UNIVERSITY OF TARTU
Institute of Computer Science
Computer Science Curriculum

Karl-Gustav Kallasmaa

Personalized concept-based image classification explanation framework

Master's Thesis (30 ECTS)

Supervisor: Radwa ElShawi, PhD

Tartu 2023

Personalized concept-based image classification explanation framework

Abstract: In recent years, the adaptation of machine learning models has proliferated. Explaining those models is essential for the end-users to install trust and mitigate potential algorithmic biases. Most current interpretability techniques predominantly rely on pixel or feature importance, making it challenging to intuitively explain these results to humans. This Thesis introduces a novel local concept-based explanation framework designed to explain image classification models. The framework empowers users to create personalized explanations through intelligent concept suggestions. These chosen concepts are used to train a shallow decision tree that is used to explain the image classifier. Additionally, the framework allows users to request a re-explanation by modifying the concepts and do receive a counterfactual explanation. The frameworks' effectiveness was tested by explaining the ResNet-50 image classifier decisions on the ADE20K dataset. The framework demonstrated a higher fidelity than LIME for this dataset and model. The intuitiveness and meaningfulness were measured through human-centric evaluations. These experiments showed that the frameworks' explanations are more intuitive than LIME.

Keywords: explainable ai, concept-based explanations

CERCS: P176

Personaliseeritud mõistete põhine piltide klassifitseerimise seletamise raamistik

Lühikokkuvõte: Viimastel aastatel on masinõppe mudelid leidnud üha laiapõhisemat kasutust. Mudelite selgitamine on väga oluline, selleks et lõppkasutajaid neid usaldaksid ning selleks, et oleks võimalik vältida algoritmilist kallutatust. Enamik praeguseid seletusmeetodeid tuginevad eelkõige üksikutel pikslitel või üksikutel muutujatel, muutes nende tulemuste inimestele seletamise keeruliseks. Käesoleva magistritöö eesmärgiks oli tutvustada uuendusliku lokaalsete selgituse raamistiku. Raamistik võimaldab kasutajatel luua isikupärastatud seletusi pildiklassifitseerimise mudelile kasutades nutikat kontseptide pakkumise protsessi. Kasutaja poolt valitud kontseptsioone kasutatakse pinnapealse otsustuspuu treenimiseks. Hiljem kasutatakse seda otsustuspuud mustakasti mudeli seletamiseks. Lisaks sellele võimaldab raamistik kasutajatel selgitusi uuesti seletada lubades neil muuta raamistikule kättesaadavaid mõisteid. Raamistikul on võimekust pakkuda ka kontrafaktilisi selgitusi. Raamistiku tulemuslikkust testiti kasutades ResNet-50 mudelit ning ADE20K andmestiku. Eksperimendi tulemusena selgus, et selle andmestiku ning mudeli puhul on raamistiku usaldusväärsus LIME omast suurem. Selgituste intuitiivsuse ning tähenduslikkuse mõõtmiseks kasutati inimkeskseid hindamismeetodeid. Eksperimendi tulemusel selgus et raamistiku intuitiivsus on kõrgem kui LIME oma.

Võtmesõnad: seletatav tehisintellekt, mõistete põhised seletused

CERCS: P176

Table of Contents

1 Introduction

Many applications that billions of people use daily rely heavily on machine learning models because of their superior performance. The main downside of such models is their inexplicability, preventing their adoption in critical and regulated environments. According to the General Data Protection Regulation (GDPR), the data controller must ensure that if they use automated decision-making software, they must be able to provide information to the data subject about the logic involved in the process [1]. Sufficiently explaining such models' behavior would empower people to understand their inner workings better, comply with regulations, and build trust in them.

There is no universal standard for determining whether the explanation is satisfactory [2]. Any given prediction is explainable in several ways, and choosing the most effective interpretability technique is a challenging task because many factors influence how well the individual explainee perceives the explanation, such as the dataset and the characteristics of the model.

This Thesis is motivated by the increased pressure to provide intuitive explanations to ever-complex models. Many current interpretability techniques fail to do so because they rely too much on individual feature/pixel importance.

This Thesis proposes and assesses a novel explanation framework designed to explain and re-explain black box image classification decisions using human-defined concepts. The framework gives users personalized explanations by allowing them to specify which concepts they want to see in the explanation. In addition, the framework offers counterfactual explanations, showcasing the minimal changes required to classify an image with the desired label.

This Thesis proposes the following research questions to assess the framework in more detail:

1. Do concept-based explanations produce more faithful explanations than feature attribution methods?
2. Do decision trees produce more intuitive explanations than LIME?
3. How meaningful are the extracted concepts?

The first research question evaluates how well the framework can explain the black box models' thought process because that is the main aim of any explanation framework. For a subset of the images, the fidelity of the concept-based explanation framework will be compared to the fidelity of LIME.

The second research question explores how intuitive are the explanations produced by this framework. Intuitive explanations are explanations that the people receiving them can easily understand and interpret. The interpretability of the proposed framework will be measured through human evaluations by showing two explanations, one made by the framework and the other created by the LIME, to a group of users and asking them to select a more intuitive explanation from these two for a total of ten images.

The final research question addresses the quality of the concept suggestions process. The framework relies heavily on user-selected concepts, so it is essential to validate the effectiveness of the concept proposal process. Meaningfulness is measured through human evaluations by showing a group of users ten images and asking them to select the most meaningful concepts from the proposed list.

This Thesis has the following structure. Section two gives an overview of different interpretability methods and their advantages and shortcomings. The next section introduces the improved personalized concept-based explanations framework. Section four gives an overview of the experimental results. Section five highlights the limitations and further improvement areas of the framework. Finally, the Thesis is summarized in the Conclusion chapter.

Two text-generation software solutions will be used for this Thesis. Firstly, Grammarly will be used to fix basic grammar mistakes and as a feedback mechanism on text readability. Secondly, Chat GPT will be used for debugging the code in the experimental section. The code used in this Thesis comes exclusively from official documentation and publicly accessible repositories, available to anyone through a conventional search engine.

2 Background

Explaining machine learning models is done through different interpretability frameworks, whose aim is to describe the inner workings of a model in a way that is understandable to a human. For example, using interpretability tools, it is possible to see what parts of an image were used by a classification model to label a particular image as a "dog". The frameworks are usually tasked with explaining black box models, which are machine learning models with a very complex decision-making process, such as deep neural networks, and it is unclear why a given prediction was made. Explaining models' predictions allows us to enhance their performance and comply with different regulations.

This chapter aims to give an overview of different interpretability methods, their advantages and disadvantages, and example use cases. The structure of this section closely follows the classification used by Molnar [3].

In Section 2.1, an overview of interpretable models, also known as transparent models, is given. Transparent models are models whose parameters can be directly interpreted or where interpretability requires little extra work. Examples of such methods include linear regression, logistic regression, and decision trees.

Section 2.2 gives an overview of model agnostic methods. These methods can be applied to any pre-trained model, regardless of its architecture and prediction type. Examples of such methods include LIME (Local Interpretable Model-Agnostic Explanation) and Sharply values.

2.1 Transparent methods

Transparent methods are a collection of interpretability methods where the model's parameters can be interpreted directly or with little extra work. This section covers three popular transparent models: linear regression, logistic regression, and decision trees.

2.1.1 Linear regression

Linear regression is a regression that uses Equation 1 as a prediction model.

$$y = a_0 + \sum_{i=1}^{k} a_i x_i + \in \qquad (1)$$

Choosing values for a set of parameters to explain the models' prediction is unnecessary for linear models, as the explanation process is clear. Model *M* achieved prediction *y* because the

sum of the independent variable and the dot product of the regression parameters and input X_i resulted in prediction y.

Understanding the direct impact of each input feature is also simple and intuitive. The sign of each regression parameter communicates whether a particular feature contributes to the rise of the target value or not, and the numerical value of the regression parameter expresses how important each feature was.

Linear regression is a valid tool for making predictions when the relationship between input features and the target variable is intuitive, meaning that the target variable is a linear combination of input features. Sidney-Gibbons and Sidney-Gibbons [4] measured the performance of a general linear model (GLM), a support vector machine (SVM), and a single-layer artificial neural network (ANN) in predicting cancer using descriptions of nuclei sampled from breast masses. In their findings, SVM achieved the highest accuracy (0.96), but the difference between the model with the highest accuracy and lowest accuracy (ANN) was 2%. The authors concluded that more complex algorithms like neural networks and SVMs, do not necessarily produce more accurate predictions.

When considering linear regression, the following constraints need to be addressed. Firstly, the task needs to be a regression task because a linear model is not capable of converting predictions to class probabilities or other output types. Secondly, linear models assume that the target variable follows the normal distribution. Finally, linear explanation models can only be used if features do not correlate with one another.

The two main advantages of using linear regression are the ease of interpretability and the ease of implementation. The two main disadvantages are the assumption of linearity [5] and the assumption of normality [6].

