kernel, with single pixel stride and no padding. This results in a local aggregation of prototypes, which only reduces the size of the latent space by a small amount, thus not hurting the spatial granularity of the final segmentation output.

We test different pooling options in our experiments in Chapter 6.

Diverse Image Representations & tanh-Loss

To get diverse prototype representations and avoid the case of all features aligning onto the same prototype, we adopt the tanh-loss L_T from the original PIP-Net architecture, which regulates that each prototype is at least present once per training batch:

$$L_T = -\frac{1}{D}\sum_{d}^{D}\log\left(\tanh\left(\sum_{b}^{B} z_{\text{global\_max\_pool}}\right)+\epsilon\right) \tag{5.2}$$

In the original model, L_T was applied on the globally max pooled feature vector, meaning that in a batch each prototype should at least be present *once in the entire image*.

If we were to apply the same principle on each feature vector in latent space, we would impose unrealistic requirements, i.e. in a batch, each prototype should be present at least *once in each latent position*. As it's practically impossible that all semantic features would be present in each part of the image individually, we keep the original requirement of each prototype being present once in the entirety of the image. For this reason, we reintroduce the global max pooling layer, on which the tanh-loss will be applied.

Sparse Classification & Upscaling

For classification, we classify each feature vector z of the locally pooled features. We reformulate the fully connected classification layer as a 1x1 convolutional layer, which works in the same way but is more convenient in the case of image segmentation. We keep PIP-Net's [38] original design choice of restricting the classification weights w_c to only take non-negative values:

$$w_c \in \mathbb{R}_{\geq 0}^{D\times N_{\text{Cls}}} \tag{5.3}$$

After bilinear upscaling of the output class scores, we compute a standard cross entropy classification loss L_C. To achieve small explanation sizes, so that only a few prototypes contribute to an explanation, the classification weights are optimized for sparsity using PIP-Net's output score transformation:

$$o = \log\left((zw_c)^2+1\right) \tag{5.4}$$

Further details about the PIP-Net architecture can be found in the original publication [38].

5.2.2 Segmentation Specific Extensions

In segmentation, the annotation data in form of the true segmentation maps holds a lot of information. In addition to the standard classification loss per pixel, this dense information could be utilized to improve the process of learning prototypes. Specifically, we try to improve the self-supervised contrastive representation learning using the segmentation data.

In its basic version, the contrastive representation learning is based on 2 differently augmented views of the same image, that should be assigned the same latent representation. Because of these augmentations, the network learns to generalize over similar looking inputs of the same images, i.e. assign the same prototype to similar looking image patches. As the augmentations are done in the input image's space and are selected to only use augmentations that appear natural to the human eye, the learned notion of similarity broadly

resembles the human perceived similarity. However, when visualizing image patches belonging to the same prototype, we can see that this method is not perfect, as patches belonging to the same prototype can look very dissimilar.

In our approach, we try to improve this representation learning using the information of the true segmentation labels.

The basic idea is to look at image patches that are assigned to the same prototype but have a very different class label. These image patches should be forced to separate (as in "not align"), i.e. not have the same prototype assigned to each other.

This approach would theoretically prevent class-shared prototypes, as similar image patches from different classes would be forced to separate. However, this is not a big loss and according to previous experimental results (see Section 6.1.2) could even be advantageous.

From a theoretical point of view, the alignment and tanh loss in PIP-Net are just an adapted version of a classic contrastive loss. In a contrastive loss setup, there is usually a positive pair with similar features and a negative sample which is dissimilar from the pair. In a broad sense, the contrastive loss optimizes the image pair to be close in latent space while being distant to the negative sample. As it is usually hard to find meaningful negative samples, PIP-Net discards the negative sample, and uses the tanh-loss instead. The alignment-loss still ensures, that similar looking features are grouped together, while the tanh-loss ensures diverse representations.

The additional "separation" loss based on the segmentation labels could ideally replace or at least complement the tanh-loss in achieving a better diverse image representation.

While first tests didn't succeed, we still think this is an interesting approach to improving the feature representation learning, which could be explored in the future.

6 Experiments

6.1 Agreement Task on Image Classification

The Agreement Task is conducted according to the design principles explained in Chapter 3. For the task of image classification, we closely follow the Agreement Task presented in HIVE [28]. If not specified otherwise, we follow the same test principles as the original task.

