Concept Detection in Deep Neural Networks

Prabal Ghosh[1][0009−0004−3449−5811]

Universite Cote d'Azur, Sophia Antipolis, France
prabal5ghosh@gmail.com

1 Internship at INRIA Sophia Antipolis

I worked with The Maasai team during my internship at INRIA Sophia Antipolis. Maasai is a research project team at Inria Sophia-Antipolis, focused on developing models and algorithms for Artificial Intelligence. This team is a collaborative effort involving the LJAD (Mathematics, UMR 7351) and I3S (Computer Science, UMR 7271) laboratories of Université Côte d'Azur.

This internship was conducted under the supervision of Professor Frédéric Precioso, Dr. Diane Lingrand, and Dr. Rémy Sun. My research focused on Explainable Artificial Intelligence (XAI) with the aim of interpreting Convolutional Neural Networks (CNNs).

2 Introduction

In recent years, there have been significant advancements in deep learning, with a strong emphasis on improving the robustness and interpretability of deep neural networks (DNNs). These efforts are particularly important for facilitating interactions between domain experts and DNNs. Researchers have explored various approaches, including the use of abstract concepts, to achieve these objectives. The field of explainable artificial intelligence (XAI) seeks to add transparency to the currently powerful but often opaque deep learning models. Local XAI methods offer explanations for individual predictions using attribution maps, which highlight "where" significant features are located within the data. However, these methods do not clarify "what" these features represent. On the other hand, global explanation techniques illustrate the broader concepts that a model has learned to encode. Each of these methods delivers only a partial understanding, thereby leaving the task of interpreting the model's reasoning largely to the user.

In [5] authors have introduced LRP. Layer-wise relevance propagation (LRP) is a technique in deep learning used to understand the significance of individual neurons or features in a neural network's decision-making process. It facilitates the explanation of predictions in complex models like deep neural networks. LRP involves a forward pass where input data passes through layers to make predictions, followed by a backward pass to determine the relevance of each neuron

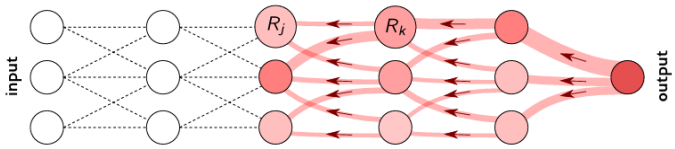


Fig. 1: Sentiment Polarity Categorization Process.[3]

based on the model's output. Relevance scores are then propagated backward through the network, guided by specific rules, assigning relevance to neurons in each layer based on their contribution to the next layer's activation. LRP's rule-based approach ensures meaningful relevance distribution. Its interpretability benefits allow insights into influential features or parts of input data in predictions.

Concept Relevance Propagation (CRP) is an advanced technique within the field of explainable artificial intelligence (XAI). CRP combines local XAI methods, which focus on identifying "where" important features appear in individual predictions through attribution maps, with global XAI methods that explain "what" concepts the model has learned overall. By merging these perspectives, CRP offers a dual view that highlights both specific and general features within the data, providing a richer understanding of the model's decision-making process.

One of the key aspects of CRP is its ability to identify relevant concepts from the internal activations of the network. These concepts are then used to interpret the model's decisions, offering insights that are more accessible and meaningful to humans. This enhanced interpretability helps in understanding the model's reasoning, making it easier to trust and validate its outputs. CRP can be applied

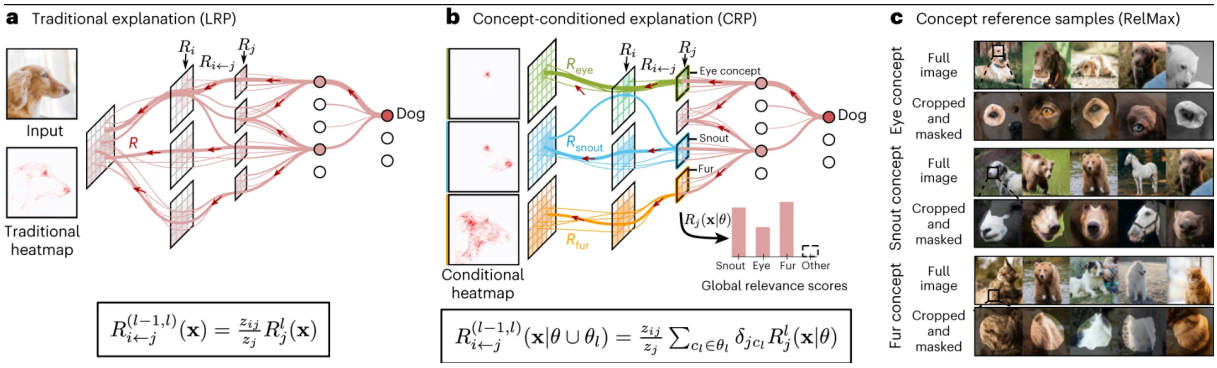


Fig. 2: Sentiment Polarity Categorization Process.[1]

