Evaluation of FEM and MLFEM AI-explainers in Image Classification tasks with reference-based and no-reference metrics

A. Zhukov, J. Benois-Pineau, R. Giot

Univ. Bordeaux, CNRS, Bordeaux INP, LaBRI, UMR 5800, F-33400
Talence, France
alexey.zhukov@etu.u-bordeaux.fr
jenny.benois-pineau@u-bordeaux.fr
romain.giot@u-bordeaux.fr

Abstract
The most popular methods and algorithms for AI are, for the vast majority, black boxes. Black boxes can be an acceptable solution to unimportant problems (in the sense of the degree of impact) but have a fatal flaw for the rest. Therefore the explanation tools for them have been quickly developed. The evaluation of their quality remains an open research question. In this technical report, we remind recently proposed post-hoc explainers FEM and MLFEM which have been designed for explanations of CNNs in image and video classification tasks. We also propose their evaluation with reference-based and no-reference metrics. The reference-based metrics are Pearson Correlation coefficient and Similarity computed between the explanation maps and the ground truth, which is represented by Gaze Fixation Density Maps obtained due to a psychovisual experiment. As a no-reference metric we use “stability” metric, proposed by Alvarez-Melis and Jaakkola. We study its behaviour, consensus with reference-based metrics and show that in case of several kind of degradations on input images, this metric is in agreement with reference-based ones. Therefore it can be used for evaluation of the quality of explainers when the ground truth is not available.

Contents
1 Introduction 2
2 State-of-the-art in Evaluation of explainers 3
3 Evaluted Explanation methods 5
   3.1 Feature Explanation Method (FEM) 5
   3.2 Multi Layered Feature Explanation Method (MLFEM) 7
   3.3 Gradient-weighted Class Activation Mapping (Grad-CAM) 8
1 Introduction

The current environment and pace of innovation in machine learning research leave little room for introspection and questioning. The most popular methods and algorithms for AI are, for the vast majority, black boxes. Black boxes can be an acceptable solution to unimportant problems (in the sense of the degree of impact) but have a fatal flaw for the rest. More than the final solution of a problem, we also need the reasoning and the factors taken into consideration to produce the result.

We can easily imagine a scenario where, the process to come up with the result, is more important than the result itself. For instance, for tools that are meant to assist humans in taking a decision, it is useless to have this tool producing a one dimensional guess to the answer. It is much more useful to point out the relevant factors in taking a decision.
We also can use explainability methods to assist us in creating better black box algorithms by an iterative process. First we create an initial algorithm, then peer into the inner working of this algorithm, we can identify shortcomings and naturally or artificially exploitable weakness, then finally tweak the model (or the training data) to fix this shortcoming.

This process can be particularly useful to detect when a network is over parameterized to a problem. For instance, when it is clear, through visualization, that some layers of the network are scrambling previously ordered information, there is an opportunity to simplify the network [10], [15]. Nowadays, the explainability of AI tools is a strongly researched subject [26] in various classification tasks. In particular a bunch of methods have been developed for explanation of image classifiers on the basis of Deep Neural Networks (DNNs). We can cite here Grad-CAM [23], LRP [18] which are probably the most popular benchmarks amongst others. These tools allow for identifying, in the input, the data which have contributed the most into prediction by classifiers.

The large variety of explanation methods proposed today have to be evaluated and compared between them. While the quality of classifiers is evaluated with usual metrics such as accuracy, recall, precision, f-Score on annotated validation and test datasets, the evaluation of the quality of explainers remains an open problem.

In [5], the evaluation of explanation maps obtained by DNN explanation methods is proposed. It consists in comparison of them with Gaze Fixation Density Maps (GFDM) obtained from humans who observed images in a given classification task. The comparison is realised with two largely used metrics for comparison of saliency maps: i) Pearson Correlation Coefficient (PCC) and ii) Similarity (SIM) [14]. The intuition behind this comparison is that the explainer is good in case of true positives if it shows the area in the image which attracted human attention, as Deep Neural networks are biologically inspired models and are supposed to imitate human brain in decision making. Other metrics have been recently proposed in [3].

In the present we evaluate recently proposed by us explanation methods FEM and Multilayered FEM (MLFEM) [1], [5] and the reference Grad-CAM [23] with the so-called ”stability metric” from [3]. We also study how this metric correlates with the GFDM-based metrics we previously proposed.

The remainder of this report is organized as follows. In section 2 we summarize the SOA in evaluation of explainers. In 3 we shortly review FEM and MLFEM explanation methods. In section 4 we explain the metric and the evaluation methodology. Results and discussion are presented in 7. Section 8 concludes this work and outlines its perspectives.

2 State-of-the-art in Evaluation of explainers

The first methods to evaluate explainers were purely qualitative and performed by humans [23]. The interpretation of the explainer results has thus been made by humans. A step ahead to the automatic evaluation of explainers has been recently made in [5] when comparing the explanation maps obtained by a method with Gaze Fixation Density Maps (GFDMs) computed upon gaze fixations of human observers. The latter were instructed to observe images with the same classification task in mind, which was required from a CNN classifier. If the classification result by a CNN is
correct, then the concordance of a human GFDM and a post hoc explanation map do correlate, it means that the explainer is of a good quality. The latter could be measured by metrics of comparison of saliency maps such as in [14]. Nevertheless, such an evaluation requires the availability of ground truth, as GFDMs.

Both, interpretation by humans or comparison with human interest expressed by Gaze Fixation density maps can be qualified as reference-based evaluation. While it is quite tedious to conduct human judgement experiment for evaluation of an explainer, the comparison of explanation maps with human GFDMs is quite realistic nowadays. Indeed, a large amount of publicly available databases with Gaze Fixations or their approximations in action-prehension paradigm do exist, such as Hollywood-2 [17], DHF1K [25], IRCCyN [4] (for video), or MexCulture [19], Salicon [11] and others.

It is a common knowledge that Deep NN (DNN) models are quite data dependent and cannot be applied without transfer learning on the data which do not follow the distribution on which the DNN was trained. In case of their explainers, we suppose that a good explainer is universal. It has to explain a Deep NN model trained on any data. Thus if its quality can be assessed on an available ground truth, it can be considered for other explanation tasks.

Another research trend is to design an evaluation strategy which quantifies the sensitivity of an explainer as of any other method to the perturbations in the data. In this case, to evaluate an explainer, human judgment or GFDMs, that is "a reference" is not needed. Thus we can qualify the metrics proposed in this case as no-reference. One group of these methods constitute the fidelity metrics [9]. These evaluation methods are based on the principle that if the perturbations are induced in the parts of input data highlighted as important by explainers, then the classification score will change. In which case the explainer can be considered as good. The metrics Deletion Area Under Curve (DAUC) and Insertion Area Under Curve (IAUC) proposed in [21] are based on this principle. DAUC tracks the changes in the classification score of the image where the areas are masked progressively accordingly to their importance in the explanation map from (highest score pixels to the full masking). The metric is computed as the area under curve of the class score of an image as a function of masked proportion of its pixels. The lower is this metrics, the better is the explainer, as masking of important (accordingly to the explainer) parts in the input should yield the decrease in initial class score. The computation of IAUC follows the opposite strategy. Here a strongly blurred image is considered first, then are added pixels accordingly to their importance in the explanation map. Higher is this metric better is the explainer.

Other metrics such as "Increase in Confidence" or "Average Drop" or "Average Drop in Deletion" have been proposed in [6]. All of them are based on the changes of the class score for a given image after it has been modified accordingly to the importance of pixels in explanation maps. In [9] the authors criticize DAUC, IAUC for the fact that they use only rank of the score and not its value and propose Deletion Correlation (DC) and Insertion Correlation (IC) metrics. They are computed as correlation coefficient between the difference of scores due to masking or adding details and the score of saliency of pixels masked/added progressively. All these metrics are based on the influence of perturbations in images accordingly to explanation maps, on the classification score.

The metric Sparsity, also proposed by the authors of [9] qualifies the distribution of importance scores in the explanation maps per se. The sparsity is computed as
\[
\frac{S_{\text{max}}}{S_{\text{mean}}}
\]
with \( S \) — saliency score of the pixels in the explanation map. The authors claim that higher sparsity makes the map more interpretable by humans as it is concentrated on a small amount of elements.