2.1.2 Logistic regression

Logistic regression is a classification model that uses Equation 2 as a prediction model. In this model the output is obtained by passing the linear model output through a sigmoid function *S*. This sigmoid function outputs a probability that indicates the likelihood that the target variable belongs to the "true" class.

$$p = S(a_0 + \sum_{i=1}^{k} a_i x_i) \qquad (2)$$

$$S(x) = \frac{1}{1 + e^{-x}}$$

Explaining the prediction of a logistic prediction is very similar to linear regression. Model *M* achieved prediction *y* because the sum of the independent variable and the dot product of the regression parameters and input X_i, given as an input to a sigmoid function, resulted in prediction *y*.

Interpreting the effect of each feature in the logistic regression is not as intuitive as for linear models. In linear regression, the effect of feature x_i is captured by the regression parameter a_i. Proof 1 demonstrates that for logistic regression, an increment of one unit in the regression parameter a_i corresponds to an increase in the output variable value by $S(a_i)$.

Logistic regression could be used if the classification task can be formulated as a binary classification problem. This prerequisite originates from the fact that the sigmoid function used by the logistic regression will always output values between zero and one.

The main advantage of logistic regression over other methods is that the output is a probability. This allows us to measure the certainty of the model's prediction. The disadvantage of logistic regression is the fact that it is more challenging to interpret the role each variable played in the model's prediction because the interpretation of weights is multiplicative and not additive [3].

2.1.3 Decision trees

A decision tree is a model used for classification tasks that segment predictions into distinct groups based on the instance features. For example, if feature x_i is less than n, the instance belongs to class *y*. If not, the model uses the next feature x_{i+1}, for further division. The explanation of the decision tree is a path influenced by the feature values from the root to a leaf node. In essence, the decision tree made a specific prediction because the given feature value resulted in the outputted decision path.

The importance of the features can be calculated using the Gini index. The Gini index indicates how often the feature was selected for a split and how large its overall discriminative value was for the classification task.

Gini impurity at node t can be calculated using Equation 3, where p_i is the fraction of samples from class i and to the total number of samples. [7]

$$i(t) = 1 - \sum_{i=0}^{t} p_i^2 \qquad (3)$$

$$p_i = \frac{n_i}{n}$$

The Gini importance can be calculated using Equation 4. From the Gini index calculated at node t, the Gini indexes of its children are subtracted. The weight of each of its children is multiplied by the probability of picking those children at random. Equation 4 is applied to every feature, and all the calculated values are scaled to between zero and one, resulting in the final relative feature importance. [7]

$$I_G(f) = \sum_{j=0}^{k} \Delta i_f(j) \qquad (4)$$

$$\Delta i(t) = i(t) - \sum_{j=0}^{k} p_j i(i_j)$$

$$p_j = \frac{n_j}{n}$$

Decision trees can only be used for model interpretability if the interpreted model is a classification model because the decision tree outputs a class, not a real number.

The two main advantages of decision trees are the speed at which the interpretability model can be trained and the ease of interpretability. The two main disadvantages of decision tree-based models are overfitting and the unjustifiably large effect of small feature changes on the prediction outcome.

2.2 Model-Agnostic methods

Model agnostic methods are a set of interpretability methods that do not depend on the architecture of the pre-trained model and on the type of problem the algorithm is used for.

This chapter is divided into six subsections, each dedicated to an important model-agnostic method. Section 2.2.1 introduces Partial Dependence Plots (PDP), a method used to analyze the model's output dependence on a single feature. Following this, in section 2.2.2, Sharply values are described. This method shows the top features contributing to the decision and the top features contributing to a different decision. In section 2.2.3, Local Interpretable Model-Agnostic Explanations (LIME) is covered. LIME is a critical interpretability method because its performance is often used to measure the relative effectiveness of other interpretability methods. The following section, section 2.2.4, covers the Global Surrogate method.

The last two sections, 2.2.5 and 2.2.6, are the most important sections in this chapter because these explanation methods are used in the proposed frameworks. Section 2.2.5 describes counterfactual explanations. Counterfactual explanations aim to find an instance very similar to the one the explainee is interested in explaining but with a different label. Section

2.2.6 introduces concept-based explanations, a technique whether the explanations are provided using human-defined concepts, such as “TV” to explain an image classified as “living room”.

2.2.1 Partial Dependence Plot

Partial Dependence Plot (PDP), introduced by Friedman [8], is a technique that is used to show how the model partially depends on a single input feature or a collection of input features. In practice, no more than three features can be selected from the feature set for the PDP plot because humans cannot interpret more than three dimensions.

Let z_l denote the features the user is interested in and let i denote a single feature fromz_l. To estimate the partial dependence of the model on feature i a value of the function f_l^* is calculated for every value that i has in the training set, and the average overall training instances is noted. An approximation of the partial dependence function is calculated using Equation 5 [8].

$$f_l^*(z_l) = \frac{1}{N}\sum_{i=1}^{N} f_l^*\left(z_l, z_{i,/l}\right) \qquad (5)$$

Molnar [3] used PDP to visualize how the number of rented bikes depends on the temperature, humidity, and wind speed, shown in Figure 1. In this dataset, PDP plots communicate that the number of rented bikes increases as the outside temperature rises, but this dependency is only present up to a point, after which the number of bikes rented plateaus and later declines.

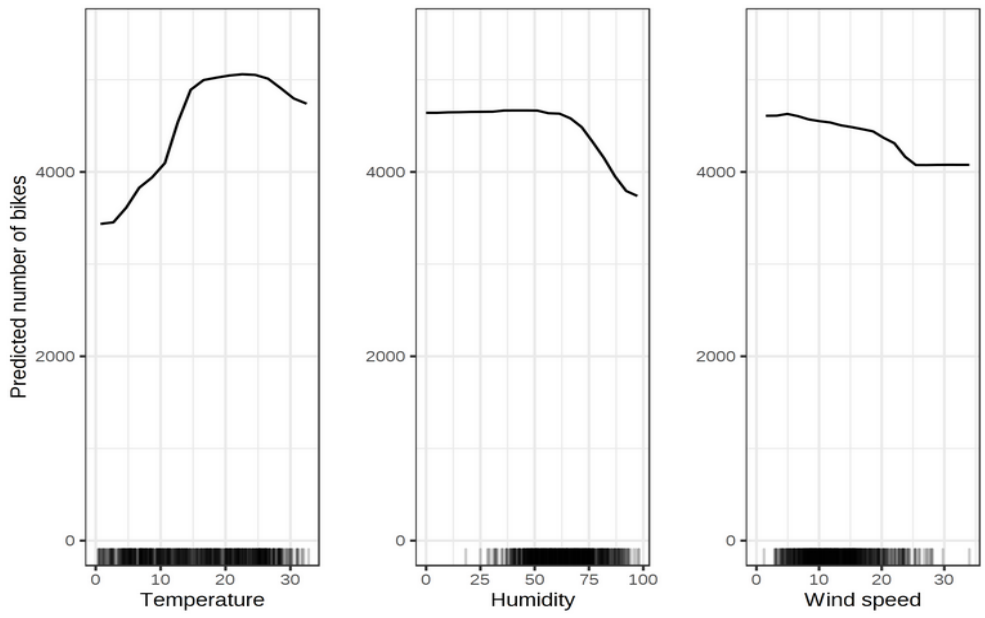


Figure 1: PDP on the number of bikes dependence on temperature, humility, windspeed [3]

PDPs could be used if the explainee is interested in a few feature effects on the predicted outcome.

PDPs have the following two advantages. Firstly, the explanations are deterministic, which means that given the same dataset, the model produces the same explanation every time. Secondly, the PDP calculations for categorical features can be relatively inexpensive because there are usually far fewer categories than training instances.

PDPs have the following disadvantages. Firstly, the number of features that can be visualized is limited to at most three because humans cannot visualize in more than three dimensions. This is a significant disadvantage, and therefore, the usage of PDPs on image datasets is not feasible. Secondly, PDP assumes that the features are independent, meaning that the change in one feature does not correspond to a change in any other feature.

