Neural Computing and Applications manuscript No.
(will be inserted by the editor)

Playing to distraction: towards a robust training of CNN classifiers through visual explanation techniques

David Morales · Estefania Talavera · Beatriz Remeseiro

Received: date / Accepted: date

Abstract The field of deep learning is evolving in different directions, with still the need for more efficient training strategies. In this work, we present a novel and robust training scheme that integrates visual explanation techniques in the learning process. Unlike the attention mechanisms that focus on the relevant parts of images, we aim to improve the robustness of the model by making it pay attention to other regions as well. Broadly speaking, the idea is to *distract* the classifier in the learning process by forcing it to focus not only on relevant regions but also on those that, *a priori*, are not so informative for the discrimination of the class. We tested the proposed approach by embedding it into the learning process of a convolutional neural network for the analysis and classification of two well-known datasets, namely Stanford cars and FGVC-Aircraft. Furthermore, we evaluated our model on a real-case scenario for the classification of egocentric images, allowing us to

Partial financial support was received from HAT.tec GmbH. This work has been financially supported in part by European Union ERDF funds, by the Spanish Ministry of Science and Innovation (research project PID2019-109238GB-C21), and by the Principado de Asturias Regional Government (research project IDI-2018-000176). The funders had no role in the study design, data collection, analysis, and preparation of the manuscript.

David Morales
Andalusian Research Institute in Data Science and Computational Intelligence, University of Granada, 18071 Granada, Spain.
HAT.tec GmbH. c/o Universitt der Bundeswehr, Werner-Heisenberg-Weg 39, 85579 Neubiberg, Germany.
E-mail: [redacted]

Estefania Talavera
Department of Computer Science, University of Groningen. Nijenborgh 9, 9747 AG Groningen, Netherlands.
E-mail: [redacted]

Beatriz Remeseiro
Department of Computer Science, Universidad de Oviedo. Campus de Gijón s/n, 33203 Gijón, Spain.
E-mail: [redacted]

obtain relevant information about peoples' lifestyles. In particular, we work on the challenging EgoFoodPlaces dataset, achieving state-of-the-art results with a lower level of complexity. The results obtained indicate the suitability of our proposed training scheme for image classification, improving the robustness of the final model.

Keywords Visual explanation techniques · Learning process · Convolutional neural networks · Image classification · Fine-grained recognition · Egocentric vision

1 Introduction

Nowadays, the potential of convolutional deep learning models for the task of image classification has been proven. Research in this field has followed different directions namely, new architecture and framework proposals [1, 2], training methods [3, 4], multi-tasking [5, 6], attention mechanisms [7, 8], explainability and interpretability [9, 10], among others.

New techniques such as attention mechanisms allow to force the model to pay attention to certain features, whilst explainable artificial intelligence techniques allow to interpret the model and know what is happening during the learning process. However, to the best of our knowledge, the combination of both approaches has not been explored. Inspired by this lack of combination, we aim to improve the training procedure by interpreting the model and focusing it on certain regions of interest. To this end, our proposed approach is based on modifying the classical training procedure to include online information and thus adapt the learning process based on the features on which the network is focused.

More specifically, we propose a new training scheme that benefits from the saliency maps provided by visual explanation techniques. Our hypothesis is that, by the end of the training phase, the model should use as many features as possible to make a robust prediction. In this sense, we apply a visual explanation algorithm to identify the regions on which the model bases its decisions. After identifying those relevant areas, we partially occlude them trying to *distract* the model in some way and forcing the detection of other regions that, a priori, are weak (i.e., not so informative for the discrimination of the class). Our intention is to highlight that the model should not forget what the occluded regions mean, but it should learn to recognize other features to make a decision. This is ensured as the occluded images are combined with the original ones during the learning process.

We think fine-grained image classification problems could benefit the most from this approach, as they have many classes that differ from each other in small details, and our training approach forces the network to find them. For this reason, we evaluated the proposed training scheme on two well-know datasets namely Stanford cars [11] and FGVC-Aircraft [12], composed of 16,185 and 10,000 images respectively, and used in fine-grained recognition. In

addition, we carried out some experiments on top of different backbone architectures to demonstrate that our proposal improves the performance regardless of the respective network.

Furthermore, we evaluate the robustness of our model in a real-scenario case study: recognizing the food-related scene that an egocentric image depicts. The analysis of egocentric images is an emerging field within computer vision that has gained attention in recent years [13]. Images captured by wearable cameras during daily life allow recording information about the lifestyle of the users from a first-person perspective [14, 15]. The analysis of this information can be used to improve peoples' health-related habits [16]. In particular, the analysis of food-related egocentric images can be a powerful tool to analyze peoples' nutritional habits, being the focus of previous research [17, 15]. In this context, we carried out some experiments on the EgoFoodPlaces dataset [15], which is composed of 33,801 images and describes food-related locations gathered by 11 camera wearers throughout their daily life activities.

The contributions of this research work are three-fold:

1. A novel training scheme for CNN image classification that makes use of visual explanation techniques, with the main aim of improving the robustness and the generalization ability of the trained models.
2. The experiments carried out demonstrate the competitiveness of our training scheme, which outperforms the classical approach on two public datasets commonly used in fine-grained recognition tasks, regardless of the backbone architecture.
3. Our proposed method achieves competitive results in a real-case scenario that addresses the classification of egocentric photo-streams depicting food-related scenes.

The rest of the paper is organized as follows. Section 2 includes an overview of related works. Section 3 presents the proposed training approach. Section 4 introduces the two datasets for fine-grained recognition, describes the experiments carried out and analyzes the obtained results. Section 5 describes and evaluates the case study focused on egocentric vision. Finally, Section 6 closes with our conclusions and future lines of research.

2 Related Work

While the very first machine learning systems were easily interpretable, the last years have been characterized by an upsurge of opaque decision systems, such as deep neural networks (DNNs) [18, 19]. DNNs are the state-of-the-art on many machine learning tasks due to their great generalization and prediction skills. However, they are considered *black-box* machine learning models. In this context, there has been a growing influx of work on explainable artificial intelligence. Post-hoc local explanations, which refer to the use of interpretation methods after training a model, and feature relevance methods are increasingly the most adopted approaches to explain DNNs [18]. In this section, we

review some methods that produce *visual explanations* for decisions of a large class of DNN-based models, making them more transparent and reliable.

Most of these visual explanation techniques provide heat maps to identify the regions of the input images that networks look at when making predictions, allowing the data to be interpreted at a glance. Note that these heat maps are also referred to in the literature as sensitivity maps, saliency maps, or class activation maps. Class activation mapping (CAM) [20] is a well-known procedure for generating class activation maps using global average pooling in CNNs. Their authors expect each unit to be activated by some visual pattern within its receptive field. The class activation map is nothing more than a weighted linear sum of the presence of these visual patterns at different spatial locations. By simply upsampling the class activation map to the size of the input image, they can analyze the most relevant image regions to identify the particular category. However, CAM can only be used with a restricted set of layers and architectures.