All these metrics are objective in the sense that they do not put the human in the loop (except sparsity) and have their reason to exist. Nevertheless, it is interesting to come back to the general approach in design of image processing and analysis algorithms, such as their stability with regard to the level of noise and or degradation on the input images. Hence we consider the stability of explanation maps with regard to the noise and degradation usual in digital images.

In his work, Bodria \[3\] cites various methods to test the stability of the explainers for black-box models. The method based on the Lipschitz constant, proposed by \[2\] seemed to us the most appropriate to measure stability of the explanation maps in case when the classifier is stable to the noise.

The main potential advantages of the method are that it is quite general, and therefore does not depend on the original design of the system, and is easy to implement. The method does not require the references, i.e. Gaze Fixation Density Maps, which makes it easier to implement. For its operation, it is enough to have a wide layer of control (initial) data and their corrupted versions. The creation of corrupted data also does not pose great financial and technical problems for the current capabilities of the industry and the scientific community. Thus, thanks to a more detailed study and a wider range of experiments, we have the opportunity to obtain a potentially high-precision and low-cost tool to test the stability of explainers. Furthermore, despite Bodria speaks about evaluation of ”black-box” explainers, the methodology is generic.

3 Evaluated Explanation methods

In this section we will describe the methods of explanation that we have used in our work.

3.1 Feature Explanation Method (FEM)

The essence of the method is the reverse tracking of the most high features from the last layer of features, namely from the convolution layer of a CNN classifier. It can be used to identify network solutions at the generalization stage. At the generalization stage, a fragment of the input images for classification is transmitted in the forward direction over the trained network. In CNN, convolution layers act as feature extractors, and the last fully connected layers act as their classifiers. The upper levels of convolution extract low-level features from the input data, while the deeper ones extract higher-level semantic features. Consequently, in the method features from the last layer of convolution are extracted and analyzed. The features are picked up after the activation layer and immediately before the fully connected layers. An overview of the explainer FEM is illustrated in figure 1 below.

For the video sequences the process is carried out in such a way that by pushing input chunks of \((W \times H \times T)\)-sized sequences of duration \( T \) and resolution \( W \times H \) through the convolution and pooling layers of the network, the input frames become
Figure 1: Overview of the explainer FEM. Features are extracted from the last convolutional layer after the activation function (upper part). Binary features maps are generated. Importance weights are calculated. Importance map is computed as a normalized linear combination with channel weights and visualized as a heat-map on the original image [1].

feature frames. Their number decreases by 2 times when combined. The method is applied to each feature frame, then the resulting \( N \).

After that, the algorithm generates a binary map for each channel of the feature tensor in order to assign an importance value for each feature by channels. In order to detect the strongest features, the developers of the method hypothesize that the feature values in feature maps follow Gaussian distributions independently by channel. From this we can conclude that the means are positive, since they extract features after the commonly used non-linearity ReLU, which converts negative values into 0.

On the last convolutional layers features are the most "vivid", but only some of them are of value. According to the Gaussian distribution hypothesis, the most valuable for studying is the correct distribution queue, which corresponds to rare but strong features. In this way, limits are created for maps of features of \( x_{i,c} \) in accordance with the \( K \)-sigma rule. Mean \( \mu_c \) and standard deviation \( \sigma_c \) are calculated for each channel \( c \). Then a binary map per channel is built which marks the strong features as in:

\[
b_c(R(x_i, c)) = \begin{cases} 
1 & \text{if } x_{i,c} \geq \mu_c + K * \sigma_c \\
0 & \text{otherwise}
\end{cases}
\]  

After that, the importance map of \( M \) of the explainer is calculated as a linear combination of all binary channel maps \( b_c \) using the weights \( \mu_c \) of these channels with its subsequent normalization to \([0; 1]\). When normalizing, Min-Max feature scaling is used. Finally, the normalized importance map \( M' \) is upscaled to the original image/video.
frame dimension $W \times H$ by a bi-linear interpolation. An example of explanation maps obtained by FEM method is illustrated in figure 2 below.

![Figure 2: An example of FEM map: (a) Original image, (b) Superimposed with FEM](image)

### 3.2 Multi Layered Feature Explanation Method (MLFEM)

The Multi Layered Feature Explanation Method (MLFEM) method is based on the FEM method. While in FEM the analysis of activations is performed at the last convolution level only, MLFEM extends it to all convolutional layers of a CNN. Since each layer of CNN embeds information at a different scale, the authors of [5] suggest that calculating FEM at multiple layers and combining them will improve the quality of explanation maps.

In reality, the FEM presented earlier can be applied to any CNN layer. Applying FEM to each layer of CNN consisting of $L$ convolutional layers will give $L$ different maps of the importance of features. Since all importance maps are interpolated in the FEM method, we get $L$ input resolution maps as a result. The information provided by the maps depends on the convolutional layer in the network, and they need to be combined into a single pixel importance map.

![Figure 3: FEM applied on every convolutional block of a typical ResNet50 architecture. Resolution is higher for the first layers.](image)
The network passes the input image through several convolution layers, giving results independent of the position of these layers. They are designed to capture spatio-local features. With each step deeper into the network, the convolutional layer captures more and more abstract concepts in the image (see figure 3). The very first layers usually perform edge detection, while later ones extract abstract concepts such as “face”, “car”, etc. This is the rationale for the idea of repeatedly applying the same method of explanation on different layers of the network.

For combination of each of $l = 1, ..., L$ maps the authors of MLFEM method propose to train a shallow convolutional network which uses as the ground-truth Gaze Fixation Density Maps and an Euclidean Loss function [5]. They show that such a fusion method is model-agnostic. An example of pixel importance map obtained by MLFEM method is given in figure 4 for the same image as in figure 2. It can be seen that the MLFEM map is much more detailed and captures the important details in the image, such as wheels of the car in this case.

Figure 4: An example of MLFEM map: (a) Original image, (b) Superimposed with MLFEM

3.3 Gradient-weighted Class Activation Mapping (Grad-CAM)

Information about space is preserved naturally in convolutional features, but is lost in fully connected layers. It follows from this that we can find the best compromise between high-level semantics and detailed spatial information on the last convolutional layers. This is the assumption of one of the most popular explainers, the Grad-CAM method [23]. Neurons in these layers search for semantic information (part of an object) related to a specific class in the image. To understand the influence of each neuron for our solution, Grad-CAM uses gradient information coming into the last convolutional layer of CNN.

Gradients are set to zero for all classes except the desired class, which will be set to 1. Then this signal is transmitted back to the corrected maps of convolutional features of interest to us, which we combine to calculate the rough localization of Grad-CAM (blue heat map), which represents where the model we are testing should look, to make a certain decision. Finally, we dot-multiply the heat map using controlled back
Figure 5: Grad-CAM Overview: Given an image and a class of interest as input, we need to pass the image through a part of the CNN model, and then through calculations depending on the task we choose to get an initial estimate for the category. [23]

...propagation to obtain a controlled Grad-CAM visualization with high resolution and taking into account a certain concept. The overview of the method is schematized in figure 5. In figure 6 an example of Grad-CAM map is given. For the sake of visual comparison, in figure 10 we present an image and the result of human observation of it, the gaze fixation density map of several observers, and on in the figure 8 we show maps resulting from the three methods: FEM, MLFEM and Grad-CAM. We can see that the Grad-CAM map is less precise and covers a lot of background.

Figure 6: (a) Original image, (b) Superimposed with GRAD-CAM
4 Evaluation of explanation methods with stability metric

In this section we introduce stability metric for evaluation of explanation methods and present our methodology if their evaluation.

4.1 Stability metric

Stability metric for evaluation of black-box explanation methods was proposed in [3]. Despite the methods we wish to evaluate are the so-called white-box methods, that is the analysis of internal layers of the DNN classifiers is needed, we apply this metric as it is model independent. This metric is based on the Lipschitz constant.

Given an image space $X$, trained classifiers $C$ and explanation methods $D$, we consider the explainer function $f(x, C, D)$ as a mapping from $X \in R^{+2} \times C \times D$ to the metric space $E$ representing explanation maps. For the sake of simplicity, as we do not
change the classifier and the explanation method we will further denote an explanation map for the given image $x \in X$ by $e = f(x)$.

