This looks like that... And that, and that. Developing visualisation software for ProtoPNet

Monika Ramel

MSc in Arificial Intelligence
Supervisor: Nico Potyka
School of Computer Science and Informatics, Cardiff University

Contents

1 Introduction

Despite the ongoing pressure to prioritize rapid product development over the security and comprehension of models that are becoming increasingly integral to our daily lives, the scientific community has been reliably committed to responsible practices and has considerably advanced the field of Explainable Machine Learning. One remarkable example that not only delivers a higher standard of interpretability than other, commonly used approaches but also achieves performance comparable to those of traditional models is ProtoPNet. The innovative design of using visual prototypes allows for a more intuitive understanding of the model's decision making process. To facilitate further insights into the model's behaviour as well as enhance the transparency and trust in the model, this project ventures to create a specialised visualisation tool, while simultaneously conducting introductory experiments showcasing the enhanced features.

2 Background

2.1 Overview of Machine Learning

Machine learning is a branch of artificial intelligence focusing on building systems capable of learning from data to make predictions or decisions without being explicitly programmed the decision paths. The goal is to achieve algorithms that can adapt to new situations by learning from examples and identifying patterns[7]. Machine learning is based on the concept of training models using data. This typically involves providing model with a labelled dataset, where the correct answers (labels) are known, so that the model can learn the relationship between inputs and outputs. The trained model can then generalise this knowledge on new, unseen data. The primary approaches to machine learning would be supervised learning, where the model is trained on a labeled dataset, where each input is paired with a correct output and the algorithm learns what a correct output is based on these examples; unsupervised learning, where model works with an unlabeled dataset to discover the underlying data structures on its own and reinforcement learning, where an agent is being taught based on how it interacts with its envirionment[12]. Certain decisions can be rewarded - ”reinforced” - or punished.

Despite the growing success and wide application of machine learning models worldwide, there are still several key challenges and risks that are inherent to the medium of machine learning and need further research to make it possible for models to be implemented in a way that mitigates those challenges consistently and reliably.

One of the most common problems in ML is overfitting[22], where the model performs well on the training data, but poorly on new, unseen data. This happens when the model learns to overly rely on the training data instead of generalising from it. This includes an over-reliance on noise and outliers, which becomes all the more difficult to diagnose the more complicated the system is. Ensuring appropriate generalisation is instrumental for making the systems robust enough to handle real world settings. Next very common problem is scalability, as ML models need to process vast amounts of data, the operations become not only extraordinarily computationally expensive in comparison with any other data analysis methods, but due to the vastly increased complexity, ensuring the correctness of the model's intepretations can often be beyond what traditional statistics can feasibly give insights on[21] [4].

From this follows the most critical challenge of ML and deep learning particularly, which this study aims to tackle - interpretability[5]. Many of the models, especially the most widely used ones currently are considered a "black box" because of the inability to see or feasibly understand its decision-making process. With this inability to understand the model also comes a very justifiable inability to trust the model, especially in high-stake decisions, but it is also crucial for taking accountability and avoiding situations like that in the future.

Machine learning is a rapidly evolving field that continues to push boundaries in various disciplines. However, with increasing model complexity, in particular deep learning, the demand for interpretable models is also growing. This has driven many advancements in the field of explainable AI in the recent years[5]. As more complex architectures like neural networks become more and more widely used tool globally and in many areas, the issue if interpretability becomes a central focus.

2.2 Deep Learning and Neural Networks

Deep learning is a subfield of machine learning that focuses on neural networks with multiple layers to model complex patterns in large datasets[10]. The advantage of deep learing models over other, traditional methods is the fact that they excel at automatically learning hierarchal representations of the data, which has lead to many breakthroughs in fields like image recognition, natural language processing and speech recognition[10].

The most basic form of a deep learning model is a neural network, which is a system inspired by the function of the human brain. Such a network consists of layers of interconnected nodes (meant to mimic neurons), each performing some calculation on the data it receives. These neurons are then organised into layers: an input layer, one or more hidden layers and an output layer. The network then learns by adjusting the strength of the connections (weights) between neurons based on the accuracy and error rates in its predictions. During the training, network processes inputs and computes outputs and the difference between the predicted and actual function (measured by a loss function) is used to adjust the weights through back-propagation. This process is also optimised by algorithms like gradient descent to allow the network to progressively reduce errors and improve its predictions.

Neural networks come in various architectures and the one that this study aims to examine is a convolutional neural network (CNN).

In terms of architectural features, CNN would typically include layers of convolutions where the previously mentioned filters are applied, followed by pooling layers, which downsample the data to reduce dimensionality while still retaining important information. After several such layers of convolution and pooling, the data is passed through fully connected layers to make a final prediction. This type of network is able to capture increasingly complex patterns with each layer[9].

Despite the excellent achievements that have been assured using deep learning models, their lack of transparency has been criticised.

2.3 Explainable AI

As artificial intelligence systems have been introduced to areas with decision that can have a high risks associated with potential errors such as healthcare, legal proceedings and autonomous vehicles, the need for transparency in these systems have become more urgent and important than ever. This need gave rise to the entire field of explainable AI, which seeks to enable humans to comprehend how the model arrived to a decision. A particular challenge in the explainability efforts is retaining the efficiency, effectiveness and the ability of AI to provide insights. For a model to be considered a viable alternative to already existing, high-accuracy ones it has to be on-par with them in terms of performance. The two main techniques in interpretable AI are Post-Hoc interpretability - applied after a model has been trained in order to not alter the model itself (ex. SHAP, Grad-CAM), which have been widely criticised[17][2] and Intrinsic interpretability, where models are designed with interpretability at the heart of them. Examples of the latter include decision trees and rule based models, so that a human expert can follow the decision rationale, as well as prototype-based models, which are designed to make decisions based on comparisons to learned prototypes, which are specific, interpretable examples from the training data.

2.4 Prototype-based Networks

ProtoPNet, a particular implementation of a CNN that this study will examine, has introduced a significant modification to the traditional convolutional architecture by incorporating a prototype layer into the network[3]. As opposed to the standard, fully connected layers mapping high-level features to predictions, ProtoPNet adds to these fully conected layers a prototype layer. The learned prototypes are patches of images from the training set that the network finds most relevant for making decisions. Instead of outputting abstract class probabilities, ProtoPNet comares parts of the input image to these learned prototypes and the network then makes decisions based on the similarity between them. This makes ProtoPNet uniquely able to show which prototypes influenced the decision.

3 Context of the research

3.1 Interpretability in machine learning and current state of the art

The field of explainability in machine learning, as of now, primarely emphasizes post-hoc explanations. Whether it's gradient based methods[20], activation maps[23], local surrogate models[16] or SHAP[11], all of these methods have the limitation of explaining the predictions in hindsight. In other words, they give insight into the potential factors correlating with the prediction, they have no access to the decision-making process itself. They are unable to provide any confirmation whether the underlying reasoning aligns with the data patterns, or if it's merely relying on correlation, which provides little help in being able to determine the model trustworthiness. Knowing this is crucial, especially when introducing ML models into high-risk decision making process such as partaking in court cases. This can be attributed to a key limitation acknowledged by David Alvarez-Melis and Tommi S. Jaakkola, which is the lack of stability of those post-hoc explanations[1]. Small changes in the input can lead to significantly different explanations, which makes the results unreliable[1]. Therefore, there begun a notion of introducing self-explaining models that embed their explanations within the architecture itself.

One of such endeavours is ProtoPNet. It's an excellent candidate for further exploration as it was one of the first attempts at addressing the black box nature of neural networks, while still retaining the high accuracy of deep learning[3]. Unlike traditional models, in which it's impossible to access the underlying decision-making process, ProtoPNet introduced a unique prototype layer making it possible to access specific examples from the training data to approximate which of those representative pieces have the most influence over the decision and in what way. The goal was to make its classifications more transparent, especially for a human user, since not only does it link the input to the previously learned examples, it does so in a visual way.