6.1.1 Preliminary Notes

In the context of this master's thesis, we weren't able to employ human-based evaluations in the scale of previous methods, but had to rely on fewer participants. For example, HIVE [28] recruited 50 participants, each of which were shown 10 samples for each evaluation. In comparison, we showed 50 samples per user, for a total of 3 to 5 users. Because of our low number of participants, results have to be regarded with some caution, however some clear tendencies can be observed.

Our participants were selected from different range of machine learning backgrounds, ranging from no experience at all, over a high-level understanding of modern AI, to people actively engaged in the field of deep learning and XAI.

For each participant, we showed a total of 50 samples, each having an equal chance of being a correct or incorrect prediction. This information is communicated to the participants beforehand. As opposed to [28], we don't predefine the number of correct/incorrect predictions, e.g. 50 % of *all* predictions being correct, so that users can't be biased towards the end of the evaluation. Additionally, we use different randomly generated samples for each participant, to eliminate the possibility that e.g. 50 random samples show really favorable or easy explanation examples.

We also don't give users feedback on their assessment, neither after a single example nor after completing the evaluation for a certain explanation method. This way, participants won't be able to learn over the course of multiple rounds of evaluations.

Before an evaluation, participants are given a short introduction about how to evaluate explanations for different explanation types. We also inform users about explanation specific details, e.g. that PIP-Net allows for class-shared prototypes, so that classes shown in the original image and prototype image must not necessarily match.

As our evaluation tasks were conducted under our own supervision, we were able to get additional qualitative information from the participants, e.g. how difficult a task feels like or *why* the user thought a prediction to be correct/incorrect.

6.1.2 PIP-Net for Image Classification on CUB Dataset

Table 6.1: Results of HIVE's Agreement Task [28] on PIP-Net [38]

Dataset	Explanation Method		
CUB [54]	PIP-Net (Vanilla)	PIP-Net (No Proto Image)	PIP-Net (Only Class Image)
Correct	55.5 ± 4.6 %	64.0 ± 6.0 %	**86.7 ± 8.6 %**
Incorrect	**75.4 ± 3.4 %**	74.5 ± 2.6 %	54.3 ± 8.7 %
Total samples	250	200	200

We evaluated the original PIP-Net prototype network [38] for the task of image classification on the CUB bird dataset [54] on 4-5 users, showing each 50 samples for a total of 200-250 samples.

The results are shown in the 1st data column of Table 6.1.

Confirmation Bias does not necessarily exist

When we compare the results of PIP-Net on the Agreement Task to the explanations methods tested in HIVE [28], we can see a much higher accuracy on incorrect predictions. Here, on average PIP-Net performed roughly 35 %-points better (75 % vs. 40 %) than the explanation methods tested in HIVE. Even though sample size and number of participants are lower than in HIVE [28], the results were very similar for all participants, as can be seen by the standard deviation of only 2.3 %. This result suggests that it is indeed possible for explanation methods to make it clear when there is an incorrect prediction. Thus, we see a strong indication that confirmation bias does not necessarily exist in use cases like the ones captured in the Agreement Task. We believe, that this big improvement is due to the novel notion of similarity of PIP-Net, which has not been used before in prototype networks.

Class-shared Prototypes might hurt Interpretability

When looking at the accuracy on correct predicted samples, we however see a slight drop compared to other explanation methods. One plausible reason for this could be the class-shared prototypes of PIP-Net. Because in PIP-Net, some prototypes are shared between classes (e.g. wheel prototypes are viable for both cars and buses), the class shown in the original image might be different from the class shown in the prototype's original image. This could confuse participants, as the shown prototype might look similar to the image region, while the full images look different from each other. As something would seem wrong in the prediction, users might believe that the model's prediction is incorrect. Even though we informed users about this characteristic, qualitatively we could still observe users being confused on examples which contained prototype images of a different class.

In conclusion, our results strongly indicate that PIP-Net performs notably different from previously tested prototype networks and could be a more viable option to use in AI-assisted decision-making tasks, as the overall metric result is significantly improved over previous methods.

6.1.3 Ablations: Original Image of Prototypes

As described in Section 3.4.2, showing the prototype's original image could be a potential threat of users using prior knowledge instead of the explanation itself to solve the task.