**Lipschitz mapping** is a mapping that increases the distance between arguments by no more than $L$ times, where $L$ is called the **Lipschitz constant** of the mapping. More formally, let us consider a metric space $X$ with the metric $p_X$ and a metric space $E$ with the metric $p_E$.

A mapping $f$ of a metric space $(X, p_X)$ to a metric space $(E, p_E)$ is called Lipschitzian if there is such a constant $L$ (the Lipschitz constant of this mapping) that $p_E(f(x), f(x')) \leq L \cdot p_X(x, x')$ for any $x, x' \in X$. This condition is called the **Lipschitz condition**.

According to the work of Bodria [3], the stability metric aims at validating how consistent the explanations are for similar inputs. The higher the value, the better is the explanation model to present similar explanations for similar inputs. Stability can be evaluated by exploiting the Lipschitz constant. In their work Bodria et al. propose to compute the maximal Lipschitz constant in a certain neighbourhood $N_x$ of the given data point $x$:

$$L_x = \max_x \left\| \frac{e_x - e_{x'}}{\|x - x'\|} \right\|, \forall x' \in N_x$$

(2)

where $x$ is the explained instance (our image), $e_x$ the explanation (saliency map) and $x'$ are the data similar to our data $x$.

The idea of using the Lipschitz constant is that if the image is corrupted by noise, then the neural network will be probably forced to look for other elements or objects in the image to classify it.

Therefore, when the norm of difference between the original image and the corrupted image grows, the norm of difference of explanation maps will grow also. Thus Lipschitz constant should not increase accordingly to the equation. 2. Thus if the explanation method is stable, then with the growing noise, Lipschitz constant will show a stable behaviour. We will explore it on well classified images and badly classified (because of the noise) images. This will give us several options:

1. The Lipschitz constant will show a drop (or no change), as well as stabilization with strong image noise, which is the correct result.

2. The increase of the Lipschitz constant indicates an error in the method of explanation.

In the following we remind the two metrics: PC and SIM which we previously proposed for explainers in a reference-based evaluation paradigm, by comparing the obtained explanation maps and the ground truth: Gaze Fixation Density Maps(GFDMs).

### 4.2 Metrics based on Gaze Fixation density maps

First of all we remind the definition of Gaze Fixation Density maps.
4.2.1 Gaze Fixation Density Map

A Gaze Fixation Density Map is a means to identify relevant part of an image to a human. The general principle of GFDM computation consists in conducting a psycho-visual experiment, when human observers observe visual content and their gaze fixations are recorded by an eye tracking device. Then on each fixation a Gaussian surface is computed, which scale parameter $\sigma$ is computed from the geometry of the experiment to represent the projection of the most sensitive retina area, the fovea, into the image plane. Summing up and normalising by maximum multi-Gaussian surface from different observers on the same image, the GFDM is obtained, this is the case of GFDMs used e.g. in [20] and available in referenced MexCulture Dataset. In SALICON [11] dataset ”gaze fixations” were obtained by mouse clicks using visual action anticipation paradigm. Some examples of GFDMs from SALICON dataset (see image example in figure 9 [11] are given in figure 10 below.

![Figure 9: Example of 5 images from SALICON dataset](image)

The GFDMs were used in [5] as the ground truth for comparing explanation maps with it with PCC, equation 3 and SIM, equation 4 metrics.

4.2.2 PCC and SIM metrics

In [5] evaluation of explanation maps was proposed by comparison with GFDMs. Two saliency maps comparison metrics were used: Pearson Correlation Coefficient (PCC) and similarity (SIM). These are two metrics used for measuring statistical correlation between two signals (PCC) and between distributions (SIM). In the first case the maps are considered to be stochastic signals, in the second case they are 2D distributions.

- Pearson Correlation Coefficient (PCC) is computed between explanation map and our available ground through, i.e. (GFDM) obtained in a psycho-visual experiment when humans observe source images in the same classification task; Pearson
Correlation Coefficient is defined as:

\[
corr(x, y) = \frac{\sum_u^W \sum_v^W (x(u, v) - \bar{x})(y(u, v) - \bar{y})}{\sqrt{\sum_u^W \sum_v^W (x(u, v) - \bar{x})^2} \sqrt{\sum_u^W \sum_v^W (y(u, v) - \bar{y})^2}}
\]  

(3)

- Similarity metric between the same maps; The similarity metric is defined as such:

\[
sim(x, y) = \sum \min(x(u, v), y(u, v))
\]

(4)

with \((u, v)\) being spatial coordinates.

### 4.3 Comparison methodology

The comparison methodology comprises several stages:

1. \(n\) noisy images with different noise level are generated from each source image with controlled distortions. We used additive Gaussian noise, Guassian blur, perspective deformations, uniform brightness distortions;

2. a pre-trained model (ResNet50) is applied to classify source and distorted images;

3. each of the explanation methods is applied to create a heat map for each source image and all generated distorted images;

4. the classification data obtained are divided into two groups: well-classified images (the labels of the original and noisy images are the same) and poorly classified (the labels did not match).

We then compute three metrics: i)Lipschitz constant for comparison of original heat map and the heat map of noisy images accordingly to the equation 2 (Stability metric), ii)PCC, iii)SIM

The mean value together with standard deviation are computed for each of three metrics Stability, (PCC), (SIM), over each set of images: well classified and badly classified as a function of the level of the noise. We track their behaviour as a function of distortions. Finally we will analyse the agreement between these three metrics by computing Pearson correlation coefficient between them. In the following sections the generation of images corrupted with different distortions is presented.

### 5 Generating of corrupted images with controlled distortions

In order to evaluate the influence of distortions on the explainers in image classification tasks, we select a variety of natural images and apply a set of parameterised distortions such as additive Gaussian Noise, Gaussian Blur, Uniform Brightness distortion and Perspective distortion.
5.1 Additive Gaussian Noise

We consider independent additive Gaussian noise. Thus each pixel value will be incremented by a randomly generated shift $u$.

$$I'(u, v) = I(u, v) + \alpha(u, v)$$

Here

$$\alpha(u, v) \sim \frac{1}{\sigma \sqrt{2\pi}} \times \exp\left(-\frac{t^2}{2\times\sigma^2}\right)$$

with $t$ is the independent variable, $\sigma$ is the shape parameter.

The same generated shift is applied for each color channel, but it is different for each pixel. Thus we consider i.i.d noise processes in each pixel. The effect is the appearance of black-and-white disturbances (noise). We parameterise the strength of the noise by the maximal absolute shift value and deduce the $\sigma$ parameter of our Gaussian from it.

To chose the maximal shift value $k$ we propose the following reasoning.

Lipschitz condition, see equation 2 is hold in a certain neighbourhood of the data point $x$. Let us denote the radius of this neighbourhood by $\epsilon$. Thus the norm of difference of the original image $x$ and corrupted $x'$ satisfies

$$\|x - x'\| \leq \epsilon$$

On the other hand

$$\|x - x'\| = \sqrt{\sum_{j=1}^{H} \sum_{i=1}^{W} (x_{i,j} - x'_{i,j})^2}$$

Here $x_{i,j}$ and $x'_{i,j}$ are the pixel values of original and corrupted image respectively in each color channel. Thus we can re-write our norm as

$$\|x - x'\| = \sqrt{\sum_{j=1}^{H} \sum_{i=1}^{W} \alpha_{i,j}^2}$$

Let us major $\alpha_{i,j}^2$ by $k^2$. Hence, from equation 6, our $k$ should satisfy :

$$k^2 \times H \times W \leq \epsilon^2$$

thus being in the range

$$-\frac{\epsilon}{\sqrt{H \times W}} \leq k \leq \frac{\epsilon}{\sqrt{H \times W}}$$

consequently, all $\alpha_{i,j}$ with $|\alpha_{i,j}| \leq |k|$ should be in this range too. In the following we will omit $i, j$. Let us now deduce $\sigma$ parameter for our Gaussian distribution for noise generation. Our maximal noise magnitude is $k$, applying ”two sigma rule” we can write

$$\sigma = k/2$$

Thus 95% of generated noise values will be less than $k$ in magnitude.

