# Tekst Introduction

Nowadays AI is apparent in every part of our society. Nearly every large company gathers and deals with data. You have all probably heard of Chat GPT, you have all used netflix’s movie recommendations or Google photos automatic photo labelling. This rise of AI can mostly be attributed to the increase in available computing power. Nowadays models can have a very large number of parameters. The sheer size and complexity of these models make them very powerful, however, it also makes it very difficult to understand their decisions.

This lack of transparency can become a problem when the model makes peculiar decisions. For example, on this slide you can see that when Chat GPT is prompted to write a python function to check if someone would be a good scientist based on a description of their race and gender, it returns a function which describes a good scientist as a male white individual. Similarly, google photos when it first came out accidentally labelled some afro americans as gorillas. These examples show that existing biases and ethical problems are perpetuated via the data into the algorithm.

A novel research field Explainable AI aims to mitigate these problems by either generating explanations of existing models or by making models intrinsically explainable. Besides highlighting ethical concerns, these explanations could also help a data scientist to get a better understanding of the model and improve its decision-making.

# Tekst Research problem

Explainable AI can be focused on a variety of models. For my Thesis I have dived into explaining Convolutional Neural Networks, or CNNs in short. In particular, I have looked at the image classification task. Therefore my Research problem is as follows:

Given a Convolutional Neural Network (CNN) trained on an image classification task: How can we allow users to explicitly compare expectations of the CNN’s reasoning to its actual behaviour?

What this means is that a user needs to be able to explicitly investigate any expectation they have. For example, given a CNN trained to classify zebras, a user needs to be able to explicitly check any expectation they might have.

The approach that I have taken to tackle this problem is to develop an interactive dashboard in which users can upload any pre-trained CNN and dataset and generate explanations for said CNN. In particular, the type of explanations I have chosen to use are concept-based explanations. This type of explanations is chosen since they best mimic how humans recognize images.

In the remainder of the presentation, I will first explain what exactly an image classification task is and what a CNN is. Next, I will introduce concept-based explanations, in particular, the methods that are used by my tool. Finally I will introduce the tool that I developed and give a quick evaluation of my tool and its contribution to the research field.

# Preliminaries

An image classification task is intuitively actually quite simple. Given a set of images 𝑋 and a set of labels 𝑌: Find a decision function 𝑓:𝑋 →𝑌 that minimizes the error. This error can be as simple as the number of incorrectly classified images. The decision function f that needs to be found is trained with existing data, however, it should work on new unseen data.

The type of model used to learn the decision function f is usually Convolutional Neural Networks. These models consist of a number of layers. The first part of the model consists of convolutional layers. These layers aim to find feature maps. In the early layers these feature maps are very simple. For example, vertical stripes. In higher layers these feature maps get combined to be more complex maps. For example, a feature map can check whether eyes are present. These feature maps and combining them together is part of what makes CNNs so useful for image analysis tasks. In the fully connected part of the CNN, the resulting feature maps get flattened into a single vector and the model learns to classify images based on this feature vector.

Now that you know what an image classification task is and what a CNN is I will explain what concept-based explanations for these networks are.

# Concept-based explanations

Concept-based explanations explain the decisions of a CNN in terms of human-understandable concepts. A concept is anything that can be defined via a set of images. For example, a set of images of black-and-white stripes represents the concept black-and-white stripes. Similarly a set of images of wheels represents the concept wheels. These concepts can either be user-defined or automatically extracted from a dataset.

The visualization on the slides give a quick overview of how these concepts work. In part a of the image you can see two rows of images. The top row consists of images of stripes. This set of images represents the concept Stripes. The bottom row consists of random images. In figure b you can see that we have a dataset of zebras for which we try to generate explanations. In particular, we want to know how important the concept stripes is for the classification of a zebra. In subfigure c, we see a pre-trained CNN. This CNN is divided into two parts. Part f consists of all of the CNN up to layer l. Part h, is the remaining part. The user needs to choose a layer for which to generate the explanations. As explained with the feature maps, lower layers correspond to lower level objects such as texture, whereas higher layers correspond to higher level objects like wheels. As a general rule of thumb, a middle layer can be chosen to have a good tradeoff between low level objects and high level objects. So now we have a pretrained CNN and a concept stripes. The images representing the concept stripes are fed into the first part of the CNN as are the random images. These images are now represented by points in the activation space of layer l of the CNN. As can be seen in figure d. We then take the vector, called a Concept Activation Vector (CAV) that points in the direction of the striped images. This vector is found by training a linear model and taking the vector orthogonal to the decision function. Using this vector we can compute an importance score.

