華中科技大學

计算机视觉
结课报告

专业: 计算机电脑

班级: CS2002

学号: I201920029

姓名: 冯就康

电话: 15623031879

邮箱: sokhorng526@gmail.com

Contents

1. Introduction 3
2. Review of Existing Literature 4
   2.1 Layer-wise relevance propagation 4
   2.2 Shapley Additive Explanations 6
   2.3 Occlusion Tests 7
   2.4 Inversion 8
   2.5 Logic Rules 11
   2.6 Deep Taylor Decomposition 13
   2.7 Deconvolutional Neural Network 14
3. Conclusion 17
4. References 17

An Analysis of Explainability Methods for Convolutional Neural Networks

Abstract

Convolutional Neural Networks are a type of deep learning model that have achieved remarkable performance in various computer vision tasks, such as object recognition, face detection, and semantic segmentation. However, CNN are often regarded as black-box models, meaning that their internal workings are not transparent or understandable to humans. This poses a challenge for trust, validation, and debugging of CNNs, especially in safety-critical or high-stakes applications. Therefore, it is essential to develop methods and techniques to make CNNs more interpretable and explainable to reveal the rationale and logic behind CNN model decisions. This literature review is a survey and analysis of the available explainability methods for showing the reasoning behind CNN's decisions

1. Introduction

Deep learning is advanced in its efforts to make accurate decisions. The Convolutional Neural Network (CNN) is known for its ability to encapsulate array features and is particularly effective in image recognition across various fields. However, like many deep learning models, CNN does not explain its decisions, making it a black-box model. This lack of transparency means that experts are left in the dark about the reasoning behind their decisions, leading to a decrease in trust due to unexplainable classifications. Despite its accuracy for a single class, there is no guarantee that the algorithm’s decisions will always be as anticipated. If the network learns incorrect features of an image, it may not always classify it correctly.

While this might not be a significant issue in some fields, the lack of trust is a problem in high-risk applications such as medical diagnoses, mortgage approvals, and autonomous driving. If CNNs are making these decisions, users need to be confident that they will make the right assessment so they can receive the appropriate treatment, secure a new home, or ensure safety on the road. In line with this, the European Union has enacted a regulation requiring automated decisions to include an explanation of the decision-making process.

While there is research on the black box nature of many models, this paper specifically focuses on CNNs. It is a survey and analysis of existing explainability methods aimed at enhancing the trustworthiness of CNN decisions. The report will cover **Layer-wise Relevance Propagation**, **Shapley Additive Explanations** (SHAP), **Occlusion Tests**, **Inversion**, **Logic Rules**, **Deep Taylor Decompositions,** and **Deconvolutional Neural Networks** as ways to elucidate the classification decision. This report provides the pros and cons of each method, offering a better understanding of what explainability method is best suited for.

2. Review of Existing Literature

2.1 Layer-wise relevance propagation

Methodology

Layer-wise relevance propagation (LWRP) is a technique that shows how much each pixel influences the outcome of a convolutional neural network (CNN). It calculates the probability of each pixel's influence by breaking down the image into pixels. Pixels that show the background are not very influential, but pixels that show specific parts of the object are influential. Unlike many other CNN explain ability techniques, LWRP not only makes a heat map of pixel influence but also shows which pixels support or oppose the outcome, which is called positive and negative evidence, respectively. This makes CNN's outcome more reliable and the explanation more accurate, because it considers both the useful and the useless pixels.

LWRP is based on the idea that a neural network makes its outcome through multiple layers, not just one layer. It adds up the influence of each output pixel to make a relevance map for that layer, then goes backward to the input image to find out how much each pixel contributed to the outcome. This technique starts from the CNN's outcome and uses the weights and activations of the neurons to find out the positive and negative evidence values of the pixels around and connected to the neurons.

$$f(x) = \sum_{d\epsilon l+1} R_d^{(l+1)} = \sum_{d\epsilon l} R_d^{(l)} = \sum_d R_d^{(l)}$$

As shown in the above Equation, the total relevance of each pixel is the same between layers. This rule makes sure that the explanation is consistent with the outcome throughout the propagation. This way, it is possible to understand what each layer ignored or focused on. The LWRP analyzes the outcome that there is a broken tooth in the gear and the influence of each pixel in the last layer of the CNN. It then

spreads the influence backward and fills the heat map, accordingly, resulting in a clear heat map of the most important features.