To generate Gaussian noise we will parameterise $k$, and compute $\sigma$ and thus will know the neighbourhood radius $\epsilon$ in equation 2 too. The algorithm of the generation of each noise value $\alpha_{i,j}$ is the following:

Algorithm

For each pixel $p = (u, v)$ of the original image $I(u, v)$

1. Generate $Z$ - a random number in the range from 0 to 1
2. Compute inverse $\alpha(u, v) = F^{-1}(Z)$ of cumulative distribution function $F$ of our Gaussian noise parameterized by $\sigma$, see equation 9, for $Z$

3. If $|\alpha(u, v)| \geq k/2$, then go to 1

4. Add $\alpha(u, v)$ accordingly the model of independent additive noise, equation 5 and crop according to channel range: $I''(u, v) = \min(255, \max(0, I(u, v) + \alpha(u, v)))$

End For each Pixel.

Due to the usage of two sigma rule in computation of our $\sigma$ parameter from $k$, the internal loops ("go to 1") are rare.

An example of images with generated noise for different $k$ parameters is given in figure 11 below. Higher the magnitude $k$ of the noise is, more the image is corrupted.

![Figure 11: Original image corrupted with Gaussian noise with different maximal shift parameter](image)

(a) (b) (c) (d)

Figure 11: Original image corrupted with Gaussian noise with different maximal shift parameter: (a) original image, (b) k=50, (c) k=125, (d) k=200

5.2 Gaussian Blur

The second distortion we apply is the Gaussian blur.

$$I'(u, v) = I(u, v) * g(\mu, \nu)$$

(10)

where $g(\mu, \nu)$ is the Gaussian filter kernel: $g(\mu, \nu) = \frac{1}{A} \times \exp \left( -\frac{\mu^2 + \nu^2}{2 \times \sigma^2} \right)$ with $A$ normalization factor and $*$ is a convolution operation. We vary the scale parameter $\sigma$ of the Gaussian filter and the size $s$ of the filter mask to generate the same number of corrupted images as for Gaussian noise distortion 5.1. The same filter $g$ is applied to three components R,G,B of colour images. We give examples of blurred images in figures 12, and 13 below.

![Figure 12: Original image (a) corrupted with Gaussian blur with the same $\sigma = 1.5$ and different kernel sizes](image)

(a) (b) (c) (d) (e)

Figure 12: Original image (a) corrupted with Gaussian blur with the same $\sigma = 1.5$ and different kernel sizes: (b) 5x5, (b) 7x7, (d) 9x9, (e) 11x11
5.3 Uniform Brightness Distortion

A uniform brightness distortion consists in adding a random shift $\beta$ to all three colour components of the image. The choice of $\beta$ value is realised randomly accordingly to the method described in 5.1 parameterised by the maximal shift magnitude parameter $k$. The new colour value in each channel is computed as

$$I'(u, v) = \min(255, \max(0, I(u, v) + \beta))$$

(11)

Examples of brightness distortion are given in figure 14.

Figure 14: Original image corrupted with Brightness with different maximal shift parameter: (a) original image, (b) k=50, (c) k=125, (d) k=200

5.4 Perspective Distorsion

Finally we apply the geometric distortion of image plane applying a perspective transformation $F : I'(u', v') = I(F(u, v))$, as it is the most general case, compared to e.g. pure zoom transformation. It is expressed as

$$u' = H \times u$$

(12)

in homogeneous coordinates, see [24] for more details. The eight parameters of the homography $H$ are defined by pairs of corresponding points in the source and target images $\{(u_1, v_1), (u_1', v_1')\}, \{(u_2, v_2), (u_2', v_2')\}$. Note we used here OpenCv$^1$ implementation of Homography estimation. To parameterise these transforms we use a trapeze figure with the bases parallel to the horizontal image border. Its upper base $\{(u_1, v_1)\}, \{(u_2, v_2)\}$

\[\text{https://docs.opencv.org/4.x/d9/dab/tutorial_homography.html}\]
is shorter than lower one \{(u_3, v_3)\}, ..., \{(u_4, v_4)\}. Then it is rotated three times by 90°. To get the target four points \{(u'_1, v'_1)\}, ..., \{(u'_4, v'_4)\} we scale the original coordinates of trapeze with a "zoom factor" \(l > 1\). The missing pixels values in the target image are bi-linearly interpolated. In this way we generate perspective distortion without holes in the target image which would yield parasite features when passing through a CNN classifier. Examples of distorted images are illustrated in figure 15 below.

![Distorted Images](image.png)

Figure 15: Original image (a) corrupted with Change of Perspective with one zoom \(l=10\) and different orientations of trapeze: (a) − > (b) = (c) top, (a) − > (d) = (e) bottom, (a) − > (f) = (g) left, (a) − > (h) = (i) right

6 The neural network for image classification

To test our explainers, we will be using ResNet50[22] for image classification, as a mean to find a balance between simplicity and performance for a more complicated dataset. The reference Grad-CAM [23], FEM [8] and MLFEM [5] will be applied to this network. ResNet50 contains 16 residual blocks, see illustration in figure 16, in MLFEM method we apply FEM to the output of each block, after the activation function. This will give us 16 different applications of FEM that we will merge together, see illustration in figure 17. The block of ResNet is the natural unit of choice to apply FEM. MLFEM being a white box method, we are free to choose the place of application. The goal is to maximize the different semantic meaning that can be extracted. In case of ResNet50 we take advantage of the structure of the network and apply FEM not at each layer but at each block. ResNet50 is trained with the Adam optimizer [13], with a cross entropy loss.
7 Experiments and Results

In this section, the results of the experiments will be presented. The three methods of explanation we consider, Grad-CAM, FEM and MLFEM, have been already evaluated in [5] with reference-based Similarity and Pearson correlation coefficient metrics. In our experiments we compute these metrics for them to assess the behaviour - agreement or not of a non-reference stability metric (Lipschitz constant) on a selected dataset. In the following the used dataset is described first and proposed generation of distorted data. Then, stability metric computation results are given. Finally, we assess the agreement of non-reference stability metric with PCC and SIM quality metrics on our images by computing correlation of the Lipschitz metric with PCC and SIM.
7.1 Original image dataset

The original image dataset consists of 50 images of the Salicon dataset [12]. We have retained this volume because of hardware constraints we have in our computation. Nevertheless, we should state that our software is fully parametrized and more large-scale experiment is seamlessly possible.

The Source SALICON dataset is composed of 15000 images 10000 being supplied with GFDMs. It is also itself a subset of another dataset, MS COCO [16]. MS COCO is a dataset containing images of 80 different categories of objects in context. As such, it is common to have multiple categories present for each image. The categories are not uniformly represented, humans for example are greatly over-represented, it is good to keep this fact in mind since it might affect results later.

To construct the GFDMs, the subjects have participated in psychovisual experiment with free viewing conditions, that is they were invited to ”look around” in the image and not searching for a particular object. Their gaze fixations were recorded as mouse clicks using psycho-motor paradigm of action anticipation by visual search as presented in [7].

We have retained 50 images with different categories in each. The presence of objects of more than one category in a source image is possible. Furthermore, all these 50 source images were correctly classified by our ResNet50.

7.2 Distorted image dataset

For each image from original dataset and for each of considered degradations (Additive Gaussian Noise, Gaussian Blur, Uniform Brightness shift, Perspective distortion) we have generated the same number of distorted images(40). Thus the whole image set for one degradation was 2000. In the following we present the degradation parameters for each considered degradation.

7.3 Experiment: Additive Gaussian noise

7.3.1 Input data and parameterisation

In our experiments we used the following data and parameterisation.

1. For each of 50 original images we generate five maximal shift values which give 95% of noise values accordingly to the two sigma rule, see section 5.1 (k): [25..200] with a step of 25.

2. The number of generated noisy images for each $k$ value is a parameter of our method $M$ ($M = 5$).

The number of corrupted images is thus 2000.

7.3.2 Results of the experiment

The behaviour of stability metric is expressed as the mean value and standard deviation of Lipschitz constant measured on 250 images for each value of parameterized maximal shift $k$. It is illustrated, for three explanation methods, in figures 18(a), 18(b),
18(c). We plot the mean value of $L$ over image set corrupted with Gaussian noise as a function of the level of the noise $k$. We can state that - generally with the increase of noise level Lipschitz constant for all the methods stabilizes, this confirms our hypothesis that higher is the noise, greater is the distance between explanation maps.