to various types of DNNs and is particularly effective in convolutional neural net-

works (CNNs) used in computer vision. It enables the creation of concept atlases, which are visual representations of the various concepts learned by the model, and supports concept-composition analyses that detail how different concepts are combined within the network. Additionally, CRP facilitates quantitative investigations into concept subspaces, aiding in fine-grained decision-making.

The implementation of CRP involves leveraging identified concepts within the model, potentially through an auxiliary loss or concept embedding model, which can lead to improvements in decision accuracy. While the technique is versatile and can be adapted to different datasets, model architectures, and application domains, it does come with challenges. The complexity of identifying and utilizing concepts from internal activations, as well as the increased computational requirements, can pose hurdles.

Concept Relevance Propagation (CRP) Mathematical Approach

Layer-wise Relevance Propagation (LRP)

Forward Pass: The activations a_i and pre-activations z_j are computed for a neural network layer:

$$z_{ij} = a_i w_{ij} \tag{1}$$

$$z_j = \sum_i z_{ij} \tag{2}$$

$$a_j = \sigma(z_j) \tag{3}$$

where w_{ij} are the weights and σ is the activation function.

Relevance Propagation: Relevance R_j is distributed back to the inputs proportionally:

$$R_i \leftarrow_j = \frac{z_{ij}}{z_j} R_j \tag{4}$$

Relevance Aggregation: The relevance for each neuron is aggregated:

$$R_i = \sum_j R_i \leftarrow_j \tag{5}$$

CRP Extension to LRP

Conditional Relevance Propagation: Introduces conditions θ to identify neural network elements (e.g., neurons or channels) associated with specific concepts.

Conditional Relevance Distribution: Relevance is distributed conditionally based on the specified concepts:

$$R_{i \leftarrow j}^{(l-1,l)}(x|\theta \cup \theta_l) = \frac{z_{ij}}{z_j} \sum_{c_l \in \theta_l} \delta_{jc_l} R_j^l(x|\theta) \tag{6}$$

where δ_{jc_l} is the Kronecker delta that filters neurons based on the conditions θ_l.

Binary Masking: The conditions θ are applied through binary masking of relevance tensors, making the process efficient.

Efficient Implementation: Implemented using frameworks like Zennit-CRP.

Spatial and Channel Axes in CNNs Conditional Relevance for CNNs: In convolutional neural networks, the conditions can be applied to specific channels representing latent concepts:

$$R^{(l-1,l)}_{i\leftarrow(p,q,j)}(x|\theta \cup \theta_l) = \frac{z_{i(p,q,j)}}{z_{(p,q,j)}} \sum_{c_l \in \theta_l} \delta_{jc_l} R^l_{(p,q,j)}(x|\theta) \tag{7}$$

where (p, q, j) addresses a voxel in the activation tensor.

Global and Local Relevance Aggregation Global Relevance: Summation over input units to measure overall concept relevance:

$$R^l(x|\theta) = \sum_i R^l_i(x|\theta) \tag{8}$$

Localized Analysis: Restricting relevance aggregation to regions of interest:

$$R^l_{\mathcal{I}}(x|\theta) = \sum_{i\in\mathcal{I}} R^l_i(x|\theta) \tag{9}$$

Dependencies Between Concepts Concept Dependencies: Identifying dependencies between concepts across layers:

$$R^{(l-1,l)}_{(u,v,i)\leftarrow(p,q,j)}(x|\theta) = \frac{z_{(u,v,i)(p,q,j)}}{z_{(p,q,j)}} R^l_{(p,q,j)}(x|\theta) \tag{10}$$

$$R^{(l-1,l)}_{i\leftarrow j}(x|\theta) = \sum_{u,v}\sum_{p,q} R^{(l-1,l)}_{(u,v,i)\leftarrow(p,q,j)}(x|\theta) \tag{11}$$

Concept Relevance Propagation (CRP) is a powerful tool that enhances the transparency and interpretability of deep neural networks. By addressing both the "where" and "what" aspects of model explanations, CRP offers a comprehensive understanding of the model's decision-making process, making it a valuable asset in improving both the accuracy and trustworthiness of AI models.