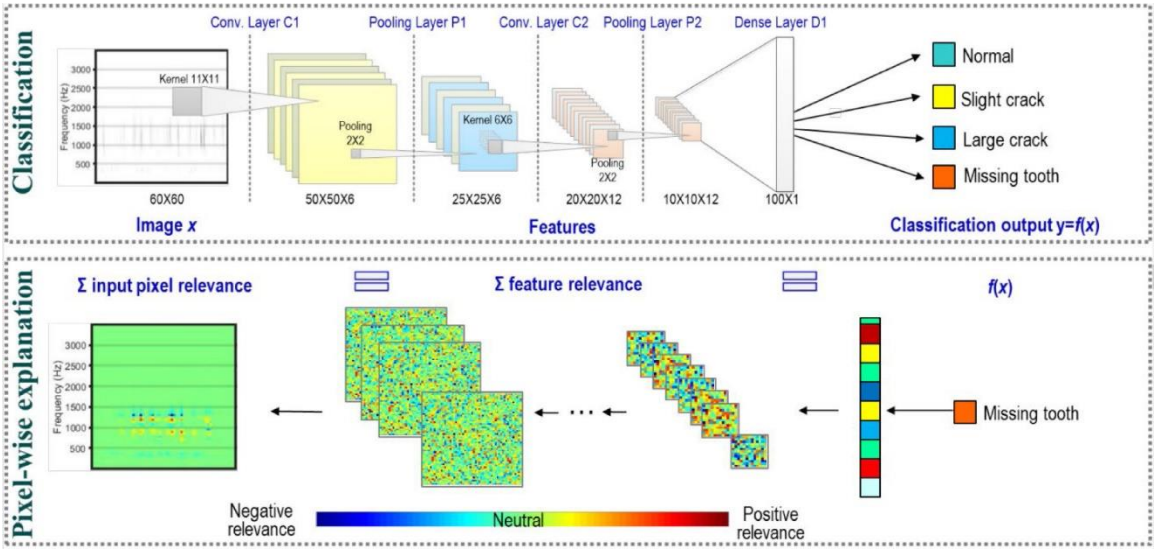

Figure 1 LWRP explanation

Method Analysis

LWRP is a technique that shows both positive and negative evidence for the outcome of a convolutional neural network (CNN). This makes the explanation more detailed because it does not only show how much each region matters but also which regions agree or disagree with the outcome. This could increase the trust in the model's accuracy because the explanation shows the whole picture. Also, the model looks at the influence of each pixel on the outcome, so the explanation should be accurate in showing which regions agree or disagree with the outcome. Lastly, the total influence of each pixel is the same between layers, which means that each layer is equally important for the outcome. This should reduce biases during the explanation process.

A main drawback of LWRP is that if the technique cannot find any influence for some pixels, it may just spread the influence on nearby pixels. Another drawback of looking at the influence of each pixel is that the explanation may rely on such a small area that there could be more overlap of possible outcomes. A single pixel could suggest different outcomes, so it would be hard to know if the explanation is right. With the influence of each pixel changing the influence of its neighbors, this could have a domino effect.

2.2 Shapley Additive Explanations

Methodology

Shapley Additive Explanations (SHAP) is a technique that explains the outcome of a black box model after it is made. SHAP examines each feature input and evaluates how much it matters for the outcome. This technique uses cooperative game theory to make a linear representation of the non-linear outcome. This is done by sampling the feature inputs in different combinations to see how they interact and affect the outcome. The result of this technique is a value for each feature input that shows how much it contributes to the output outcome. Feature sampling makes SHAP a good technique for both local and global explanations depending on which features are sampled.

Some variations of SHAP estimate the values in a less costly way, such as Kernel SHAP and Deep SHAP. Kernel SHAP lowers the cost by combining Linear LIME and Shapely values to make the explanation linear. SHAP can also be combined with other explanation techniques to lower the cost. For example, combining SHAP with the DeepLIFT technique, a technique similar to LWRP, resulted in Deep SHAP. Deep SHAP calculates the SHAP value for each input separately by assuming that the CNN model is linear. These two variations of SHAP, as well as others, allow SHAP to be used for explanation despite its high cost.

Method Analysis

SHAP is a technique that can explain the outcomes of a black box model both locally and globally, by sampling the feature inputs and evaluating their importance. It can also work on different types of neural networks because it does not depend on the model's structure. However, a big drawback of SHAP is that it requires a lot of computing power when there are many feature inputs. This is because the technique samples and tries different combinations of the feature inputs, and the computing power grows very fast with the number of feature inputs. However, some variations of SHAP can solve this problem.

