A Clustering-based Approach for Interpreting Black-box Models

(Discussion Paper)

Luca Ferragina[1,†], Simona Nisticò[1,*,†]

[1]DIMES, University of Calabria, 87036 Rende (CS), Italy

Abstract

Classification and regression tasks involving image data are often connected to critical domains or operations. In this context, Machine and Deep Learning techniques have achieved astonishing performances. Unfortunately, the models resulting from such techniques are so complex to be seen as black boxes, even when we have full access to the model's information. This is limiting for experts who leverage these tools to make decisions and lowers the trust of users who are somehow subjected to their outcomes.

Some methods have been proposed to solve the task of explaining a black box both in a non-specific data domain and for images. Nevertheless, the most used explanation tools when dealing with image data have some limitations, as they consider pixel-level explanations (SHAP), involve an image segmentation phase (LIME) or apply to specific neural architectures (Grad-CAM).

In this work, we introduce CLAIM, a model-agnostic explanation approach, that interprets black boxes by leveraging a clustering-based approach to produce interpretation-dependent higher lever features. Additionally, we perform a preliminary analysis aimed at probing the potentiality of the proposed approach.

Keywords

eXplainable AI, Post-hoc explanations, Local Explanations, Model-agnostic Explanations

1. Introduction

The pervasive use of Machine and Deep Learning models in everyday life processes has raised the problem of models' trustworthiness. The root of this problem lies in the level of complexity characterizing them. Indeed, such complexity makes it difficult to understand the logic followed by the model to perform its prediction, and this is true not only for final users but also for Machine and Deep Learning experts who have difficulties inspecting and debugging their own models. When predictive models take part to decisions that affect users' lives, the above-stated problem involves even a legal dimension. The GDPR enshrines the right of explanation [1], which requires to be able to provide to users an intelligible explanation for the model outcome.

All the above-stated issues have led to the birth of the eXplainable Artificial Intelligence (XAI) field, which collects all the research efforts in providing instruments for a more-aware use of artificial intelligence solutions. Many types of approaches have been developed to face the

SEBD '24: 32nd Symposium on Advanced Database Systems June 23-26, 2024 - Villasimius, Sardinia, Italy

*Corresponding author.

†These authors contributed equally.

luca.ferragina@unical.it (L. Ferragina); simona.nistico@unical.it (S. Nisticò)

0000-0003-3184-4639 (L. Ferragina); 0000-0002-7386-2512 (S. Nisticò)

model explainability problem [2, 3]. From the taxonomy described in [1], it emerges that some works focus on building models which are interpretable by design, whose more challenging aspect is to provide explainability without affecting too much model performances.

Others, known as post-hoc explanation methods, aim to explain already designed and trained models. This class of methods differs from the others because there are various levels of information availability ranging from having access only to the model output to having complete access to all its information. In this work, we will focus on the latter class of explanation methods. Our contributions can be summarized as follows:

- we analyze the main algorithms for explaining models working on image data;
- we introduce **CLAIM**, **CL**ustering-based **A**pproach for **I**nterpreting black-box **M**odels;
- we provide preliminary experimental results.

The structure of the paper is the following. In Section 2 we describe the background and discuss related works. In Section 3 we introduce the CLAIM algorithm and provide an example to illustrate its working principles. In Section 4 the results of the experiments are reported. Finally, Section 5 concludes the paper.

2. Preliminaries and Related Works

Regarding post-hoc explainability, various settings arise from different levels of information availability. One of them considers model-agnostic explanations in which only the information provided by the model output is exploited to understand its behaviour.

To explain the prediction for a certain data instance, some Post-hoc methodologies perturb the input, collect information about how the outcome changes, and then exploit it to estimate the level of importance of each feature. Among them, SHAP [4] is a game-theory-inspired method that attempts to enhance interpretability by computing the importance values for each feature. While SHAP applies to different data types, the RISE [5] method focuses on image data. To explain a prediction, it generates an importance map indicating how salient each pixel is for the model's prediction by probing the model with randomly masked versions of the input image to observe how the output changes.