3.2 ProtoPNet

The core of ProtoPNet's design is the concept of prototype learning. During the training process, the models learns a set of prototypes for each class, which can be thought of as the most representative features of that class. When presented with a new input, the model compares parts of the input to these prototypes, assigning similarity scores based on how closely the input image matches the learned examples. These scores are then used to make predictions. While traditional CNNs rely on complex layers of transformations, ProtoPNet's prototype layer offers decision making that has a certain level of basis in explicit comparisons. This significantly increases the level of transparency of the model.

As mentioned earlier, the arguably the most important and intriguing characteristic is its ability to provide visual explanations for it's predictions. When a model classifies and image, it not only outputs the predicted class, but also highlights the specific part of the input that were the most similar to the specific part of the learned prototype. This approach is allows human users to be able to intuitively understand what might have happened during the decision-making process, and do so with a much higher precision and reliability.

The current limitation of the investigative efforts is, that the chosen prototypes, while significant might not provide the complete picture due to the way they are currently being extracted. Amongst many prototypical parts and decisions, displaying only one prototype is equivalent to displaying only one decision. For example: the model found a high degree of similarity between the beaks of the input bird image and the prototypical bird image. Is the beak the only determining factor? What was the model comparing and activating when it comes to other prototypes? Were the highest activations overwhelmingly amongst the predicted class, or were the activations distributed more evenly amongst all of the classes? Are there any outliers, where the model finds something particularily important or unimportant and how do they look like? Would they correlate with what a human thinks? The need to investigate these characteristics at larger scale is apparent.

Despite the achievements of ProtoPNet, it has to be noted that both ProtoPNet, as well as its derivative works have mostly relied on traditional methods of evaluation such as accuracy, as well as the basic visualisations provided by the original authors to critique and establish their own findings. When this approach is dissected, it may appear quite shallow, especially when the goal is to provide a more nuanced understanding of the model. The central issue is that the visualisation code that is meant to select the most important prototypes to display is based solely on one value, a raw similarity score. The model computes it by measuring the closeness between a part of the input image and a prototype it has learned during training. While this representation does offer a way of ranking the relevance of the prototypes, it deserves a closer evaluation, as it is is not an accurate representation of the entire decision-making process of the model, thousands of intertwined activations are at play with every decision. What the existing visualisations fail to account for is the fact that each prototype's influence on the final classification is not determined by the similarity score alone. In the current visualisation setup, it is possible that the similarity score might be high, while the overall importance to the classification score might be near 0 due to the weights applied to it. This practice might therefore lead to misinterpretations, where prototypes that are not relevant, are not only reported as such, but the user has no background on how exactly relevant it is in comparison with the other candidates. While the correlation between high similarity score and high weighted similarity score might exist, it might not be enough to justify all of the choices for the most important single prototype.

To address this issue, this paper will propose a more robust way of evaluating which prototypes to display by introducing weighted similarity scores, as well as showing the context of these values against other top scoring candidates. The introduction of weighted similarity scores not only improves the accuracy of assessing which prototype is worthy of selection but also enhances the interpretability of the model. By visualising the prototypes that actually carry weight in the decision-making process, we provide users with a clearer, more faithful representation of how ProtoPNet arrives at its prediction.

3.3 Overview of ProtoPNet's derivative works

ProtoPNet's unique approach has inspired a range of derivative works, building up on and expanding the idea of interpretable deep learning by introducing new mechanisms to enhance or extend the model's transparency, decision making process, or both. These efforts such as ProtoTree[13], PiPNet[14] or ProtoPShare[18], reflect the increasing recognition of interpretability as a crucial aspect in the development of machine learning models. While these models introduce valuable innovations, they still grapple with some of the same challenges seen in ProtoPNet, particularily in terms of evaluation and visualisation. This chapter will provide an overview of some example works in order to highlight the importance of introducing enhanced visualisations.

ProtoTree is one of the most direct extensions of ProtoPNet. It enhances the interpretability by introducing a decision tree where each node learns a prototype and decisions propagate down the tree based on the similarity between the input and the prototype at each node[13]. This approach makes the entire process inherently intepretable and understandable to humans. Path taken through the tree provides a clear explanation of how the model arrives at its final prediction. The major flaw of this approach is that only a similarity of a prototype gets taken into account, not the overall contribution, once again possibly overstating the significance of a single prototype over the collective contributions of all prototypes.

PiPNet builds on ProtoPNet by introducing prototype integration pathways, where multiple pathways of feature integration are used to improve the robustness and generalization of the model's decisions[14]. By adding these pathways, PiPNet aims to capture a broader set of features from different regions of the input, making it less reliant on a single prototype and potentially improving accuracy in more complex settings. This approach allows the model to consider diverse and sometimes complementary prototypes that may collectively contribute to a decision, rather than focusing on a single one. However, PiPNet's complexity introduces challenges in maintaining clear interpretability. The multiple pathways make it harder to pinpoint exactly which prototypes had the most influence in the final decision, potentially diluting transparency as compared to simpler models like ProtoPNet.

ProtoPShare extends ProtoPNet by sharing prototypes across multiple classes, effectively reducing the total number of prototypes required and enhancing the model's efficiency[18]. By allowing prototypes to represent features relevant to more than one class, ProtoPShare introduces a more resource-efficient method for maintaining interpretability while keeping the number of prototypes manageable. This shared structure can improve model scalability and ensure that it doesn't overfit by learning overly specific features. However, the downside is that shared prototypes can reduce specificity, potentially leading to misclassifications when certain prototypes apply too broadly across classes. This trade-off may introduce ambiguity into the decision-making process, as it becomes less clear which class-specific features are influencing the model's predictions.

One significant limitation seen across all of these models is the inability of these models to provide a wider view of the decision process, focusing rather on either a few or even a single, selected contribution, leaving us with fragmented reasoning. For instance, a model might reach its decision through a combination of several moderately important prototypes, that are not high enough to rise as the "top" similarity - something that every single mentioned model is doing. This narrow view can be especially problematic in not only this scenario, but especially when the classification is wrong, where assessing not only the classification path is important, but also the activations and decisions that have been made for the correct, but unselected class. By adding even this one additional line of reasoning - displaying the prototypes of the not predicted, but also correct class, the human interpreting the input can study what has failed within the wrong alignment and by what margin the model is wrong. This then can potentially be used as a debugging tool, or a tool informing the data collection process for supplementing the model with gaps.

These models also lack insight into the overall certainty of a decision. By conveying only one or two prototypes and only considering them in terms of the predicted class, they fail to note the broader context. The user is unable to see whether the model considered any alternative classifications and what was their perceived plausibility. This problem becomes all the more severe, the more prototypes are being introduced during learning. In such cases, the prototype importance is more likely to fluctuate and therefore it would be more difficult to pinpoint whether coincidental, correlation data (such as background features) is being taken into account, significantly affecting trust in the model. An additional benefit of being able to see activations across all possible classes would make it possible to study trends behind the wrong and right decisions even further - does the model tend to be confidently wrong or is it not sure of any class in particular and chooses one with the highest activation, but only by a small margin? Do the predictions with similar activation patterns have commonalities and justifications also visible to the human eye? For example, does a majority of the confidently, correctly classified classes have vivid colours or high contrast? Such insights would be possible to reach only if we revise the existing approach.

3.4 Aims and objectives

As identified previously, advancing intepretability is crucial in machine learning models. This project aims to advance the understanding of high performing models such as ProtoPNet and enhance their transparency by developing a framework for visualising both its architectures. This venture will set a foundation for further, systematic study of individual prototypes and their interconnections, which will in turn allow for greater insights into model robustness. Factors like lighting conditions for example may or may not be a problem to various degrees. By visualising the prototypical components and structures of these models we can better identify and address such issues, potentially increasing further reliability.

The study seeks to investigate the broader activation patterns across all classes rather than solely focusing on the predicted class. The statistical distributions could help identify key areas of weakness in the model, such as prototypes that overly rely on background features, leading to classification errors, as well as unveiling the questions of whether there exist underlying trends in activations between certain classes or images of certain characteristics within the classes.