**importance**

Don’t worry, this slide might look a bit daunting, however, I will explain the formulas in a more intuitive way. The goal of this formula is to compute the importance score of a single concept. This is achieved by first computing the directional derivative. This is done by the first formula. In particular, we compute the rate of change of the logit of the class k when we move a tiny step in the direction of the concept v\_c. This corresponds to the directional derivative. A high value means that the class k is sensitive to concept v\_c and therefore that this concept is used by the CNN in classifying the images. This is exactly what the second formula tries to quantify. This formula computes the fraction of images of the class that are positively influenced by the concept. So lets’s say the outcome is 0.7. This would mean that for 70% of the images of a zebra the classification of said image is sensitive to the concept stripes.

**ACE**

Up to now I have explained how to define a concept, how to represent it and how to compute the importance of said concept. Now I will explain how these concepts can be automatically extracted from a dataset. The visualization on the slide shows a quick overview of this process. First, an image of a class is segmented into multiple different resolutions. In this case three different resolution levels. This results in segments of different sizes. Three different resolution scales are used such that we receive low level up to high level concepts. Next, these segments are clustered into clusters of similar images. Outliers are removed and the remaining clusters correspond to sets of images that represent a concept. In this case, we see that the concept window, wheel and sky are found. From the sets of images Concept Activation Vectors can be formed as explained before.

**Comparison**

As explained, there are two types of concept-based explanations, user-defined concepts or automatically extracted concepts. They both have their own advantages and disadvantages. User defined concepts can perpetuate human biases, when users only look into their own beliefs they might miss important concepts that they did not think about. Moreover, it is very time intensive to define concepts yourself. You need to gather a set of minimum 40 images that represent a concept. In particular, when the concepts correspond to small parts of an object, for example the ear shape of an elephant, it is hard to gather these images. The advantages are that the resulting concepts are, by definition, easy to understand. Moreover, defining your own concepts allows you to explicitly check the importance of said concept to the model’s behaviour. On the other hand, automatically generated concepts do not suffer from human biases and are quick to generate. However, the resulting concepts can be difficult to understand and there is no guarantee that the resulting concepts align with human understandable concepts. Moreover, it does not allow for explicitly checking the importance of a concept when the concept is not found by the model.

**Contributions of our tool**

Since user-defined concepts and automated concepts have very different advantages and disadvantages, we argue for a combination of the two. In particular, the automated part solves the human bias and gives an improve in concept-definition time when it finds concept that you would have defined anyways. Moreover, it still allows you to explicitly compare the importance of the model’s reasoning to its behavior.

Besides combining automated concept extraction with user-defined concepts, our tool also supports an iterative concept-definition process instead of a linear process. This is important because users might change their beliefs about which concepts are important when inspecting the generated explanations and data. For example, users might not think about the background of an image when defining the concepts for the toucan class, however, after further inspection, all toucans seem to sit on trees in the jungle. By supporting an iterative concept-definition loop we allow users to add or remove desired concepts.

Now you know the motivation of our tool, its contributions to the research field and its technical backbone. I can finally show you the interactive dashboard itself.

# TOOL

This Tool consists of two pages. This first page is the concept-definition page. In this page, users can define, interact with and visualize the Concept Bank. This page consists of three parts.

**a**

First the settings part, which can be seen in the blue square in page a. In the settings section, users can set the working directory of the tool. This is where the defined concepts are stored such that they can be quickly refound when the user continues the session at a later time. The model selection input setting allows the user to upload the pretrained CNN with a path to the model or by default the Inception model is used. Next, the data input setting can be used to upload the dataset such that concepts can be extracted. This dataset can be independent of the training data used for the CNN. Next, the class for which concepts need to be extracted needs to be defined as well as the layer of the CNN for which the explanations need to be generated. As you can see, green checkmarks appear when the chosen input makes sense. For example, whether the class is in the data and whether the layer is in the CNN.

**B**

In part b of the tool we can interact with the Concept Bank. In particular we can automatically extract concepts via the corresponding button. We can remove existing concepts. This will usually be needed since the automated concept extraction may find concepts that are not understandable for humans or it might find duplicate concepts. Moreover, users can also define their own concepts. They need to specify for which class the concept is and they can then upload a set of images via the upload button. Finally users can import a concept bank they have defined before, clear the current Concept Bank to start over or save the concept bank on hard drive.

**C**

Finally in part c of the tool, the Concept Bank can be visualized. In this visualization you can see multiple things. Every visualization of a concept consists of two rows. The upper row is the segment that is extracted by the automated concept extraction method. The bottom row is the location of the segment in the original image. For example, the upper concept, concept 17 consists of segments that represent the colorful eye of the toucan. Next to the name of the concept you can also see the importance score. In this case the score is 1, which means that this concept positively influences the classification of every image of a toucan. Besides that is the p-value of the TCAV score, which tests whether the concept is consistent. Finally, the accuracy of finding the CAV is denoted.