2.3 Occlusion Tests

Methodology

Occlusion is a technique that explains the outcome of a convolutional neural network (CNN) by hiding or changing parts of an image and seeing how the prediction or the model's certainty changes. This shows which parts of the image are most important for the outcome, without knowing how the model's layers work. The information from each occlusion is used to make a saliency map, which is a visual way of showing how the model makes its decision. Occlusion can be done by making the value of each pixel in a region the same. Also, parts of an image can be covered by another image to hide important features, as shown in **Fig. 2**.

Figure 2 Occlusion of different parts of an image via iterate image overlap

Occlusion is very important for measuring how accurate a CNN decision is because research has shown that hiding parts of an image can make the decision less accurate. This is especially true if the model is trained on images that are not hidden at all. So, occlusion is used not only to explain CNN's decisions but also to make CNN's decisions better. By training images that are partly hidden, the model can handle situations where the main object is not fully visible. So, occlusion can make users trust CNN more, not only by making a saliency map that shows how the model decides but also by training the model on a more challenging dataset.

Occlusion is a technique that explains the outcome of a convolutional neural network (CNN) by hiding or changing parts of an image and seeing how the prediction or the model's certainty changes. There are many types of occlusions, but they all follow the same idea of hiding parts of the image and analyzing the change in the model outcome. Some examples of occlusion methods are Randomized Input Sampling for Explanation (RISE) and Time Series Randomized Evolving Input Sample Explanation (TimeREISE). RISE is an occlusion method that makes a saliency map by randomly hiding parts of the input image. RISE samples each image to make a whole dataset to test and the saliency map shows the importance of each of the

sampled images. TimeREISE is different from RISE because it uses time series classification to explain outcomes. TimeREISE also uses random masks like RISE, but it hides parts of the image over time and features so that it can work on time series data. This method also changes how much of the image is hidden so that the number of important data points for explanation can change. Another occlusion method is Similarity Difference and Uniqueness (SIDU). SIDU makes a saliency map based on the similarities, differences, and uniqueness values from the changes of the last layer in the CNN. This solves the problem of RISE picking random changes instead of finding the best ones for explanation.

Method Analysis

Occlusion is a technique that explains the outcome of a black box model after it is made by hiding or changing parts of an image and seeing how the prediction or the model's certainty changes. One of the benefits of occlusion is that it can work on different types of neural networks because it does not depend on the model's structure. This flexibility could make it faster to trust machine learning models for important applications. Also, some occlusion methods, like TimeREISE, can work on data that changes over time, which could make these explanations more useful for different domains.

However, as mentioned in the part about LIME, black box explanation methods do not show how a model makes its decision. Occlusion makes an explanation by randomly hiding parts of the image, which means that the explanation is not always the same.

2.4 Inversion

Methodology

Inversion of a CNN is a technique that tries to rebuild the original image from a convoluted image. Each layer of the network removes some of the images that it does not need for classification, so rebuilding the image after each layer can show what CNN thought was important to recognize the image and how it decided. This can help to understand the neural values in each layer of the network to show what features each layer focuses on.

There are two ways to do inversion. The first is optimization-based inversion, which uses changes to rebuild the image after each convoluted layer, which needs a white box model for change calculation. The other way is the training-based inversion, where the machine uses another model, called the inversion model, to rebuild the image to see the filtering of the convoluted layer. Rebuilding the image after the

convoluted layers still looks a lot like the original image, so the network is working to remove the background and zoom in on the main object of the image, as shown in **Fig. 3**. There are different ways to do inversion based on the math model used, but even with the different models, the rebuilt image is the same.

Functions show convoluted images and can be inverted mathematically. The original way of showing the image and the inversion method are used to change how the rebuilt image will look. However, the main shape of the main building is still there in all of the rebuilt images in **Fig. 3**, so in all cases, it is clear which parts of the image CNN thought were important for image classification. Details of how a function is made lead to the differences in the rebuilt images. As with any function that is integrated without limits, there are many solutions, as shown in **Fig. 4**. Just as there are many constant values, there are many possible rebuilt images. This makes the differences in the rebuilding of the image in **Fig. 3**.