We test this hypothesis by evaluating 2 different ablations, which were again performed using PIP-Net [38] on the CUB dataset [54]:

1. Omitting the prototype's original image in the explanation
2. Only showing an image that represents the predicted class

The results are shown in the 2nd and 3rd data column of table 6.1, referred to as "NoProtoImage" and "OnlyClassImage", respectively.

Ablation "NoProtoImage"

For the ablation "NoProtoImage", users concordantly reported that the task felt more difficult than the baseline task. This is expected, as the prototype region often doesn't contain the complete information about the class in the image. This also indicates that previously users utilized the prototype's original image for their assessment, which is now missing, making the task feel more difficult. Surprisingly, even though the perceived difficulty of the task increased, the accuracies stayed mostly in line with the baseline results. We can even see a slight improvement on correct predictions, which could potentially be connected to the previously mentioned case of class-shared prototypes. As now the comparison can only be made using the prototype, a prototype from a different class might not be as obvious compared to when the whole image is shown. Continuing the example of car and bus wheels, when only a wheel is shown, it might not be as distinctive that the wheel might be from the other class.

Ablation "OnlyClassImage"

In the 2nd ablation "OnlyClassImage", we can see a big difference compared to the baseline. The accuracy on correctly predicted samples increases, while for incorrect predictions it decreases. Also, variance increases, which make these results less statistically significant. Still, we can interpret some tendencies. When users are only shown the class image, especially for the case of bird classification, many classes that might be distinct, still look similar enough, so that a prediction would be believed as correct. In contrast, when prototypes highlight specific features, a difference in a detail might be more obvious, which could lead to a better accuracy on incorrect predictions.

We note, that omitting the prototype's original image would not make sense in reality, as it an easy resource to add useful information to a prototype explanation. If our interpretations regarding class-shared prototypes are correct, we would suggest, that in terms of utility in AI-assisted decision-making tasks, it's better to use prototypes that only relate to a single class.

6.2 PIP-Net for Segmentation

In this section, we take a look at the performance of our extension of PIP-Net [38] to the task of image segmentation.

6.2.1 Implementation Details

As done for the original PIP-Net network, we train our PIP-Net segmentation network with ResNet-50 [23] and ConvNeXt-tiny [33] classification backbones. The maximum number of prototypes are 2048 and 768 respectively for both backbones. Both networks are pretrained for 30 epochs, to get meaningful latent representations, and then trained on all losses for 100 more epochs. Compared to PIP-Net on classification, more pretraining and training epochs were needed for learning convergence.

We evaluated our model on the PASCAL VOC 2012 dataset [17] with images resized to a size of 224×224 pixels, as done in the original PIP-Net training.

The network is trained with an Adam optimizer, using the learning rates of 0.05 for the classification layer, 0.0001 for the ResNet50 backbone and 0.0005 for the ConvNeXt-tiny backbone.

Details about the architecture are described in Chapter 4, whereas other implementation details can be found in the original paper [38] in Section 6.1.

6.2.2 Results

We report the metrics mean Intersection over Union (mIoU) and pixel accuracy and compare our model's against their non PIP-Net FCN baselines in Table 6.2.

Table 6.2: Accuracy results for PIP-Net extensions to segmentation, compared to the FCN baseline. Results are averaged over 3 runs with different seeds.

Model	Mean Pixel Acc.	mIoU
FCN-8s ResNet50 vanilla	**90.1**	**63.2**
FCN-8s ResNet50 PIP-Net	85.9	42.6
FCN-8s ConvNeXt-tiny PIP-Net (best)	89.5	56.0

As can be seen, our PIP-Net implementation for FCN-8s ResNet50 performs worse than its baseline. However, by exchanging the Resnet50 backbone by a ConvNeXt-tiny backbone, we can achieve accuracies that are more comparable to the baseline results.

We want to point out, that hyperparameter tuning was not a priority in our work, as we fully focused on optimizing interpretability. We are confident that accuracies can be increased by changing hyperparameters, but leave this to future work.

6.2.3 Local Pooling Layer

Next, we want to observe the effects of the local pooling layer, which we introduced in Section 5.2.1. We show the result for different combinations in Table 6.3.