One of the key objectives is to move beyond the simplistic view of binary classification outcomes - right or wrong - and instead develop deep visualisations that capture the degree of certainty with which model arrives at its decision. By visualising not only the top-ranked prototype, but also the distribution of activations across all prototypes for all classes, this study aims to offer a richer picture of the decision-making process, which could allow researchers to observe whether a model is making decisions based on narrow margin or if it confidently selects one class over the others and under what circumstances does the certainty change (even though the decision might still remain correct). These insights can help establish reliability of the model's predictions as well as unveil potential areas of improvement within the selection of the training and/or prototype data.

To retain a level of human-readability, the contributions that will be given the most detailed visualisations will be a comparison between a predicted class and a correct class (in cases where the prediction is wrong). This will enable users to asses how close the model was to making the correct decision, as well as which parts were activated in each of the cases respectively. Understanding these dynamics might offer a way of understanding how and why the mistakes have happened, potentially pointing out which prototypes need adjustment.

A key objective of this study is to provide tools that enable researchers to analyse how exactly adverserial attacks affect the model at an activation level as well as provide a tool for acknowledging and analysing weaknesses in the existing models. By enabling the visualisations of the weighted activations and similarity scores for each prototype across all classes both before and after attack, the software will allow for a detailed examination of the exact impact an attack has on the model's decision process or what potential biases and reliances are present. This approach will offer insights into whether for example an attack has made the model less confident overall - by bringing all the contributions to a similar level in terms of distribution - or whether it has pushed the model to confidently misclassify the image by significantly increasing the activations for an incorrect class. Thanks to this ability to compare pre and post-attack distributions directly the researchers will not only be able to see if the attack introduced confusion, or if it exploited a particular vulnerability in the model, but also to pinpoint which classes or images in particular are more prone to error during this type of an attack.

Another potential insight that might be uncovered by the study is being able to uncover potential overfitting or underfitting. For instance, if certain prototypes consistently activate for both correct and incorrect classifications, this might suggest that the model has learned some unknown correlation in a way that hinders its ability to generalise correctly. Such insights could help inform strategies for training.

Lastly and most importantly, this study aims to provide the scientific community with a practical and usable interface that can be interacted with and modified easily for the sake of future experiments and use. The goal is not only to provide theoretical insights, but an easy to use tool that can be expanded on and easily integrated into any setup.

3.5 Review of relevant literature

Other attempts at enhancing visualisations as well as attempting to draw connections between what the model ”sees” and what humans see have been made. This chapter will tackle the works most relevant to this attempt.

The paper ”This looks like that... Does it?”[8] explores the extent of interpretability in ProtoPNet. The authors assess effectiveness of the prototypes in actually enhancing the interpretability of model's decisions. In particual, they question whether the chosen prototypes correspond to human-understandable features or rather align with the model's internal abstractions.

One of the paper's significant critiques is that there exists an assumption that proximity in the feature space correlates with human interpretability. The authors argue that this assumption can be misleading, as high similarity scores may not correspond to features that are meaningful from a human perspective. It is important however, to consider that prototypes don't necessarily have to align perfectly with human thinking to enhance interpretability. Prototypes that reflect the model's internal abstractions can still be valuable, especially in the debugging process.

The paper discusses a JPEG experiment where the authors critique ProtoPNet's ability to interpret compressed images. This critique instead of pointing to a specific flaw within ProtoPNet highlights a broader issue in computer vision. Models do often struggle with images that have been altered in a significant way such as compression. JPEG compression can introduce artifacts that alter the image's features in a way that makes it difficult for the model to adapt to, especially if the model has not been trained on a variety of different compressions. This challenge is not unique to ProtoPNet and is quite well known in computer vision[6][19]. The authors use the jpeg experiment to argue that ProtoPNet is not as interpretable and robust as claimed, but the sole fact that they were able to extract the data about prototype similarity scores and reach meaningful conclusions about the model's vulnerabilities through that does prove that the model does have a meaningful level of interpretability. This would not have been able to be achieved with a black box model, where the only data available would have been for example the accuracy scores, which might be deceptively high and might have gone unnoticed during traditional evaluations of black-box models. ProtoPNet, after extraction of the relevant data, was able to offer insights into its internal patterns, which while not perfect in terms of how widely interpretable the findings are, is highly interpretable for human experts in the field. This indicates that the model does offer valuable insights, even if they're not perfect. This insight would not be possible without the inclusion of the prototypes.

While the authors do critique the use of similarity score by highlighting cases where the similarity score is high, but the prototype itself doesn't correspond to anything relevant from a human perspective, deeming it superficial and raising concerns about techniques that may seem transparent on the surface, but do not provide sufficient insight after a closer examination.

In the context of this research, "This looks like that... Does it?" is definitely the central piece of literature. The paper's emphasis on the misalignment between the model's internal representations and human-understandable features directly inform the goals of this study. The insights provided by the paper also suggest that interpretability in prototype-based networks should extend beyond the identification of representative prototypes, but that it should encompass how these prototypes interact with the input across all classes, potentially revealing trends, uncertainties and the collective behaviour of the network

Another work that excellently, critically examines how well prototypes in CNN-based models human-understandable features and whether they provide sufficient transparency in decision making. While the previous work directly informed this study, the "This looks like that, because..."[15] is more of an aspirational study, that will inform future directions and great visualisation efforts that are worth mentioning. The authors scrutinize human-interpretability of models such as ProtoPNet, emphasizing the frequent disconnection between the model-derived prototypes and features that humans consider significant. They ague that while prototypes are promising tools for XAI they often reflect internal abstractions that are not easily aligned with the human perceptions. The findings of "This Looks Like That, Because ..." are excellent and particularly relevant to the current research. This study complements the approach taken in the project by focusing of the visual alignment between machine-derived prototypes and human-understandable features. While approach taken in this study still relies on similarity and activation, it expands on these insights by incorporating weighted similarity scores and visualising alternative class activations - key areas that Nauta et al. have noted but haven't explored in depth. By addressing these limitations the projects offers a more nuanced view for future evaluations. This approach also considers the broader context of model confidence and uncertainty, which have not been deeply explored in earlier works.

By using box plots, the team provides a valuable tool for analysing the overall behaviour of the models by displaying how frequently certain characteristics (or prototypes in case of prototype-based networks) contribute to decisions. The box plot offers a comprehensive view of global trends, showing how the distribution of a particular characteristic (like colour, texture or shape) behaves across various classes. This enables a nuanced assessment of whether the model is relaying on consistent and meaningful features.

Future iterations of this study could expand on the visualisations by encorporating the methodologies introduced by Nauta et al. to further enhance transparency of the visualisations. Integrating them would significantly enhance the ability to evaluate ProtoPNet and its decision making, however while this research has contributed greately into an alternative way of thinking about global characteristics, enhanced insights informed by the study would be outside of the scope for now.

4 Significance of the research and research problem

The primary research problem addressed is the inadequacy of current visualisation techniques in prototype-based neural networks. Existing methods rely on a limited set of characteristics like raw similarity scores to determine which prototypes to display, while not acknowledging that the prototypes with higher similarity scores don't always map onto the most influential prototypes, potentially misleading the person assessing the results. Additionally, only one of the significant prototypes is ever present during the assessment. This might once again give a wrong impression of the decision making process giving an impression that for example the model has made a decision based solely on bird's wings, when the next n top prototypes focus for example on a beak, or even worse, what if the most significant prototype is focusing on a feature that is, or at least appears to be significant, while the next top n prototypes reveal, that the most activated feature for the predicted class is actually a superficial background feature? Such a problem would be able to be uncovered using the software. The research becomes particularly significant when examining model behaviour in edge cases - such as when the model makes incorrect classifications. In these situations being able to compare the activations for multiple prototypes and multiple classes might unveil what exactly are the features confusing the model. This research sets out to answer several key questions:

Are the most similar prototypes always the ones corresponding to the most influential prototypes? How well do the most similar prototypes reflect actual decision-making of the model?