RELMAX, or Relevance Maximization, is an advanced technique within explainable artificial intelligence (XAI) aimed at providing insights into the inner workings of deep neural networks (DNNs). It focuses on identifying the most relevant examples within a training dataset based on their contribution to the model's decisions. Unlike Activation Maximization (ActMax), which finds examples that produce high activations for specific neurons or layers, RELMAX seeks to determine examples that are highly relevant to the model's decision-making process. This relevance is traced by backpropagating relevance scores through

the network, ensuring that the relevance is conserved at each layer, which clarifies the influence path from the output to the input. By identifying these key examples, RELMAX enhances the interpretability and debugging of the model, making it easier to understand which parts of the data the model relies on most heavily. This is particularly useful for uncovering potential biases or areas where the model might be overfitting.

In addition to improving transparency and trust in the model's decisions, RELMAX is useful for feature selection and engineering by highlighting the most relevant features. This technique provides deeper insights into the model's decision process, aids in debugging by pinpointing critical examples and features, and helps detect biases in the training data and model. Despite its benefits, implementing RELMAX can be complex and computationally intensive, particularly for large and deep networks. Nonetheless, it remains a powerful tool for enhancing the interpretability and performance of deep neural networks, making it a valuable asset in various applications. This Concept Relevance Propagation technique is mainly used as a supervised approach in the paper.

The paper **"Extraction of an Explanatory Graph to Interpret a CNN"** introduces an innovative approach to enhance the interpretability of convolutional neural networks (CNNs) by constructing an explanatory graph. This method employs Expectation-Maximization (EM) algorithms and Gaussian mixture models to identify relevant pixels from the nodes of the graph, thereby elucidating the CNN's decision-making process.[5]

An explanatory graph is a structured representation that captures the interactions and dependencies between different features identified by a CNN. In essence, it serves as a map of the network's internal workings, where nodes represent specific parts or patterns detected by filters in various convolutional layers, and edges denote the spatial and functional relationships between these parts. The goal of constructing an explanatory graph is to provide a clearer understanding of how different layers of the network contribute to the final output, thereby enhancing the transparency and interpretability of CNNs.

Top-Down Iterative Learning of the Explanatory Graph

The process of extracting an explanatory graph is conducted in a top-down iterative manner, which is crucial for capturing the hierarchical nature of CNNs. This method involves several key steps:

Initialization: The process begins with the top convolutional layer of the CNN. Parts are disentangled from this layer to construct the top layer of the explanatory graph. Nodes in this layer represent high-level features detected by the top-layer filters. Inputs: feature map X of the Lth conv layer, inference results R in the upper conv-layer.

Position Inference: For each node in the top layer, the method performs position inference to determine whether the part indicated by the node appears in the feature maps of the input images, and if so, its location. This inference is based on matching the feature maps to the explanatory graph.

Recursive Construction: Using the position inference results from the top layer, the method recursively constructs the lower layers of the explanatory

graph. Each node in a lower layer is connected to nodes in the neighboring upper layer, maintaining consistent spatial relationships. This ensures that the explanatory graph captures the hierarchical dependencies between features across different layers.

Parameter Optimization: For each node in the Lth layer, the method learns two key parameters: the prior location of the node and the set of parent nodes in the upper layer. The prior location encodes the expected position of the node, while the parent nodes represent the spatial relationships with nodes in the upper layer. These parameters are iteratively refined using an Expectation-Maximization (EM) algorithm to better fit the feature maps.

Mathematical Formulation: The learning process can be formalized through an objective function that maximizes the likelihood of the feature maps given the explanatory graph. The objective function for the Lth layer is formulated as:

$$\arg\max_{\theta} \prod_{I \in \mathcal{I}} P(X_I \mid R_I, \theta)$$

where θ represents the parameters of the nodes in the Lth layer, X_I denotes the feature map of image I, and R_I denotes the position inference results of the nodes in the upper layer.

The feature map X can be regarded as a distribution of neural activation entities, where each activation unit x is identified by its spatial position p_x and channel number d_x. The compatibility of a node V with an activation unit x is measured by the spatial relationship between V and its connected nodes in the graph. This relationship is modeled using a Gaussian distribution:

$$P(p_x \mid V, R, \theta) = \mathcal{N}(p_x \mid \mu_V, \sigma_V^2),$$

where μ_V is the prior position of V, and σ_V^2 represents the variance. The overall compatibility is the product of the spatial compatibilities of the node with each of its connected nodes in the upper layer.