Table 6.3: Results for FCN-8s ConvNeXt-tiny using different configurations of local pooling layer. Results are averaged over 3 runs with different seeds.

Kernel size	Stride	Latent Space Size	Mean Pixel Acc.	mIoU
-	-	26	87.7	45.6
2x2	2x2	13	89.1	53.6
2x2	1x1	25	89.2	53.0
3x3	1x1	24	**89.5**	**56.0**

We can see, that using pooling we can increase PIP-Net's accuracy. This is probably due to the availability of multiple prototypes in a feature vector, which allows for better classification, compared to only one-hot vectors. The best accuracies are achieved with a 3x3 pooling with single pixel stride. Interestingly, for 2x2 pooling a significantly smaller latent space size, as an effect of the bigger stride, doesn't hurt accuracy.

6.2.4 Other Observations

Clutter Prototypes

During our tests, it could be observed that consistently over 90 % of all feature vectors in the training or validation set were assigned to the same prototype. These feature vectors usually belong to image patches representing parts of the "background" class and have almost no similarity. We believe, that all feature vectors, that do not fit to any other prototype will land in these "clutter prototypes". This phenomenon can also be observed for the original PIP-Net on image classification tasks, even though this is not as relevant, as only a single prediction per image is done.

No abstained Patches in Segmentation

The original PIP-Net architecture for classification allowed for prototypes to be abstained, if they do not share a connection with any class. In the case of segmentation, it could be observed that there were no abstained prototypes. We believe that this is due to the fact, that every feature vector in latent space is classified. This means that even seemingly irrelevant prototypes like the earlier mentioned clutter prototypes will be relevant for a classification, simply because (almost) all feature vectors present will be used for an classification at each respective location in the output segmentation map.

6.3 Agreement Task on Image Segmentation

In Chapter 4, we extended HIVE's Agreement Task [28] to image segmentation. On image segmentation, we evaluate our extension of the prototype network PIP-Net [38]. In a previous experiment (see Section 6.1), we already achieved promising results for PIP-Net on image classification. Here, we test if PIP-Net still performs

well on the Agreement Task on the task of image segmentation. We compare our method to the related works ProtoSeg [45] by Sacha et al. and L-CRP [15] by Dreyer et al.

6.3.1 Preliminary Notes

We performed the Agreement Task on image segmentation with the same procedure as for image classification. We advise reading Section 6.1 for details.

One notable difference is, that since we now sample predictions for individual pixels from images, it could happen that a picture is shown multiple times. We prevent this possibility, even between correct and incorrect samples, to not create confusion for the participants.

To allow for fair testing between all methods, we adapt the explanations of each method accordingly. For the tests on the Pascal VOC dataset [17] preliminary tests showed, that the standard prototype patch size was too small to show any meaningful information. Because of this, we increased the prototype patch size by a factor of 2.

For ProtoSeg [45], the prototypes are variable in size. To avoid huge prototypes, that give away too much context, so that users could make use of prior knowledge, we crop prototype bigger than the predefined patch size accordingly.

Even though L-CRP works using concepts instead of prototypes, we found that explanations are similar. We adapt the explanation to fit the Agreement Task. Also, in the context of the evaluation, we will use the terms "concept" and "prototype" interchangeably.

6.3.2 Results

Table 6.4: Results of multiple explanation methods for image segmentation on our extension of HIVE's Agreement Task [28].

Dataset	Explanation Method		
Pascal VOC 2012 [17]	PIP-Net Seg.	ProtoSeg [45]	L-CRP [15]
Correct	**81.4 ± 8.8 %**	68.3 ± 5.6 %	41.5 ± 7.8 %
Incorrect	**84.4 ± 12.1 %**	75.0 ± 12.4 %	78.0 ± 0.8 %
Total samples	150	100	100

When looking at the results in Table 6.4, we can see that our extension of PIP-Net to segmentation performs best for both correct and incorrect predictions. ProtoSeg performs slightly worse than PIP-Net, which is in line with PIP-Net (for classification) performing better than ProtoPNet. Although it must be noted, that the differences in segmentation are far smaller than in the case of classification. L-CRP performs notable worse than the other 2 methods, mainly because it could be observed that its visualized concepts share very little similarity with the relevant image patches. The high accuracy of L-CRP on incorrect predictions is due to many predictions being believed as incorrect, which we suggest is due to the lack of visual similarity. This emphasized, that it's important to consider the accuracies for both correct and incorrect predictions.