A class-discriminative localization technique called gradient-weighted class activation mapping (Grad-CAM) was proposed in [21]. In fact, it is a generalization of CAM that can be applied to a significantly broader range of CNN families. Grad-CAM uses the gradients of any target concept flowing into the final convolutional layer to produce a coarse localization map, highlighting the regions of the image that are relevant for the prediction. Given an image and a class of interest (e.g., *tiger cat*) as inputs, Grad-CAM forward propagates the image through the convolutional part of the model and then through task-specific computations to obtain a raw score for the category. The gradients are set to 0 for all classes except for the desired class (*tiger cat*), which is set to 1. This signal is then backpropagated to the rectified convolutional feature maps of interest, which are combined to compute the coarse Grad-CAM localization that represents where the model looks at to make the corresponding decision. Finally, they point-wise multiply the heat map with guided backpropagation, thus obtaining also guided Grad-CAM visualizations, which are both high-resolution and concept-specific.

Another visual explanation method was presented in [22], in which input images are perturbed by occluding all their patches, in an iterative process, and classifying the occluded images. This idea allows the authors to analyze how the top feature maps and the classifier output change, revealing structures within each patch that stimulate a particular feature map. However, the use of this method requires generating multiple occluded samples and their classification, making it computationally expensive.

Ribeiro et al. [23] proposed the local interpretable model-agnostic explanations (LIME) technique, which allows to explain the predictions of any classifier in an interpretable and faithful manner. Given the original representation of the instance being explained, they get new samples by perturbing the original representation. They use those samples to approximate the classifier with an interpretable model. Just as the method above, the use of multiple samples implies to apply the classifier several times given one instance.

Some of these visual explanation techniques generate noisy sensitivity maps. In this context, Smilkov et al. [24] proposed SmoothGrad, a technique to reduce the noise in the sensitivity maps produced by visual explanation techniques based on gradients. Their idea was to sample images similar to the original ones by adding some noise. Then, they produced intermediate sensitivity maps for each image and took the average of them as the final sensitivity map.

Finally, it is worth highlighting some applications of the saliency maps generated by visual explanation techniques. Schöttl [25] used Grad-CAM maps to improve the explainability of classification networks. More specifically, the idea was to introduce some measures obtained from the Grad-CAM maps in the loss function. Cancela et al. [26] proposed a saliency-based feature selection method that selects the features that contain a higher discrimination result, allowing to provide robust and explainable predictions in both classification and regression problems.

2.1 Egocentric photo-streams

Following, we review some recent works on egocentric photo-streams, mainly focused on the classification of food-related scenes, such as our case study.

Egocentric image analysis is a field within computer vision related to the design and development of algorithms to analyze and understand photo-streams captured by wearable cameras [15]. These cameras are capable of capturing images that record visual information of our daily life, known as *visual lifelogging*, to create a visual diary with activities of first-person life. The analysis of these egocentric photo-streams can improve peoples' lifestyle by analyzing social patterns [27], social interactions [28], or food behavior [29].

In recent years, there is a growing interest in egocentric photo-streams giving their potential for assisted living. For instance, Furnari et al. [30] presented a benchmark dataset containing egocentric videos of eight personal locations and proposed a multi-class classifier to reject locations not belonging to any of the categories of interest for the end-user.

As for food-related scene recognition, Sarker et al. [17] addressed this task by proposing a multi-scale atrous CNN [31] to analyze lifelogging images and determine people's recurrences in food places throughout their day. Later, Talavera et al. [15] presented the EgoFoodPlaces dataset, composed of more than 33,000 images organized in 15 food-related scene classes. This dataset was recorded by 11 users while spending time on the acquisition, preparation, or consumption of food. The dataset was manually labeled into a total of 15 different food-related scene classes like *bakery shop*, *bar*, or *kitchen*. Taking into account the relation of the studied classes, a taxonomy for food-related scene recognition was introduced. Furthermore, the authors proposed a hierarchical classification model based on the aggregation of six VGG16 networks [32] over different subgroups of classes, emulating the proposed taxonomy. This is, to the best of our knowledge, the state-of-the-art in the recognition of food-related scenes in egocentric images.

3 Methodology

We propose a novel training approach to improve the robustness of CNNs in image classification. Figure 1 illustrates the different steps of the proposed scheme, which are subsequently explained in depth.

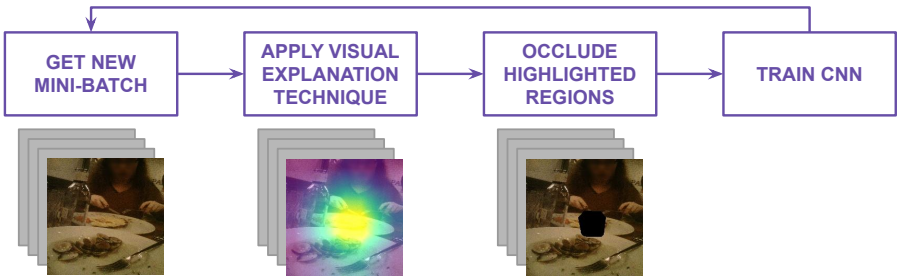


Figure 1: Workflow of our alternative training scheme, which (1) gets a new mini-batch of input images, (2) applies a visual explanation technique to generate the heat maps, (3) occludes the regions highlighted in the previous step, and (4) trains the CNN classifier.

Let consider the classical mini-batch gradient descent [33] training algorithm where, on each training step, the mini-batch is first fed into the neural network, then the gradient is computed, and finally, the calculated gradient is used to update the weights of the network. We propose to modify the training step to apply the new scheme over each mini-batch with a probability $p \in (0, 1)$; i.e., with a probability $1 - p$, the images in the mini-batch kept unchanged and the classical training step is performed as usual. Note that the probability p belongs to the open interval $(0, 1)$. $p = 0$ would mean that our training scheme is not applied (i.e., the classical training procedure is used instead). $p = 1$ would mean that only the modified images are used, making model convergence difficult. Preliminary experimentation suggests applying the method with values of $p \leq 0.5$ to guarantee that both occluded and original images are used in the learning process. Therefore, with a probability $p \in (0, 1)$, our training scheme is applied as follows:

1. Using the current weights of the network, we do inference over the current mini-batch and apply a visual explanation method to get a heat map for each image in the mini-batch. These heat maps highlight the regions where the current model focuses its attention to classify the corresponding image.
2. After that, we occlude the areas corresponding to those highlighted regions, forcing the model to look at other regions in the image. For each image in the mini-batch, we normalize its heat map and get a weight $w \in [0, 1]$ for each pixel. Next, we select all the pixels whose weight w is over a threshold th. The selected pixels are erased by setting them to 0, calling this approach 0-occlusion. As a result, we obtain the occluded images of the mini-batch.