Furthermore, MLFEM method is the best in this sense as it is very much stable across generated dataset. It’s ±σ interval for any $k$ is tighter than for two other methods. The behaviour is similar on well classified and badly classified images.

We also studied the stability of Lipschitz constant, see table 1 as a function of the noise level. To do this we compute $s = \frac{|L_i - L_{i+1}|}{L_i} * 100\%$. We note that the best stabilisation ($s = 0.856\%$ for Well and $s = 7.847\%$ for Badly classified images) is observed for MLFEM method. We nevertheless stress that for high levels of noise $k = 175...200$, the saturation effects are stronger in the corrupted images, but still for the intermediate levels of noise MLFEM and FEM stability remains better than that one of GRAD-CAM.

We measured the same - mean and standard deviation for our references-based metrics: PCC and SIM computed for the three explanation methods as a function of noise. Their behaviour, see figures 18(d), 18(e), 18(f) and 18(g), 18(h), 18(i), is stable for different levels of noise, with a slight decline for very high levels of noise ($k$ starting from 175 approximately). We also studied the stability of PCC, table 2 and of SIM, table 3 as a function of the noise level. The behaviour is similar for well-classified and badly classified images. Hence we can conclude that these metrics are less sensitive to the noise level, and thus to the slight differences in explanation maps and GFDMs.

Consensus of metrics measured by Pearson correlation coefficient between them is presented in the table 4. It can be seen that, the no-reference stability metric demonstrates consensus with referenced-based metrics. Thus, the Lipschitz constant can be used to determine the quality of explainers even in the absence of ground truth (GFDMs).
Figure 18: Gaussian Noise: Behaviour of Lipschitz constant, PCC and SIM measures as a function of noise level: (a) FEM-Lipschitz, (b) MLFEM-Lipschitz, (c) GRAD-CAM-Lipschitz, (d) FEM-PCC, (e) MLFEM-PCC, (f) GRAD-CAM-PCC, (g) FEM-SIM, (h) MLFEM-SIM, (i) GRAD-CAM-SIM

| k     | FEM Well | FEM Badly | MLFEM Well | MLFEM Badly | GRAD-CAM Well | GRAD-CAM Badly |
|-------|----------|-----------|------------|-------------|---------------|----------------|
| 25 → 50 | 45.855% | 45.841% | 45.855% | 46.939% | 46.882% | 46.130% |
| 50 → 75 | 31.196% | 29.052% | 31.196% | 27.582% | 30.950% | 26.466% |
| 75 → 100 | 24.320% | 17.939% | 24.320% | 21.279% | 21.408% | 21.726% |
| 100 → 125 | 21.390% | 15.573% | 21.390% | 15.893% | 22.952% | 14.467% |
| 125 → 150 | 12.671% | 13.201% | 12.671% | 12.211% | 6.883% | 14.262% |
| 150 → 175 | 7.121%  | 11.171% | 7.121%  | 10.726% | 10.295% | 11.680% |
| 175 → 200 | 0.856%  | 8.903%  | 0.856%  | 7.847%  | 6.799% | 7.949% |

Table 1: Gaussian noise: Stability of Lipschitz constant for FEM, MLFEM, GRAD-CAM as a function of noise level
### Table 2: Gaussian noise: Stability PCC for FEM, MLFEM, GRAD-CAM as a function of noise

| k     | FEM Well (%) | FEM Badly (%) | MLFEM Well (%) | MLFEM Badly (%) | GRAD-CAM Well (%) | GRAD-CAM Badly (%) |
|-------|--------------|--------------|----------------|-----------------|-------------------|-------------------|
| 25 → 50 | 11.618%      | 5.524%       | 2.122%         | 14.132%         | 2.606%            | 5.132%            |
| 50 → 75 | **0.040%**   | 3.747%       | 2.827%         | 1.091%          | 6.030%            | 18.643%           |
| 75 → 100 | 11.420%      | 9.085%       | 5.510%         | 10.638%         | 24.461%           | 2.207%            |
| 100 → 125 | 7.594%      | 13.677%      | 18.515%        | **0.889%**      | 6.496%            | 4.589%            |
| 125 → 150 | 13.470%     | 7.523%       | 8.491%         | 17.356%         | 25.322%           | 2.645%            |
| 150 → 175 | 22.819%     | 10.756%      | 5.921%         | 2.387%          | 85.030%           | 19.792%           |
| 175 → 200 | 26.432%     | 3.987%       | 19.989%        | 9.803%          | 24.676%           | 12.224%           |

### Table 3: Gaussian noise: Stability of SIM for FEM, MLFEM, GRAD-CAM as a function of noise.

| k     | FEM Well (%) | FEM Badly (%) | MLFEM Well (%) | MLFEM Badly (%) | GRAD-CAM Well (%) | GRAD-CAM Badly (%) |
|-------|--------------|--------------|----------------|-----------------|-------------------|-------------------|
| 25 → 50 | 1.513%       | 1.179%       | 0.175%         | 1.662%          | 1.766%            | 0.133%            |
| 50 → 75 | 3.148%       | 0.842%       | 1.048%         | 0.930%          | 1.585%            | 2.108%            |
| 75 → 100 | 1.125%      | 1.997%       | 1.314%         | **0.191%**      | 0.496%            | 2.541%            |
| 100 → 125 | 0.609%     | 1.011%       | 0.173%         | 0.825%          | 3.862%            | 0.289%            |
| 125 → 150 | 1.140%     | 0.443%       | **0.118%**     | 0.490%          | 6.851%            | 0.265%            |
| 150 → 175 | 0.588%     | 0.412%       | 0.927%         | 0.750%          | 8.539%            | 0.337%            |
| 175 → 200 | 4.152%     | 0.691%       | 3.075%         | 0.550%          | 8.758%            | 0.066%            |

### Table 4: Gaussian noise: Pearson Correlation Coefficient value between different metrics: L-stability, PCC - Pearson Correlation Coefficient with GFDMs, SIM - similarity with GFDMs

|            | L -> PCC | L -> SIM | PCC -> SIM |
|------------|----------|----------|------------|
| Well FEM   | 0.81333871 | 0.80195250 | 0.97573735 |
| Well MLFEM  | 0.60197973 | **0.80978277** | 0.88800587 |
| Well GRAD-CAM | 0.41669108 | 0.46893878 | 0.78719912 |
| Badly FEM  | 0.78998537 | 0.74460457 | 0.93865378 |
| Badly MLFEM  | 0.53253558 | 0.78474459 | 0.80990318 |
| Badly GRAD-CAM | 0.59826123 | 0.69360754 | 0.67705911 |
7.4 Experiment: Gaussian Blur

7.4.1 Input data and parametrisation

In this experiment we used the following data and parametrisation.

1. For each of 50 original images we generate a corrupted image for each mask of size: (5x5), (7x7), (9x9), (11x11), see section 5.2 with scale parameter $\sigma$ values from $[1.25, 1.5, 1.75, 2, 2.5, 3, 3.5, 4, 5, 6]$.

2. The number of generated noisy images for each $\sigma$ value was taken as ($M = 4$).

Thus in our experiment we use the total number of 2000 of images corrupted by Gaussian blur.

7.4.2 Results of the experiment

The behaviour of explanation methods is illustrated in figures 19(a), 19(b), 19(c) in terms of stability metric $L$. Gaussian blur shows similar behaviour compared to Gaussian noise, but a wider range of standard deviation can be noticed. In this experiment, the MLFEM algorithm demonstrates the best stability results for the Lipschitz constant over the distorted database, see table 5. For comparison, in tables 6 and 7 figures for PCC and SIM metrics are given respectively.