What role do less prominent prototypes play in the final classification? In the current models, prototypes that do not have strictly the highest similarity are overlooked, even though they might still contribute to the dedcision.

How does the model handle uncertainty in classification? By visualising the activations and weighted similarity scores across all possible classes, this research aims to shed light on whether the model is condifent in its decision or whether it considered alternative classification.

Can enhanced visualisations help identify patterns in misclassifications? By comparing the highest activated prototypes in both correct and predicted classes, this research could help identify weaknesses in the model and inform strategies for improving its performance.

Based on the background and the discussed research problem, we can derive following hypotheses to guide the study:

H1: Prototypes with the highest raw similarity scores are not always the most influential in the final decision.

H2: Visualising prototype activations across all classes, rather than just predicted class will reveal additional insights into model uncertainty and alternative classifications considered during prediction.

H3: Even in the cases of correct classification, it is possible to observe the model basing its decision on undesirable influences thanks to the enhanced visualisations.

5 Methodology

With these interpretability questions in mind, this project will aim to tackle the current limitations in how prototype activations and model decisions are visualised by implementing an enhanced visualisation framework that provides a more comprehensive understanding of how multiple prototypes are utilised during classification. This section will detail how exactly these questions are aimed to be addressed.

5.1 Research design and approach

The study will use ProtoPNet trained on a clean, unprocessed CUB-200-2011 dataset. This clean dataset provides a strong baseline for assessing the interpretability under normal conditions. This will allow further work to be built upon it and serve as a comparison ground for future experiments and research directions. To address the research questions and hypotheses in a systematic way, the experiment is devided into four main stages:

5.1.1 Prototype visualisation enhancements

The primary objective of this phase is to enhance the visual representation of prototypes utilised by ProtoPNet in making classification decisions to address the current gap in the literature, where only single prototype is used to reach conclusion, which provides an incomplete picture of the model's decision-making process.

To address this, the first step involves extracting multiple top-n prototypes per classification decision. This approach aims to provide a comprehensive representation of the most influential features that contribute to a given decision. Enhanced visualisations are then displayed alongside to display the relationships between the input images and their top-n activated prototypes via activation heatmaps. This allows the researchers to examine which components of the image are taken into account when assessing the similarity. A graphical user interface is also necessary to facilitate the user exploration.

This phase aims to generate detailed visualisations that depict how exactly the addition of multiple prototypes enhances the understanding of the model's decisions.

5.1.2 Quantitative analysis of weighted similarity score

To deepen the analysis of how different prototypes influence classification decisions, a weighted similarity score will be used instead of raw similarity scores.

In ProtoPNet, similarity scores are calculated for each prototype by comparing parts of the input image with learned prototypes. These raw similarity scores will be multiplies by the model's weights to provide an approximation that is closer to the model's actual reasoning. A series of visualisations will be implemented to display distributions of not only the new scores, but the weighted scores alone and similarity scores alone, to further examplify the inadequacy of those measures as lone determiners of which prototypes should be displayed. These visualisations will include bar charts and histograms to help illustrate prototypes carrying the most weight in the final decisions.

5.1.3 Misclassification and uncertainty analysis

By visualising the prototype activations across all classes, this phase aims to reveal any underlying dynamics that may be important into assessing how they lead to errors and assess how confidently the model makes its correct and incorrect predictions. As a proof of concept, a subset of misclassified input samples will be identified and commented on. Visualisations are developed to display prototype's weighted activations, similarity scores and weights themselves for both predicted and correct classes, which will help to conceptualise how strong the lead of the misclassified class is over the correct class. This will also help to visualise whether a particular prototype has contributed to a disproportionate extent.

A makeshift measure of confidence analysis is also introduced. Confidence is estimated by analysing the skeweness of the graph of all the activation scores to approximate the confidence with which a certain class has been picked.

5.1.4 Comparative analysis of predicted vs correct classes

For each misclassified instance, detailed visualisations are generated that highlight most activated prototypes for both the predicted and correct class, allowing for a side-by-side comparison and an intuitive understanding of where the model's reasoning went wrong. This helps to manually identify instances where there is a misalignment between the model's decision logic and the human understanding of the features, for example in an event of the model giving a heightened importance to the background or other irrelevant feature.

5.2 Approach Considerations

Due to the computational limitations of the hardware available during this study, it was necessary to utilise a pre-trained ProtoPNet models with aavailable prototype activations for analysis. While training ProtoPNet from scratch would offer more flexibility, the computational demands of training, pruning and validation processes would make it beyond the available resources for this project. The use of existing models allows the research to focus on the primary objective: improving the interpretability through a deeper analysis of prototype activations and developing visualisations.

The models in this study are trained on a clean CUB-200-2011 dataset. The decision to use clean data, even though the authors do provide a model trained on compromised data as well is to provide a more controlled and realistic environment for evaluating how ProtoPNet behaves under normal conditions. This allows to create a baseline to later compare any further findings to if it were ever necessary to extend this study beyond into for example investigating the changes in performance after adversarial attacks or simply low quality data. While this decision does enhance resource efficiency and allows to focus on interpretability, there are limitiations and challenges to be acknowledged. The lack of flexibility that comes with having a pre trained model means lack of training flexibility and customisation of hyperparameter tuning, prototype pruning. Any model adujstments that the research team had not disclosed also cannot be verified or altered. While these limitations do not detract from the study's ability to evaluate ProtoPNet's core behaviour, it is important to keep in mind that certain findings can possibly be attributed to the specifics of how this particular model had been trained. A certain, low level risk with using a clean, well performing dataset is that there might not be enough variation in the dataset to observe any changes and therefore to showcase the capabilities of the software reliably. On the other hand, if conclusions are able to be reached even in a high accuracy, realistic environment, that could potentially strengthen the case for the use of the visualisation software. While the models used in this study do provide some insights into ProtoPNet's behaviour, they are limited to only a single dataset and may not fully generalise to other datasets or variant architectures.

To address these limitations, the analysis focuses on detailed, instance-level comparisons of individual predictions.

5.3 The Intended User and The Interface

The four archetypes of users that would be the most likely to use the project would be academic researchers, machine learning engineers and students. All four of these groups would have differing needs and in the event of a conflict, the first group would be prioritised, however all four have informed the design decisions.

The key user group, academic researchers, are very likely to be well-versed in the possible model behaviours, as well as meaning of any terminology used. They are highly skilled in designing and extending machine learning models, however they are less likely to have in-depth knowledge of different visualisation frameworks in comparison to software developers or UX experts. Their primary need would be the depth, flexibility and transparency of the visuals. The code then needs to use widely known libraries to allow for maximum modularity and flexibility without distracting from the primary research and the insight that is to be gained from streamlining the process and excessive reworking needs to be avoided.

Another possible requirement from such a program would be clarity. A tool that would make it possible to asses performance or patterns at a glance would not only serve as a possible research tool, but also a debugging tool. By automating gaining visual insights, the data is much easier to be compared against eachother. This design need was satisfied by the choice of PyQt5 for GUI for its ease of use as well as cross-platform compatibility. Even though it's not the most modern-looking or advanced, Matplotlib was chosen for the plotting component, since it's widely used not only by the computing researchers, but also by the scientific community as a whole, which future-proofs the project for streamlining any possible collaborations between the scientific communities. Interactivity - the system allows users to select images, classes and different types of visualisations as well as interactivity within those visualisation that allows for more detailed inspections. To summarise, the design process puts value on leaveraging familiar and commonly used libraries to maximise the potential usage by fellow scientists.

5.4 Implementation Design

To effectively address the goals and facilitate a more nuanced view of ProtoPNet's behaviour, the model should include the following key features:

5.4.1 Prototype Activation Visualisation

At the heart of ProtoPNet's interpretability is the concept of prototypes that represent parts of images and the similarity of these images to input images (both as judged by a human and the model) and the role they play in classification. As identified previously, understanding how multiple prototypes interact is crucial, therefore the designed software has to be able to visualise and provide the means to assess more than only the single top prototype.