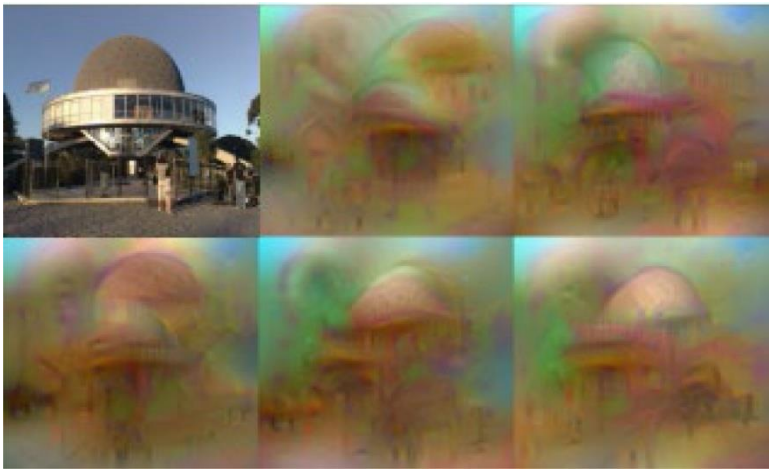

Figure 3 Five inversion variation of the original image in the top left corner

Fig. 5 shows how the image is rebuilt after each layer. The first image on the top left is the image after the first convoluted layer. The next images are the outcomes of the following layers in the network. After each layer, the rebuilt image looks more different from the original image, but it shows the most important features for classification. In this example, the face is the most important feature. **Fig. 5** shows the main purpose of neural network inversion. By rebuilding the image after each layer in the CNN, it is possible to see what each layer focused on to recognize the object in the image. This can also help to find the neural values of each layer.

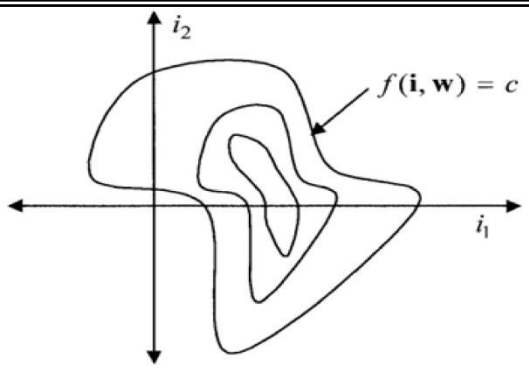

Figure 4 A family of solutions of an original function. Represents how there are many ways to invert the original image

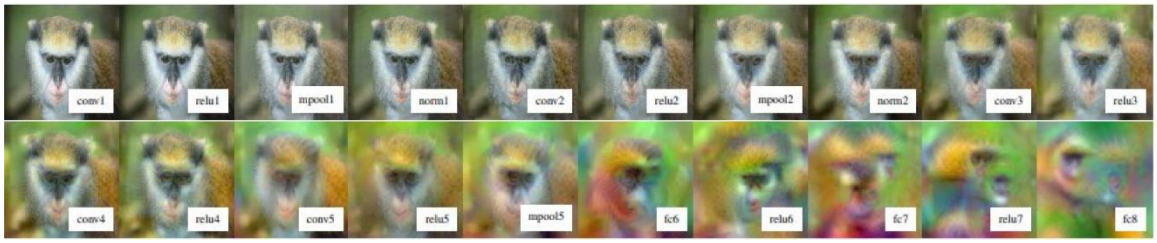

Figure 5 Inversion of the image after layer of the CNN

Method Analysis

Inversion is a technique that tries to rebuild the original image from a convoluted image. This can make it easier to see which parts of the image are removed by each layer of the network, because it can compare the rebuilt image to the original one and see how the network decided. The first layers start to remove the background parts and the last layers only keep the most important shapes of things in the image. Also, the math behind inversion is easy once the computer has the image as a function. This makes the explanation easy for non-math experts to understand.

However, this has a problem because many functions can show the image. This means that when the network decides again if the function is very different, the explanation could be very different too. The explanation not being the same could make the user trust the network less. Also, inversion creates a problem of finding the best way to change the direction of the calculation and turn the outcomes into inputs.

2.5 Logic Rules

Methodology

One way to explain the decisions of CNNs is to use fuzzy logic rules, which are sentences that express the reasoning in a logical way. Unlike other explanation methods that produce images, fuzzy logic rules give more detailed descriptions of the decisions by highlighting the relevant parts of the image. The main format of these rules is a conditional statement that provides a linguistic explanation of the decision. These conditional statements can also make logical deductions in the future. The rules are derived from the initial data set and can be modified during the training process of the CNN or by using both objective and subjective data.