Consensus of metrics is presented in the table 8. Based on the data obtained, it can be concluded that in the case of Gaussian blur, the no-reference stability metric demonstrates consensus with referenced-based metrics. Therefore, the Lipschitz constant can be used to determine the quality of explainers.
Figure 19: Gaussian Blur: Behaviour of Lipschitz constant, of PCC and of SIM measures as a function of noise level: (a)FEM-Lipschitz, (b)MLFEM-Lipschitz, (c)GRAD-CAM-Lipschitz, (d)FEM-PCC, (e)MLFEM-PCC, (f)GRAD-CAM-PCC, (g)FEM-SIM, (h)MLFEM-SIM, (i)GRAD-CAM-SIM

Table 5: Gaussian blur: Stability of Lipschitz constant for FEM, MLFEM, GRAD-CAM as a function of distortion level

| k     | FEM Well | FEM Badly | MLFEM Well | MLFEM Badly | GRAD-CAM Well | GRAD-CAM Badly |
|-------|----------|-----------|------------|-------------|---------------|----------------|
| 1.25 → 1.50 | 10.862%  | 9.975% | 10.862% | 9.301% | 9.598% | 12.052% |
| 1.50 → 1.75 | 11.060% | 1.870% | 11.060% | 3.076% | 8.711% | 1.666% |
| 1.75 → 2.00 | 7.219% | 11.292% | 7.219% | 6.514% | 3.407% | 8.990% |
| 2.00 → 2.50 | 9.816% | 11.751% | 9.816% | 10.007% | 5.988% | 6.521% |
| 2.50 → 3.00 | 9.822% | 0.580% | 9.822% | 2.568% | 9.239% | 2.231% |
| 3.00 → 3.50 | 0.426% | 4.170% | 0.426% | 3.825% | 1.328% | 4.732% |
| 3.50 → 4.00 | 0.613% | 0.811% | 0.613% | 0.535% | 0.910% | 1.119% |
| 4.00 → 5.00 | 3.582% | 1.365% | 3.582% | 0.637% | 3.206% | 0.358% |
| 5.00 → 6.00 | 5.125% | 0.723% | 5.125% | 0.602% | 1.864% | 0.576% |
### Table 6: Gaussian blur: Stability PCC for FEM, MLFEM, GRAD-CAM as function of distortion level

|   | FEM   | MLFEM | GRAD-CAM |
|---|-------|-------|----------|
|   | Well  | Badly | Well  | Badly | Well  | Badly |
| 1.25 → 1.50 | 4.635% | 5.674% | 3.572% | 14.866% | 0.110% | 29.992% |
| 1.50 → 1.75 | 2.652% | 9.656% | 2.878% | 6.285% | 3.989% | 4.746% |
| 1.75 → 2.00 | 13.083% | 20.896% | 2.228% | 9.690% | 5.766% | 7.102% |
| 2.00 → 2.50 | 15.262% | 18.384% | 9.143% | 11.352% | 2.455% | 18.775% |
| 2.50 → 3.00 | 2.886% | 15.628% | 3.688% | 4.833% | 14.585% | 10.505% |
| 3.00 → 3.50 | 1.404% | 1.177% | 0.310% | 1.171% | 2.751% | 1.545% |
| 3.50 → 4.00 | 2.765% | 4.112% | 4.905% | 2.844% | 2.075% | 3.865% |
| 4.00 → 5.00 | 1.658% | 2.075% | 5.285% | 1.397% | 5.395% | 6.529% |
| 5.00 → 6.00 | 7.317% | 19.055% | 5.470% | 3.615% | 2.933% | 20.196% |

### Table 7: Gaussian blur: Stability of SIM for FEM, MLFEM, GRAD-CAM as a function of distortion level

|   | FEM   | MLFEM | GRAD-CAM |
|---|-------|-------|----------|
|   | Well  | Badly | Well  | Badly | Well  | Badly |
| 1.25 → 1.50 | 1.510% | 0.525% | 1.510% | 0.103% | 2.009% | 2.022% |
| 1.50 → 1.75 | 3.863% | 0.001% | 3.863% | 0.832% | 2.950% | 0.636% |
| 1.75 → 2.00 | 1.450% | 2.534% | 1.45% | 0.669% | 1.572% | 1.013% |
| 2.00 → 2.50 | 2.442% | 1.72% | 2.442% | 1.316% | 0.366% | 3.004% |
| 2.50 → 3.00 | 1.111% | 1.121% | 1.111% | 0.383% | 2.275% | 1.443% |
| 3.00 → 3.50 | 0.426% | 0.599% | 0.426% | 0.589% | 0.468% | 0.996% |
| 3.50 → 4.00 | 0.848% | 0.543% | 0.848% | 0.376% | 1.540% | 0.008% |
| 4.00 → 5.00 | 0.075% | 0.801% | 0.075% | 0.206% | 0.129% | 0.589% |
| 5.00 → 6.00 | 1.243% | 0.961% | 1.243% | 0.383% | 0.412% | 1.121% |

### Table 8: Gaussian blur: Pearson Correlation Coefficient value between different metrics: L-stability, PCC - Pearson Correlation Coefficient with GFDMs, SIM - similarity with GFDMs

|   | L -> PCC | L -> SIM | PCC -> SIM |
|---|-----------|----------|------------|
| Well FEM | 0.27157465 | 0.6967906 | 0.742997 |
| Well MLFEM | 0.54681184 | 0.84820559 | 0.7448735 |
| Well GRAD-CAM | 0.37102968 | 0.71505468 | 0.65992497 |
| Badly FEM | 0.30078848 | 0.79844088 | 0.67166912 |
| Badly MLFEM | 0.61705396 | 0.87294826 | 0.76930315 |
| Badly GRAD-CAM | 0.27192725 | 0.81145439 | 0.57960577 |

25
7.5 Experiment: Uniform Brightness Distortion

7.5.1 Input data and parametrisation

In our experiments we used the following data and parametrisation.

1. For each of 50 original images we generate five maximal shift values which give 95% of noise values accordingly to the two sigma rule, see section 5.3 $(k)$: $[25..200]$ with a step of 25.

2. The number of generated noisy images for each $k$ value is a parameter of our method $(M)$.

Thus in our experiment we use the total number of images corrupted by constant (Gaussian) shift $M = 2000$.

7.5.2 Results of the experiment

This experiment demonstrates interesting results related to well-classified and badly-classified images. The behaviour of the Lipschitz constant is similar compared to previous experiments (Gaussian noise and Gaussian blur), see figures 20(a), 20(b), 20(c). But it is worth noting that with this distortion, some randomness of the final data is manifested, as the stability over distorted data is lower for all methods, see figures 20(d), 20(e), 20(f) and 20(g), 20(h), 20(i) and tables 9, 10 and 11. Perhaps this is due to the disappearance of objects of attention in the images with even not too strong distortion, nevertheless, the MLFEM method demonstrates itself as more reliable.

Consensus of metrics is given in the table 12. From these figures it can be concluded that in the case of Uniform Brightness Distortion, the consensus values are lower than in previous experiments, but high enough to use a non-reference stability metric in the absence of ground truth (GFDMs).
Figure 20: Uniform Brightness Distortion: Behaviour of Lipschitz constant, of PCC and of SIM measures as a function of noise level: (a)FEM-Lipschitz, (b)MLFEM-Lipschitz, (c)GRAD-CAM-Lipschitz, (d)FEM-PCC, (e)MLFEM-PCC, (f)GRAD-CAM-PCC, (g)FEM-SIM, (h)MLFEM-SIM, (i)GRAD-CAM-SIM

|      | FEM   | MLFEM  | GRAD-CAM |
|------|-------|--------|----------|
| k    | Well  | Badly  | Well     | Badly  | Well  | Badly |
| 25 → 50 | 38.171% | 45.046% | 38.171% | 39.673% | 46.882% | 46.130% |
| 50 → 75 | 30.754% | 33.71%  | 30.754% | 32.5%   | 30.950% | 26.466% |
| 75 → 100 | 23.058% | 10.23%  | 23.058% | 15.322% | 21.408% | 21.726% |
| 100 → 125 | 16.279% | 32.858% | 16.279% | 28.557% | 22.952% | 14.467% |
| 125 → 150 | 16.3%  | 16.552% | 16.3%   | 12.706% | 6.883%  | 14.262% |
| 150 → 175 | 12.952% | 31.773% | 12.952% | 27.942% | 10.295% | 11.680% |
| 175 → 200 | 12.815% | 0.391%  | 12.815% | 8.915%  | 6.799%  | 7.949%  |