Explanations can also be given by examples, in which case counterfactuals, which are instances similar to the considered sample that bring the model to produce a different outcome, are provided to users as justifications. Among the methods of this category it is possible to find LORE [6], which learns an interpretable classifier in a neighbourhood of the sample to explain. This neighbourhood is generated by employing a genetic algorithm, using model outcomes as labels, and then extracting from the interpretable classifier an explanation consisting of a decision rule and a set of counterfactuals. DiCE, proposed in [7], generates counterfactual examples that are both actionable and diverse. The diversity requirement aims at increasing the richness of information delivered to the user. MILE [8] exploits an adversarial-like neural network architecture to learn a transformation able to change the black-box model outcome for the considered sample. It can provide simultaneously two kinds of explanation: a counterfactual, which is the result of the transformation application, and a score for each object feature, derived from the transformation.

Finally, some methodologies explain models via local surrogates, which are self-interpretable functions that mimic the decisions of black-box models locally. LIME [9] explains black-box models by converting the data object into a domain composed of interpretable features, and then perturbing it in that domain and querying the black-box to learn a simple model (the local surrogate) using this generated dataset. A variant of LIME called $\mathcal{S}$-LIME, has been proposed in [10] to solve the problem related to out-of-distribution generated samples. In particular, it exploits semantic features extracted through unsupervised learning to generate the neighbourhood. The type of explanations provided changes in Anchor [11] where if-then rules having high precision, called anchors, are created and utilised to represent locally sufficient conditions for prediction. Other approaches, known as model-specific methods, exploit the peculiarities of the class of models they are targeted to. For example, some methods focus on neural networks and exploit information carried by the gradient. The authors of [12] propose two methodologies to explain ConvNets through visualization, that leverage the computation of the gradient related to the class score for the input image considered. Grad-CAM [13, 14] uses the gradients flowing into the final convolutional layer of any target concept, which can be related to classification or other tasks, to produce a coarse localization map highlighting important regions in the image for predicting the concept.

Among the aforementioned methods, the ones that specifically address the issue of providing a post-hoc explanation of black-box models dealing with image data are Grad-CAM and RISE. However, Grad-CAM has the weakness of applying only to a very specific type of models, i. e. convolutional neural networks.

As for RISE, being designed for computing the importance of every single feature, it provides explanations that are not beneficial since there is a consistent number of features. Generally, to make these explanations more user-friendly, as long as image data is considered, the explanation is shaped as a heatmap $\mathbf{h}$ that assigns a continuous importance value to each pixel. Regrettably, this is not sufficient, since the so obtained explanations should be so scattered to be still incomprehensible for users.

Even if it is not specifically tailored to image data, one of the most widespread methods for model-agnostic explanation is LIME, because of its versatility and ease of use. When dealing with images, LIME considers as interpretable features for the surrogate model the output obtained from a segmentation algorithm. The issue with this approach is that the segmentation and explanation steps are separated from each other, therefore the aggregation of different pixels is based on their importance for the model. This fact may lead to a rough explanation that identifies the most important portions (according to the black box) of the image with low precision.

In the following section, we introduce CLAIM, an algorithm for post-hoc explaining black-box models specifically designed for image data. CLAIM faces the issues described above by aggregating pixels that the model considers of similar importance, i. e. that produce a similar effect when they are perturbed.

3. Methodology

Let $f : \mathbb{R}^d \to \mathbb{R}$ be a black-box assigning a real value to each data point belonging to the input space. Theoretically, f may be the function obtained from a machine learning method achieving any specific task. Thus, for example, $f(\mathbf{x})$ may represent the result of a regression analysis on $\mathbf{x}$, the probability that $\mathbf{x}$ belong to a certain class, the anomaly degree of the point $\mathbf{x}$, and so on.

Given a sample $\mathbf{x}$, in order to understand which features are the most important, according to the model f, for the elaboration of the output $f(\mathbf{x})$, we investigate how feature-wise perturbations of $\mathbf{x}$ affect the value of $f(\mathbf{x})$. Thus, for each fixed $i \in \{1, \dots, d\}$, we consider

$$p_n^{(i)} = f(\mathbf{x}) - f(\mathbf{x} + n\varepsilon \mathbf{e}_i), \tag{1}$$

where ε is a *perturbation step*, $n \in \{-N, \dots, -1, 1, \dots, N\}$ determines the number of perturbations we are performing and $\mathbf{e}_i$ is a vector whose components are all equals to 0 except for the i-th that is equal to 1.