6.3.3 Influence of the Dataset

During our evaluations on the Pascal VOC dataset [17], we observed some challenges compared to the CUB dataset [54], which was used for the Agreement Task on classification.

Even with showing all class labels of the dataset, as described in Section 4.2, there were some misclassifications that were not caused by a bad explanation, but rather due to ambiguous labelling of the dataset. For example, a pixel showing part of a table, that has been labeled as "background" due to not being one of the main objects in the scene, was classified as "dining table". In this case, we can not blame the model, as an intuitive explanation for a dining table was given, which was also in line with the human perception. This shows, that coarse-grained datasets like Pascal VOC might not be well suited for evaluation tasks like the Agreement Task.

Additionally, we could observe that sometimes users didn't solve the task by comparing the highlighted image region around the pixel with the prototype, but instead recognized the true class based on the original image. With this knowledge, users tried to align the content of the prototype with the class they recognized in the original image. A potential fix here could be to also omit the original image, but we fear, that this will make the evaluation task too abstract for users and too disconnected from real life use cases.

7 Discussion and Future Work

In this work, our goal was to obtain a more interpretable image segmentation and gain insights towards making the task of image segmentation more explainable.

With our main approach of gaining important insights in the context of real-life use cases, we first revisited the evaluation of explanation method for image classification. We tested a new prototype explanation method called PIP-Net [38] on the human-based evaluation metric "Agreement Task" of the HIVE framework [28]. In contrast to previously tested methods, PIP-Net achieved impressive results, especially because it was able to highlight incorrect predictions to the users, which all previously tested methods failed to do.

We think, this is due to PIP-Net's unique way of how its prototype representation is learned. We believe, that PIP-Net's notion of similarity aligns closely to how we as humans perceive similar images. This property is especially useful in AI-assisted decision-making tasks that are captured by the Agreement Task. A prototype explanation usually shows the image that is most representative for the prototype that the relevant patch has been assigned to. If we assume an intuitive notion of similarity, we can assume that when the relevant image patch and its corresponding prototype (referring to the image) do not look alike, something went wrong in the network's decision-making process. The results of the Agreement Task show, that this thought process, which is a natural way of how prototype explanations could be interpreted, works much better on PIP-Net explanations compared to other prototype explanation. As a good notion of similarity, where shown prototypes align with human-perception, is an important factor, PIP-Net realizes a noteworthy step towards the reliable usage of explanation methods in real-life use cases.

During our tests on the Agreement Task, we employed some ablations to its original version. We showed, that visualizing solely a training image, that represents the predicted class, already delivers good results on the Agreement Task. On previous prototype explanations, the prototype was shown along with its original image, which represents the mentioned class image. As this class image can be visualized for any kind of network, we evaluated how much of the Agreement Task scores are due to the actual prototype explanation. For PIP-Net we could see that the explanations without the prototype's original performed similarly. For the future, it would be important to see, how other prototype methods, e.g. those previously tested in HIVE [28], would perform on the modified Agreement Task without the prototype's original image. Also, we hypothesized, that PIP-Net's ability to learn class-shared prototypes might hurt human interpretability, as the prototype's original image might contain a different class than the prediction. Here, it would be interesting to implement an option for PIP-Net to restrict class-shared prototypes, and compare both versions on human-studies.

Moving on to image segmentation, we saw that we can easily adapt PIP-Net to work on segmentation using a Fully Convolutional Network (FCN) architecture. Using a local pooling layer, we can restore PIP-Net's characteristic scoring sheet reasoning, which otherwise would've been lost. The pooling layer also increases performance in terms of pixel accuracy and mean IoU. We think, this is because the local pooling layer allows for more diverse latent representations, which are easier to classify than simple one-hot encoded vectors. Still, during development, we noticed, that optimizing these sparse features is notable slower and possibly more difficult than with using standard feature vectors of baseline architectures. For the future, we encourage

trying out alternatives for sparsification of feature vectors, which could be more temperate compared to strict one-hot encoding.