The fuzzy logic rules can have different structures, such as ordered or unordered, depending on the order of the conditions. For a local explanation, there are two kinds of rules. A logic rule is a direct explanation of how the CNN identified a part of the image or a filter. A counterfactual rule shows how to change the conditions of the part or the filter to reverse it. Combining both kinds of rules can create a decision rule, which is a local explanation of the decision made. The local explanations contribute to the final decision and can be traced back to understand the decision better. For example, **Fig. 6** shows a decision rule e for rejecting a loan. The two rules below it, denoted by are counterfactual rules that would change the decision of the black box model. As shown, this explanation method uses mathematical logic. To make the rules more interpretable by humans, they can be written as conditional statements or organized into a decision tree.

$$e = \langle r = \{age \leq 25, job = none, amount > 5k\} \rightarrow deny,$$
$$\Phi = \{(\{age > 25, amount \leq 5k\} \rightarrow grant),$$
$$(\{job = clerk, car = yes\} \rightarrow grant)\}\rangle$$

Figure 6 An example of decision and counterfactual rules

Fuzzy logic can be described as computing with words instead of numbers and symbols. Fuzzy logic groups similar elements together to form a granule, which is described by keywords. When making decisions, such as in image classification, the model can group elements of the image into granules and use the keywords as an explanation of the decision. This explanation also includes mathematical representations in the form of fuzzy graphs, which are approximations of the real function. **Fig. 7** shows the actual function as the line and the fuzzy approximation of the function as the boxes around it. In image classification, functions can represent

images and each box can be a related granule that explains the rationale behind the decision.

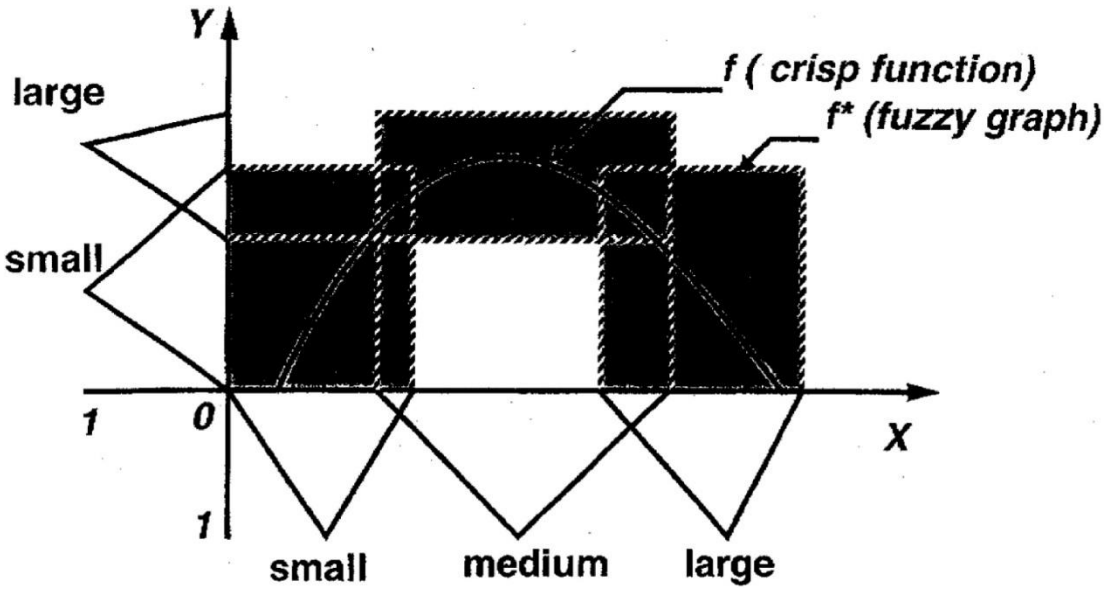

Figure 7 The actual function versus the fuzzy approximation

The architecture of the CNN can incorporate fuzzy logic by adding a fuzzy logic layer, which is usually near or instead of the fully connected layer. The fuzzy logic layer can extract the logic rules from the classification layer and propagate them backward to the input layer to find the words that describe the decision-making process. The activation of certain filters in each layer adds words to the final logical rule, so following the words backward can show which filters are activated in each layer and what the neural network considers important for classifying the image.