The value in Equation (1) expresses how much the output of the black-box model f varies if we perturb by $n\varepsilon$ the feature i of the input data point $\mathbf{x}$. By collecting the variations obtained on the feature i with all the different $n \in \{-N, \dots, -1, 1, \dots, N\}$, we obtain an embedded representation $\mathbf{p}^{(i)} \in \mathbb{R}^{2N}$ relative to the feature i, whose components are expressed by

$$\mathbf{p}^{(i)} = \left[p_{-N}^{(i)}, \dots, p_{-1}^{(i)}, p_1^{(i)}, \dots, p_N^{(i)} \right].$$

The same reasoning applied to each feature of $\mathbf{x}$ produces a finite set of embedded points

$$P = \left\{ \mathbf{p}^{(1)}, \dots, \mathbf{p}^{(d)} \right\} \subseteq \mathbb{R}^{2N}$$

such that each $\mathbf{p}^{(i)}$ represents how the model behaves when we perturb $\mathbf{x}$ on the feature i. Each of the $2N$ dimensions of the space we build contains information about how the pixels of the sample behave when they are subjected to a fixed perturbation.

The norm of a certain $\mathbf{p}^{(i)}$ in the $2N$-dimensional space represents a score measuring the importance of the feature i for the elaboration of the output provided by f on the sample $\mathbf{x}$. Indeed, if $\|\mathbf{p}^{(i)}\|$ is relatively low, it means that perturbations on the feature i of the input do not substantially modify the output provided by f, thus the contribution of the feature i to the output $f(\mathbf{x})$ is poor. On the other hand, if it is high, it means that perturbing the feature i has a huge effect on the output, thus i must be a very important feature according to the model.

In order to provide a more understandable heatmap we apply a clustering algorithm on P with the aim of aggregating features with similar behaviors. Here we consider the K-MEANS algorithm [15] that builds up a set of K centroids $C = \{\mathbf{c}^{(1)}, \dots, \mathbf{c}^{(K)}\}$ that are representative of the K clusters $\{C_1, \dots, C_K\}$ in which P is partitioned.

From our perspective, each cluster C_k with $k \in \{1, \dots, K\}$ constitutes a subset of features of the input point $\mathbf{x}$ in which the model f behaves similarly in presence of perturbations. Thus, each C_k can be seen as a *macro-feature* and the norm of the relative centroid $\|\mathbf{c}^{(k)}\|$ indicates the importance of this macro-feature for the output provided by f on $\mathbf{x}$. To better visualize the explanation obtained, we build a heatmap $\mathbf{h}$ whose i-th element is given by

Algorithm 1: CLAIM

Input: Black-box f, point $\mathbf{x}$, perturbation step ε, number of perturbations N, number of macro-features K

Output: An heatmap $\mathbf{h}$ highlighting the most important macro-features of $\mathbf{x}$ according to f

1 **foreach** *feature* $i = 1, \ldots, d$ **do**
2 **foreach** $n \in \{-N, \ldots, -1, 1, \ldots, N\}$ **do**
3 Compute the perturbation $p_n^{(i)}$ using Equation (1);
4 Apply the K-MEANS algorithm to the set P obtaining clusters $\{C_1, \ldots, C_K\}$;
5 Build the value of the feature i of the heatmap $\mathbf{h}$ using Equation (2);

$$h_i = \|\mathcal{C}(\mathbf{p}^{(i)})\| \tag{2}$$

where we indicate by $\mathcal{C} : P \to C$ the function that assigns each point in P to the centroid of the cluster to which it belongs. To better illustrate each step of CLAIM (also reported in Algorithm 1), in the next section we provide a detailed example.

3.1. Motivating Example

When dealing with a black-box model, an analysis of the value of performance metrics such as accuracy, is not always sufficient to effectively assess its quality.

One common situation is the one in which the presence of some *bias* in the data used for the training of a model f may affect the output provided by f, making it focus on non-relevant features. This is potentially dangerous, because, if also the data used for the quality assessment presents the same issue, the resulting model performances do not reflect its poor quality.