3. Finally, we train our model making use of the occluded mini-batch.

Algorithm 1 shows the pseudo-code of our proposed training method according to the 0-occlusion approach. Note that once the mini-batch is modified, the training step continues as usual (i.e., the gradient is calculated and the weights are updated). We think it is important to highlight that the model should not forget what the occluded regions mean, but learn to recognize other parts of the image to make a decision. This is guaranteed as the occluded images are used only for some mini-batches, according to the p hyper-parameter, while the original ones are used for the rest of them.

```
Data: trainingSet, model, p, th
Result: the model trained using the proposed approach
for miniBatch in trainingSet do
    r = random(0,1);
    if r ≤ p then
        for (image, label) in miniBatch do
            heatMap = visualExplanation(image, label, model, lastConvLayer);
            heatMap = minMaxNorm(heatMap);
            selectedPixels = [heatMap > th];
            image[selectedPixels] = 0;
        end
    end
    train(model, miniBatch);
end
```

Algorithm 1: Pseudo-code of the proposed training scheme using 0-occlusion.

The proposed approach is compatible with any of the visual explanation methods presented in Section 2 and, in general, with any method that generates a heat map to explain the decision of a CNN for a given target image. Among all these techniques, we choose Grad-CAM [21] because it uses the flow of the gradients from the last convolutional layer to compute the heat maps, making it computationally less expensive than other methods like LIME [23] or SmoothGrad [24]. These other techniques apply inference several times on images generated by perturbing the target image to compute the heat maps. In other words, Grad-CAM does inference once per image while other techniques do inference several times per image, which makes the former more appropriate for the problem at hand.

Summarizing, the heat maps provided by Grad-CAM highlight the relevant regions of the image for predicting the ground truth class. By occluding them, the model is forced to look at other regions to make the decision. The initial regions should not be forgotten by the model, but other parts of the images should also be taken into account in the learning process. In this manner, the model improves its robustness and generalization capabilities.

4 Experimental framework and results

In this section, we present two datasets used to evaluate the proposed method. Next, we describe the implementation details as well as the two experiments carried out, including the evaluation metrics considered. Finally, we report and analyze the results obtained in both experiments: (1) a comparison between the proposed method and some variants of it, and (2) a comparison with standardized baselines.

4.1 Datasets

We evaluated our proposed method on two well-known datasets: the Stanford cars dataset [11], and the fine-grained visual classification of aircraft (FGVC-Aircraft) benchmark dataset [12]. Both datasets were used as part of the fine-grained recognition challenge FGComp 2013, which ran jointly with the ImageNet Challenge 2013[1].

The Stanford cars dataset contains 16,185 images of 196 car models covering sedans, SUVs, coupes, convertibles, pickups, hatchbacks, and station wagons; and it is officially split into 8,144 training and 8,041 test images. The FGVC-Aircraft dataset contains 10,000 images of aircraft, with 100 images for each of 100 different aircraft model variants; and it is officially split into 6,667 training and 3,333 test images.

4.2 Implementation details

The techniques and parameters used for experimentation are explained in the following. We used the Adam optimization algorithm [34] with the following parameters: learning rate $\alpha = 0.00005$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 0.0000007$. Regarding the training step, we used a batch size of 16 and the images were resized to 224×224. The outputs were monitored using the validation accuracy to apply an early stopping strategy, based on which the training process finished after 30 epochs with no improvement. Additionally, we applied the following classical data augmentation techniques: horizontal flip, rotation $[-40, 40]$, random channel shift $[-30, 30]$, and image brightness change $[0.5, 1.5]$.

The proposed method was implemented on TensorFlow [35] and Keras [36], and the code is available for download[2]. Starting from the training algorithm provided in Keras, we modified the training step to apply our method over each mini-batch with a probability p, as described in Section 3. According to some preliminary experiments, we applied the proposed method with a probability $p = 0.25$, and the threshold for the occlusion step was set to $th = 0.85$.

[1] http://image-net.org/challenges/LSVRC/2013/

[2] URL available after paper acceptance

4.3 Experimental setup

This section describes the two experiments designed to evaluate our training scheme. Both experiments were applied to each dataset individually and compared with other approaches. As for the experimentation itself, we kept the original split in training and test sets for the two considered datasets (see Section 4.1). For validation purposes, we randomly divided the original training dataset into two parts: 75% training and 25% validation. Then, we trained the model and evaluated it on the isolated test set, using the performance metrics described in Section 4.4. This validation procedure was repeated five times. We report the average performance and the standard deviation calculated across the five runs.

4.3.1 Experiment 1

The objective of this experiment is to test several setups of our training scheme and compare them with a baseline. In particular, we used a ResNet50 [37], a very popular network successfully applied to different image classification tasks. The different configurations are detailed as follows:

1. **Baseline.** In order to compare our method with a baseline, we trained a ResNet50 using the classical training approach (i.e., without applying the proposed method). We called this model fine-tuned ResNet50 (FT-ResNet50) because it is a model pre-trained on the ImageNet dataset [38], whose parameters were fine-tuned with the corresponding dataset.
2. **Our approach.** We trained a ResNet50 using the proposed training method, which is based on Grad-CAM visualizations and illustrated in Figure 1. More specifically, we used the weights from the ResNet50 model pre-trained on ImageNet [38], and then we fine-tuned them using the corresponding dataset and our training scheme. Note that, during the learning process, the Grad-CAM algorithm was applied to the last convolutional layer of the ResNet50, as indicated in [21].
3. **Other setups.** Aiming at demonstrating the adequacy of the 0-occlusion approach, we also conducted some experiments in which the pixels were set to a random value (R-occlusion) and 1 (1-occlusion).

4.3.2 Experiment 2

This experiment aims to demonstrate the adequacy of our training scheme regardless of the backbone architecture considered. In this sense, we applied it to two well-known backbone architectures, different from ResNet50, using the following configurations:

1. **Baselines.** We trained two architectures commonly used in the literature, InceptionV3 [39] and DenseNet [40], using the classical approach. We called them FT-InceptionV3 and FT-DenseNet, respectively, because they were pre-trained on ImageNet and fine-tuned with the corresponding dataset.

2. **Our approach.** We trained the two backbone architectures considered, InceptionV3 and DenseNet, using the proposed training scheme (see Figure 1). As in the previous experiment, we used the weights from these two architectures pre-trained on ImageNet, and then we fine-tuned them with the corresponding dataset and our training scheme. Regarding the Grad-CAM algorithm, it was applied to the last convolutional layer of the networks as described in [21].