Method Analysis

Logic rules not only show the important part of the image for the decision but also explain how that part influenced the decision. Their clarity makes the explanation easier to comprehend. It can also be easier to notice changes in the model by looking at changes in the if-statements. With other explanation methods, there could be small changes in parts of the image, but the changes could be harder to notice than with logic rules. This method should be easy to understand because words explain the rules, which makes the explanation suitable for a non-expert in the field. However, the reasons for the decision made for image recognition could be more complicated than simple if-statements can show. When explaining convolution, complex math and ideas could make the if-statements hard to follow. While a math expert might get the rules, a non-expert, or even an expert in a non-math field, like a doctor, might not get the explanation, which makes it less useful.

2.6 Deep Taylor Decomposition

Methodology

Deep Taylor decomposition is a technique that explains the outcome of a convolutional neural network (CNN) by breaking it down into smaller calculations. It does this by using backpropagation and Taylor expansions on each neuron in the network. The explanation begins at the output and assigns relevance to each neuron that is activated in that layer, then goes back, and does the same thing on the previous layer. This continues until the model reaches the input image and gives relevant values to each pixel of the image. This results in a saliency map that shows how much each pixel matters for the outcome, as shown in **Fig. 8**. This is similar to LWRP, but with the deep Taylor method, the breakdown in each layer is easier to do. Like LWRP, deep Taylor decomposition also needs the total relevance of each layer to be the same.

As shown in **Fig. 8**, the outcome of deep Taylor decomposition is the importance of each pixel for the outcome. In the figure, the dark pixels show the background pixels that did not matter for the outcome. This saliency map is simpler and shows the positive and negative evidence from LWRP. However, both methods have benefits depending on the needs of the domain and application.

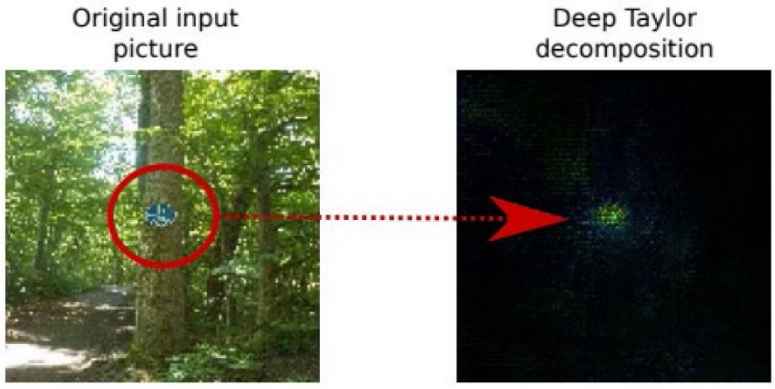

Figure 8 Saliency map from deep Taylor decomposition

Method Analysis

Deep Taylor decomposition is a technique that breaks down the outcome of a convolutional neural network (CNN) into smaller calculations. It has an advantage over LWRP, which is another technique that does the same thing because the calculations are easier. However, this makes the explanation more unstable, which reduces the user's trust in the technique compared to LWRP. This is more of a

problem for complex models, like deep neural networks. Also, deep Taylor decomposition makes simpler saliency maps than LWRP, as shown in **Fig. 8**. These saliency maps do not show the pixels that did not matter for the outcome, but they do show the pixels that did matter.

2.7 Deconvolutional Neural Network

Methodology

Convolutional Neural Networks (CNNs) process an image by eliminating irrelevant background details, utilizing a kernel and the convolution operation to identify key features for categorization. This extraction is performed by the convolutional and pooling layers. Deconvolutional networks can take these CNN-extracted features and use deconvolutional and unpooling layers to identify the feature shapes from the CNN, assigning a probability to each shape based on its importance for image classification, as determined by neuron activations. This process is illustrated in **Fig. 9**. The dog image on the far left is the input. The CNN processes this input, extracting crucial features. The CNN's prediction is made in the figure's center, and the deconvolutional network on the figure's right side reverses the filtering and pooling processes to create a probability map, explaining the CNN's prediction. The CNN's filtering layers identify and retain key characteristics, while the deconvolutional network's unfiltering layers reverse this process by flipping the filters in both directions.