2.2.2 Sharply values

Sharply values were introduced by Sharpley [9] to attribute payouts to players depending on their contribution to the game. The Sharply value for player i can be calculated using Equation 6, where $|N|$ is the number of players, $|S|$ the number of players in coalition S, $v(S)$ the total expected sum of payout the players in S can obtain by cooperation. [9]

$$\varphi_i = \sum_{S \subseteq N \setminus \{i\}}^{n} \frac{|S|!\,(|N| - |S| - 1)!}{|N|!} \cdot [v(S \cup \{i\}) - v(S)] \qquad (6)$$

Štrumbelj and Kononenko [10] used the Sharply values idea to allocate importance to features, treating each feature as a player and the difference between the correct value and predicted value as the payout. Calculating the Sharply value for each feature using Equation 6 would be very expensive because the number of collisions is an order of $O(2^N)$, where n is the number of features. This is especially true for images where a single one-megabyte image (1024 x 512) would have over 524 thousand features, resulting in $2^{(1024\times512)}$ collisions. Therefore Equation 7 is used instead as an approximation, where $\varphi_m(x)$ approximates how the prediction of x depends on the i-th feature. $\varphi_m(x)$ is the sum of all marginal contributions over M samples, which is calculated as a difference between model predictions, where in one case, y_i^+ feature i is present, and in another case, y_i^- feature i is not present. Instead of using raw instance x, a new variable y, a permutation of x features, is used to minimize the effect of correlations between features. In Equation 7, z is used as a random instance of X. However, the ordering of the features in z is the same as in y because it is required to ensure that all features are present only once when calculating the model output value.

$$\varphi_i^*(x) = \frac{1}{|M|}\sum_{m=1}^{M}\varphi_m \qquad (7)$$

$$\varphi_m = f(y_i^+) - f(y_i^-)$$

$$y_i^+ = (y_1, y_2, \dots, y_{i-1}, y_i, z_{i+1}, z_{i+2}, \dots, z_p)$$

$$y_i^- = (y_1, y_2, \dots, y_{i-1}, z_i, z_{i+1}, \dots, z_p)$$

$$y = Per(x)$$

$$z \sqsubseteq X$$

Sharply values can be used for any model to express the feature's importance and effect. Ayub, Yang and Zhou [11] used Sharply values to quantify each feature's effect in predicting the trust people place in autonomous vehicles (AV). The three most important prediction features, ranked by Sharpley values, are shown in Figure 2.A. They are the benefits the AVs bring, the associated risks, and the general excitement about them. Sharply values can also indicate if the values contributed negatively or positively to the classification, as illustrated in Figure 2.B.

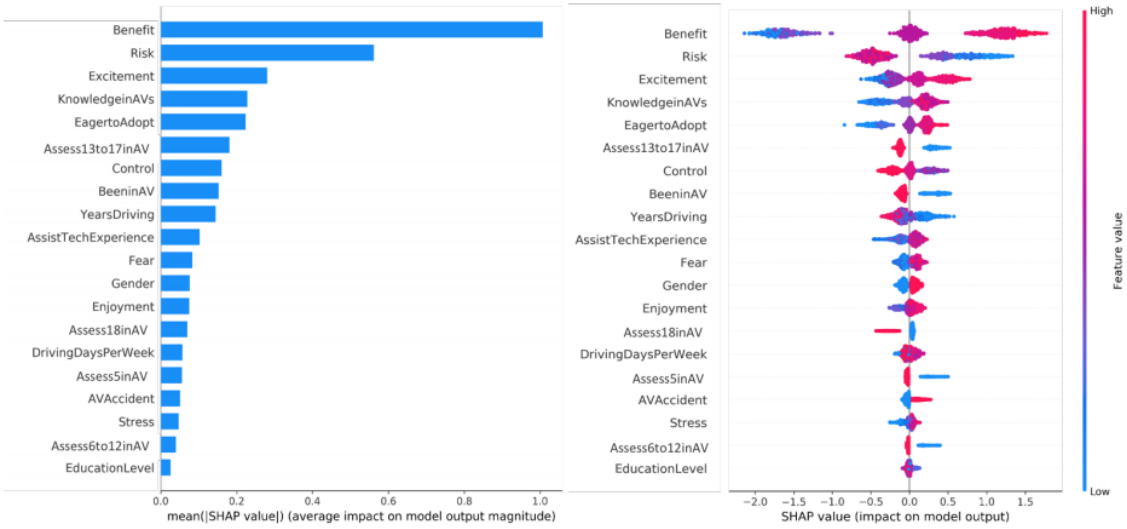


Figure 2.A: Sharply feature importance [11] Figure 2.B: Sharply value effect, by feature [11]

Sharpley values are a great explanation tool when the explainee is interested in an individual feature's impact on the prediction. In action, they can also show what features contributed positively to the prediction and negatively. This is especially powerful when dealing with a binary classification task.

Sharply values have the following two advantages. Firstly, Sharply values are fair, meaning that they reflect proportionally how much each feature contributed to the prediction/loss, which neglects the possibility of showing some features as more important than they are. Secondly, if the Sharply value is not approximated and is calculated on the same dataset, then given two identical instances will result in two identical Sharply value vectors.

Sharply values have the following three disadvantages. Firstly, they are computationally expensive, which makes them unusable on real-world datasets, when computation speed and cost also need to be considered. Therefore Sharply values are usually approximated. Secondly, Sharply values do not communicate the relationship between the target variable and the feature, but rather the relative importance of the features, which can result in misinterpretation. Thirdly, Sharply values assume that all features are used in the explanation because there cannot be any "profit" that is not allocated. One way to neglect the third disadvantage is using the "other" feature, a synthetic feature that attempts to capture everything not covered by the requested feature subset.

2.2.3 LIME

Local Interpretable Model-Agnostic Explanation (LIME) is an interpretability method, first introduced by Ribeiro, Singh and Guestrin [12], that attempts to explain the models' predictions by training a local surrogate model. This local surrogate model is a transparent model trained to predict the black box model. Because of this, the black box model can be interpreted by interpreting the surrogate model.

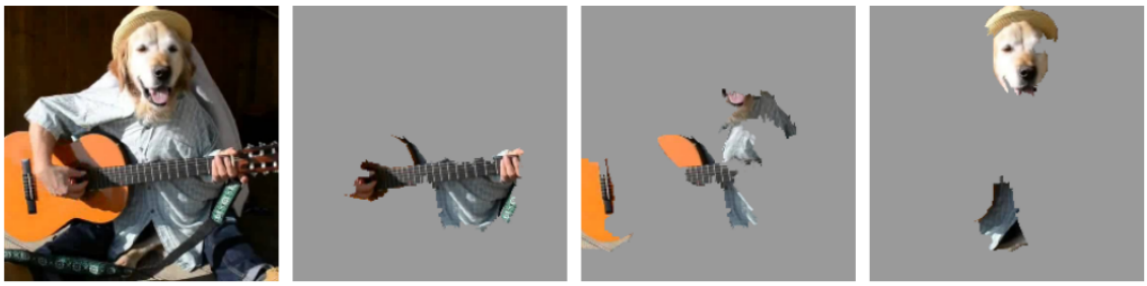
LIME wants to minimize locality-aware loss $L(f, g, \pi_x)$, where f is the machine learning model that needs explaining, g a transparent model, and π_x a proximity measure from instance x to another instance z. $L(f, g, \pi_x)$ measures how unfaithfully g approximates f in a locality defined by π_x [12].

Assuming that the explainee has chosen instance x that needs an explanation, the number of features used in lasso regression, and a model f that can predict the value for x, LIME works as follows. For every testing sample i, the algorithm draws a sample z_i' from the testing data using a uniform distribution, calculates the predicted value for z_i' using model f, and calculates the exponential kernel between z_i' and x using distance function D. Ribeiro, Singh and Guestrin [12] used Euclidean distance for tabular data, cosine distance for text data, and L2 distance for images. After this, LIME trains a lasso-regression function using k features, the set of z as instances, $f(z)$ as labels, and π_x as initial weights. After training the K-Lasso model, LIME uses its weights as a feature importance indicator. An illustration of the LIME algorithm is visible in Figure 3.

Algorithm 1 Sparse Linear Explanations using LIME

Require: Classifier f, Number of samples N
Require: Instance x, and its interpretable version x'
Require: Similarity kernel π_x, Length of explanation K
  $\mathcal{Z} \leftarrow \{\}$
  for $i \in \{1, 2, 3, ..., N\}$ **do**
    $z'_i \leftarrow sample\_around(x')$
    $\mathcal{Z} \leftarrow \mathcal{Z} \cup \langle z'_i, f(z_i), \pi_x(z_i) \rangle$
  end for
  $w \leftarrow$ K-Lasso$(\mathcal{Z}, K)$ ▷ with z'_i as features, $f(z)$ as target
  return w

Figure 3: LIME algorithm [12]



(a) Original Image (b) Explaining *Electric guitar* (c) Explaining *Acoustic guitar* (d) Explaining *Labrador*

Figure 4: LIME explanation [12]

Ribeiro, Singh and Guestrin [12] chose to use LIME on Google's pre-trained Inception neural network by choosing a random image (Figure 4. a). Algorithm 1 returns a set of weights, which communicate the importance of each pixel. However, to visualize the results more effectively, they choose to show the image's superpixels, which share common characteristics of the top 3 classes. The rest of the pixels were grayed out. Figure 4. b shows the pixels the Inception network used to predict this image as an "electric guitar" in Figure 4. b. The explanation increases the network's authority because the superpixels map to the guitar's fingerboard.

LIME could be used when the user is interested in using a single explanation framework for many types of models, from images to tabular data.

LIME has two advantages. Firstly, it can be applied to any data type, such as tabular and images, making it usable in different project settings. Secondly, LIME can be configured by the explainer, allowing them to verify that different hyperparameter combinations yield similar results.

LIME has four disadvantages. Firstly, as White and Garcez [13] demonstrated, LIME does not measure the fidelity of its regression. Therefore, LIME can produce misleading

explanations. Secondly, using LIME may require substantial time investments. Explaining the prediction of a simple random forest can take three seconds on a laptop while explaining the Inception network's prediction for image classification can take ten minutes on the same device [12]. Thirdly, the explanations produced by LIME are not deterministic, caused by the inherent randomness in the dataset selection process. This means that if the explainee is interested in interpreting the model's prediction, given the same input, n times, then the explainee receives n different, and potentially contractionary, explanations. Finally, LIME explanations depend on the selected hyperparameters, such as the used distance function, allowing the explainer to achieve more suitable explanations for their goals.