4.4 Evaluation

In order to evaluate the performance of the proposed models and make a fair comparison with other approaches, we computed some popular metrics in image classification tasks: accuracy, precision, recall, and F-score (F1). These metrics are defined as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{4}$$

where TP, FP, TN, and FN stand for true positives, false positives, true negatives, and false negatives, respectively.

4.5 Results

In this section, we report and analyze the results obtained in the two experiments described above.

4.5.1 Experiment 1

Table 1 shows the results obtained for the different configurations. As can be observed, our training scheme provides very competitive results regardless of the setup used for the occlusion. Analyzing the four metrics considered, the three setups outperform the baseline method (FT-ResNet50), which was trained with the classical learning procedure, in both datasets. Focusing on our proposal (0-occlusion), it achieves a gain of more than 2 percent in the Standford cars dataset and about 2 percent in the FGVC-Aircraft dataset. In order to demonstrate the relevance of this improvement, we applied a statistical t-test that allows us to determine if there is a significant difference between

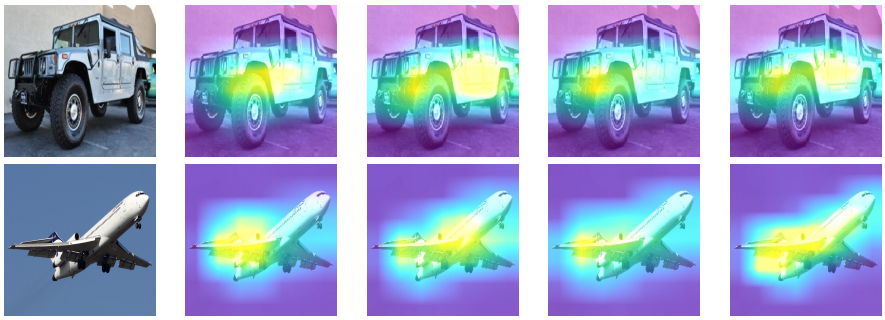
the baseline (FT-ResNet50) and our proposal (0-occlusion). Notice that we used a paired sample, two-tailed t-test. As a result, we can confirm that our proposal significantly outperforms the baseline in terms of accuracy, with a significance level of 0.05.

If we analyze the behavior of the three different setups considered for the proposed training scheme, we can see that both 0-occlusion and 1-occlusion provide better results than R-occlusion, with a very slight difference in favor of the former (0-occlusion). The experiments show that, when using random values for the occlusion procedure, the model does not benefit so much from the *distraction* applied to the model, by forcing it to look at new regions in the input images. This behavior is discussed in detail below, with some qualitative results that aim at illustrating the impact of the proposed method.

Stanford cars				
	FT-ResNet50	0-occlusion	R-occlusion	1-occlusion
Accuracy	0.849 ± 0.009	$\mathbf{0.871 \pm 0.007}$	0.860 ± 0.009	0.869 ± 0.008
Precision	0.855 ± 0.007	$\mathbf{0.876 \pm 0.007}$	0.866 ± 0.008	0.873 ± 0.008
Recall	0.849 ± 0.009	$\mathbf{0.870 \pm 0.008}$	0.860 ± 0.009	0.868 ± 0.009
F1	0.848 ± 0.009	$\mathbf{0.870 \pm 0.008}$	0.859 ± 0.009	0.867 ± 0.009
FGVC-Aircraft				
	FT-ResNet50	0-occlusion	R-occlusion	1-occlusion
Accuracy	0.731 ± 0.013	$\mathbf{0.749 \pm 0.005}$	0.739 ± 0.012	0.743 ± 0.005
Precision	0.746 ± 0.011	$\mathbf{0.762 \pm 0.005}$	0.755 ± 0.010	0.759 ± 0.004
Recall	0.731 ± 0.013	$\mathbf{0.749 \pm 0.005}$	0.739 ± 0.012	0.743 ± 0.005
F1	0.731 ± 0.014	$\mathbf{0.748 \pm 0.005}$	0.739 ± 0.012	0.743 ± 0.005

Table 1: Classification performance, averaged across five runs, of the different approaches on the Stanford cars [11] and FGVC-Aircraft [12] datasets. Best results are in bold.

Figure 2 depicts two representative images of the two datasets used for experimentation, Stanford cars and FGVC-Aircraft, along with the heat maps generated by Grad-CAM for the different methods analyzed: the baseline FT-ResNet50 and the three setups for the proposed training approach. As can be observed, the models trained with the proposed approach, regardless of the setup, base their decisions on more features than the one trained using a classical approach (FT-ResNet50). While the baseline method seems to base its decisions just on a local area of the image, the models trained with the proposed approach seem to react to almost the whole object. Analyzing the different configurations, we can see that both 0-occlusion and 1-occlusion show a similar behavior, reacting to the whole object, which explains the achieved results in both cases. However, the R-occlusion version behaves differently since it reacts to many features but with a low level of confidence. That is, occluding the selected pixels with a fixed value (0 or 1) allows us to achieve better results than occluding the relevant regions with a random value. The reason for this behavior could be that, when using a fixed value, the model learns to ignore these areas and looks at other regions, whereas the model



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

Figure 2: (a) Input images from the Stanford cars (top) and FGVC-Aircraft (bottom) datasets, (b) heat maps generated by Grad-CAM for the baseline FT-ResNet50, and heat maps generated by Grad-CAM for the model trained with the proposed training scheme using (c) 0-occlusion, (d) R-occlusion, and (e) 1-occlusion.

does not benefit as much from this idea when using a different value each time. It is worth noting that using 0-occlusion is somewhat similar to the well-known dropout [41], a regularization technique in which some connections are disabled during the learning phase. This would explain why this approach gets slightly better results than the 1-occlusion version.

Finally, Table 2 shows the number of epochs and the seconds per epoch needed to train the baseline (FT-ResNet50) and our proposal (0-occlusion). As can be observed, our training scheme requires more computational time per epoch and more epochs to converge than the classical procedure. Regarding the increment in terms of seconds per epoch, it is lower than 19%. Note that this time only depends on the image resolution and the hardware, so it is the same for both datasets. With respect to the increment in the number of epochs, it is $\approx 32\%$ for the Stanford cars dataset and $\approx 53\%$ for the FGVC-Aircraft dataset. Nevertheless, it is worth noting that, for application purposes, this computational time is not decisive since the training procedure is carried out only once before the model is put into production, after defining its architecture and setting its hyper-parameters. As our method is applied during the learning process, the computation time in the test phase is not affected.