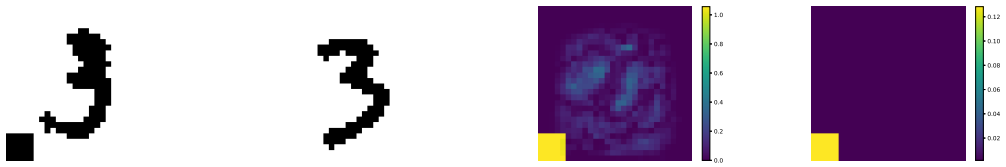
To reproduce this kind of scenario, we set up the following experiment. We consider as f a logistic regression model that must classify the images belonging to the MNIST [16] data set as 3 or 9, more specifically, given an image $\mathbf{x}$, f outputs the probability of $\mathbf{x}$ belonging to the class 9. Then, we train f with samples of classes 3 and 9 in which we add a black rectangle in the bottom left corner to all the training images belonging to the class 3, as shown in Figure 1a. In the following, we refer to the images with this modification as *biased*.

At inference time we pass to the model an input sample $\mathbf{x}$ representing a non-biased 3 (Figure 1b). The model fails in recognizing it as a 3 since it outputs $f(\mathbf{x}) = 0.97$.

This happens because the added sign is fully discriminating between classes in the training set, thus, the model is clearly focusing only on the portion of the image where the sign is (or should be) located, as depicted in Figure 1c showing the magnitude of the model's weights.

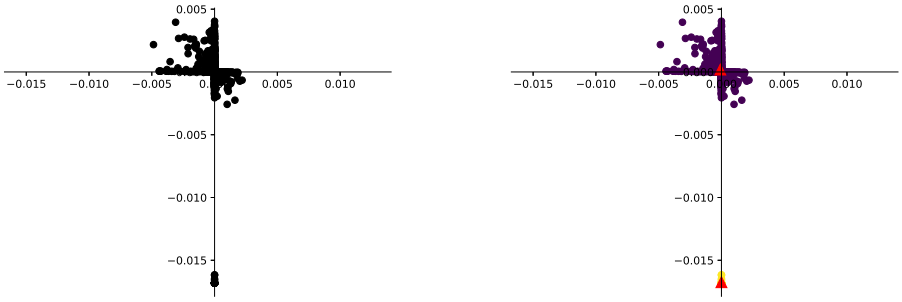
We now apply CLAIM to explain the behaviour of f on the image $\mathbf{x}$, in particular, we set $\varepsilon = 0.5$, $K = 2$, and $N = 1$.

Figure 2a shows the bi-dimensional data space P built in line 3 of Algorithm 1 and Figure 2b shows the two centroids (the red triangles) and the splitting into the two clusters (purple and yellow points) obtained in line 4. We can observe that the cluster in yellow is the one farther from the origin, which means that its centroid has a larger norm and, thus, it is relative to the macro-feature that contributes the most to the output of f.



(a) Biased train sample. (b) Unbiased test sample. (c) Model's weights. (d) CLAIM heatmap.

Figure 1: Explanation of the classification outcome for an unbiased sample of class 3 (Figure 1b) classified by the black box as belonging to class 9. For the latter two figures, the colour indicates the level of importance for the corresponding pixel/cluster.



(a) (b)

Figure 2: Plot of the 2-dimensional data set collected by CLAIM to explain the black-box outcome for an unbiased element of class 3. Figure 2a depicts the points carrying the collected information. Figure 2b shows how the samples have been clustered in 2 groups, the red triangles highlight cluster centroids.

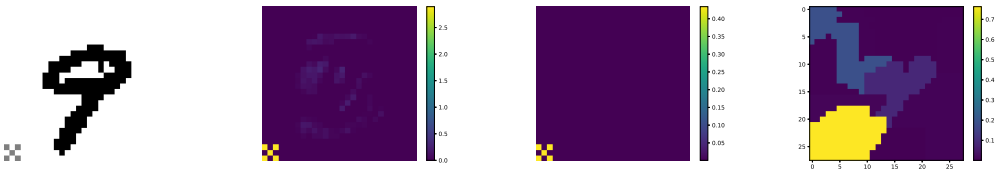
Figure 1d shows the heatmap provided in output, we can see that it is almost identical to the model's weight, which means that CLAIM has been able to identify with great precision the most expressive (according to f) macro-feature of $\mathbf{x}$.

4. Experimental Results

The example described before is quite simple since the bias is inducted by a patch with a regular shape that is quite easy to handle for most algorithms. Therefore, in order to analyze the behavior of CLAIM in more challenging scenarios, in the following we consider settings similar to the one described in Section 3.1 but in which the shapes of the patches are more elaborated.