2.2.4 Global Surrogate

The global surrogate method is an explanation method that attempts to employ a transparent model to approximate the predictions produced by the uninterpretable model. As the name suggests, global surrogate models do not focus on a single instance but the instances in general. Surrogate models are used to approximate the black box model if the computation on the model is expensive, i.e., requiring a significant number of elementary operations, or slow, i.e., requiring numerous IO operations.

Training the surrogate model is done as follows. The first step is to select two subsets, X^* and X^{**}, from the training examples. The second step is to compute the predictions y^* and y^{**}, given X^* and X^{**}, respectively, using the black box model M. The next step is to select a transparent model M^*, such as a linear model or a decision tree, that will be used to predict y^*. After M^* has been selected, it needs to be trained to predict y^*, given X^*. Now that the interpretable model has been trained, it should be used to calculate testing labels l, i.e. predicting labels using M^*, given X^{**}, so that the quality of the surrogate model can be measured, such as calculating the absolute loss, which is the sum of the absolute differences between vector l and vector y^{**}. Finally, the explainee can interpret model M's predictions by interpreting the transparent model M^*.

Global surrogate models could be used if the explainee is interested in effectively explaining the model as a whole and not an individual instance.

The global surrogate method has two main advantages. Firstly, it is not restricted to a particular transparent model, allowing the explanation provider to explore many different transparent models. Secondly, the global surrogate method allows us to calculate how good the

interpretability model is at predicting the black box model predictions by calculating the absolute loss between vector l and vector y^{**}.

Global surrogate models have three main disadvantages. Firstly, the model is designed to interpret the predictions of the black box model but not the labels. This means the surrogate model interprets the black box model but does not explain the general relationship between the features and the label. Secondly, the surrogate model inherits all the disadvantages in their chosen transparent model. Finally, the model favors global interpretability over individual explanations, which can result in an illogical explanation.

2.2.5 Counterfactual

Counterfactual explanations try to explain a machine learning model's prediction using hypothetical counterfactuals. For example, suppose the explainee is interested in discovering why their mortgage application was not approved. In that case, the model's prediction can be explained by telling them that their mortgage application would have been approved if their monthly income would be either $300 higher or the loan amount was $1000 lower. These "what-if" explanations can offer valuable and actionable insights as they provide a clear set of steps the explainee needs to take to reach their desired outcome.

Although many correct counterfactual explanations exist, some are better than others because they are either closer to the original instance or more actionable. For example, reducing the desired mortgage amount by $1000 might be more feasible than increasing monthly income by $300. Watcher, Mittelstadt, and Russell [14] suggest data controllers use “unconditional counterfactual explanations”, which are counterfactual explanations that achieve the desired outcome with few changes to the original instance. In doing so, the explanation aligns with their interest in not disclosing any trade secrets to competitors.

There are many ways to generate counterfactual explanations. A naïve approach would be randomly changing the instance feature values until the desired outcome is received. However, this is not the most optimal strategy because random changes might not reflect meaningful instance change. A more efficient approach would be to define a loss function between the current prediction y and the desired prediction y' and systematically work to minimize this function.

Mothilal et al. [15] introduced a model-agnostic method for generating counterfactual explanations called Diverse Counterfactual Explanations (DiCE). They sought to minimize a

combined loss function over all k generated counterfactuals using gradient descent and Equation 8.

$$C(x) = arg_{c_1,\ldots,c_k} min\ \frac{1}{k}\sum_{i=1}^{k} yloss(f(c_i), y) + \frac{\lambda_1}{k}\sum_{i=1}^{k} dist(c_i, x) - \lambda_2 dpp\_diversity(c_1, \ldots, c_k) \quad (8)$$

$$dist\_cont(c, x) = \frac{1}{d_{cont}}\sum_{p=1}^{d_{cont}} \frac{|c^p - x^p|}{MAD_p} \quad (9)$$

$$dist\_cat(c, x) = \frac{1}{d_{cat}}\sum_{p=1}^{d_{cat}} I(c^p \neq x^p) \quad (10)$$

$$dpp_{diversity(c_1,\ldots,c_k)} = det\ (K) = det\ (\frac{1}{1 + dist(c_i, c_j)}) \quad (11)$$

The first term, $yloss(f(c_i), y)$, aims to reduce the discrepancy between the desired label y and the model-predicted label $f(c_i)$. Instead of the more intuitive l_1-loss or l_2-loss, they opted to use hinge loss. They chose to do so because such losses penalize the distance between $f(c_i)$ and y too much, whereas, for a valid counterfactual, it is sufficient for the feature value to be close to a certain threshold rather than as close to the desired label y.

The second term, $\frac{\lambda_1}{k}\sum_{i=1}^{k} dist(c_i, x)$, aims to minimize the distance between the original instance feature values and counterfactual feature values. For continuous features, they used Equation 9, where d_{cont} is the number of continuous features, $|c^p - x^p|$ the l_1 distance between two feature values, and MAD is the median absolute deviation for that feature. They used Equation 10 for categorical features, where d_{cat} is the number of categorical features, and $I(c^p \neq x^p)$ equals zero if the two feature values are identical or one otherwise.

The third term, $dpp\_diversity(c_1, \ldots, c_k)$, uses Equation 11 and measures the determinantal point process diversity (DPP). This is accomplished by calculating the determinant of K, where K is the kernel matrix of a given counterfactual.

Finally, the loss function has two hyperparameters, λ_1 and λ_2 that balance the effects of different loss components. Mothilal et al. [15] found that setting λ_1 to 0.5 and λ_2 to 1 yields good results.

Counterfactual explanations could be used if the explainee is more interested in a particular label. In other words, if the explainee is not interested in why a particular image was classified as a "bedroom" but in what needs to happen for it to be classified as an "office".

The main advantage of this explainability method is the ease of interpretability. For the explainee, it is apparent what steps must be taken to classify the current instance counterfactually. The main disadvantage of this method is that it does not produce a single answer. Usually, many variables can be changed to achieve the counterfactual label, and there is no universally accepted method for choosing the "best" counterfactual explanation.

2.2.6 Concept-based

Concept-based models (CBM) aim to distill the workings of a model into abstract concepts the explainee is familiar with. To illustrate this, let us consider a black box model trained on tabular data to predict someone's credit score. This model, which can be composed of hundreds of features, can be explained to the bank's customer using everyday concepts such as "income", "age", and "number of credit cards". The model could also be explained using different, more industry-specific concepts, such as "credit utilization", making it more relevant to the teams within the bank. More formally, given an array of concepts, we can use one-hot encoding to note the presence or absence of these concepts. After the dataset has been transformed into one-hot vectors, we can use any other explainability method, such as previously discussed LIME or Sharply values.

Concept-based examples can also be combined with counterfactual examples. For example, instead of telling the explainee that to get a favorable decision on their loan application, they would need to increase their monthly salary by $300, concept "monthly salary over $3000" could be used. The loan application will be approved when their monthly salary is over $3000.

Unlike tabular data interpretation, which can rely on straightforward feature values, image interpretation requires indirect methods of interpretation due to the inherent high-dimensionality and contextual nature of images. The number of features in images is significantly larger than in typical data because each pixel constitutes as a potential feature. However, individual pixel values are rarely helpful or interpretable in isolation due to their dependency on the context provided by adjacent pixels. For that reason, concept-based explanations can be beneficial because they help to consolidate a large group of pixels meaningfully. For example, to explain an image containing a "cat" the framework could group 10% of image pixel values and call reference it as concept "ear".

The main challenge for concept-based methods is ensuring that the used concepts are semantically meaningful to the explainee. Marconato, Passerini, and Teso [16] tackled this problem by introducing GlanceNets, a new CBM that uses techniques from disentangled

representation learning and open-set recognition to achieve alignment. Their research found that the proposed technique achieved substantially better alignment than Concept bottleneck models (CBNM) [17]

Concept-based methods could be used if the explainee is interested in simplified explanations.

The main advantage of concept-based methods is the ease of interpretability. Assuming the explainee is familiar with all the concepts used for the explanation, it is relatively easy for them to understand the explanation and to judge whether the explanation is reasonable. The main challenge in concept-based image classification is the concept-mapping process. Firstly, annotating an image with custom concepts requires significant human resources. Secondly, the annotation process can introduce errors because it cannot precisely be defined when one concept ends, and another begins.

3 Local Concept-based Interpretability framework

This chapter gives a high-level overview of the framework used to provide personalized concepts-based image explanations to a black box classification model. The sections in the chapter are used to describe the framework's main steps in detail.

The framework has four main steps and two optional ones. In the first step, the explainee chooses the image they would like to be explained. In the second step, the framework finds the closest images to it. In the third step, the framework extracts concepts from the closest images. In the fourth step, the framework proposes some concepts to the user that will later be used to explain the image classification. The fifth step requires the user to choose which explanation, either decision tree or counterfactual, they would like to see, and depending on that, the framework will generate it. In the sixth step, the explainee can refine the concept set by excluding some of the proposed concepts. If they choose to do so, the framework will update the explanation using the new concept set. All the previously stated steps have been visualized in Figure 5.