4.5.2 Experiment 2

Table 3 shows the results obtained when applying our training scheme to the other two backbone architectures selected: InceptionV3 and DenseNet. According to the figures, our approach outperforms the corresponding baseline for both datasets regardless of backbone considered. While analyzing the behavior of our training scheme when using InceptionV3, we can observe that it achieves an improvement of more than 1 percent for the four performance

	FT-ResNet50		0-occlusion	
	Standford cars	FGVC-Aircraft	Standford cars	FGVC-Aircraft
Number of epochs	98.8 ± 6.78	113.4 ± 10.97	130 ± 10.38	174 ± 13.87
Seconds per epoch*	153 ± 0.00		175 ± 0.00	

* Network input size: $224 \times 224 \times 3$. Hardware: NVIDIA T4 Tensor Core GPU.

Table 2: Number of epochs and seconds per epoch, averaged across five runs, needed to train the two different approaches on the Stanford cars [11] and FGVC-Aircraft [12] datasets.

measures. In terms of accuracy, this improvement over the baseline is of 1.3 percent on the Stanford cars dataset and 1.5 percent on the FGVC-Aircraft dataset. Regarding the DenseNet backbone, the improvement with respect to the baseline is about 1.1 percent for all the metrics on both datasets.

Stanford cars				
	FT-InceptionV3	0-occlusion-InceptionV3	FT-DenseNet	0-occl-DenseNet
Accuracy	0.778 ± 0.023	$\mathbf{0.791 \pm 0.020}$	0.883 ± 0.010	$\mathbf{0.894 \pm 0.011}$
Precision	0.788 ± 0.021	$\mathbf{0.798 \pm 0.020}$	0.888 ± 0.009	$\mathbf{0.898 \pm 0.011}$
Recall	0.777 ± 0.023	$\mathbf{0.791 \pm 0.020}$	0.882 ± 0.010	$\mathbf{0.893 \pm 0.012}$
F1	0.776 ± 0.023	$\mathbf{0.790 \pm 0.021}$	0.882 ± 0.010	$\mathbf{0.893 \pm 0.012}$
FGVC-Aircraft				
	FT-InceptionV3	0-occlusion-InceptionV3	FT-DenseNet	0-occl-DenseNet
Accuracy	0.618 ± 0.029	$\mathbf{0.633 \pm 0.026}$	0.767 ± 0.026	$\mathbf{0.780 \pm 0.025}$
Precision	0.630 ± 0.030	$\mathbf{0.641 \pm 0.029}$	0.786 ± 0.024	$\mathbf{0.794 \pm 0.023}$
Recall	0.618 ± 0.028	$\mathbf{0.633 \pm 0.026}$	0.767 ± 0.026	$\mathbf{0.780 \pm 0.025}$
F1	0.616 ± 0.029	$\mathbf{0.630 \pm 0.026}$	0.768 ± 0.026	$\mathbf{0.780 \pm 0.025}$

Table 3: Classification performance, averaged across five runs, making use of different backbones on the Stanford cars [11] and FGVC-Aircraft [12] datasets. Best results are in bold.

5 Case study

This section describes an application of the proposed method to a real-world scenario. In particular, we consider the task of food-related scene classification in egocentric images, as detailed below.

5.1 Dataset

We evaluated our proposed method on the EgoFoodPlaces dataset [15], which is composed of 33,810 egocentric images gathered by 11 users and organized in 15 food-related scene classes. By making use of a wearable camera[3], the

[3] `http://getnarrative.com/`

users regularly recorded an amount of approximately 1,000 images per day. The camera movements and the wide range of different situations that the users experienced during their days, lead to challenges such as background scene variation or changes in lighting conditions. The dataset was manually labeled into a total of 15 different food-related scene classes namely, *bakery shop, bar, beer hall, cafeteria, coffee shop, dining room, food court, ice cream parlor, kitchen, market indoor, market outdoor, picnic area, pub indoor, restaurant, and supermarket.* Table 4 depicts the distribution of images among the collected classes, with a great imbalance between them.

Class	Bakery shop	Bar	Beer hall	Cafeteria	Coffee shop	Dining room	Food court	Ice cream parlor	Kitchen	Market indoor	Market outdoor	Picnic area	Pub indoor	Restaurant	Supermarket	Total
#Images	144	1632	672	1689	2313	3639	204	107	3837	1181	1388	921	511	10310	5262	**33810**

Table 4: Distribution of images per class in the EgoFoodPlaces dataset [15].

5.2 Experimental results

This section describes the results obtained when addressing the task of food-related scene classification with our proposed training scheme.

The implementation details are the ones described in Section 4.2 with two exceptions: (1) the resolution of the input images, which in this case is 250×250 as in [15]; and (2) the application of class oversampling to the fourth largest class (i.e., *dining room*) in order to alleviate the imbalance problem.

Concerning the experimentation, we used the split described in [15], which includes a division into events for the training and evaluation phases, to make sure that there are no images from the same scene/event in both phases. The validation procedure, in this case, consisted of three partitions, with the following distribution: training set (70%), validation set (10%), and test set (20%). Then, the model was trained and evaluated on the test set. This validation procedure was repeated five times. We report the average performance and the standard deviation calculated across the five runs.

Finally, we considered the four performance metrics detailed in Section 4.4: accuracy, precision, recall, and F1 score. Note that, for the per-class metrics (precision, recall, and F1), we calculated the macro- and weighted-averages, as suggested in [15]: *macro* gives equal weight to all classes, while *weighted* is sensitive to imbalances. It is worth noting the relevance of these two average values due to the unbalanced nature of the dataset.

5.2.1 Classification performance

For the evaluation of our proposal, we followed the experimental setup described in Section 4.3, but using the EgoFoodPlaces dataset to train the ResNet50 architecture with the classical procedure (FT-ResNet50) and with our training scheme (0-occlusion). Additionally, we compared our results with the ones reported in [15], the state-of-the-art approach for this dataset.

Table 5 reports the results obtained for the different approaches. As can be seen, training a ResNet50 with our proposed scheme (0-occlusion) allows us to achieve a higher accuracy than the one obtained with the baseline (FT-ResNet50). Moreover, the proposed method also achieves higher weighted averages for the other three metrics considered (precision, recall, and F1). It is worth noting that, due to the high imbalance of the dataset, the weighted metrics are more informative than the macro values. Concerning the latter, the differences between both methods are minimal, with the same values for precision and F1, and a slightly higher macro recall in favor of the baseline.

	Hierarchical approach [15]	FT-ResNet50	0-occlusion
Macro Precision	0.56	**0.59 ± 0.03**	**0.59 ± 0.05**
Macro Recall	0.53	**0.55 ± 0.03**	0.54 ± 0.06
Macro F1	**0.53**	**0.53 ± 0.04**	**0.53 ± 0.06**
Weighted Precision	0.65	0.67 ± 0.02	**0.68 ± 0.03**
Weighted Recall	**0.68**	0.67 ± 0.03	**0.68 ± 0.04**
Weighted F1	0.65	0.64 ± 0.03	**0.66 ± 0.04**
Accuracy	**0.68**	0.67 ± 0.03	**0.68 ± 0.04**

Table 5: Classification performance, averaged across five runs, of the different approaches on the EgoFoodPlaces dataset [15]. Best results are in bold.