Table 9: Uniform Brightness Distortion: Stability of Lipschitz constant for FEM, MLFEM, GRAD-CAM as a function of distortion level
| k   | FEM | MLFEM | GRAD-CAM |
|-----|-----|-------|----------|
|     | Well| Badly| Well  | Badly | Well | Badly |
| 25 → 50 | 6.288% | 0.656% | 3.947% | 11.957% | 8.209% | 32.55% |
| 50 → 75 | 7.745% | 34.037% | 13.94% | 8.888% | 6.201% | 14.435% |
| 75 → 100 | 5.401% | 29.962% | 7.859% | 18.492% | 17.683% | 25.974% |
| 100 → 125 | 4.188% | 28.433% | 1.851% | 3.919% | 0.535% | 60.425% |
| 125 → 150 | 9.446% | 3.867% | 4.324% | 15.757% | 11.102% | 14.383% |
| 150 → 175 | 2.139% | 15.828% | 4.407% | 15.857% | 13.889% | 16.699% |
| 175 → 200 | 1.133% | 7.095% | 5.081% | 6.257% | 1.187% | 0.488% |

Table 10: Uniform Brightness Distortion: Stability of PCC for FEM, MLFEM, GRAD-CAM as a function of distortion level

| k   | FEM | MLFEM | GRAD-CAM |
|-----|-----|-------|----------|
|     | Well| Badly| Well  | Badly | Well | Badly |
| 25 → 50 | 3.268% | 3.082% | 3.268% | 2.027% | 0.048% | 5.657% |
| 50 → 75 | 0.546% | 4.661% | 0.546% | 0.842% | 2.071% | 2.4% |
| 75 → 100 | 2.044% | 5.661% | 2.044% | 1.657% | 4.961% | 5.424% |
| 100 → 125 | 2.0% | 5.907% | 2.0% | 0.363% | 0.207% | 8.908% |
| 125 → 150 | 0.09% | 1.718% | 0.09% | 2.563% | 2.173% | 3.459% |
| 150 → 175 | 0.684% | 3.226% | 0.684% | 1.577% | 2.688% | 1.727% |
| 175 → 200 | 4.156% | 2.992% | 4.156% | 1.408% | 0.752% | 1.33% |

Table 11: Uniform Brightness Distortion: Stability of SIM for FEM, MLFEM, GRAD-CAM as a function of distortion level

| L -> PCC | L -> SIM | PCC -> SIM |
|----------|----------|------------|
| Well FEM | -0.00296529 | 0.17647169 | 0.4315179 |
| Well MLFEM | 0.05596645 | 0.23795837 | 0.28704762 |
| Well GRAD-CAM | -0.04494992 | 0.18619149 | 0.32821639 |
| Badly FEM | 0.41958505 | 0.59771389 | 0.93045171 |
| Badly MLFEM | 0.60064776 | 0.68983374 | 0.89775104 |
| Badly GRAD-CAM | 0.2570528 | 0.58340737 | 0.82306687 |

Table 12: Uniform Brightness Distortion: Pearson Correlation Coefficient value between different metrics: L-stability, PCC - Pearson Correlation Coefficient with GFDMs, SIM - similarity with GFDMs
7.6 Experiment: Perspective Distortion

7.6.1 Input data and parameterisation

In our experiments we used the following data and parameterisation.

1. For each of 50 original images we generate a corrupted image for each direction of the narrow part of the trapezoid: (top), (bottom), (left), (right), see section 5.4 $l$: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10].

2. The number of generated noisy images for each $l$ value is a parameter of our method ($M = 4$).

Thus in our experiment we use the total number of distorted images as 2000.

7.6.2 Results of the experiment

With this distortion, the Lipschitz constant shows lower values compared to other degradations even with minimal perspective changes in the images (figures 21(a), 21(b), 21(c)). A drop in the Lipschitz constant indicates an increase in distortion, which at some point in degradation scale can lead to stability of the constant (maps and images themselves are too different from the original ones). Indeed, in these degradations important areas in images "move". Thus the difference between the original image and the degraded one becomes stronger, thus stronger is the difference between the explanation maps. Indeed, explanation maps capture strong features which are situated on deformed and displaced borders of objects. Therefore, gaze fixation density maps differ a lot from good explanation maps on the distorted image. The stability of metrics is presented in tables 13, 14 and 15. From table 13 one can see that the Lipschitz constant does not change practically.

Consensus of metrics is presented in the table 16. Based on the data obtained, it can be concluded that in the case of Perspective Distortion, the no-reference stability metric also as with Gaussian noise or with Gaussian blur demonstrates consensus with referenced-based metrics. Thus, the Lipschitz constant can be used to determine the quality of explainers.
Figure 21: Perspective Distortion: Behaviour of Lipschitz constant, of PCC and of SIM measures as a function of distortion level: (a) FEM-Lipschitz, (b) MLFEM-Lipschitz, (c) GRAD-CAM-Lipschitz, (d) FEM-PCC, (e) MLFEM-PCC, (f) GRAD-CAM-PCC, (g) FEM-SIM, (h) MLFEM-SIM, (i) GRAD-CAM-SIM

|     | FEM |           | FEM |           | FEM |           |
|-----|-----|-----------|-----|-----------|-----|-----------|
|     |     | Well      |     | Badly     |     | Badly     |
| 1 → 2 | 3.129% | 1.339%    | 3.129% | 7.46%     | 4.61% | 4.574%    |
| 2 → 3 | 5.872% | 0.303%    | 5.872% | 0.423%    | 5.152% | 2.89%     |
| 3 → 4 | 2.81% | 0.449%    | 2.81% | 0.513%    | 2.511% | 2.797%    |
| 4 → 5 | 2.383% | 3.316%    | 2.383% | 7.133%    | 3.7% | 8.115%    |
| 5 → 6 | 3.603% | 9.616%    | 3.603% | 1.2%      | 3.132% | 4.003%    |
| 6 → 7 | 0.732% | 2.867%    | 0.732% | 5.909%    | 2.141% | 4.359%    |
| 7 → 8 | 2.059% | 0.578%    | 2.059% | 0.679%    | 0.036% | 2.529%    |
| 8 → 9 | 0.278% | 3.435%    | 0.278% | 3.71%     | 0.088% | 5.045%    |

Table 13: Perspective Distortion: Stability of Lipschitz constant for FEM, MLFEM, GRAD-CAM as a function of distortion level
| k     | FEM       |         |         |         |         |         |
|-------|-----------|---------|---------|---------|---------|---------|
|       | Well | Badly |         | Well | Badly |         |         |
| 1 → 2 | 2.968% | 6.585% | 3.88% | 21.392% | 2.87% | 12.984% |
| 2 → 3 | 2.327% | 13.287% | 6.559% | 4.207% | 0.653% | 10.636% |
| 3 → 4 | 4.086% | 0.115% | 2.966% | 10.811% | 2.908% | 0.507% |
| 4 → 5 | 2.896% | 13.596% | 6.042% | 5.009% | 16.893% | 34.549% |
| 5 → 6 | 23.819% | 10.252% | 2.075% | 8.726% | 1.476% | 23.611% |
| 6 → 7 | 10.889% | 1.785% | 5.306% | 2.514% | 15.983% | 15.313% |
| 7 → 8 | 10.149% | 3.718% | 1.228% | 2.493% | 2.577% | 2.677% |
| 8 → 9 | 2.027% | 6.381% | 10.644% | 7.168% | 4.849% | 3.175% |

Table 14: Perspective Distortion: Stability of PCC for FEM, MLFEM, GRAD-CAM as a function of distortion level

| k     | FEM       |         |         |         |         |         |
|-------|-----------|---------|---------|---------|---------|---------|
|       | Well | Badly |         | Well | Badly |         |         |
| 1 → 2 | 0.387% | 1.244% | 0.387% | 1.41% | 2.299% | 2.125% |
| 2 → 3 | 2.476% | 1.418% | 2.476% | 0.497% | 1.239% | 0.125% |
| 3 → 4 | 1.85% | 0.25% | 1.85% | 0.654% | 0.871% | 1.75% |
| 4 → 5 | 0.959% | 2.408% | 0.959% | 0.44% | 2.547% | 0.486% |
| 5 → 6 | 1.695% | 2.524% | 1.695% | 1.003% | 1.55% | 0.258% |
| 6 → 7 | 0.055% | 1.109% | 0.055% | 0.896% | 1.016% | 0.777% |
| 7 → 8 | 0.546% | 1.47% | 0.546% | 1.358% | 0.734% | 0.455% |
| 8 → 9 | 2.091% | 0.683% | 2.091% | 0.501% | 1.421% | 2.414% |