EM Algorithm for Parameter Optimization

The EM algorithm is employed to iteratively refine the parameters of the explanatory graph. During each iteration, the algorithm performs the following steps:

Expectation Step

Compute the posterior probability of each node given the feature maps and current graph parameters. This involves calculating the likelihood of the node positions and updating the assignment of activation units to nodes.

Maximization Step

Update the parameters μ_V and E_V (the set of parent nodes) to maximize the expected log-likelihood of the feature maps given the current assignments. This step involves adjusting the prior positions and spatial relationships to better fit the data.

The iterative process continues until convergence, resulting in an optimal explanatory graph that accurately represents the spatial and hierarchical relationships within the CNN.

Part Localization

Part localization is a critical aspect of the explanatory graph, where nodes are assigned to activation peaks in the feature maps to infer the locations of object parts. Each node V is assigned to the unit $\hat{x}$ in the feature map that maximizes the score:

$$S_{V\to\hat{x}} = \arg\max_{x\in X, d_x=d} S_{V\to x},$$

where $S_{V\to x}$ denotes the score of assigning node V to activation unit x. This score combines the activation strength and spatial compatibility, ensuring that nodes are assigned to positions that best represent the underlying parts[4].

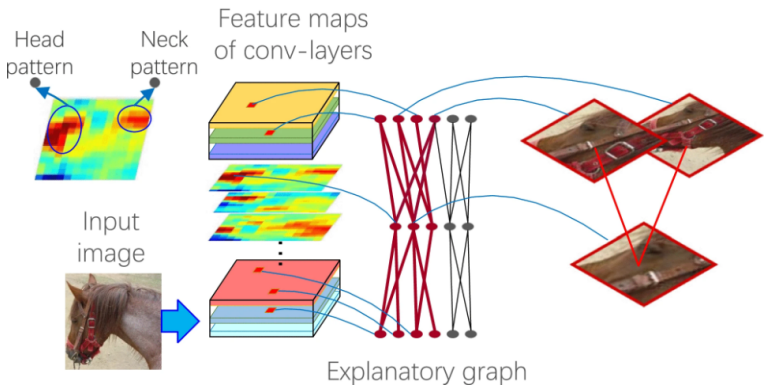


Fig. 3: An explanatory graph represents the compositional hierarchy of object parts encoded in conv-layers of a CNN..[5]

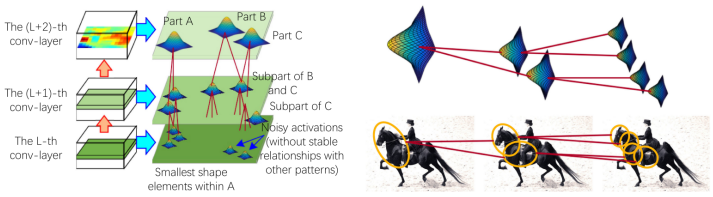


Fig. 4: Sentiment Polarity Categorization Process.[5]

The main advantage of this approach is its unsupervised nature, which does not require part annotations, making it more scalable and generalizable. The explanatory graph highlights the relationships and interactions within the network, providing a clearer understanding of the CNN's internal workings. This

enhanced transparency aids in debugging, improving the model, and identifying potential biases by pinpointing areas of high relevance.

Overall, the paper presents a powerful tool for interpreting CNNs by using EM algorithms and Gaussian mixture models to construct an explanatory graph. This graph not only provides insights into the model's decision-making process but also facilitates a better understanding of feature importance and part localization, contributing to more trustworthy and interpretable AI systems.

3 What I had to do

The aim is to replace the Expectation-Maximization (EM) algorithms and Gaussian mixture models used in the explanatory graph construction with the Concept Relevance Propagation (CRP) approach. This integration seeks to leverage CRP's ability to identify and propagate relevant concepts within a CNN in an unsupervised manner.