If we analyze the results achieved by the state-of-the-art [15] and compare them with the proposed method, we can see that our approach achieves better results in four out of the seven performance measures, whereas the remaining three are equal. We find important to point out that our approach makes use of only just one classifier (ResNet50), while the model presented in [15] uses a hierarchical ensemble composed of six VGG16 networks. Therefore, the complexity of our model is significantly lower, not only because we have one single classifier but also because our backbone model ResNet50 has a lower number of parameters than their VGG16 networks. Therefore, we can conclude that our proposed method is able to achieve similar performance results with a less complex architecture and a computationally less expensive approach.

Finally, the impact of the different approaches on the individual classes is presented in Table 6. As can be seen, our method (0-occlusion) shows a behavior very similar to the baseline approach (FT-ResNet50), with slightly higher rates in seven classes and three ties. Analyzing the figures obtained with the hierarchical approach [15], our method achieves better results in eight classes. More specifically, the results in which our approach outperforms the

state-of-the-art correspond to the four most represented classes (*restaurant*, *supermarket*, *kitchen*, and *dining room*). Also noteworthy is the improvement achieved for the class *food court*, which could not be classified by the hierarchical model (true positive rate of 0.00). However, there are five classes for which the hierarchical model gets a better performance, including *beer hall*, *cafeteria*, and *coffee shop*. We deduce that this is due to the benefits of classifying them in a hierarchical fashion.

Class	Bakery shop	Bar	Beer hall	Cafeteria	Coffee shop	Dining room	Food court	Ice cream parlor	Kitchen	Market indoor	Market outdoor	Picnic area	Pub indoor	Restaurant	Supermarket
Hierarchical app. [15]	0.39	0.31	**0.89**	**0.45**	**0.59**	0.58	0.00	**0.52**	0.89	**0.70**	0.28	0.00	**0.85**	0.70	0.85
FT-ResNet50	**0.63**	0.31	0.24	0.38	0.49	0.66	**0.53**	0.50	0.89	0.60	**0.57**	0.00	0.78	**0.73**	0.90
0-occlusion	0.60	**0.32**	0.26	0.35	0.48	**0.72**	0.43	**0.52**	**0.90**	0.60	0.53	0.00	0.80	**0.73**	**0.92**

Table 6: True positive rate per class, averaged across five runs, of the different approaches on the EgoFoodPlaces dataset [15]. Best results are in bold.

Going deeper into the results obtained and given the characteristics of the EgoFoodPlaces dataset, we can draw some additional conclusions. Firstly, we can observe that the classification improves when using our approach for (1) classes where the scene to recognize is right in front of the camera users (e.g., *restaurant*), and (2) classes that tend to share descriptors even if recorded at different locations (e.g., *dining room* or *supermarket*). Those results inherit that the model is able to learn the relevant features in the scene when it is self-contained, which is closely related to the fine-grained datasets evaluated in Section 4.

Analyzing the images we can also see that, in some classes (e.g., *food court*, *cafeteria*, *market outdoor*), there is more background than foreground information necessary for the identification of the scene; that is, the image is composed of characteristics that an observer would not find relevant for the distinction of an event. Therefore, the main difficulty in learning these scenes is that not only the locations vary but also they are composed of elements common to other scenes. In this case, including other relevant regions along with a limited amount of samples available per class might represent imposed noise and lead to a lower performance in our approach compared to the baseline. This issue could be addressed with the extension of the dataset.

5.2.2 Model inspection

We analyzed not only the classification performance of our training scheme but also its ability to make predictions. In particular, we aimed to find out if the proposed approach is able to improve the robustness of a CNN classifier and make it sensible to more features. For this reason, we carried out two additional experiments: (1) we analyzed the behavior of the models making

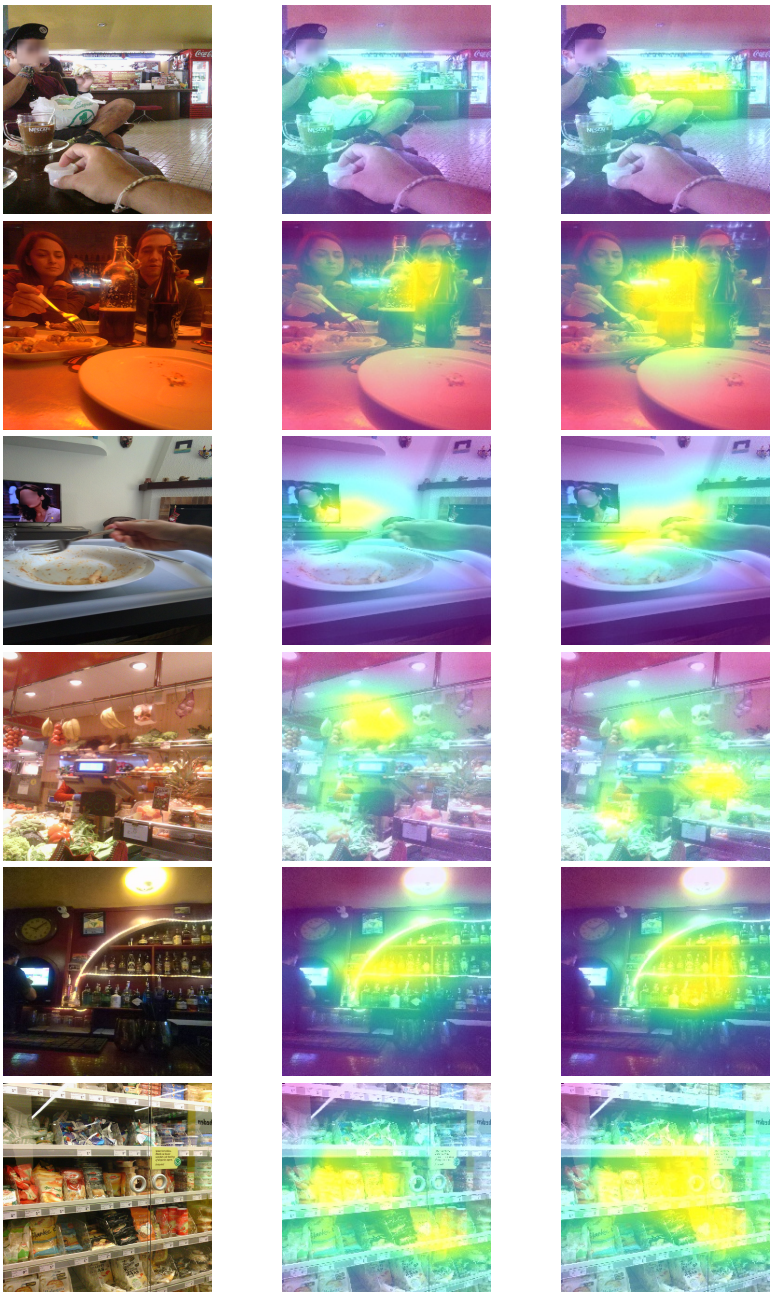
use of a visual explanation algorithm, and (2) we randomly erased some areas of the test images before evaluating the models on them.