To evaluate our PIP-Net for segmentation, we compare it against ProtoSeg [45] and L-CRP [15] on our extension of HIVE's agreement task. While segmentation accuracy of our FCN based prototype network is lower compared to the other tested methods, PIP-Net for segmentation outperforms the other two methods in terms of interpretability. ProtoSeg performs slightly worse, whereas L-CRP struggles to produce similar prototypical images. We saw that using the Pascal VOC dataset [17] predictions were generally easier to explain than e.g. the CUB dataset [54] used in classification. This is probably due to the coarse granularity of the dataset, whose effect on usage of human prior knowledge probably can't be completely mitigated. The employment of our Agreement Task for segmentation on more difficult segmentation datasets (in terms of classification of segments) would be interesting for the future.

This highlights an interesting limitation of our Agreement Task. It predominantly tests the interpretability of segmentation networks in terms of classifying individual segments. But another important and in our opinion more difficult subtask of image segmentation is the localization, i.e. the precise placement of the segment boundaries. An explanation method that intuitively explains the localization aspect of image segmentation, e.g. why a segment boundary has been placed exactly there and not a few pixels to the side, does to the best of our knowledge not exist. As practically all deep learning approaches to image segmentation are based on the principle of pixel wise classification, we think that an intuitive localization explanation is a difficult challenge. A first approach for the future could be design of an inherently interpretable version of segmentation specific modules, e.g. the expansive upscaling path of U-Net [41]. By this, we could for example find out, which features from which latent location were responsible for the classification at a certain pixel and how these latent feature influences traversed through the upscaling module.

In conclusion, we extended both the prototype-based explanation method "PIP-Net" and the human-based "Agreement Task" evaluation metric to the task of image segmentation. Our extension of PIP-Net conserves the interpretability characteristics of its original version and outperformed all other tested methods in the Agreement Task for segmentation. We discovered some important aspects, which could be the basis for a fundamentally new way to interpretable image segmentation. By all of this, we hope to make an important contribution towards more explainable image segmentation.

Bibliography

[1] Reduan Achtibat et al. “From attribution maps to human-understandable explanations through Concept Relevance Propagation”. In: *Nature Machine Intelligence* (2023).

[2] Julius Adebayo et al. “Sanity Checks for Saliency Maps”. In: *Advances in Neural Information Processing Systems (NeurIPS)* (2018).

[3] Moritz Böhle, Mario Fritz, and Bernt Schiele. “B-cos Networks: Alignment is All We Need for Interpretability”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2022).

[4] Moritz Böhle, Mario Fritz, and Bernt Schiele. “Convolutional Dynamic Alignment Networks for Interpretable Classifications”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2021).

[5] Wieland Brendel and Matthias Bethge. “Approximating CNNs with Bag-of-local-Features models works surprisingly well on ImageNet”. In: *International Conference on Learning Representations (ICLR)* (2019). Poster Presentation.

[6] Zachariah Carmichael et al. “Pixel-Grounded Prototypical Part Networks”. In: *Computing Research Repository arXiv* (2023).

[7] Siddhartha Chandra and Iasonas Kokkinos. “Fast, Exact and Multi-Scale Inference for Semantic Image Segmentation with Deep Gaussian CRFs”. In: *European Conference on Computer Vision (ECCV)* (2016).

[8] Chun-Hao Chang et al. “Explaining Image Classifiers by Adaptive Dropout and Generative In-filling”. In: *Computing Research Repository arXiv* (2018).

[9] Chaofan Chen et al. “This looks like that: deep learning for interpretable image recognition”. In: *Advances in Neural Information Processing Systems (NeurIPS)* (2019).

[10] Liang-Chieh Chen et al. “DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2018).

[11] Liang-Chieh Chen et al. “Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation”. In: *European Conference on Computer Vision (ECCV)* (2018).

[12] Liang-Chieh Chen et al. “Rethinking Atrous Convolution for Semantic Image Segmentation”. In: *Computing Research Repository arXiv* (2017).

[13] Marius Cordts et al. “The Cityscapes Dataset for Semantic Urban Scene Understanding”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016).

[14] Rachel Lea Draelos and Lawrence Carin. “Use HiResCAM instead of Grad-CAM for faithful explanations of convolutional neural networks”. In: *Computing Research Repository arXiv* (2021).