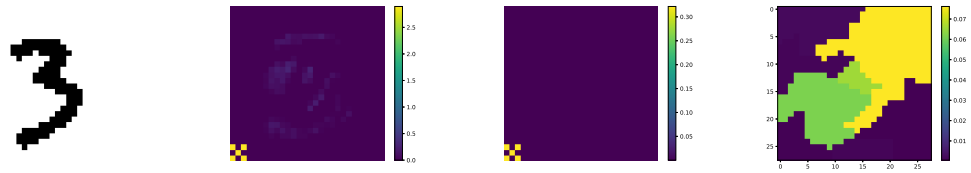
To have guidance in assessing the adherence of the explanation to the behaviour of the explained model, in all the experiments we consider again a logistic regression as a black-box f. We also compare the results obtained by CLAIM with those obtained by LIME.

In the first experiment, instead of a single square, the bias is given by five squares placed in a sort of "chessboard"-shaped patch (Figure 3a). Figure 3 reports the visual result of the experiment on an image (Figure 3a) of a 9 that, differently from the training set, contains the



(a) Test sample. (b) Model's weights. (c) CLAIM heatmap. (d) LIME heatmap.

Figure 3: A biased 9 from the test set of the first experiment and its associated explanations.



(a) Test sample. (b) Model's weights. (c) CLAIM heatmap. (d) LIME heatmap.

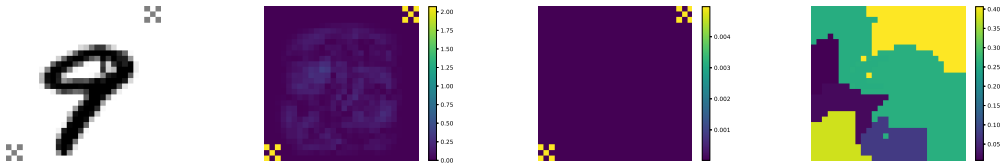
Figure 4: An unbiased 3 from the test set of the first experiment and its associated explanations.

bias. Similarly, as before, the model has mainly focused on the features involved in the bias, indeed all the weights associated with the other features are very close to 0 (Figure 3b). As we can see from Figure 3d CLAIM is able to perfectly isolate the bias aggregating all its pixels in a single macro-feature and assigning to it a large value in the heatmap. On the other hand, LIME is only able to roughly identify the portion of the image containing the bias but it fails to precisely determine its shape. This happens because the segmentation algorithm used in LIME does not exactly separate the bias from the rest of the image.

A similar behavior can be observed on an image of a 3 that does not contain the patch (Figure 4a). Even in this case in which the bias is invisible, our model succeeds in capturing it, obtaining a heatmap (Figure 4d) consistent with the weights of f (Figure 4b). As for LIME (Figure 4c), its behaviour in this sample is worse. Indeed, since the segmentation step is performed before the rest of the algorithm and only considers the image content, it is impossible for it to obtain a partition of the image that is suitable to identify this bias.

In the second experiment, we add another difficulty by inserting a disconnected bias into the training set. In particular, we place one "chessboard" patch in the bottom-left corner and one in the top-right corner. As we can see from Figure 5, the heatmap provided by CLAIM is extremely precise, since it includes in a single macro-feature both the patches and judges it as the most relevant macro-feature for f. For what concerns LIME, also in this case, the segmentation step causes some issues. In particular, the two patches belong to different portions of the images and thus they are assigned to two different macro-features. This type of result is not desirable since the pixels belonging to the two patches contribute in exactly the same way to the output provided by f and, for this reason, they conceptually belong to a unique macro-feature.

Still concerning this experiment, Table 1 shows Precision and Recall over the features con-



(a) Test sample. (b) Model's weights. (c) CLAIM heatmap. (d) LIME heatmap.

Figure 5: A biased 9 from the test set of the second experiment and its associated explanations.