The software will display not only the raw similarity scores attributed to the prototpes, but the weighted similarity scores, that directly influence the model's output.

Those which will vary for each prototype per activated class and per input image, whereas the similarity score will only vary per input image.

The user will be able to view prototype activations (similarity score), weights and the weighted activations with a special focus on those per predicted and correct classes (in cases of misclassification). This will allow a compairson of how those different values change and how relying on only one of them might not be reliable. This will help the users understand where the model's predictions deviate from the expected outcome and allow for a more critical view of how the activations work together.

5.4.2 Class-Specific Confidence insights

Understanding confidence is crucial for assessing reliability. The software must be able to give insight into the overall certainty of its decision as well as any alternative classes that competed for the top place.

By showing weighted prototype activations across all classes, the software will allow users to assess whether the model considered other classes with high overall similarity scores, but ultimately chose another class. To reiterate a point of not relying on a single prototype, there will be a sister graph that will only show the maximum or top prototype by its weighted activation. This will aid with seeing how a dominant similarity does not always translate into a classification - for example, only one of prototypes activated for a certain class can have a dominantly high activation, while the others will be relatively low, leading to not being the deciding or insightful factor in the classification. The software will plot histograms and bar charts illustrating the distribution of weighted similarity scores across all prototypes for each class. This will give the user a clear indication of how confident the model was in its prediction versus alternative predictions. This confidence can be measured through metrics like the margin between the highest and second-highest scoring classes. In the future, this could be another class added to the prototype visualisation interface besides the predicted class and the correct class. Another measure of confidence could be skeweness of the graph, where the low values would indicate a certain degree of confusion.

5.4.3 Error Analysis Capabilities and Interactive interface

For misclassified images, the software will provide visualisations that highlight which prototypes contributed the most to the incorrect decision and how those prototypes differ from the correct class prototypes. Users will be able to see whether the missclassification was due to over-reliance on certain background features of irrelevant parts of the image.

The software also needs a degree of interactivity. Since the software will potentially be used on hundreds or thousands of classes, that's now many insights it will have, therefore an ability to interact with the data, including enabling more detailed view is crucial. Every visualisation will also have the capability to be saved.

5.5 Details of implementation

5.5.1 Software Architecture

The entire project is based around the code based on "This looks like that"[3] as well as "This looks like that... Does it?"[8] who kindly made their code readily available. The logic is encapsulated in the two key files: main.py and local_analysis.py, which manage the GUI and the prototype analysis respectively forming the core of the system. Since the wider analysis of the dataset as a whole is only a small portion of this research, it has been placed in the main class as a placeholder. Were we to continue with the development, it would either have its own class. The implementation is done entirely in python to comply with the most commonly used language in machine learning as a whole. This allowed for use libraries like PyQt5 for GUI, PyTorch for handling deep learning models and Matplotlib for the visualisations.

5.5.2 Data Loading and Preprocessing

In main a CustomImageFolder class is used to handle the dataset. In this will be the CUB-200-2011 dataset. These images are then being preprocessed before being passed into the model. Each image needs to be normalised and resized to ensure it matches the input format required by the pre-trained model and by loading data in batches, the DataLoader helps handle large datasets. After that, either a saved model can be fed into the code, or a new one has to be trained independently.

5.5.3 Weighted Similarity Scores Calculation

The similarity scores from the prototype layer are multiplied by the learned prototype-class weights (obtained from the final linear layer) to determine the influence of each prototype on the model's final decision, which ensures a more proportional representation in terms of prototype importance.

5.5.4 Prototypical Image Analysis

For each prediction, the system identifies the prototypes that contributed the most either to the final prediction (predicted class) or the correct class in case of a misclassified input. This is done by ranking the prototypes based on their weighted similarity scores. These top prototypes are visualised along with the portion of the input image that activated the most strongly. Using bounding boxes, the relevant part of the input image is highlighted to show which part of the image corresponds to the most activated prototype.

5.5.5 Charts

1. Total Weighted Activations per Class
   Classes with higher total activations would suggest that the model is more confident in its predictions for those classes, which can be supported by the fact that in this dataset, every single prediction has corresponded to the class who had the highest weighted total activations (appendix).

2. Maximum Weighted Activation per Class
   For each class the software computes the top weighted activation across all prototypes and only the highest weighted activation is plotted on the chart. This histogram highlights the most influential prototype for each class and in conjunction with the previous graph, it helps to evidence the fact that there will be disparities between the distribution of the graph with only the most prominent activations and the total activations. It shows that over-reliance on one example is not a sustainable or justifiable way of assessing models. It can however inform a further investigation into which prototypes are worth further exploration and comparisons. On the other hand, if one prototype overwhelmingly dominates the decision-making process, it may raise questions about over-reliance on certain features, also contributing to the debugging process.

3. Filtered distribution of weighted prototype activations
   The filtered distribution distribution chart displays only the weighted activations that have values greater than 0.1. For both the predicted class and the correct class, weighted activations are filtered and their distributions are

shown. The reason for filtering is that if all the prototypes are included, the values close to zero become so overwhelmingly large, that the graph can become unreadable. In some cases it was borderline impossible to asses if values that were not close to 0 even existed without an extensive manual search. It allows the user to focus on the prototypes that have a meaningful contribution to the decision making process. By comparing the predicted class and the correct class activations, users can assess the disparity between how activated the wrongly predicted class was in comparison with the correct one.

This view allows the user to see how closely the two classes were and how the different, significant prototypes compare to each other. It's making it possible to reveal potential confusion or uncertainties. The red vertical line helps pinpoint just how influential the highest weighted prototype is and how much it stands out in comparison with the others.

4. Filtered distribution of Prototype Weights
   Once again, the filtering is applied to not overwhelm the user and the visualisation with the fact that a vast majority of the weights are very close to 0 to the point that the high weights would sometimes not even be visible without significant manual adjustment of the graph. In conjunction with the previous and following graph, it can identify whether the model is considering prototypes of high activation and low weights or vice versa - which characteristic is dominant in the decision.

5. Distribution of Prototype Similarity Scores
   In this histogram, the green line represents the exact values of the activations for the prototypes of the predicted class, while the yellow line represents those of the correct class. The red vertical line indicates the activation of the prototype with the highest weighted similarity score in the predicted class, making it easier to see how that particular prototype stands out. The insight of the red line corresponding to the highest activated prototype helps to diagnose cases where the most influential prototypes will correspond to the ones with the highest similarity scores, how often and to what extent. The conjunction of this and previous graph can diagnose cases where in case of misclassifications, the similarity scores are relatively high but lower comparative weights for the correct class, which might suggest that a highly significant classes are being overlooked.

6. Distribution of Total Weighted Activations Across Classes

The final graph provides an overall view of how many classes fall into what bracket when it comes to total weighted activation. A somewhat alternative view to the first graph. While the first graph equips the user with the ability to see which particular classes have what activation, this graph showcases the extent with whether certain classes have been particularily dominant over the others and to what extent.

7. Additional skeweness analysis graph.
   The first graph that appears when running the program. This graph is a foundation for any future expansions on the software targeting the entire dataset as opposed to prediction-level insights. It's analysing the skeweness of Distribution of Total Weighted Activations Across Classes across all predictions for all classes and comparing it to the accuracy rates to analyse potential correlation.

6 Results and discussion

6.1 Discussion of findings in relation to the research questions

6.1.1 Hypothesis 1

"Prototypes with the highest raw similarity scores are not always the most influential in the final decision."

Or in other words: is a single value enough?

Only one counterexample is needed to disprove an absolute statement, a statement that the other visualisations have inherently introduced with their practices. The argument that is implicitly stated every time a single prototype is used to assess the quality of the predictions, and further than that, the prototype chosen is only based on a single value. That statement is: "Such reliance on a single prototype is enough". The following example shows that there are cases, where that isn't true and even if those cases are in a minority, they exist. In fact, it is all the more important to consider those edge cases whether they do or do not have a minority status, as dealing with anomalies will help us prevent critical failures.