[15] Maximilian Dreyer et al. “Revealing Hidden Context Bias in Segmentation and Object Detection through Concept-specific Explanations”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops* (2022).

[16] Mengnan Du, Ninghao Liu, and Xia Hu. “Techniques for Interpretable Machine Learning”. In: *Communications of the ACM* (2019).

[17] M. Everingham et al. “The Pascal Visual Object Classes (VOC) Challenge”. In: *International Journal of Computer Vision (IJCV)* (2010).

[18] Thomas Fel et al. “What I Cannot Predict, I Do Not Understand: A Human-Centered Evaluation Framework for Explainability Methods”. In: *Advances in Neural Information Processing Systems (NeurIPS)* (2022).

[19] Ruth C. Fong and Andrea Vedaldi. “Interpretable Explanations of Black Boxes by Meaningful Perturbation”. In: *IEEE International Conference on Computer Vision (ICCV)* (2017).

[20] Bharath Hariharan et al. “Simultaneous Detection and Segmentation”. In: *European Conference on Computer Vision (ECCV)* (2014).

[21] Syed Nouman Hasany, Caroline Petitjean, and Fabrice Mériaudeau. “Seg-XRes-CAM: Explaining Spatially Local Regions in Image Segmentation”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops* (2023).

[22] Peter Hase, Harry Xie, and Mohit Bansal. “The Out-of-Distribution Problem in Explainability and Search Methods for Feature Importance Explanations”. In: *Advances in Neural Information Processing Systems (NeurIPS)* (2021).

[23] Kaiming He et al. “Deep Residual Learning for Image Recognition”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016).

[24] Robin Hesse, Simone Schaub-Meyer, and Stefan Roth. “FunnyBirds: A Synthetic Vision Dataset for a Part-Based Analysis of Explainable AI Methods”. In: *IEEE/CVF International Conference on Computer Vision (ICCV)* (2023).

[25] Adrian Hoffmann et al. “This Looks Like That... Does it? Shortcomings of Latent Space Prototype Interpretability in Deep Networks”. In: *Computing Research Repository arXiv* (2021).

[26] Lukas Hoyer et al. “Grid Saliency for Context Explanations of Semantic Segmentation”. In: *Advances in Neural Information Processing Systems (NeurIPS)* (2019). Poster Presentation.

[27] Christina Humer et al. “Interactive Attribution-based Explanations for Image Segmentation”. In: *EG Conference on Visualization (EuroVis)* (2022). Ed. by Michael Krone, Simone Lenti, and Johanna Schmidt. Poster Presentation.

[28] Sunnie S. Y. Kim et al. “HIVE: Evaluating the Human Interpretability of Visual Explanations”. In: *IEEE Conference on Computer Vision and Pattern Recognition Conference (CVPR)* (2022).

[29] Teddy Koker et al. “U-Noise: Learnable Noise Masks for Interpretable Image Segmentation”. In: *IEEE International Conference on Image Processing (ICIP)* (2021).

[30] Matthew L. Leavitt and Ari S. Morcos. “Towards falsifiable interpretability research”. In: *Computing Research Repository arXiv* (2020).

[31] Paul Pu Liang, Amir Zadeh, and Louis-Philippe Morency. “Foundations and Trends in Multimodal Machine Learning: Principles, Challenges, and Open Questions”. In: *Computing Research Repository arXiv* (2023).

[32] Guosheng Lin et al. “Efficient piecewise training of deep structured models for semantic segmentation”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016).

[33] Zhuang Liu et al. “A ConvNet for the 2020s”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2022).

[34] Ziwei Liu et al. “Semantic Image Segmentation via Deep Parsing Network”. In: *IEEE International Conference on Computer Vision (ICCV)* (2015).

[35] Jonathan Long, Evan Shelhamer, and Trevor Darrell. “Fully Convolutional Networks for Semantic Segmentation”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2015).

[36] Grégoire Montavon et al. “Layer-Wise Relevance Propagation: An Overview”. In: *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*. Ed. by Wojciech Samek et al. Springer International Publishing, 2019.

[37] Meike Nauta, Ron van Bree, and Christin Seifert. “Neural Prototype Trees for Interpretable Fine-grained Image Recognition”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2021).