In the first experiment, our target was to demonstrate that the regions considered as relevant by the trained models were more and bigger when applying our training scheme than when following the classical procedure. For this purpose, we applied the Grad-CAM algorithm to the last convolutional layer of the two ResNet50 models previously trained on the EgoFoodPlaces dataset: one trained using the classical procedure (FT-ResNet50), and the other one using our training scheme with 0-occlusion. As a result, we obtained the heat maps that allow us to visualize the regions that are important to the models when making a prediction for a given image. Figure 3 depicts some representative images along with their corresponding heat maps for each model. As can be observed, our model took into account bigger regions than the baseline method (FT-ResNet50) when processing the same target images. Besides, it can be seen that the model trained with our proposed method bases its decisions on more regions than when using the classical procedure. Furthermore, the regions that the baseline model took into account when making a decision were also taken into account by the proposed model. This demonstrates that when using the proposed training scheme, the model does not forget the learned features, but just learns to recognize other features.

Finally, we conducted the second experiment to test the robustness of our training scheme. For this purpose, we hid some regions of the test images by randomly erasing them, as proposed in [42]. After that, we compared how the two approaches (FT-ResNet50 and 0-occlusion) performed on the modified test set. Table 7 presents the results for this experiment. As can be observed, the proposed approach (0-occlusion) performs better than the baseline model (FT-ResNet50). This means that our model does not suffer as much when some areas of the image are erased or hidden, demonstrating its robustness. It is also worth noting that these results are consistent with the ones obtained in the previous experiment, and demonstrate that our model makes use of more and bigger regions than the baseline approach to make a prediction for a target image.

	FT-ResNet50	0-occlusion
Macro Precision	0.53 ± 0.01	$\mathbf{0.54 \pm 0.02}$
Macro Recall	0.47 ± 0.02	$\mathbf{0.48 \pm 0.03}$
Macro F1	0.47 ± 0.02	$\mathbf{0.48 \pm 0.05}$
Weighted Precision	$\mathbf{0.63 \pm 0.02}$	$\mathbf{0.63 \pm 0.03}$
Weighted Recall	0.59 ± 0.02	$\mathbf{0.65 \pm 0.03}$
Weighted F1	$\mathbf{0.59 \pm 0.02}$	$\mathbf{0.59 \pm 0.02}$
Accuracy	0.59 ± 0.02	$\mathbf{0.60 \pm 0.02}$

Table 7: Classification performance, averaged across five runs, of the baseline method and the proposed training scheme when we randomly hid some regions on the test images. Best results are in bold.



(a) (b) (c)

Figure 3: (a) Input images, (b) heat maps generated by Grad-CAM for the baseline FT-ResNet50, and (c) heat maps generated by Grad-CAM for the model trained with the proposed training scheme (0-occlusion).

6 Conclusion

This research work presents a novel training scheme that improves the robustness and generalization ability of CNNs applied to image classification. The idea is to force the model to learn as many features as possible when making a class selection. For this purpose, we apply a visual explanation algorithm to identify the areas on which the model bases its decisions. After identifying those areas, we occluded them and trained the model with a combination of the modified images and the original ones. In this manner, the model is not able to base its prediction on the occluded regions and is forced to use other areas. Consequently, the model also learns to pay attention to those regions of the target image that, *a priori*, are not so informative for its classification.

To evaluate the proposed method, we carried out different experiments on two popular datasets used for fine-grained recognition tasks: Stanford cars and FGVC-Aircraft. From the obtained results, we can confirm our initial hypothesis: our method forces the network to learn additional features that help it distinguish between very similar classes, showing its suitability for fine-grained classification problems. More specifically, and within the different evaluated configurations, the 0-occlusion approach has shown to be the most appropriate setting. Furthermore, we demonstrated the adequacy of our training scheme regardless of the backbone architecture considered.

We further experimented with a real-case study focused on the classification of food-related scenes. We analyzed the impact of our training scheme by comparing it with a baseline method and, to the best of our knowledge, with the state-of-the-art approach that follows an ensemble composed of six CNNs [15]. The results achieved with our method were comparable or even better than the ones obtained with the state-of-the art approach despite making use of just one network, thus reducing the level of complexity while maintaining a competitive performance. Furthermore, our method is computationally less expensive, as the chosen backbone (ResNet50) has fewer parameters than the VGG16 used in [15]. Finally, we carried out several occlusion and visual explanation experiments, showing that our method improves the robustness of the classifier by forcing it to base its decisions on more features.

As a future line of research, it would be interesting to apply the same methodology not only to input images but also at different convolutional levels, as it is usually done with the regularization technique known as dropout. In other words, the feature maps obtained at different levels could be analyzed and occluded in the same way that we did with the input images. This idea would force the model to pay attention to different characteristics on the feature maps, thereby improving the robustness of the model at different levels of the learning process.

Acknowledgements We would like to thank the Center for Information Technology of the University of Groningen for their support and for providing access to the Peregrine high performance computing cluster.

Conflict of interest

The authors declare that they have no conflict of interest.

References

1. B.A. Richards, T.P. Lillicrap, P. Beaudoin, Y. Bengio, R. Bogacz, A. Christensen, C. Clopath, R.P. Costa, A. de Berker, S. Ganguli, et al. A deep learning framework for neuroscience. Nature Neuroscience **22**(11), 1761 (2019)
2. S. Khan, N. Islam, Z. Jan, I.U. Din, J.J.C. Rodrigues. A novel deep learning based framework for the detection and classification of breast cancer using transfer learning. Pattern Recognition Letters **125**, 1 (2019)
3. J. Wu, S. Shin, C.G. Kim, S.D. Kim. Effective lazy training method for deep q-network in obstacle avoidance and path planning. IEEE International Conference on Systems, Man, and Cybernetics pp. 1799–1804 (2017)
4. J. Xu, Z. Zhang, T. Friedman, Y. Liang, G. Broeck. A semantic loss function for deep learning with symbolic knowledge. International Conference on Machine Learning pp. 5502–5511 (2018)
5. K. Zhang, L. Zheng, Z. Liu, N. Jia. A deep learning based multitask model for network-wide traffic speed prediction. Neurocomputing **396**, 438 (2020)
6. D.C. Luvizon, D. Picard, H. Tabia. 2D/3D Pose Estimation and Action Recognition Using Multitask Deep Learning. IEEE Conference on Computer Vision and Pattern Recognition pp. 5137–5146 (2018)
7. X. Li, W. Zhang, Q. Ding. Understanding and improving deep learning-based rolling bearing fault diagnosis with attention mechanism. Signal Processing **161**, 136 (2019)
8. D.K. Jain, R. Jain, Y. Upadhyay, A. Kathuria, X. Lan. Deep refinement: capsule network with attention mechanism-based system for text classification pp. 1839–1856 (2020)
9. W. Samek, T. Wiegand, K.R. Müller. Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models. arXiv preprint arXiv:1708.08296 (2017)
10. A. Vellido. The importance of interpretability and visualization in machine learning for applications in medicine and health care. Neural Computing and Applications pp. 1–15 (2019)
11. J. Krause, M. Stark, J. Deng, L. Fei-Fei. 3D Object Representations for Fine-Grained Categorization. 4th International IEEE Workshop on 3D Representation and Recognition pp. 554–561 (2013)
12. S. Maji, E. Rahtu, J. Kannala, M. Blaschko, A. Vedaldi. Fine-Grained Visual Classification of Aircraft. arXiv preprint arXiv:1306.5151 (2013)
13. D. Damen, H. Doughty, G. Maria Farinella, S. Fidler, A. Furnari, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price, et al. Scaling Egocentric