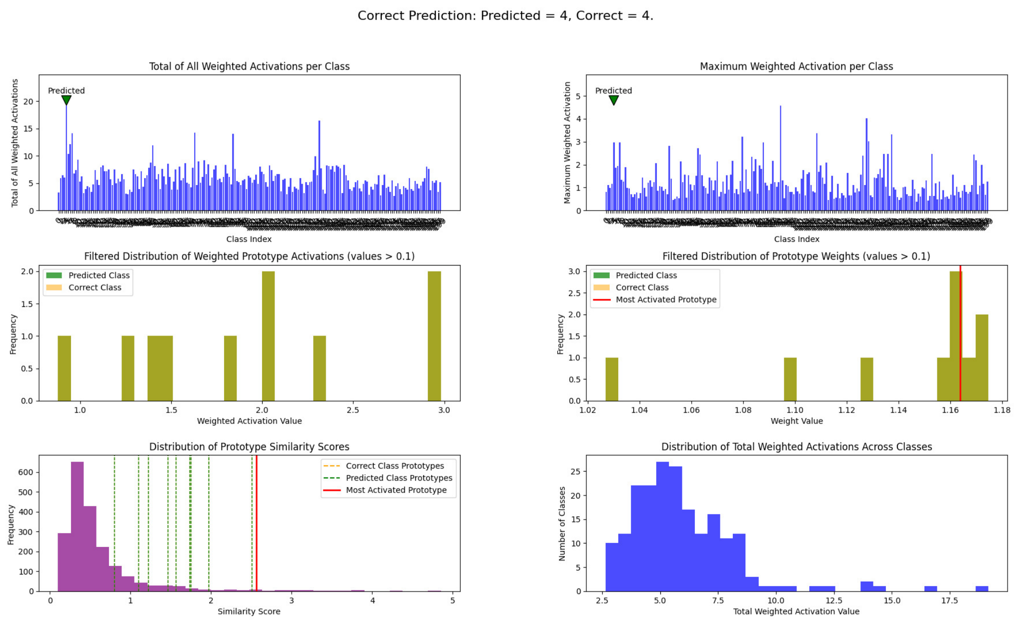

Figure 1: Unusual behaviour example

This entire prediction is the single greatest example of why a single prototype or a single value can't be a deciding factor in our judgements about the accuracy or the though process of the model. If we were to talk about a most activated prototype - no matter whether we talk about weights alone (middle right graph) or similarity scores (lower left one) - in both of those cases, the highest weighted similarity score (the column furthest to the right) has not been a result of either a highest similarity score nor has the highest weight - the values that make up the highest weighted activation score are highlighted in red - making it apparent that either one of those measures is not a direct translation to how the prototype influences the model, thus supporting the hypothesis that those scores alone are unreliable. In fact, on the histogram of weighted similarity scores itself it can be observed that a single weighted similarity score would not be a sufficient measure either, as there are two prototypes whose activations are close enough together in their influence that they're in the same histogram bin. The importance of how exactly multiple prototypes influence predictions only starts to reveal itself then looking at the first two graphs. The prototype's highest weighted activation for correctly predicted class as indicated by the green arrow in the top left graph was scored relatively low, looking at the top activations alone, it was not apparent at all that this class would be the one to be predicted. When taking into account total activations of all possible prototypes, however, it is enough to outweigh those initial single-prototype outcomes and the correctly predicted class clearly comes out on top in the left upper graph. Truly, no single predictor in this case was enough to assess an accurate model performance and only the aggregate of all of them has shown it.

6.1.2 Hypothesis 2

”Visualising prototype activations across all classes, rather than just predicted class will reveal additional insights into model uncertainty and alternative classifications considered during prediction.”

While this insight is partially supported by our findings, to have a better understanding of the intersections between uncertainty and the prototype activations, a more extensive research is needed.

This study has preliminarily used skeweness of the Distribution of Total Weighted Activations Across Classes graphs to approximate the model’s certainty, it is not a perfect measure. While it can’t directly inform us of when there are only a few prominent classes competing, it definitely does inform us of when there are no particularly prominent classes, indicating confusion or uncertainty.

The data-wide visualisations on all available models show that for this dataset, there might be a correlation between skewness and accuracy of the predictions. The higher the skeweness (thought to be associated rather with a single dominant class, since all the classes will be concentrated at the left hand-side of the graph) the higher the number of accurate predictions, with the inverse being true for the lower skeweness. All of the graphs represent a clear positive correlation between the skeweness of the activations and the percentage of correct predictions. As skeweness increases, the percentage of correct predictions also increases significantly. The differences in the shape of the graph are heavily dependent on the architecture, serving as a further example to be cautious with any claims that may potentially be dataset or architecture dependent. This example shows us the potential in extending the research and visualising different aspects of model behaviour that was introduced on a prediction scale to a global scale and comparing the findings across architectures and datasets.