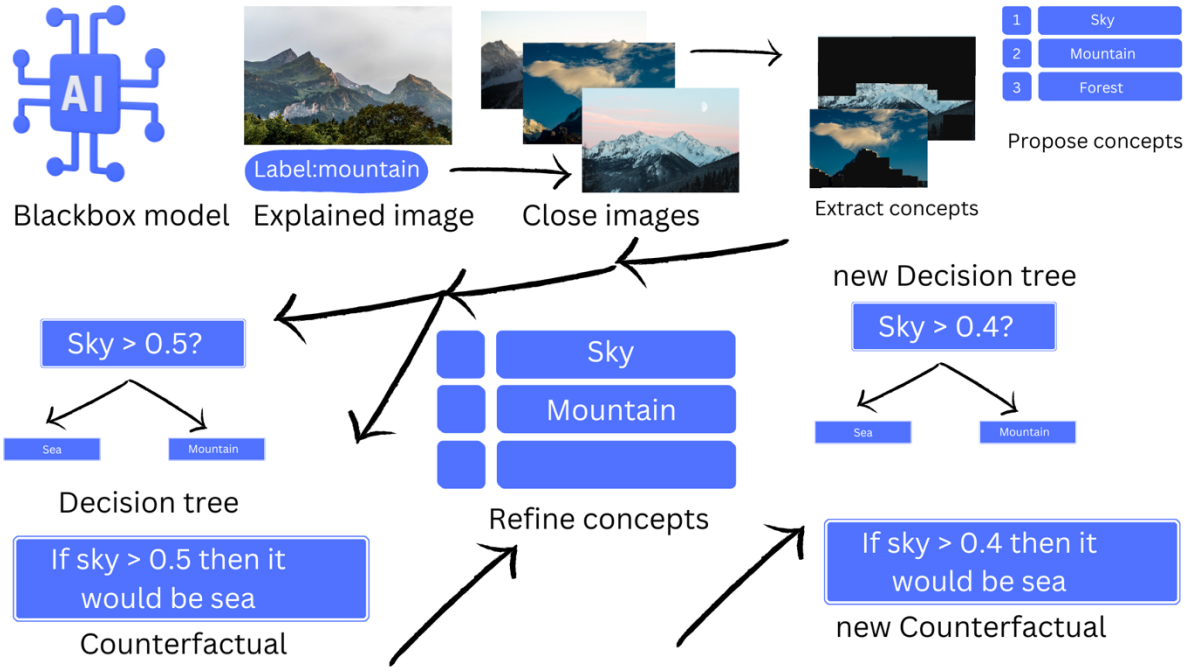


Figure 5: Local concept-based image explanation Framework.

3.1 Finding the closest image

After the user specifies the to-be-explained image, the framework finds the closest image in the training dataset. It does this by first calculating the Euclidean distance between the target image's Histogram of Oriented Gradients (HOG) and the HOG of all images in the dataset. Next each image's HOG vector is resized. The width is set to the minimum value between the

image and the target image width, and the height is set to the minimum value between the height of the image and the target height. The resizing is necessary because Euclidean distance can only be calculated between equally shaped matrices.

Next, the framework sorts the distances in ascending order and counts the occurrence of each label among the k ($k = 8$) closest instances. The label with the highest occurrence is recorded as the most popular label.

Finally, the algorithm selects the first image in the training set with that label as the synthetic instance the user wants to explain.

3.2 Extracting concepts

The next step in the framework's process focuses on extracting all concepts from the closest images. This step is quick because all concepts were extracted from images before the first explanation to improve the framework's performance at explanation time. To achieve this, the framework uses DeepLabV3+ [18] to extract the segmentation masks from the image. The DeepLabV3+ model was chosen because it demonstrated the best performance compared to other examined models. To ensure precision in the concept extraction process and reduce noise, the framework applies an additional filter: only those segments that constitute a minimum of $p(p = 5)$ percent of the entire image are considered valid concepts. This thresholding was set after exploring various thresholds, allowing the framework to discard minor, potentially irrelevant segments, focusing only on the most significant.

3.3 Proposing concepts

The next step in the framework's process is concept suggestion. Initially, the algorithm proposes concepts most frequently present in label l^*. For example, for the label "bedroom" the framework will probably suggests concepts "bed", "chair", and "windows", because they are common in many bedroom images, instead of "TV", which might be more intuitive to a particular explainee.

Selecting the initial concepts is critical because it only considers them when generating the decision-tree-based explanations, even when using different concepts would lead to a better-performing model. However, the algorithm may still use other concepts for counterfactual explanations because it will also use concepts from the counterfactual class.

Non-initial concept suggestion for decision tree-based explanations and counterfactual explanations works as follows. Firstly, the algorithm identifies a set of predictive concepts by arranging the decision tree features in decreasing order based on their local explanation's Gini

index. Secondly, the algorithm finds a set of intuitive concepts. In the case of decision tree-based explanations, the algorithm finds all concepts found in images with the label l^* ordered by their popularity. For counterfactual explanations, the framework finds intuitive concepts for both the original label l^* and the counterfactual label c^*. Subsequently, the algorithm excludes any predictive or intuitive concepts currently displayed to the user. Lastly, it merges both predictive and intuitive concepts. The concepts are combined by alternating between predictive and intuitive concepts, starting with predictive concepts. A predictive concept is proposed if a) it has not been already proposed and b) fewer than y ($y = 5$) predictive concepts have been proposed. An intuitive concept is proposed if a) it has not been already proposed and b) fewer than z ($z = 5$) intuitive concepts have been proposed. If the total number of proposed concepts is less than n ($n = 10$), then the remaining concepts are randomly proposed from the pool of available concepts.

3.4 Decision tree explainers

This section describes how concept-based decision trees are used in the image explanation. Those decision trees aim to explain the black box model using the concepts chosen by the explainee, and it has three parts: data pre-processing, training, and explaining.

In the data pre-processing step, the image dataset is first transformed into a $n\ x\ l$ zero matrix M, where n is the number of images the black box model has classified, and l is the number of unique concepts the user has chosen. Secondly, concepts are intelligently one-hot encoded. Instead of storing the presence of the concept in the matrix as 1, the algorithm quantifies how much of the image the concept covers. For instance, let us assume the algorithm is currency processing image number five, and that concept "chair" is the third one in the ordered list of users' chosen concepts. If a $20\ x\ 30$ chair is present in a $100\ x\ 120$ image, we record the proportion of the chair in this image as $M[4][2] = \frac{20*30}{100*120} = 0.05$. This approach offers a more accurate representation as it distinguishes between images that, while different, could otherwise be encoded the same way. The sum of a single row in M can be greater than one because concepts can overlap. To illustrate, part of an image can belong to the concept of "plane" but this same area can also belong to the concept of "sky".

In the second step, the encoded image data is used to train a decision tree in the training step, where 80% of the data is used for training and 20% for testing.

Finally, the explanation is presented as a sequence of features accompanied by their labels. The order of these features is based on their local feature importance. Feature local

importance is calculated as its' normalized sum of the features' global importance. The Gini index is used to quantify Global feature importance. This sorting ensures that the most locally important features are highlighted first in the explanation.

3.5 Counterfactual explanations

This section describes how concept-based counterfactual explanations are generated. Those counterfactual explanations aim to explain the black box model using the concepts chosen by the explainee. The explanation process has three components: data pre-processing, discovery, and explaining.

The first stage is data pre-processing, where image feature values undergo intelligent one-hot encoding, a process detailed in section 3.4. Additionally, binary encoding is applied to the image labels, where the label is encoded as “1” if it belongs to the counterfactual class and “0” otherwise.

The next step is discovery, where the algorithm finds two suitable counterfactual examples. These examples are synthetic, meaning they do not necessarily need to exist within the training dataset to be valid counterfactual examples. User-specified concepts can have a value between zero and one, where zero corresponds to an image not containing that concept and one where the entire image is composed of this concept. The framework generates these counterfactual examples using DiCE [15] until a counterfactual example meets the minimum stopping probability, starting at probability 1 and continuing to 0.25. The loop starts with a minimum stopping probability of 1 and continues to 0.25. The minimum stopping probability tells the algorithm the minimum existence probability that the synthetic example needs to have to be considered valid. When the algorithm finds the counterfactual example, it returns it along with the minimum acceptance probability that was used to generate it. It returns an error if the framework fails to find a counterfactual explanation with a higher than 0.25 minimum acceptance probability.

The final stage provides the explainee with an explanation in two parts. In part one, the explainee can see the intelligent counterfactual encoding vector, and in the second part, they can see the difference between the counterfactual vector and the original vector. This allows the explainee to understand the new state that must be reached to get the counterfactual decision and the difference between it and the current state.

4 Experiments and Results

This chapter describes the experiment, and its results, used to evaluate the proposed framework. The chapter consists of four sections. The first section provides an overview on how the experiment was set up. The second section covers the experiment used to answer the first research question. The third section outlines the experiment used to answer the second research question. The final section of this chapter focuses on the experiment used to answer the third research question.