	CLAIM	LIME 1	LIME 2	LIME 3
Precision	$\mathbf{1.000 \pm 0.000}$	0.072 ± 0.002	0.034 ± 0.000	0.023 ± 0.000
Recall	$\mathbf{1.000 \pm 0.000}$	0.816 ± 0.001	0.840 ± 0.003	0.865 ± 0.004

Table 1
Precision and Recall over the features considered important by the weight of the logistic regression trained on the training data set containing the disconnected bias.

sidered important by the weight of the logistic regressor. The metrics are computed on an unbiased test set and we report the mean and the standard deviation over all the samples in the test set. The first column is relative to our method while the others are relative to the heatmap of LIME in which we consider the 1, 2, or 3 most important macro-features. The numerical results confirm that CLAIM overcomes LIME in detecting such kind of bias. In particular, the Precision of LIME is always much smaller than the Recall, which means that it is judging as important a lot of pixels that actually are irrelevant for f.

5. Conclusion

In this work, we deal with the post-hoc explanation of models that work with image data and introduce the CLAIM algorithm. Our goal is to provide users with heatmaps that include higher-level features built through the guidance of the black-box model without exploiting segmentation algorithms, which only consider image content.

The preliminary analysis performed on the CLAIM's explanations leads to promising results that give rise to hope about the potential of this method in producing faithful explanations that lead users to focus only on important image regions.

In future development, we will focus on deepening the expressive power of the information it extracts to compute explanations. We will also investigate CLAIM's robustness with respect to its parameters, to analyze how they affect explanation quality. Furthermore, we plan to enlarge the experiments performed to include more competitors and to consider richer data sets.

Acknowledgments

We acknowledge the support of the PNRR project FAIR - Future AI Research (PE00000013), Spoke 9 - Green-aware AI, under the NRRP MUR program funded by the NextGenerationEU.

References

[1] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, D. Pedreschi, A survey of methods for explaining black box models, ACM Comput. Surv. 51 (2019) 93:1–93:42. URL: https://doi.org/10.1145/3236009. doi:10.1145/3236009.

[2] Q. Zhang, Y. N. Wu, S.-C. Zhu, Interpretable convolutional neural networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 8827–8836.

[3] J. Donnelly, A. J. Barnett, C. Chen, Deformable protopnet: An interpretable image classifier using deformable prototypes, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2022, pp. 10265–10275.

[4] S. Lundberg, S.-I. Lee, A unified approach to interpreting model predictions, arXiv preprint arXiv:1705.07874 (2017).

[5] V. Petsiuk, A. Das, K. Saenko, Rise: Randomized input sampling for explanation of black-box models, arXiv preprint arXiv:1806.07421 (2018).

[6] R. Guidotti, A. Monreale, F. Giannotti, D. Pedreschi, S. Ruggieri, F. Turini, Factual and counterfactual explanations for black box decision making, IEEE Intelligent Systems 34 (2019) 14–23.

[7] R. K. Mothilal, A. Sharma, C. Tan, Explaining machine learning classifiers through diverse counterfactual explanations, in: Proceedings of the 2020 ACM FAccT, 2020, pp. 607–617.

[8] F. Angiulli, F. Fassetti, S. Nisticò, Local interpretable classifier explanations with self-generated semantic features, in: DS, Springer, 2021, pp. 401–410.

[9] M. T. Ribeiro, S. Singh, C. Guestrin, " why should i trust you?" explaining the predictions of any classifier, in: Proceedings of the 22nd ACM SIGKDD KDD, 2016, pp. 1135–1144.

[10] F. Angiulli, F. Fassetti, S. Nisticò, Finding local explanations through masking models, in: IDEAL 2021, Springer, 2021, pp. 467–475.

[11] M. T. Ribeiro, S. Singh, C. Guestrin, Anchors: High-precision model-agnostic explanations, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 32, 2018.

[12] K. Simonyan, A. Vedaldi, A. Zisserman, Visualising image classification models and saliency maps, Deep Inside Convolutional Networks (2014).

[13] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, Grad-cam: Visual explanations from deep networks via gradient-based localization, in: Proceedings of the IEEE ICCV, 2017, pp. 618–626.

[14] A. Chattopadhay, A. Sarkar, P. Howlader, V. N. Balasubramanian, Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks, in: 2018 IEEE winter conference on applications of computer vision (WACV), IEEE, 2018, pp. 839–847.

[15] J. MacQueen, et al., Some methods for classification and analysis of multivariate observations, in: Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, volume 1, Oakland, CA, USA, 1967, pp. 281–297.

[16] L. Deng, The mnist database of handwritten digit images for machine learning research, IEEE Signal Processing Magazine 29 (2012) 141–142.