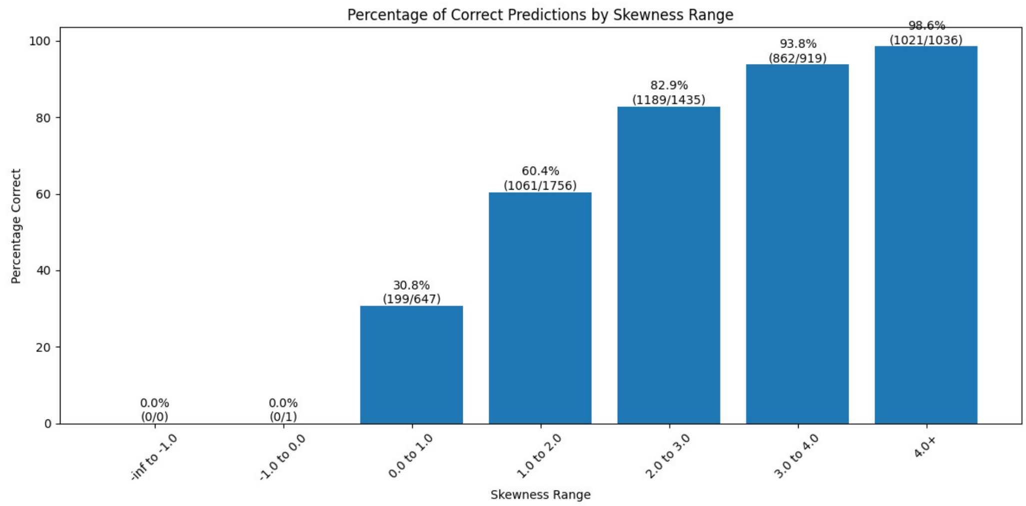

Figure 2: Skeweness on Resnet18 model

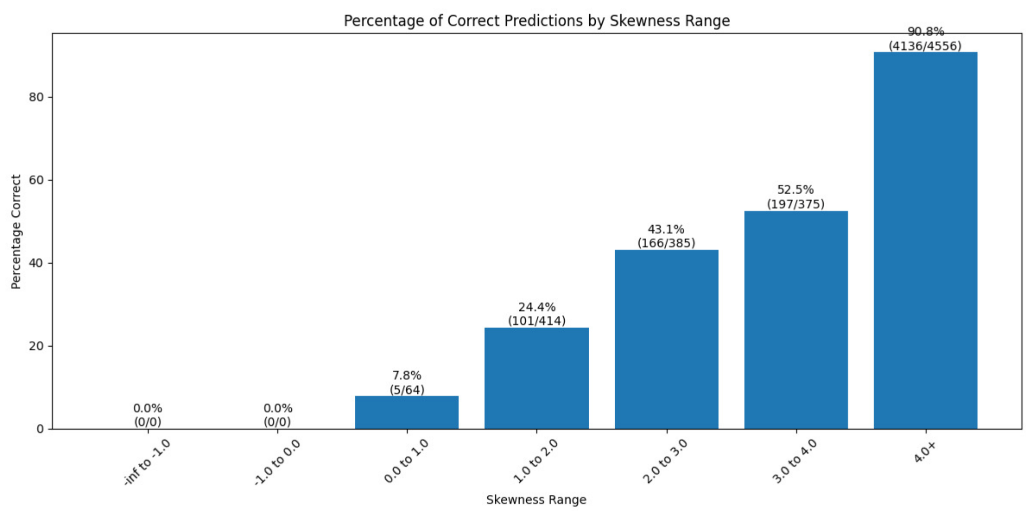

Figure 3: Skewness on Resnet34

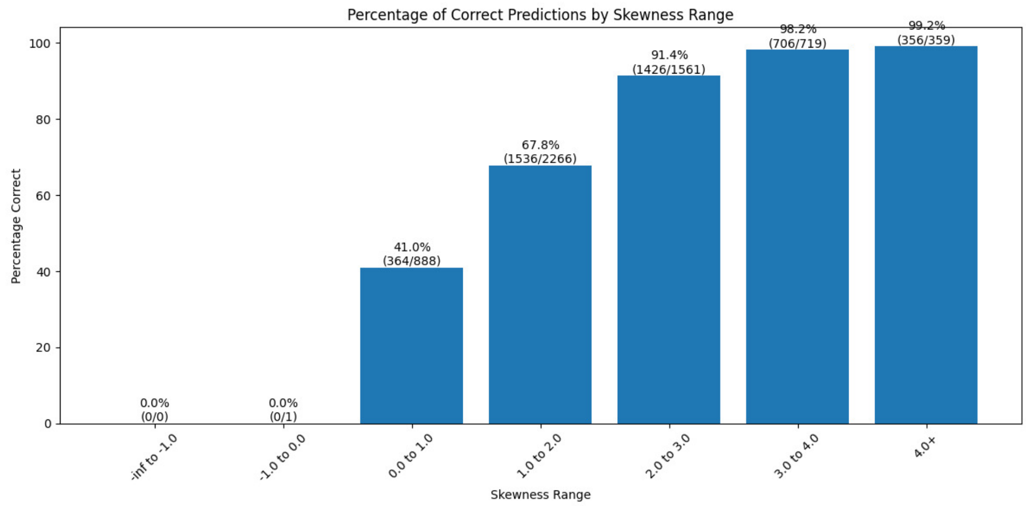

Figure 4: Skewness on VGG19

The limitation of using skewness as a proxy for certainty does not acknowledge cases where for example two classes were significantly activated, competing with eachother. There are many reasons why more work is needed in this context, for example how do these findings relate to softmax activations.

To tie it back to the hypothesis: the visualisation did get enriched by the addition of more classes and a degree of insight was gained both through the global analysis as shown in this subsection, as well as through the class-wide view of the first two graphs as shown in the figure of 6.1.1. The findings presented here showed a great degree of change between the models, so conclusions are yet to be reached about ProtoPNet as a whole. Another consideration is that the skeweness does correlate with accuracy for this particular dataset. Datasets with a certain vulnerability being exploited might show completely different outcomes and even further than that, different types of vulnerabilities might show different degree of change like in a case where a model is being exploited by making it think that something is really something else. Such a case could for example show higher certainty scores with a low degree of correctness. Such behaviour could be used to alert researchers of such an influence being present. It is still unclear however that in such a case what exact insights and differences, if any would be present

6.1.3 Hypothesis 3

”Even in the cases of correct classification, it is possible to observe the model basing its decision on undesirable influences thanks to the enhanced visualisations.”

Coincidentally, an example supporting this hypothesis can be found in the very first prediction of the very first class. The enhanced visualisation allows us to recognise that even though the first few prototypes could be somewhat justifiable or humanly understandable, the further explanations degrade in quality showing over reliance on unimportant features. It is subjective, but it can be argued that it recognises a similar wing shape, texture or colour. The second best activation however, already reveals that there exists a significant activation in the background area, which is further confirmed by the fourth and fifth prototypes. As we have established in 6.1.1, multiple prototypes have an influence over the final decision, so activations showing reliance on background in a second or fourth or even fifth most activated prototypes should not be trivialised as it will have an effect on the decision making process.

Figure 5: Correct prediction showing high activations on background features

Those visualisations show that the model’s decision was influenced in a significant degree by a prototype that doesn't align well with the seen image, even though the prediction itself is correct. This example not only supports the shown hypothesis, but also highlights how the accuracy alone is not a sufficient enough predictor of success.

6.2 Limitations of the study

The limitations were acknowledged alongside every decision where they were noticed, but a comprehensive overview of them will be presented here, and it will serve as a foundation to inform future research directions.

As acknowledged many times before, the study, the exploration and its conclusions have been limited by the data available, meaning that there is significant lack of generalisability. While there are some limited observations, they could very well be a symptom of a chosen dataset as opposed to being an absolute of how the model deals with data as a whole. Even within the dataset itself, there can be considerable differences depending on the image's features like contrast or lightning. We cannot be certain of such tendencies, so we must abstain from statements suggesting conclusions in this matter.

A survey with statistical comparisons of how different datasets and differently trained models could potentially unveil the tendencies of ProtoPNet itself and provide a more informed view of how it could be extended and what issues specifically need to be addressed to make it less vulnerable. This could provide efforts like ProtoTree be even more intentional and informed in their attempts by providing a more precise information about the gaps and challenges within the model.

While the model does enhance interpretability overall by unraveling the tendencies across predictions, it has not been measured how and to what degree it does so. It might still be the case, that the criticism posed in "This looks like that. Does it?"[8] prevails and the prototypes won't exhibit human-understandable features across predictions and the added complexity might make it more difficult to interpret. This might be the case particularly when the prototypes are activated by subtle or abstract features. A survey on human users, especially experts in the field, could greatly inform the future of this development.

A significant challenge of the software is the amount of classes that have to be accommodated in a single, limited, visual space, which reduces visibility and limits the ability to reach conclusions quickly and intuitively. Future iterations of this project could utilise a custom interaction window that would allow a more enhanced control over what's currently being displayed like an addition of an option to sort the classes by their activations or limit the view to a particular number of classes. In the current view, the classes have to be manually zoomed into to reveal their id. While the current layout still has its advantages when it comes to spotting whether particular classes are being more commonly highly activated than others, this can also be done programatically instead.

The study's heavy focus on the development of the software has limited the ability to perform an extensive statistical analysis. While the visual insights do enhance the human-interpretable aspects, a validation of those findings in a form of statistical backing could make it possible to reach strong, cross-model conclusions with much greater accuracy than a human examining the processes step by step.

6.3 Process evaluation and exploration of future research directions

A potential research direction could include an examination of how known attacks influence the model and to what extent. Thanks to the enhanced visualisations, more robust insights can be achieved.

Another potential research question would be in what way and to what extent the weighted activations contribute to a softmax confidence score. As noted during a debugging phase of the project, the highest weighted similarity score when considered in the context of the entire prediction is accounting for at most around 2-3 percent of all weighted similarity scores. Exploring this potential correlation between the confidence score and the weighted activation distribution could expose that for example models with high activation patterns strongly correlate with model's certainty allowing us to reach further conclusions and set up experiments on how this correlation could be manipulated. Through that it could then be investigated if a model's certainty increase is linked to any human-interpretable features, further increasing the intuitive understanding of the models as well as trustworthiness. While a preliminary measure of confidence has been introduced, it has been very limited in scope and served as an invitation into exploring this direction further, with more robust techniques, more models and more acknowledged confidence measures, as well as their interplay and dependencies.

In terms of the future directions for the software, many potential enhancements could be introduced. In the future endeavors, we will be working on the capabilities of the software to provide more generalised conclusions, for example by allowing the user to aggregate and compare multiple results in one graph. The user should also be able to investigate any class activations and visualisations for any class within a single prediction, not only the correct and predicted class. Very often it is the case that during examining a wrong prediction many of the top activations are close in terms of activations or even higher than those of a correct class. It would be useful to allow researchers to manually select any desired classes, particular to investigate any anomalies - any especially high or especially low activated classes.