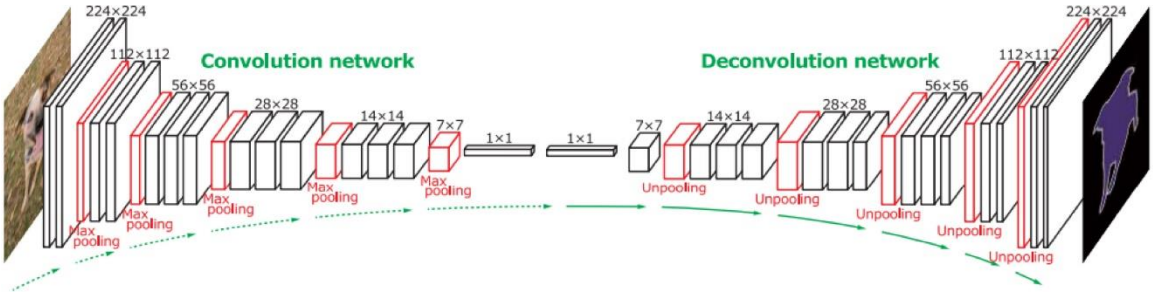

Figure 9 Deconvolutional Neural Network Structure

Fig. 10 illustrates the operational differences between the convolutional and deconvolutional networks. The CNN's pooling layers condense the image to its most important features. To generate an explanation, the deconvolutional network's unpooling layers expand this condensed image to understand the CNN's prediction process. The unpooling layers highlight the shapes with the highest activation levels in the CNN, which increases their likelihood of being used for image classification. These activation maps are sparse as the CNN filters out many image parts, so the deconvolutional layers increase the resulting activation maps' density using

mathematical operations, just as the convolutional layers do the opposite through convolutions. **Fig. 11** illustrates this, where (a) is the input image, (c) is the unpooled image, and (b) and (d) are deconvoluted images. The unpooled image displays the activated features with large spaces between them, while the deconvoluted images fill in these gaps.