Vision: The EPIC-KITCHENS Dataset. European Conference on Computer Vision pp. 720–736 (2018)
14. M. Bolanos, M. Dimiccoli, P. Radeva. Toward storytelling from visual lifelogging: An overview. IEEE Transactions on Human-Machine Systems **47**(1), 77 (2016)
15. E. Talavera, M. Leyva-Vallina, M.M.K. Sarker, D. Puig, N. Petkov, P. Radeva. Hierarchical approach to classify food scenes in egocentric photo-streams. IEEE Journal of Biomedical and Health Informatics **24**(3), 866 (2019)
16. O. Gelonch, N. Cano, M. Vancells, M. Bolaños, L. Farràs-Permanyer, M. Garolera. The effects of exposure to recent autobiographical events on declarative memory in amnestic mild cognitive impairment: A preliminary pilot study. Current Alzheimer Research **17**(2), 158 (2020)
17. M.K. Sarker, H.A. Rashwan, E. Talavera, S. Furruka Banu, P. Radeva, D. Puig, et al. MACNet: Multi-scale Atrous Convolution Networks for Food Places Classification in Egocentric Photo-streams. European Conference on Computer Vision Workshops pp. 1–11 (2018)
18. A.B. Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-López, D. Molina, R. Benjamins, et al. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. Information Fusion **58**, 82 (2020)
19. A. Bennetot, J.L. Laurent, R. Chatila, N. Díaz-Rodríguez. Towards explainable neural-symbolic visual reasoning. IJCAI Neural-Symbolic Learning and Reasoning Workshop (2019)
20. B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, A. Torralba. Learning deep features for discriminative localization. IEEE Conference on Computer Vision and Pattern Recognition pp. 2921–2929 (2016)
21. R.R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra. Grad-CAM: Visual explanations from deep networks via gradient-based localization. IEEE International Conference on Computer Vision pp. 618–626 (2017)
22. M.D. Zeiler, R. Fergus. Visualizing and understanding convolutional networks. European Conference on Computer Vision pp. 818–833 (2014)
23. M.T. Ribeiro, S. Singh, C. Guestrin. Why should I trust you? Explaining the predictions of any classifier. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining pp. 1135–1144 (2016)
24. D. Smilkov, N. Thorat, B. Kim, F. Viégas, M. Wattenberg. SmoothGrad: removing noise by adding noise. arXiv preprint arXiv:1706.03825 (2017)
25. A. Schöttl. A light-weight method to foster the (Grad) CAM interpretability and explainability of classification networks. International Conference on Advanced Computer Information Technologies pp. 348–351 (2020)
26. B. Cancela, V. Bolón-Canedo, A. Alonso-Betanzos, J. Gama. A scalable saliency-based feature selection method with instance-level information. Knowledge-Based Systems **192**, 105326 (2020)
27. P. Herruzo, L. Portell, A. Soto, B. Remeseiro. Analyzing first-person stories based on socializing, eating and sedentary patterns. International

Conference on Image Analysis and Processing pp. 109–119 (2017)
28. M. Aghaei, M. Dimiccoli, P. Radeva. With whom do I interact? Detecting social interactions in egocentric photo-streams. International Conference on Pattern Recognition pp. 2959–2964 (2016)
29. E. Talavera, A. Glavan, A. Matei, P. Radeva. Eating Habits Discovery in Egocentric Photo-streams. arXiv preprint arXiv:2009.07646 (2020)
30. A. Furnari, G.M. Farinella, S. Battiato. Temporal segmentation of egocentric videos to highlight personal locations of interest. European Conference on Computer Vision pp. 474–489 (2016)
31. L.C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, A.L. Yuille. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. IEEE Transactions on Pattern Analysis and Machine Intelligence **40**(4), 834 (2017)
32. J. Kim, J. Kwon Lee, K. Mu Lee. Accurate image super-resolution using very deep convolutional networks. IEEE Conference on Computer Vision and Pattern Recognition pp. 1646–1654 (2016)
33. O. Dekel, R. Gilad-Bachrach, O. Shamir, L. Xiao. Optimal distributed online prediction using mini-batches. Journal of Machine Learning Research **13**, 165 (2012)
34. D.P. Kingma, J. Ba. Adam: A method for stochastic optimization. 3rd International Conference on Learning Representations pp. 1–15 (2015)
35. M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al. Tensorflow: A system for large-scale machine learning. USENIX Symposium on Operating Systems Design and Implementation pp. 265–283 (2016)
36. F. Chollet, et al. Keras. `https://keras.io` (2015)
37. K. He, X. Zhang, S. Ren, J. Sun. Deep residual learning for image recognition. IEEE Conference on Computer Vision and Pattern Recognition pp. 770–778 (2016)
38. J. Deng, W. Dong, R. Socher, L.J. Li, K. Li, L. Fei-Fei. ImageNet: A large-scale hierarchical image database. IEEE Conference on Computer Vision and Pattern Recognition pp. 248–255 (2009)
39. C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna. Rethinking the inception architecture for computer vision. IEEE conference on computer vision and pattern recognition pp. 2818–2826 (2016)
40. G. Huang, Z. Liu, K.Q. Weinberger, L. van der Maaten. Densely connected convolutional networks. IEEE Conference on Computer Vision and Pattern Recognition pp. 4700–4708 (2017)
41. G.E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, R.R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint arXiv:1207.0580 (2012)
42. Z. Zhong, L. Zheng, G. Kang, S. Li, Y. Yang. Random Erasing Data Augmentation. AAAI Conference on Artificial Intelligence pp. 13,001–13,008 (2020)