4.1 Experimental Setup

A subset of the ADE20K [19] image dataset was used for this experiment. It contains 1592 images, 1258 unique, from 32 different classes. The concepts contained within each image were also from the same dataset. ResNet-50 was used as the black box model due to its popularity for image classification tasks and the complexity of its decision-making process. The code for the experiment was written in Python 3.10 and has been made freely available on GitHub [20]. Seed 42 was used to always generate the same random numbers. The questionnaires that were sent out to the participants can also be found in the same GitHub repository.

4.2 Faithfulness experiment

The first research question explores whether the proposed concept-based explanation framework generates more faithful explanations than feature attribution methods. This research question was divided into the following steps.

A random 10% of images, or 159, were first selected from the training dataset. These images were then classified using the black box model. Following this, each of these images was explained using LIME. The explanations generated by the LIME explainer were then fed back into the black box model, with the model's outputs recorded as the LIME explainer's classification predictions. Subsequently, a concept-based explanation framework was used to train a decision tree on the entire dataset, excluding the initial 159 images. Then each of the training images was classified using the decision tree. Finally, the LIME explainer and the decision tree framework's fidelities were calculated.

$$F(A,B) = \frac{1}{n}\sum_{i=0}^{n}\begin{cases}1, & A_i = B_i \\ 0, & A_i \neq B_i\end{cases} \qquad (12)$$

Fidelity is calculated using Equation 12. Given two arrays, A and B, fidelity measure the ratio of instances in A that are the same in B. In this experiment, A always represented the predictions made by the black box model, while B would either be the predictions from the decision tree explainer or those from the framework.

Based on the results obtained from this random dataset, the fidelity for the LIME framework was 29.268%. Meanwhile, the fidelity of the concept-based framework stood at 82.759%. This result indicates that the proposed framework more closely replicates the predictions of the original black box model - thus proving itself to be more faithful to the original model.

4.3 Intuitiveness experiment

The second research question of this Thesis aimed to answer whether the proposed framework produces more intuitive explanations than LIME. Intuitiveness, in this context, refers to an explanation that the explainee can effortlessly understand.

Individual intuitiveness is a subjective measurement, and to quantifiably answer this research question, an experiment in the following stages was conducted. Firstly, ten images were chosen from the dataset. Secondly, the predictions were explained using both the framework and LIME. Subsequently, these explanations, paired with their corresponding images, were incorporated into a questionnaire presented to ten participants.

Figure 6: Image labeled as “triumphal_arch”

In that questionnaire, the participants were shown those ten images, and for each of them, they had to choose which of the proposed explanations, concept-based or LIME, was more intuitive

to them. The participants were shown images belonging to the following classes: “triumphal_arch, “golfcart”, “gas_pump”, “minivan”, “streetcar”, “cab”, “garbage_truck”, traffic_light, “cinema”, and “unicycle”.

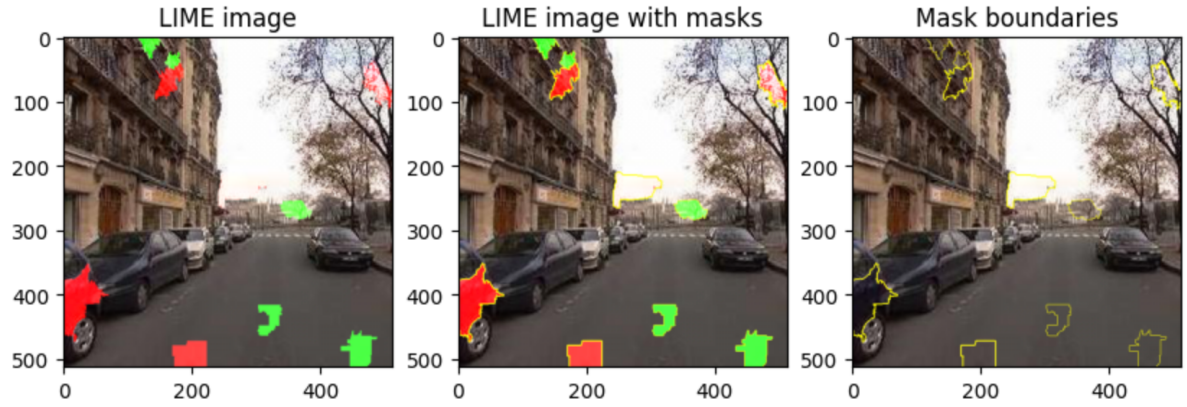


Figure 7: LIME explanation to why the image was labeled as “triumphal_arch”

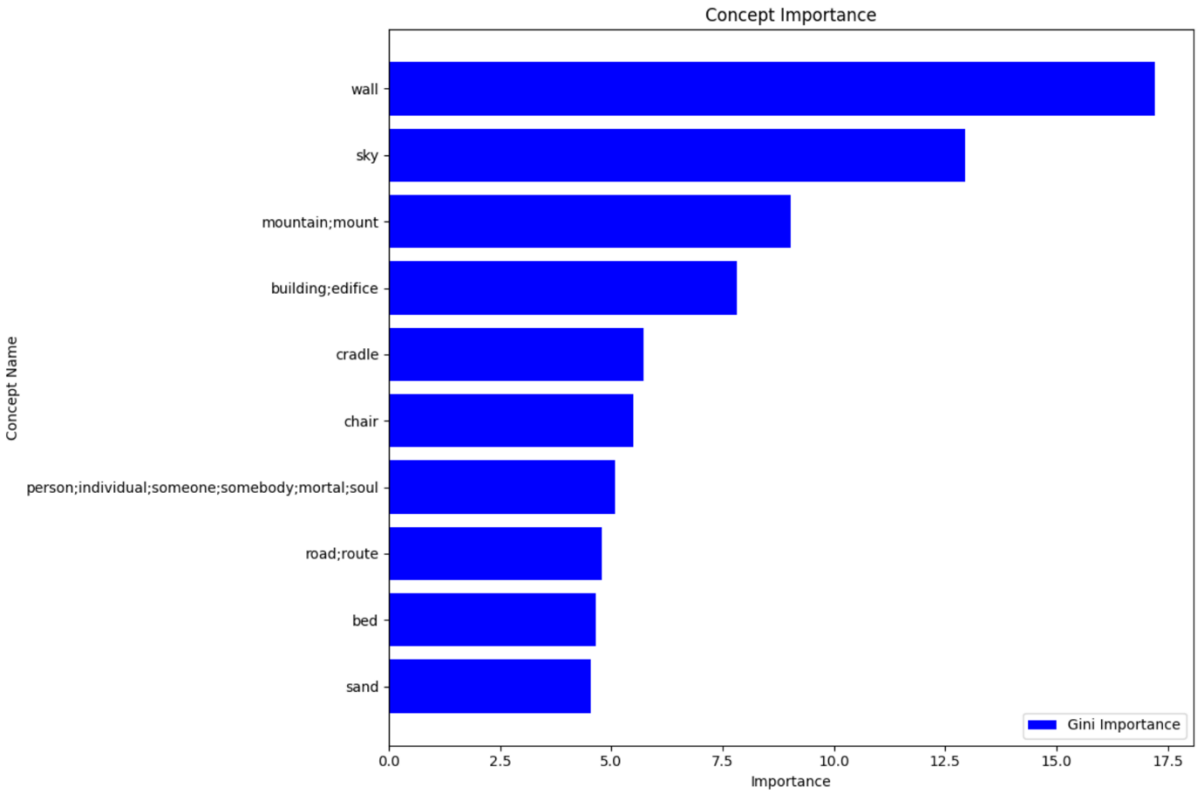


Figure 8: Framework explanation to why the image was labeled as “triumphal_arch”

The participants found the concept-based explanations more intuitive than LIME for six out of ten images. The concept-based explanations were seen as more intuitive because LIME, more often than not, highlighted seemingly random areas of the image. In contrast, the concept-based method showed the relative importance of well-known concepts. For example, for the image

shown in Figure 6 that was labeled by the black box model labeled as “triumphal_arch,” the participants preferred to see the explanation provided by the concept-based framework. As illustrated in Figure 8, the concept-based explanation shows the relative importance of important concepts such as “wall” and “sky”. LIME, on the other hand, as illustrated in Figure 7, only highlights a few seemingly random areas of the image.

Another reason the framework can produce more intuitive explanations than LIME is that it can emphasize the importance of two separate concepts when one is inside another. For example, it can show that the concepts “building” and “person” were equally important when classifying the image as “cinema”. LIME, however, will highlight the concatenated area, leaving the explainee guessing why this area is highlighted.

The concept-based method fails to perform sometimes because of the following reasons. Firstly, it needs to explain a wide array of prediction labels, from minivans to cinemas, and some of these labels are very far apart. For example, the concepts used to explain cinemas are probably very different from those used to explain cars. This can be improved by either using more training data so that the framework can more precisely learn the differences between different concepts or by implementing an ensemble method. This way, the framework would use more than one model, for example, a single model for a single label, making it more intuitive. This is particularly relevant when considering complex models like ResNet-50, which classifies images into more than 1000 classes. Explaining this model could benefit from a more nuanced, label-specific approach.