Currently, the visualisations of similarity scores and weights are done without a particular insight into which prototypes exactly are how prominent and how often - while a simple solution like adding a label with an id of each of the prototypes could strengthen the comparative studies into particular prototype involvement, this would not be useful if the user would have to manually search the prototype. This example shows a need for a further development into the interactive elements of the software. Ideally, a deep dive system could be implemented to address these and more considerations when it comes to linking the visual, picture based part of the software and the graph-based part of the software. Efforts could be made into making the graphs interactible, where clicking on a certain part of the graph could bring up a more detailed view. For example, clicking a bar of a total weighed activations per class would bring up top n prototypes displayed visually for that particular class, as it is currently done for the correct and predicted classes.

Another future direction could be generating charts and detailed statistics to show how often certain classes or prototypes are misclassified and whether there are particular prototypes of a particular class that contribute to the errors. In the future, users would be able to make targeted adjustments to improve model performance by detailed error analysis.

7 Conclusion

The study was set out to explore and enhance the interpretability of prototype-based neural networks and to do so on an example of ProtoPNet. We have identifies a need for a development of a more advanced visualisation tool than the ones used currently. By providing a more nuanced approach to prototype activation and similarity score visualisation. The aim was to address the limitations found in existing methods that overly rely on single-value assessments of model performance. The study effectively shows benefits and pressing needs to advance the explainability as we know it today. We have seen that accuracy might not be an ideal predictor of robustness of the model and we have explored that through the involvement of increased number of considered prototypes as well as a more nuanced selection systems. These insights would not have been possible with the tools available today.

The results once again reiterate and confirm the suspicion that raw similarity scores are not always the most influential in the model's decision-making process. The multiple charts available for every prediction showcased how detailed insights are essential for understanding model behaviour, which is all the more important in the cases of misclassification to identify the problem, identify prototypes and its features that might have been overlooked, but are still significantly contributing to the misclassification logic. We have also experienced the importance of enhanced precautions, where even accurate predictions might stem from an undesirable place.

Additionally, the study has shown that visualising prototype activations across all classes rather than solely focusing on the predicted class reveals further insights into what role uncertainty and confidence correlates with model accuracy. This observation opens new areas for possible exploration of trying to understand and mitigate model uncertainty.

Many possibilities have opened up from exploring only a small aspect of such a vast discipline. While AI is clearly still a few steps away from achieving full transparency, we have shown that progress is not only possible but also achievable.

References

[1] David Alvarez-Melis and Tommi S. Jaakkola. *Towards Robust Interpretability with Self-Explaining Neural Networks*. 2018. arXiv: 1806.07538 [cs.LG]. URL: https://arxiv.org/abs/1806.07538.

[2] Sebastian Bordt et al. 'Post-Hoc Explanations Fail to Achieve their Purpose in Adversarial Contexts'. In: *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*. FAccT '22. Seoul, Republic of Korea: Association for Computing Machinery, 2022, pp. 891–905. ISBN: 9781450393522. DOI: 10.1145/3531146.3533153. URL: https://doi.org/10.1145/3531146.3533153.

[3] Chaofan Chen et al. *This Looks Like That: Deep Learning for Interpretable Image Recognition*. 2019. arXiv: 1806.10574 [cs.LG]. URL: https://arxiv.org/abs/1806.10574.

[4] Jeffrey Dean et al. 'Large Scale Distributed Deep Networks'. In: *Advances in neural information processing systems* (Oct. 2012).

[5] Finale Doshi-Velez and Been Kim. *Towards A Rigorous Science of Interpretable Machine Learning*. 2017. arXiv: 1702.08608 [stat.ML]. URL: https://arxiv.org/abs/1702.08608.

[6] Max Ehrlich et al. 'Analyzing and mitigating jpeg compression defects in deep learning'. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 2357–2367.

[7] I. Goodfellow, Y. Bengio and A. Courville. *Deep Learning*. Adaptive Computation and Machine Learning series. MIT Press, 2016. ISBN: 9780262035613. URL: https://books.google.co.uk/books?id=Np9SDQAAQBAJ.

[8] Adrian Hoffmann et al. *This Looks Like That... Does it? Shortcomings of Latent Space Prototype Interpretability in Deep Networks*. 2021.

[9] Alex Krizhevsky, Ilya Sutskever and Geoffrey E Hinton. 'ImageNet Classification with Deep Convolutional Neural Networks'. In: *Advances in Neural Information Processing Systems*. Ed. by F. Pereira et al. Vol. 25. Curran Associates, Inc., 2012. URL: https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

[10] Yann LeCun, Yoshua Bengio and Geoffrey Hinton. 'Deep learning'. In: *Nature* 521.7553 (May 2015), pp. 436–444. ISSN: 1476-4687. DOI: 10.1038/nature14539. URL: http://dx.doi.org/10.1038/nature14539.

[11] Scott Lundberg and Su-In Lee. *A Unified Approach to Interpreting Model Predictions*. 2017. arXiv: 1705.07874 [cs.AI]. URL: https://arxiv.org/abs/1705.07874.

[12] T.M. Mitchell. *Machine Learning*. McGraw-Hill International Editions. McGraw-Hill, 1997. ISBN: 9780071154673. URL: https://books.google.co.uk/books?id=EoYBngEACAAJ.

[13] Meike Nauta, Ron van Bree and Christin Seifert. *Neural Prototype Trees for Interpretable Fine-grained Image Recognition*. 2021. arXiv: 2012.02046 [cs.CV]. URL: https://arxiv.org/abs/2012.02046.

[14] Meike Nauta et al. 'PIP-Net: Patch-Based Intuitive Prototypes for Interpretable Image Classification'. In: *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2023, pp. 2744–2753. DOI: 10.1109/CVPR52729.2023.00269.

[15] Meike Nauta et al. 'This Looks Like That, Because ... Explaining Prototypes for Interpretable Image Recognition'. In: *CoRR* abs/2011.02863 (2020). arXiv: 2011.02863. URL: https://arxiv.org/abs/2011.02863.

[16] Marco Tulio Ribeiro, Sameer Singh and Carlos Guestrin. *"Why Should I Trust You?": Explaining the Predictions of Any Classifier*. 2016. arXiv: 1602.04938 [cs.LG]. URL: https://arxiv.org/abs/1602.04938.

[17] Cynthia Rudin. *Stop Explaining Black Box Machine Learning Models for High Stakes Decisions and Use Interpretable Models Instead*. 2019. arXiv: 1811.10154 [stat.ML]. URL: https://arxiv.org/abs/1811.10154.

[18] Dawid Rymarczyk et al. 'ProtoPShare: Prototypical Parts Sharing for Similarity Discovery in Interpretable Image Classification'. In: *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery amp; Data Mining*. KDD '21. ACM, Aug. 2021. DOI: 10.1145/3447548.3467245. URL: http://dx.doi.org/10.1145/3447548.3467245.

[19] Samuel Felipe dos Santos, Nicu Sebe and Jurandy Almeida. 'The good, the bad, and the ugly: Neural networks straight from jpeg'. In: *2020 IEEE International Conference on Image Processing (ICIP)*. IEEE. 2020, pp. 1896–1900.

[20] Ramprasaath R. Selvaraju et al. 'Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization'. In: *International Journal of Computer Vision* 128.2 (Oct. 2019), pp. 336–359. ISSN: 1573-1405. DOI: 10.1007/s11263-019-01228-7. URL: http://dx.doi.org/10.1007/s11263-019-01228-7.

[21] Vinod Sharma. 'A Study on Data Scaling Methods for Machine Learning'. In: *International Journal for Global Academic Scientific Research* 1 (Feb. 2022). DOI: 10.55938/ijgasr.v1i1.4.

[22] Xue Ying. 'An Overview of Overfitting and its Solutions'. In: *Journal of Physics: Conference Series* 1168 (Feb. 2019), p. 022022. DOI: 10.1088/1742-6596/1168/2/022022.

[23] Matthew D Zeiler and Rob Fergus. *Visualizing and Understanding Convolutional Networks*. 2013. arXiv: 1311.2901 [cs.CV]. URL: https://arxiv.org/abs/1311.2901.