4 What I did, how I did it and why I did it the way

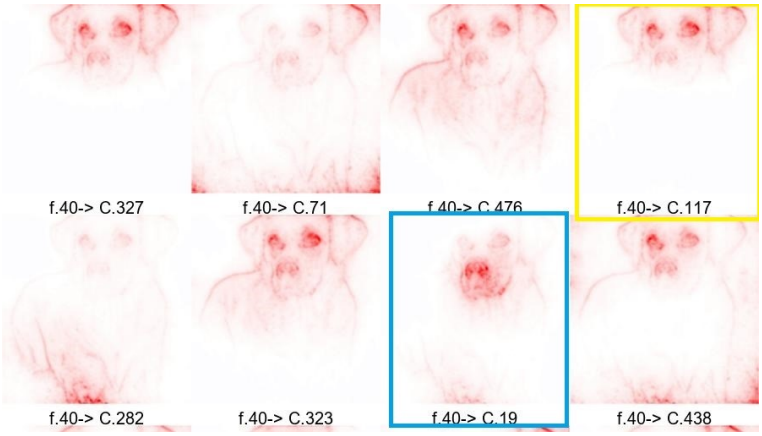
(a) Labrador-1 (b) Labrador-12

Fig. 5: Two images of Labrador dogs

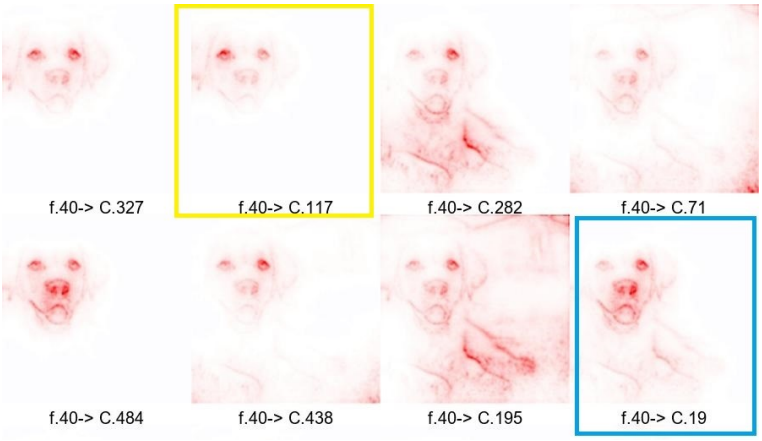
In this study, a pre-trained VGG16 model is used. Relevance scores are calculated using the CRP (Concept Relevance Propagation) approach, starting from the top convolutional layer and propagating down to the initial convolutional layer [2]. The zennit-crp library, which can be found at Zennit Documentation (https://zennit.readthedocs.io/en/latest/index.html), is utilized in this analysis

to perform efficient and flexible composite rule propagation for neural network interpretability.

Two different images of Labrador dogs[Fig. 5a,Fig. 5b] (classified under class number 208 in ImageNet) were used in this study.



(a) conditional heatmap for top conv-layer(feature.40)(Labrador-1)



(b) conditional heatmap for top conv-layer(feature.40)((Labrador-2)

Fig. 6: conditional heatmap of top 8 relevant channels for top conv-layer(feature.40)

To understand different concepts such as eyes and nose, conditional heatmaps for the top 8 channels of the top convolutional layer (Feature-40) were first plotted. Channels that indicated parts of the eyes and nose for both dogs in Figures 6a and 6b were selected. Specifically, channels 117 and 19 were chosen

for eye and nose concept detection. Subsequently, the focus was on detecting the exact channels in the inner convolutional layers that indicate these concepts.

(a) Conditional heatmap for top conv-layer (Feature-37) (Labrador-1)

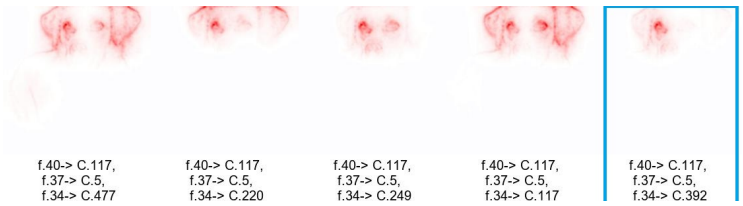
(b) Conditional heatmap for top conv-layer (Feature-37) (Labrador-2)

Fig. 7: conditional heatmaps for Labrador dogs(Feature-37)

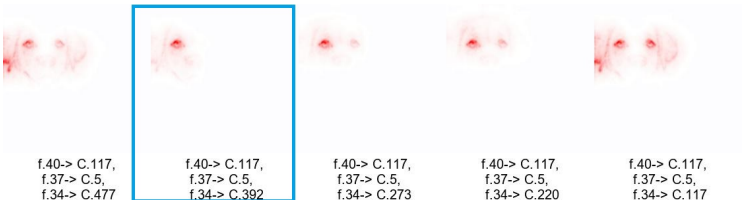
For each of the top 8 channels of feature 40, conditional relevance flows were analyzed for the previous convolutional layer (feature 37). Subsequently, conditional heatmaps for the top 5 channels in the convolutional layer (feature 37) were plotted to visualize different concepts based on the relevance flows from the top 8 channels of feature 40.