Secondly, the framework can sometimes propose concepts not present in the to-be-explained image, such as advocating the importance of the concept “mountain” even though there was not a single mountain in the image labeled as a minivan. However, it is not a far-fetched idea that a minivan image can contain a mountain, such as a picture from a competing holiday. This behavior happens because the framework defaults to using the most popular concepts for that label. Most of the effects of this phenomenon can be mitigated by improving the quality of the concept-mapping dataset, such as introducing lower-level concepts with precise boundaries.

4.4 Meaningfulness experiment

The third research question explored how meaningful the extracted concepts are. One of the first steps in the proposed framework is choosing what concepts will be proposed to the

explainee. The outcome of the explanation is greatly linked to it. Therefore, it is essential to validate that the people receiving the explanation also find the extracted concepts meaningful.

An experiment was performed to answer the third research question in the following stages. Firstly, ten random images were chosen. Secondly, a set of concepts was proposed for each of those images. When proposing concepts, the framework considered both performance and intuitiveness. Because of the format of the experiment, it was assumed that if concept i is the j-th most popular concept for label l, then it is also the j-th most intuitive concept for that label. In other words, it was assumed that concept's intuitiveness and popularity were the same. A maximum of four concepts were recorded for each of these images, with a maximum of two most predictive concepts and two most intuitive ones. If the label had less than four concepts, some concepts from that image were randomly chosen. Thirdly, for each image, the concepts the framework would have used in its prediction and the random concepts were recorded. In the experiment, the other concepts were not completely random because this way the framework's performance can be scrutinized more. In other words, it is much more valuable to study whether a participant who is shown an image of a bedroom can find the relevant concept "bed" from concepts "bed", "bookcase", and "chair" than from concepts "bed", "mountain" and "snow". Lastly, a questionnaire was sent out to ten participants where the participants were asked to identify the top concepts for in total of ten images. The results of this survey are presented in Table 1.

Table 1: Experiment three results: Image number and percent of participants who chose these concepts. If the image has less than four correct concepts, then the percentage is recorded as x.

	Concept 1 identified by %	Concept 2 identified by %	Concept 3 identified by %	Concept 4 identified by %
1	60%	100%	10%	x
2	80%	40%	30%	70%
3	100%	80%	10%	90%
4	100%	100%	0%	50%
5	70%	100%	30%	60%
6	100%	50%	100%	0%
7	40%	80%	0%	x
8	70%	100%	10%	80%
9	100%	100%	10%	10%
10	100%	20%	40%	80%

Image 3, illustrated in Figure 9.A, performed the best for the group of participants. The black box model labeled this image as “gas_pump”, and the framework would have used concepts “car”, “building”, “airplane” and “road” to describe this. The first, second, and fourth of these concepts could be used to describe an image containing a “gas_pump”. The author thinks that the third concept, “airplane”, is irrelevant because this image does not contain an airplane, a notion shared by the participants.

Figure 9.A: image 3 with label “gas_pump” Figure 9.B: image 7 with label “gas_pump”

Image seven, illustrated in Figure 9.B, performed the worst for the group. The black box model labeled this image as “gas_truck”, and the framework would have used the concepts “sky”, “building,” and “airplane” to explain this prediction while leaving out a highly relevant concept, “car”, a concept all of the participants chose as a relevant concept to this prediction.

The following approach was used to answer whether the framework can propose concepts better than just choosing concepts randomly present in the image. The framework can propose concepts better than just picking them randomly if the fraction of images where every concept used by the framework had a higher picking probability than random is greater than 50 percent. Participants were always given a choice of eight concepts, and for simplicity, it was assumed that each had a $\frac{1}{8}$ probability of being picked. The number of concepts that were picked at a higher-than-random probability and the total number of concepts the framework would have used are shown in Table 2.

Table 2: Experiment 3 results. Number of concepts that were chosen by more than $\frac{1}{8}$ of the participants, the total number of chosen concepts

Image nr	**Higher than random probability**	**Total number of concepts**
1	2	3
2	4	4
3	4	4
4	3	4
5	4	4
6	3	4
7	2	3
8	2	3
9	2	3
10	4	4

We cannot say that the framework can propose concepts better than choosing a random sample of concepts present in the image because only for 40% of the images were the participants able to recall all of the concepts used by the framework at a higher chance than at random.

It seems that there might be better approaches than choosing the most popular concepts because it is essential to ensure that the explanation does not indicate the importance of concepts that are not in the image.

5 Discussion

This chapter discusses some crucial aspects concerning the proposed framework. It has been broken down into two sections. The shortfalls of the current framework are listed in the first section, and some improvement areas are listed in the second section.

5.1 Limitations

The section discusses the limitations of the proposed framework. The current framework has five main limitations.

Firstly, using Euclidean distance, the proposed framework finds the k ($k = 8$) closest images to the to-be-explained image. Calculating Euclidian distance for the entire image data set is $O(n)$, where n is the number of images in the datasets. This time complexity is tolerable for small datasets. However, the distance function needs to be optimized for larger datasets or production environments to meet non-functional requirements, mainly real-time responsiveness.

Secondly, the framework assumes that the training dataset has an image similar to the to-be-explained image. This assumption can result in a significant shortfall if discrepancies exist between the training and testing datasets, potentially resulting in inadequate explanations.

Thirdly, the framework presumes that each image has concepts associated with it. This may be a costly and error-pruning assumption for any real-world dataset, requiring extensive human annotation and validation, thus limiting its practical applicability.

Figure 10: Experiment 2 image 5 with a label “streetcar”

Fourthly, the framework's reliance on Euclidean distance for similarity measurement can lead to biases and misinterpretations. For example, Figure 10 was mistakenly identified as a

“streetcar” due to bad lighting conditions and a minor similarity to a streetcar. Such errors highlight the limitations of a pixel-based comparison, which can be sensitive to transformations, lighting conditions, or other variations, restricting the framework's ability to generalize and adapt to different data types.

Fifthly, the framework's strategy of employing a single model to explain every possible decision represents another significant limitation. While this approach might be sufficient for simple classification tasks with few labels, it fails to live up to its potential when explaining complex models capable of classifying images into hundreds or thousands of unrelated classes. This "one-size-fits-all" approach can lead to oversimplification and loss of nuance in explanations, undermining the framework's utility in more intricate, multifaceted scenarios.

5.2 Future work

As evidenced in experiment one findings, the proposed framework can have a higher fidelity than LIME. However, there are areas where the proposed framework can be improved. This section discusses three further research areas to enhance the efficiency and effectiveness of the proposed framework. These three areas are improved closest image-finding process, using unsupervised learning for concept allocation, and using ensemble methods.

The current closest image-finding process has complexity $O(n)$. One way to make it faster is to reduce the number of classes the target image can belong to. For example, let us assume that there are four unique labels: "car"," bedroom"," living room" and "office". Also, let us assume the black box model classifies the uploaded image as "living room". An improved framework would ignore every image labeled as "car", because "car" and "living room" have relatively little in common compared to the "office" and "bedroom". This assumption would allow the enhanced framework to exclude $(n - l)$, where l is the number of closest categories considered. This has the potential to decrease the computation time significantly. However, we cannot be sure that the explanation will be as good as it is right now because without looking at every image in the training set, the algorithm cannot guarantee it will find the absolute n closest images. The author thinks that it is worthwhile to explore whether the reduced execution time is worth the potential performance impact.

Another way to improve the similar image-finding process is by using a different similarity algorithm. For example, exploring how the algorithm's performance would change if it relied on image embeddings instead of depending on Euclidian distance between the raw image values would be beneficial.

Figure 11: Quilt images

To illustrate this, let us calculate the Euclidian distance between the images shown in Figure 11 and the Euclidian distance between their embeddings. Images 11.A and 11.B are two distinct images labeled as “quilt” by the black box model, and images 11.C and 11.D their respective flipped copies. The outcomes of these calculations are shown in Table 2.

Table 2: Euclidean distance between different images and the between their embeddings.

	Image A	**Image B**	**Image C**	**Image D**
Image A		6410,572*	6786,510	1479,586
Image B	2337,572		0,557	3172,444
Image C	2436,510	0,557		3172,553
Image D	7751,586	5553,444	5553,553	

Using the image Euclidian distance method, the closest image to image one would be image D. Using the image embedding Euclidian distance method, the closest image to image one would be image C. This shows that using embeddings has the potential to be a more reliable similarity-finding method.

The framework currently depends on pre-annotated concept data, which limits the applicability of the framework. The applicability of this framework to other domains could greatly be improved if it automatically detects the presence of concepts in images. For example, an improved framework would use a foundational segmentation model, such as Segment Anything [21], to automatically extract relevant segments in images.

Lastly, exploring how to use multiple models in the explanation process would be beneficial. Currently, the framework trains a single model to explain a complex black box model that can potentially classify images into thousands of different classes, inherently sacrificing on quality. An improved version of this framework would employ multiple models, thus transforming itself into a multi-model framework.