**Second page**

Now we have defined a concept bank we can go to the concept importance page. In this page we try to classify classes based solely on the concept information. In particular, we take the inner product of the concepts with the images to get a dataset where each row is an image and each column is the activation of a concept. Based on this dataset we train a linear classifier to classify the classes based solely on these concept values.

This page also consists of three parts. First in part a) of the tool you can include which classes you want to classify based on the concept information. You can either choose a selection of some classes or you can choose all classes at once. Next, you can start the classification by choosing the layer for which concepts are found and pressing the start button.

In part b of the page, you can find the descriptive results of the classification. In particular, we show the size of the dataset used to train the classifier and the size of the dataset used to get the test results. Besides that, you can see the train and test accuracy and AUC to find the performance of the classifier. These values can be used to get the completeness of the concept bank. When the accuracy and AUC are very high for the classifier, we can see that the concept information alone is sufficient to recognize these images. When the accuracy and AUC is low, we know that additional concepts are needed.

Finally, in part c of the image, we show the concept importances and a confusion matrix. The feature importances are used to see which concepts are important for a class prediction. This allows us to determine which concepts are most important for a class prediction. You can choose for which class you want to see the concept importance. Note that the concept importance might vary depending on the classes chosen to be included in the classification problem. For example, separating a toucan from a great white shark is quite easy since they are completely different and live in different environments. However, separating a toucan from a hornbill is much harder since they look very similar and live in similar environments. Next, the confusion matrix is shown to determine regions of error. In particular, it allows for seeing which classes are hard to separate.

# Evaluation

Now that you know how the tool works, I would like to reiterate the contributions and novelty this tool makes with respect to the current research field. My tool is one of the first concept-based explanation method that explicitly views the concept-definition process as an iterative process. Additionally, my tool is the first concept-based explanation method that combines user-defined concepts with automated concept extraction methods. I have evaluated both contributions via a questionnaire. In the first part of the questionnaire, I have asked users to list a set of concepts they find important for recognizing an object in an image. I have compared the resulting list of concepts with the ones that are automatically extracted. This is done to evaluate whether using both results in a more extensive concept set and whether it could save time when defining the concepts. In the next part of the questionnaire I have evaluated whether a user can change its perceived importance of concepts after being provided with more information. This is done by asking the user to rank a set of concepts based on importance for recognizing a object. Then the user is provided with information about the object and is asked to rank the concepts again.

**Results**

The results of this questionnaire are found by averaging the answers of 30 respondents. On average half of the concepts that are defined by a user are also found by automatically extracting the concepts. Interestingly, concepts that are not found by the automated concept extraction method are usually concepts that are quite abstract. For example, Asian for a wok pan or the military for a half track.

Moreover, on average 1.1 additional concepts are found by automatically extracting concepts. The average size of the concept sets found by the users is 5.5 so this is a 20% increase. On the other hand, users define an additional 2.5 concepts that are not defined by the automated concept extraction method.

The second part of the questionnaire tested whether users could change their belief about the importance of concepts for an image classification task. This is done by testing two different scenarios. In the first scenario, users needed to rank a set of 5 concepts on classifying a stingray. After providing them with the information that there are actually two types of stingrays, one which has blue spots and one which does not. Users on average increased the perceived importance of blue spots from rank 4.5 to rank 3. The second scenario asked a similar question, however, in this scenario users were asked to rank 5 concepts for classifying an elephant. After providing them with information that elephants can have different ear shapes based on whether they are Asian or Indian elephants. Users did not change the perceived importance of the concept ear shape. It remained the same rank: namely rank 3. This is likely because users already found ear shape to be a defining concept for elephants.

# Conclusions

From these results we can draw some preliminary conclusions: Indeed combining automatically extracted concepts with user-defined concepts can result in a more extensive concept set. This can mitigate human bias since concepts are used that a user would not have included himself. Besides that, combining the two helps the user save a lot of time. Finally, in specific scenarios where the knowledge of the class is limited users can change the perceived importance of concepts based on novel information. This warrants an iterative concept-definition process. In conclusion, my tool is the first tool that combines automated concept extraction methods with user-defined concepts. It is also the first concept-based explanation method that supports an iterative concept-definition process as opposed to a linear process. The results seem to warrant both novel contributions and my tool functions as a proof of concept that both are interesting research directions to explore further.

# Questions?