To analyze the eye concept detection, channels were selected from Figures 7a and 7b. Specifically, channel 117 from the top convolutional layer (feature 40) and channel 5 from the previous convolutional layer (feature 37) were chosen for both Labrador dog images. Relevance scores were calculated for these

channels, and based on these scores, the top 5 relevant channels were identified. Subsequently, conditional heatmaps were plotted to visualize the indication of the eye concept. This process facilitated the understanding of how different channels contribute to the detection of specific features, such as the eyes, within the convolutional layers of the CNN.



(a) Conditional heatmap for top conv-layer (Feature-34) (Labrador-1)



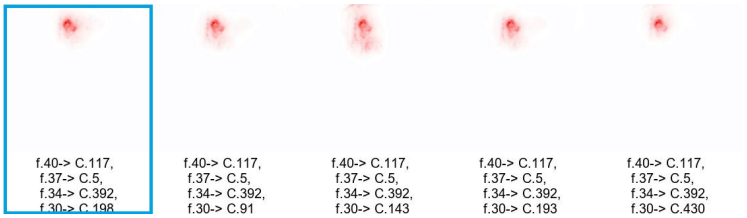
(b) Conditional heatmap for top conv-layer (Feature-34)(Labrador-2)

Fig. 8: conditional heatmaps for Labrador dogs(Feature-34). Relevance flow from channel 117 of feature40 and channel 5 of feature 37

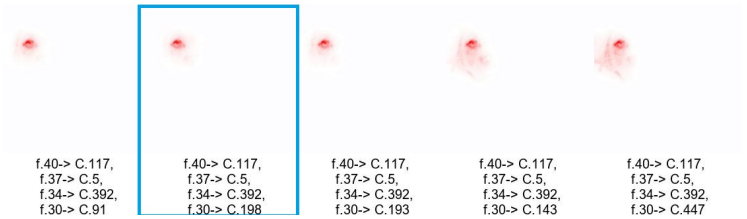
In Figures 8a and 8b, I observed that channel number 392 in feature 34 predominantly highlights the left eye region for both Labradors. Consequently, I proceeded to calculate the relevance score for the preceding convolutional layer (feature 30). My findings revealed that the top five most relevant channels consistently indicate the left eye region, as demonstrated in Figures 9a and 9b

Similarly, by analyzing the relevance flow from channel 117 of feature 40, channel 5 of feature 37, channel 392 of feature 34, and channel 198 of feature 30, I identified the top five relevant channels in the convolutional layer (feature 27). These channels consistently highlight the exact left eye region in Fig 10a and Fig 10b.

To analyze the Nose concept detection, channels were selected from Figures 7a and 7b. Specifically, channel 19 from the top convolutional layer (feature 40) and channel 64 from the previous convolutional layer (feature 37) were chosen for both Labrador dog images. Relevance scores were calculated for these channels, and based on these scores, the top 5 relevant channels were identified. Subsequently, conditional heatmaps were plotted to visualize the indication of the nose concept. This process facilitated the understanding of how different

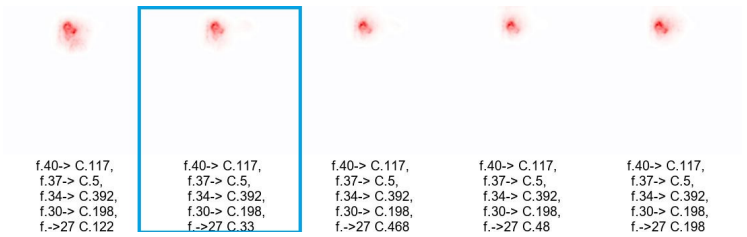


(a) Conditional heatmap for top conv-layer (Feature-30) (Labrador-1)