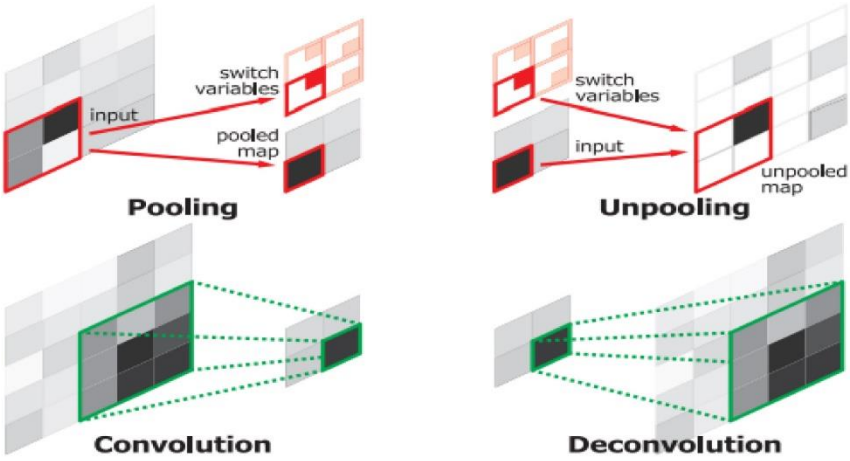

Figure 10 Pooling versus Unpooling Layer

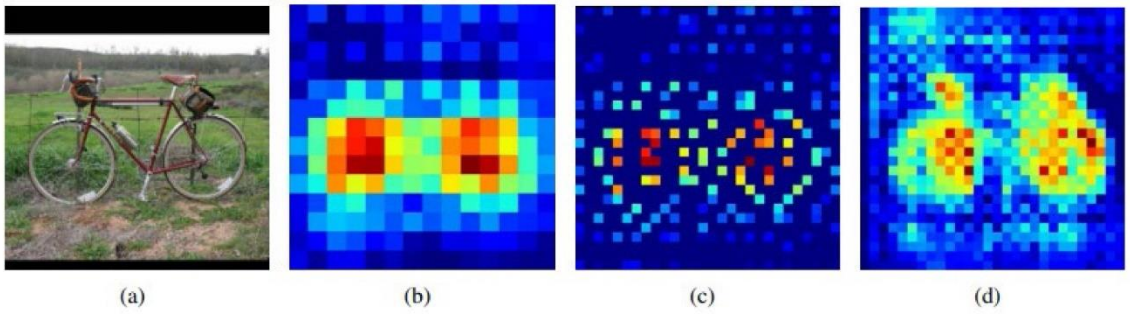

Figure 11 Densifying the Unpooled Images

Deconvolutional neural networks are directly linked to the previously discussed issue of trust in black box models in healthcare. Sometimes, a Magnetic Resonance Image (MRI) or Computed Tomography (CT) image may have low resolution due to inadequate exposure time or equipment constraints. Despite this, CNNs can still make a prediction and diagnosis based on these images. However, by adding a deconvolutional layer, the network can increase the images' density, thereby enhancing their resolution and readability, and providing greater confidence in the diagnosis. Convolutional and deconvolutional neural networks can also be used together in a two-path design, where the convolutional and pooling layers provide a detailed image based on feature activation, and the deconvolutional and unpooling

layers provide a coarse, global view of the image. In summary, deconvolutional neural networks generate a precise and dense activation map that represents the entire input image, explaining the CNN's decision-making process.

Methodology Analysis

Deconvolutional neural networks offer several benefits. They are relatively easy to comprehend for those familiar with Convolutional Neural Networks (CNNs) as they perform functions that are the inverse of those performed by CNNs. For instance, understanding pooling layers can aid in comprehending unpooling layers. Furthermore, deconvolutional neural networks provide a visual depiction of the decision explanation through a probability map and an activation map. Most importantly, deconvolutional neural networks enhance their explanation by densifying it. While other explanation methods might only display a map of the significant regions remaining at the end, deconvolutional neural networks densify the image while retracing the neural activations, resulting in an activation map that closely resembles the original image.

However, this explanation method has some limitations. The primary issue is that the deconvolutional neural network can reanalyze some pixels, leading to a checkerboard effect that alters the reconstructed image's appearance. As shown in **Fig. 11**, even with densification, it might be challenging to recognize the original image in the reconstruction without the appropriate architecture. Consequently, despite the densification of the reconstructed image, the explanation could be hard to understand and validate. Lastly, deconvolution only indicates the relative contribution of each region to the decision. In **Fig. 11**, the blue areas had a low relative contribution, while yellow and red had a higher relative contribution. This explanation does not differentiate the regions that suggested an alternative decision.

3. Conclusion

This literature review of various methods used to explain the decisions made by Convolutional Neural Networks (CNNs). It presents a review of many commonly used and innovative explainability methods, each with its own set of advantages and disadvantages. Despite the accuracy of CNNs, their inability to provide explanations for their decisions has led to a lack of confidence in their use in high-risk sectors such as healthcare and banking. The report discusses several common explainability methods including **Layer-Wise Relevance Propagation**, **Shapley Additive Explanations**, **Occlusion Tests**, **Inversion**, **Logic Rules**, **Deep Taylor Decomposition**, and **Deconvolutional Neural Networks**. Each method has unique advantages not found in the others. Expanding upon the analysis could lead to a more comprehensive ontology of explainability methods, serving as a reference for domain experts looking for the most effective method for their specific domain when using CNN.

Future research could explore the possibility of merging some of these advantages to create new explainability methods. The use of such methods could help bridge the gap in applying CNNs to high-risk applications.

4. References

1. Bach et al., 2015. On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation. [**Google Scholar**]
2. Bazen and Joutard, 2013. The Taylor Decomposition: A Unified Generalization of the Oaxaca Method to Nonlinear Models. [**Google Scholar**]
3. Cao et al., 2020. Deconvolutional neural network for image super-resolution [**Google Scholar**]
4. Kauffmann et al., 2020. Towards explaining anomalies: A deep Taylor decomposition of one-class models. [**Google Scholar**]
5. Liu et al., 2018. Learning deconvolutional deep neural network for high-resolution medical image reconstruction. [**Google Scholar**]
6. Muddamsetty et al., 2021. Visual explanation of black-box model: Similarity difference and uniqueness (SIDU) method. [**Google Scholar**]
7. Zhao et al., 2020. SHAP values for explaining CNN-based text classification models. [**Google Scholar**]
8. Osherov and Lindenbaum, 2017. Increasing CNN robustness to occlusions by reducing filter support. In: IEEE International Conference on Computer Vision, pp. 550-561. [**Google Scholar**]
9. Zhang and Zhu, 2018c. Visual Interpretability for Deep Learning: A Survey. [**Google Scholar**]
10. Yang et al., 2019. Neural network inversion in adversarial setting via background knowledge alignment. In: ACM SIGSAC Conference on Computer and Communications Security. London, UK. [**Google Scholar**]