Table 15: Perspective Distortion: Stability of SIM for FEM, MLFEM, GRAD-CAM as a function of distortion level
|               | L -> PCC | L -> SIM | PCC -> SIM |
|---------------|----------|----------|------------|
| **Well FEM**  | 0.1856449| 0.70561689| 0.58639395 |
| **Well MLFEM**| 0.34383688| 0.74330975| 0.59419122 |
| **Well GRAD-CAM** | 0.2646829 | 0.64390689 | 0.38101421 |
| **Badly FEM** | 0.60857839| 0.86638901| 0.85901772 |
| **Badly MLFEM** | **0.82047656** | **0.94822866** | **0.86921674** |
| **Badly GRAD-CAM** | 0.56267264 | 0.91313241 | 0.7129938 |

Table 16: Perspective Distortion: Pearson Correlation Coefficient value between different metrics: L-stability, PCC - Pearson Correlation Coefficient with GFDMs, SIM - similarity with GFDMs

8 Conclusion and Discussion

Hence, in this work we have studied Lipschitz constant as a non-reference metric of the quality of explanation methods. We have applied it for three explanation methods Grad-CAM, FEM and MLFEM.

We also studied the agreement of this metric with previously used reference-based PCC and SIM metrics computed by comparing explanation maps with Gaze Fixation Density Maps available for classified images. We did these studies on images corrupted with growing noise separating two cases i)images which were misclassified after their corruption by distortions, ii)images which kept their class labels correctly.

Accordingly to the results obtained, it can be concluded that the Lipschitz constant does not increase with increasing distortions of the image. Since with a serious change in the original image, the explanation heatmap must also seriously change relatively to the original heatmap, which leads to stabilization and non-growth of the Lipschitz constant. The experimental behaviour thus confirms our hypothesis. In addition, the correlation with other, reference-based, metrics demonstrates high values.

As a conclusion we have two points.

- Comparing the three methods we state that MLFEM method is the best explainer from the point of view of all three metrics, reference - based and non-reference ones;

- Due to the very good agreement between Lipshitz constant and PCC and SIM values, this non-refefence stability metric can be used in case when the Gaze Fixation Density Maps are not available for images to classify.

References

[1] Kazi Ahmed Asif Fuad, Pierre-Etienne Martin, Romain Giot, Romain Bourqui, Jenny Benois-Pineau, and Akka Zemmari. Features Understanding in 3D CNNs for Actions Recognition in Video. In *Tenth International Conference on Image*
[2] David Alvarez-Melis and Tommi S. Jaakkola. Towards robust interpretability with self-explaining neural networks. *CoRR*, abs/1806.07538, 2018.

[3] Francesco Bodria, Fosca Giannotti, Riccardo Guidotti, Francesca Naretto, Dino Pedreschi, and Salvatore Rinzivillo. Benchmarking and survey of explanation methods for black box models. *CoRR*, abs/2102.13076, 2021.

[4] Fadi Boulos, Wei Chen, Benoît Parrein, and Patrick Le Callet. Region-of-interest intra prediction for h.264/avc error resilience. In *2009 16th IEEE International Conference on Image Processing (ICIP)*, pages 3109–3112, 2009.

[5] Luca Bourroux, Jenny Benoïs-Pineau, Romain Bourqui, and Romain Giot. Multi layered feature explanation method for convolutional neural networks. In *ICPRAI (1)*, volume 13363 of *Lecture Notes in Computer Science*, pages 603–614. Springer, 2022.

[6] Aditya Chattopadhay, Anirban Sarkar, Prantik Howlader, and Vineeth N. Balasubramanian. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In *WACV*, pages 839–847. IEEE Computer Society, 2018.

[7] Philippe Pérez de San Roman, Jenny Benoïs-Pineau, Jean-Philippe Domenger, Florent Paclet, Daniel Cattaert, and Aymar de Rugy. Saliency driven object recognition in egocentric videos with deep CNN: toward application in assistance to neuroprostheses. *Comput. Vis. Image Underst.*, 164:82–91, 2017.

[8] Kazi Ahmed Asif Fuad, Pierre-Etienne Martin, Romain Giot, Romain Bourqui, Jenny Benoïs-Pineau, and Akka Zemmari. Features understanding in 3d cnns for actions recognition in video. In *IPTA*, pages 1–6. IEEE, 2020.

[9] Tristan Gomez, Thomas Fréour, and Harold Mouchère. Metrics for saliency map evaluation of deep learning explanation methods. In *ICPRAI (1)*, volume 13363 of *Lecture Notes in Computer Science*, pages 84–95. Springer, 2022.

[10] Adrien Halnaut, Romain Giot, Romain Bourqui, and David Auber. Deep dive into deep neural networks with flows. In *VISIGRAPP (3: IVAPP)*, pages 231–239. SCITEPRESS, 2020.

[11] Ming Jiang, Shengsheng Huang, Juanyong Duan, and Qi Zhao. Salicon: Saliency in context. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1072–1080, 2015.

[12] Ming Jiang, Shengsheng Huang, Juanyong Duan, and Qi Zhao. Salicon: Saliency in context. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1072–1080, 2015.

[13] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 12 2014.
[14] O. LeMeur’2013 and T. Baccino. Methods for comparing scanpaths and saliency maps: strengths and weaknesses. Behavior Research Methods, 45(1):251–266, 2013.

[15] Guan Li, Junpeng Wang, Han-Wei Shen, Kaixin Chen, Guihua Shan, and Zhonghua Lu. Cnnpruner: Pruning convolutional neural networks with visual analytics. IEEE Transactions on Visualization and Computer Graphics, 27(2):1364–1373, 2021.

[16] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. CoRR, abs/1405.0312, 2014.

[17] Stefan Mathe and Cristian Sminchisescu. Actions in the eye: Dynamic gaze datasets and learnt saliency models for visual recognition. IEEE Trans. Pattern Anal. Mach. Intell., 37(7):1408–1424, jul 2015.

[18] Grégoire Montavon, Alexander Binder, Sebastian Lapuschkin, Wojciech Samek, and Klaus-Robert Müller. Layer-Wise Relevance Propagation: An Overview, pages 193–209. 09 2019.

[19] Abraham Montoya Obeso, Jenny Benois-Pineau, Mireya Saraí García-Vázquez, and Alejandro Alvaro Ramírez-Acosta. Visual vs internal attention mechanisms in deep neural networks for image classification and object detection. Pattern Recognit., 123:108411, 2022.

[20] Abraham Montoya Obeso, Jenny Benois-Pineau, Mireya Saraí García-Vázquez, and Alejandro Alvaro Ramírez-Acosta. Visual vs internal attention mechanisms in deep neural networks for image classification and object detection. Pattern Recognit., 123:108411, 2022.

[21] Vitali Petsiuk, Abir Das, and Kate Saenko. RISE: randomized input sampling for explanation of black-box models. In BMVC, page 151. BMVA Press, 2018.

[22] François Rousseau, Lucas Drumetz, and Ronan Fablet. Residual Networks as Flows of Diffeomorphisms. Journal of Mathematical Imaging and Vision, May 2019.

[23] Ramprasaath R. Selvaraju, Abhishek Das, Ramakrishna Vedantam, Michael Cogswell, Devi Parikh, and Dhruv Batra. Grad-cam: Why did you say that? visual explanations from deep networks via gradient-based localization. CoRR, abs/1610.02391, 2016.

[24] Richard Szeliski. Computer Vision Algorithms and Applications. Second Edition. Springer Cham, New York, 2022.

[25] Wenguan Wang, Jianbing Shen, Fang Guo, Ming-Ming Cheng, and Ali Borji. Revisiting video saliency: A large-scale benchmark and a new model. In Proceedings of the IEEE Conference on computer vision and pattern recognition, pages 4894–4903, 2018.
[26] Bolei Zhou, Aditya Khosla, Àgata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. *CoRR*, abs/1512.04150, 2015.