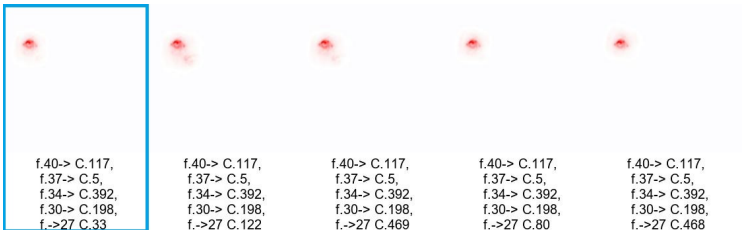


(b) Conditional heatmap for top conv-layer (Feature-30)(Labrador-2)

Fig. 9: conditional heatmaps for Labrador dogs(Feature-30). Relevance flow from channel 117 of feature40 and channel 5 of feature 37 and channel 392 of feature 34



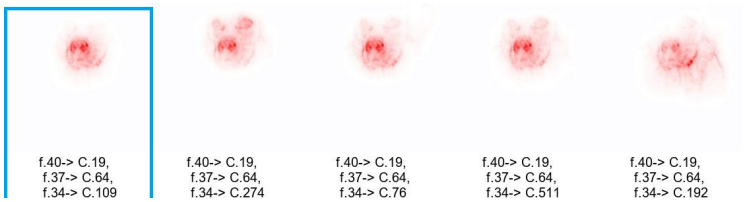
(a) Conditional heatmap for top conv-layer (Feature-27)(Labrador-1)



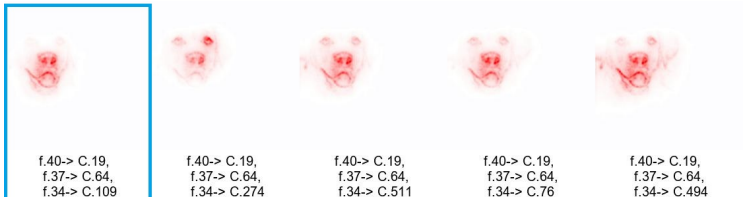
(b) Conditional heatmap for top conv-layer (Feature-27)(Labrador-2)

Fig. 10: conditional heatmaps for Labrador dogs(Feature-27). Relevance flow from channel 117 of feature40 and channel 5 of feature 37 and channel 392 of feature 34 and channel 196 of feature 30

channels contribute to the detection of specific features, such as the noses, within the convolutional layers of the CNN.



(a) Conditional heatmap for top conv-layer (Feature-34) (Labrador-1)



(b) Conditional heatmap for top conv-layer (Feature-34)(Labrador-2)

Fig. 11: conditional heatmaps for Labrador dogs(Feature-34). Relevance flow from channel 19 of feature40 and channel 64 of feature 37

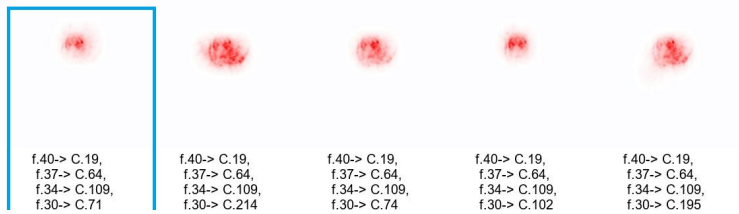
In Figures 11a and 11b, I observed that channel number 109 in feature 34 predominantly highlights the nose and mouth region for both Labradors. Consequently, I proceeded to calculate the relevance score for the preceding convolutional layer (feature 30). My findings revealed that the top five most relevant channels consistently indicate the nose region, as demonstrated in Figures 12a and 12b

Similarly, by analyzing the relevance flow from channel 19 of feature 40, channel 64 of feature 37, channel 109 of feature 34, and channel 71 of feature 30, I identified the top five relevant channels in the convolutional layer (feature 27). These channels consistently highlight the exact nose region in Fig 13a and Fig 13b.

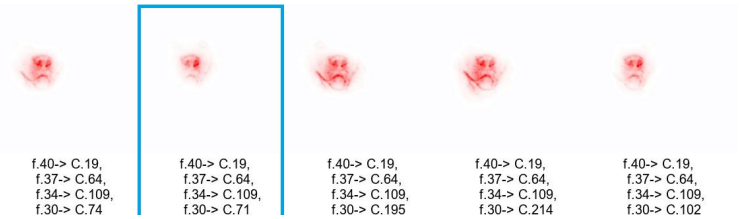
By using Concept Relevance Propagation (CRP), we can recognize different concepts in various channels across different layers.