The multi-model framework could be illustrated through the following example. Image classes could be divided into n groups based on the concepts with them or their inherit similarity of the labels. For example, images belonging to "bedroom" and "office" would be in group one, and images from "mountain" and "rock" would be in group two. This allows irrelevant images and concepts to be excluded from the training process. Every group could use whatever framework and hyperparameter configuration it needs to minimize its loss function. There also needs to be a process that knows how to map between the image label and image groups. Using this setup has the potential to unlock more intuitiveness and meaningfulness at an increased implementation and computation cost.

By improving the closest image-finding process, utilizing an automatic concept tagging process, and exploring a multi-model approach, this framework can potentially explain image classification model predictions even more powerfully.

6 Conclusion

This Thesis aimed to propose a novel explanation framework designed to explain the black box image classification model through human-defined concepts and to assess its effectiveness. This Thesis had the following three research questions:

1. Do concept-based explanations produce more faithful explanations than feature attribution methods?
2. Do decision trees produce more intuitive explanations than LIME?
3. How meaningful are the extracted concepts?

The Thesis used a sample of the ADE20K image dataset, containing 1592 images, 1258 unique, from 32 classes. The used black box model was ResNet-50.

Ten percent, or 159 images, were chosen for the first research question. These images were then classified using the black box model, LIME, and the concept-based framework. For this model and this dataset, LIME fidelity was 29.27%, and for the concept-based framework, it was 82.76%. This experiment proved that concept-based explanations can produce more faithful explanations than feature attribution methods.

For the second research question, human evaluation was performed. Ten people were asked to choose which of the explanations, concept-based or LIME, was more intuitive to them. This was repeated for ten images. On average, the participants found concept-based explanations more intuitive than LIME because the concept-based method gave them good high-level concept explanations. In contrast, LIME highlighted irrelevant areas of the image. However, LIME's performance was sometimes seen as more intuitive because the concept-based framework sometimes showcased concepts that were not present in the image. This experiment showed that the proposed framework could produce more intuitive explanations than LIME.

For the third research question, human evaluation was performed. Ten participants were asked to evaluate a set of ten images. They were asked to choose which of the proposed concepts they think are the most meaningful in explaining the image with a given label. For 40% of the images, the participants could identify all of the framework's chosen concepts with a probability greater than random. However, for the majority of images, they failed to do so. The current reliance on the most predictive and popular concepts may not be the optimal strategy, as it is crucial to avoid suggesting concepts that are not present in the image. This experiment's findings do not conclusively establish that the concepts extracted by the framework are more meaningful than randomly extracted ones. This conclusion is supported

by the fact that the framework-proposed concepts were identified with a higher-than-random probability for only four out of the ten images.

The framework has a few limitations:

1. It uses the Euclidian distance, which can be costly to calculate. In addition to that, it cannot capture the meaning of the images as it relies on raw pixel similarity.
2. The framework assumes that the training and testing dataset are homonymous.
3. The training process relies on pre-defined concepts.

These limitations can be mitigated by exploring the following research areas:

1. Using image embeddings as a similarity metric
2. Using a foundational model for segment extraction
3. Using a multi-modal approach

In conclusion, this Thesis successfully presented a novel explanation framework capable of explaining black box image classification models. The results were promising, showcasing its faithfulness and intuitiveness. However, it is essential to recognize that further work is needed to improve its performance, meaningfulness, and domain coverage.

References

[1] "General Data Protection Regulation Article 13," 2016.

[2] T. Miller, "Explanation in artificial intelligence: Insights from the social sciences," *Artificial Intelligence,* vol. 267, pp. 1-38, 2019.

[3] C. Molnar, Interpretable machine learning, 2023, pp. 34-36.

[4] J. A. M. Sidey-Gibbons and . C. J. Sidey-Gibbons, "Machine learning in medicine: a practical introduction," *BMC Medical Research Methodology,* no. 64, pp. 1, 15, 2019.

[5] G. Bansal, "What are the four assumptions of linear regression?," [Online]. Available: https://blog.uwgb.edu/bansalg/statistics-data-analytics/linear-regression/what-are-the-four-assumptions-of-linear-regression/. [Accessed 6 08 2023].

[6] A. F. Schmidt and C. Finan, "Linear regression and the normality assumption," *Epidemiol,* 2018.

[7] B. H. Menze, M. Kelm, R. Masuch, U. Himmelreich, P. Bachert, P. Wolfgang and F. A. Hamprecht , "A comparison of random forest and its Gini importance with standard chemometric methods for the feature selection and classification of spectral data," *BMC Bioinformatics volume,* vol. 10, no. 213, pp. 2-3, 2009.

[8] J. H. Friedman, "Greedy function approximation: A gradient boosting machine.," p. 1220, 2001.

[9] L. S. Shapley, "A value for n-person games," *Contributions to the Theory of Games 2.28 ,* pp. 307-317, 1953.

[10] E. Štrumbelj and I. Kononenko , "Explaining prediction models and individual predictions with feature contributions," vol. 41, pp. 655-657, 2014.

[11] J. Ayoub, X. J. Yang b and F. Zhou, "Modeling dispositional and initial learned trust in automated vehicles with predictability and explainability," *Transporation Research Part F: Psychology and Behaviour,* vol. 77, p. 15, 2021.

[12] M. T. Ribeiro, S. Singh and C. Guestrin, ""Why Should I Trust You?": Explaining the Predictions of Any Classifier," *The 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,* pp. 3-10, 2016.

[13] A. White and A. d. Garcez, "Measurable Counterfactual Local Explanations for Any Classifier," 24th European Conference on Artificial Intelligence, 2019.

[14] S. Wachter, B. Mittelstadt and C. Russell, "Counterfactual Explanations without Opening the Black Box: Automated Decisions and the GDPR," *Harvard Journal of Law & Technology, Forthcoming.,* vol. 31, p. 844, 2017.

[15] R. K. Mothilal, D. Mahajan, C. Tan and A. Sharma, "Towards Unifying Feature Attribution and Counterfactual Explanations: Different Means to the Same End," pp. 2-6, 2021.

[16] E. Marconato, A. Passerini and S. Teso, "GlanceNets: Interpretable, Leak-proof Concept-based Models," Advances in Neural Information Processing Systems, 2022.

[17] W. K. Pang, T. Nguyen, Y. S. Tang, S. Mussmann, E. Pierson, B. Kim and P. Liang, "Concept Bottleneck Models," Proceedings of the 37th International Conference on Machine Learning, 2020.

[18] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff and H. Adam, "Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation," 2018.

[19] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso and A. Torralba, "Scene Parsing through ADE20K Dataset," Institute of Electrical and Electronics Engineers, 2017.

[20] K.-G. Kallasmaa, "Experiment code and questionnaires," 11 08 2023. [Online]. Available: https://github.com/KGKallasmaa/master-thesis.

[21] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár and R. Girshick, "Segment Anything," Meta AI Research, 2023.

Appendix

I Proofs

Proof 1:

Theorem: given the following logistic regression function:

$$y = a_0 + \sum_{i=1}^{k} a_i x_i$$

$$p = S(y)$$
$$S(x) = \frac{1}{1 + e^{-x}}$$

The influence of x_i to the odds p is $S(a_i)$
Let us calculate value of p, if x_i increases by *1*

$$z = y - a_i x_i + a_i(x_i + 1)$$

$$p_+ = S(z)$$

Let us calculate the ration between p_+ and p

$$ratio = \frac{p_+}{p} = \frac{S(y - a_i x_i + a_i(x_i + 1))}{S(y)} = \frac{S(y + a_i)}{S(y)} = S(a_i)$$

Proof 2:

Given the desired number of synthetic instances x, and the rule reduction rate t, find the minimum number of times y the initial synthetic dataset size needs to be increased, so that the number of synthetic instances, after the instances removed by the reduction rate, is equal to the length of x.

The expected number of instances in the synthetic dataset after n iterations

$$\sum_{i=2}^{n} xy * (i-1) - txy(i-1)$$

If we set $n = 2$, and assume that $t \neq 1$, then the optimization problem would look as follows.

$$xy - txy = x$$
$$y \geq 1$$
$$y$$

$$y - ty = 1$$
$$y = \frac{1}{1-t}$$

If we assume that there's only one instance that satisfies the constraint, then the minimum y required is.

$$y = \frac{1}{1 - \frac{1-x}{x}} = \frac{x}{2x-1}$$

II License

Non-exclusive license to reproduce thesis and make thesis public

I, Karl-Gustav Kallasmaa, (author's name)

4 herewith grant the University of Tartu a free permit (non-exclusive license) to
reproduce, for the purpose of preservation, including for adding to the DSpace digital archives until the expiry of the term of copyright,
Personalized concept-based image classification explanation framework,
(title of thesis)
supervised by Radwa ElShawi, PhD.
(supervisor's name)

5 I grant the University of Tartu a permit to make the work specified in p. 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons license CC BY NC ND 3.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright.

6 I am aware of the fact that the author retains the rights specified in p. 1 and 2.

7 I certify that granting the non-exclusive license does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

Karl-Gustav Kallasmaa
11/08/2023