[38] Meike Nauta et al. “PIP-Net: Patch-Based Intuitive Prototypes for Interpretable Image Classification”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2023).

[39] Meike Nauta et al. “This Looks Like That, Because ... Explaining Prototypes for Interpretable Image Recognition”. In: *Machine Learning and Principles and Practice of Knowledge Discovery in Databases (PKDD/ECML) Workshops* (2021).

[40] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. “Learning Deconvolution Network for Semantic Segmentation”. In: *IEEE International Conference on Computer Vision (ICCV)* (2015).

[41] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. “U-Net: Convolutional Networks for Biomedical Image Segmentation”. In: *Medical Image Computing and Computer-Assisted Intervention (MICCAI)* (2015).

[42] Olga Russakovsky et al. “ImageNet Large Scale Visual Recognition Challenge”. In: *International Journal of Computer Vision (IJCV)* (2015).

[43] Dawid Rymarczyk et al. “Interpretable Image Classification with Differentiable Prototypes Assignment”. In: *European Conference on Computer Vision (ECCV)* (2022). Ed. by Shai Avidan et al.

[44] Dawid Rymarczyk et al. “ProtoPShare: Prototypical Parts Sharing for Similarity Discovery in Interpretable Image Classification”. In: *ACM SIGKDD Conference on Knowledge Discovery & Data Mining (KDD)* (2021). Ed. by Feida Zhu, Beng Chin Ooi, and Chunyan Miao.

[45] Mikołaj Sacha et al. “ProtoSeg: Interpretable Semantic Segmentation with Prototypical Parts”. In: *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)* (2023).

[46] Christian Schorr et al. “Neuroscope: An Explainable AI Toolbox for Semantic Segmentation and Image Classification of Convolutional Neural Nets”. In: *Applied Sciences* 5 (2021).

[47] Ramprasaath R. Selvaraju et al. “Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization”. In: *IEEE International Conference on Computer Vision (ICCV)* (2017).

[48] Hua Shen and Ting-Hao Kenneth Huang. “How Useful Are the Machine-Generated Interpretations to General Users? A Human Evaluation on Guessing the Incorrectly Predicted Labels”. In: *AAAI Conference on Human Computation and Crowdsourcing (HCOMP)* (2020).

[49] Rakshith Shetty, Bernt Schiele, and Mario Fritz. “Not Using the Car to See the Sidewalk: Quantifying and Controlling the Effects of Context in Classification and Segmentation”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019).

[50] Jamie Shotton et al. “TextonBoost for Image Understanding: Multi-Class Object Recognition and Segmentation by Jointly Modeling Texture, Layout, and Context”. In: *International Journal of Computer Vision (IJCV)* (2009).

[51] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. “Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps”. In: *International Conference on Learning Representations (ICLR) Workshops* (2014). Ed. by Yoshua Bengio and Yann LeCun. Poster Presentation.

[52] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. “Axiomatic Attribution for Deep Networks”. In: *International Conference on Machine Learning (ICML)* (2017). (Visited on 04/29/2023).

[53] Kira Vinogradova, Alexandr Dibrov, and Gene Myers. “Towards Interpretable Semantic Segmentation via Gradient-weighted Class Activation Mapping (Student Abstract)”. In: *AAAI Conference on Artificial Intelligence* (2020).

[54] C. Wah et al. *The Caltech-UCSD Birds-200-2011 Dataset*. Tech. rep. CNS-TR-2011-001. California Institute of Technology, 2011.

[55] Haofan Wang et al. “Score-CAM: Improved Visual Explanations Via Score-Weighted Class Activation Mapping”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops* (2020).

[56] Jiaqi Wang et al. “Interpretable Image Recognition by Constructing Transparent Embedding Space”. In: *IEEE/CVF International Conference on Computer Vision (ICCV)* (2021).

[57] Carlos Zednik. “Solving the Black Box Problem: A General-Purpose Recipe for Explainable Artificial Intelligence”. In: *Computing Research Repository arXiv* (2019).

[58] Jianming Zhang et al. “Top-down Neural Attention by Excitation Backprop”. In: *International Journal of Computer Vision (IJCV)* (2018).

[59] Bolei Zhou et al. “Learning Deep Features for Discriminative Localization”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016).