5 Conclusion

In the explanatory graph, each convolutional layer is represented as a layer of the graph. Different object parts are constructed by using Gaussian Mixture Models (GMM) on the activation peaks, evaluating the Gaussian distributions, and clustering them. In this approach, the locations of nodes and the connections

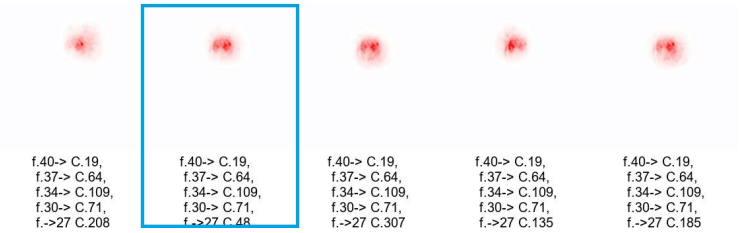


(a) Conditional heatmap for top conv-layer (Feature-30) (Labrador-1)

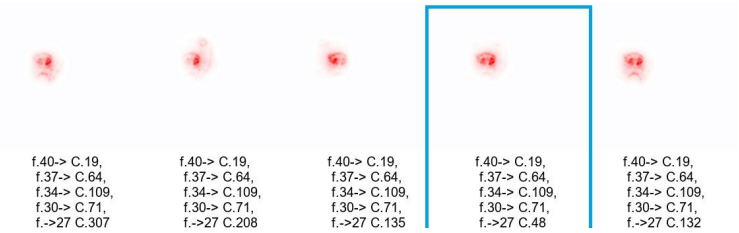


(b) Conditional heatmap for top conv-layer (Feature-30)(Labrador-2)

Fig. 12: conditional heatmaps for Labrador dogs(Feature-30). Relevance flow from channel 19 of feature40 and channel 64 of feature 37 and channel 109 of feature 34



(a) Conditional heatmap for top conv-layer (Feature-27) (Labrador-1)



(b) Conditional heatmap for top conv-layer (Feature-27)(Labrador-2)

Fig. 13: conditional heatmaps for Labrador dogs(Feature-30). Relevance flow from channel 19 of feature40 and channel 64 of feature 37 and channel 109 of feature 34 and channel 71 of feature 30

between the nodes of a layer with its parent layer are updated using Expectation-Maximization (EM) algorithms iteratively, which is a very tedious and time-consuming process. This algorithm requires numerous iterations and consumes a significant amount of time. In the explanatory graph, top layer nodes indicate large object parts, while the latent or lower layer nodes indicate smaller object parts, such as eyes, nose, and ears.

In Concept Relevance Propagation (CRP), the top layers also indicate large combined concepts, and the inner lower convolutional layer channels indicate smaller concepts (like eyes, nose, etc.). The channels of a layer serve as nodes, and they are connected by edges or connections with the nodes of the parent convolutional layer. I have verified that in the inner layers, a channel detects the same concept for different images. For example, in different Labrador dogs, several channels in layer 27 detect the same object, such as an eye or nose.

In the next step, I will use a threshold value for the relevance of the layer channels so that I can specify these groups of channels as a node representing an object part, and the connection is established. With this modification, I can create an explanatory graph for Convolutional Neural Networks (CNN) in an unsupervised manner. Currently, I am working on determining the appropriate threshold value and the grouping of channels within a layer.

References

1. Reduan Achtibat, Maximilian Dreyer, Ilona Eisenbraun, Sebastian Bosse, Thomas Wiegand, Wojciech Samek, and Sebastian Lapuschkin. From attribution maps to human-understandable explanations through concept relevance propagation. *Nature Machine Intelligence*, 5(9):1006–1019, 2023.
2. Christopher J. Anders, David Neumann, Wojciech Samek, Klaus-Robert Müller, and Sebastian Lapuschkin. Software for dataset-wide xai: From local explanations to global insights with Zennit, CoRelAy, and ViRelAy. *CoRR*, abs/2106.13200, 2021.
3. Grégoire Montavon, Alexander Binder, Sebastian Lapuschkin, Wojciech Samek, and Klaus-Robert Müller. *Layer-Wise Relevance Propagation: An Overview*, pages 193–209. Springer International Publishing, Cham, 2019.
4. Quanshi Zhang, Ruiming Cao, Feng Shi, Ying Nian Wu, and Song-Chun Zhu. Interpreting cnn knowledge via an explanatory graph. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.
5. Quanshi Zhang, Xin Wang, Ruiming Cao, Ying Nian Wu, Feng Shi, and Song-Chun Zhu. Extraction of an explanatory graph to interpret a cnn. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(11):3863–3877, 2021.