Exploring Use of Objects in Image Classification

Helena Montenegro
FEUP, MIEIC
Porto, Portugal
up201604184@fe.up.pt

Juliana Marques
FEUP, MIEIC
Porto, Portugal
up201605568@fe.up.pt

***Abstract*—Deep learning has become the state-of-the-art in tasks regarding multi-dimensional data, such as image classification, thanks to its capacity to learn how to extract features from such data. As deep learning models try to mimic the behavior of the human brain with the use of deep artificial neural networks, to understand how good they are at achieving this task, we need to compare their reasoning in the decision-making process with that of a human. In specific, in the context of image classification, there is a need to evaluate how objects contained in images are used by these models to achieve a decision. To accomplish this purpose, first, we perform baseline experiments to study how classification networks may implicitly use objects contained in images to achieve a prediction, using a gradient-based method of interpretability to verify which parts of an image contribute the most to the prediction. Then, we explicitly provide the type of objects contained in the images to the classifier to evaluate whether providing this information improves its performance. In this last experiment, we use an object detection method to extract information about the objects contained in the images. With this approach, we arrived at the conclusions that normal image classification networks implicitly use objects contained in images in their predictions and that their performance can be improved by explicitly providing information about the objects, from which we can infer that there is still room for improvement in the feature extraction process done by a basic classification network.**

***Index Terms*—Interpretability, Deep Learning, Computer Vision**

I. Introduction

The appearance of deep learning has triggered significant progress in the area of machine learning, specially when it comes to tasks involving multi-dimensional data, such as images or video. Deep learning has shown great potential in tasks such as image classification thanks to its capacity to perform automatic feature extraction in the same network that performs the target task, which allows to optimize the extraction process to fit the task, whereas in traditional machine learning feature extraction has to be a separate and possibly manual process. As such, deep classification networks have become the state-of-the-art for image classification, having achieved great results that sometimes even surpass the human capacity, as is the case of a classification network for breast cancer screening, recently developed by McKinney *et al.* [1], whose predictions surpass those of human experts.

Deep learning models intend to mimic the behavior of the human brain through the use of deep artificial neural networks composed by units, representative of neurons, and links between units which propagate signals, similarly to synapses. In order to understand how good these models are at copying the human brain, there is a need to analyze the similarity between the reasoning of these models and the reasoning of a human, in the decision-making process.

One problem inherent to deep learning models is that it's difficult to interpret or even replicate the computations made by a network during training or to achieve a decision. As we don't know the reasons that led to a decision, it's difficult to assess whether the features that a model has learnt are relevant to the respective domain. Even though these models achieve high levels of accuracy on a specific dataset, they might fail when applied to real scenarios due to a misguided feature extraction process, which focus on context rather than on the objects or parts of an image that are relevant to a domain. As such, lack of interpretability leads to a lack of trust in the models' predictions and to a consequent lack of acceptance of these models in real scenarios, specially when wrong decisions have significant consequences [2]. To address these problems, interpretability in deep learning has recently become a trending topic in the scientific community, leading to the development of various sorts of explanations to explain a model's decisions.

With this work, we intend to use interpretability methods to explore how objects contained in an image are used by image classification networks, in the context of indoor scenes. The human process of recognition of indoor scenes is often associated to the objects contained in the scene. For example, a bed is normally found in a bedroom, leading humans to infer that rooms with beds are bedrooms. As such, we seek to evaluate if image classifiers identify and use information about objects, in similarity to humans. More specifically, this work has two main goals:

- Verify whether a basic classification network implicitly uses the objects contained in an image to classify it.
- Assess whether we can improve the image classification results by explicitly providing information about the objects contained in the image, which are detected through an object detection network.

The following document is structured as follows: Section II presents a literature review focused on the main concepts inherent to the topic of this work: classification, interpretability and object detection. Section III describes the approach used to address the goals of this work, containing a description of the experimental setup and the obtained results. Section IV provides a discussion about the approaches used to achieve each goal and the respective results, reflecting on the limitations of this work. Section V presents the main conclusions of this work and possible future work.

II. Literature Review

Since this work is focused on classification, this section will start with an overview of methods to achieve this task, followed by a review of methods of interpretability, which can be used to determine whether objects are implicitly used in the decision-making process of an image classification network. As we need to obtain the objects contained in an image, to be able to provide them explicitly to a image classification network, we will also explore object detection methods in this section.

A. Image Classification

Image classification refers to the process of assigning a label to an image, according to its contents. Deep learning classification algorithms are currently the state-of-the-art, since they learn to extract features from images in the same network that performs the classification. These networks are often convolutional neural networks (CNN), which alternately use convolution and pooling layers.

Since image classification has been very studied in the scientific community, there are various standard models available, such as VGG [3], Inception [4] and ResNet [5], which have been trained on generic datasets like ImageNet [6] and whose weights can be reused in other domains through a process of transfer learning. The advantages of using transfer learning, in comparison to training a network from scratch, is that, since the models are pre-trained in generic data, the training process is much faster and leads to an improvement of the network's performance.

The CNN that we have used to achieve feature extraction is VGG-16 [3], which is composed by 16 weight layers: 13 convolution layers organized in 5 blocks at the end of which there is a pooling layer, followed by 3 fully-connected layers that perform the classification task based on the features extracted in the previous layers. In this work, the fully-connected layers were replaced by a new classifier which was trained from scratch to fit the specific classification task that it's meant to achieve.

B. Interpretability

Interpretability in deep learning aims to explain a model's behaviour or the reasons that led to its results in a way that is understandable to a human. Since in this work the goal of interpretability is to understand what parts of an image are used by a standard image classification network to classify it, this section will focus on methods that provide visual explanations.

One family of methods that achieve this purpose are based on sensitivity analysis, which consists in perturbing the input given to a network and verify if there are significant changes to the output, to infer the importance of a part of an image in the classification process. A possible perturbation can be the occlusion of portions of the images to see if the respective classification scores go down [7, 8].

Another type of methods are the gradient-based methods [9–13], which use information about the gradients to identify the parts of an image that contribute the most to a decision. The method that we've implemented is the Grad-CAM algorithm [11], which shows the parts of the image that contribute to a class score, through the calculation of the gradient of the class score y with respect to the feature map activations A seen in the last convolutional layer: $\frac{\partial y}{\partial A}$. This method is a generalization of CAM (Class Activation Mapping) [10], which identifies discriminative regions in images but that can only be applied to CNN which do not contain fully-connected layers.

One final method that has been introduced for visual explanations of image classification networks is deconvolution [7], which consists in developing a deconvolution network composed by deconvolution and unpooling operations, representing the inverse of the original convolutional neural network. By applying the deconvolution to the features obtained in the original network, it's possible to reconstruct and visualize the features that the network has learnt. The problem with this method, in comparison with Grad-CAM, is that it doesn't allow to understand which of the learnt features were used for the classification score of a specific class [11].

C. Object Detection

Object detection is a technique that allows us to determine where objects are in a given image or video (object localization) and which category each object belongs to (object classification).

In the last years, the most significant improvements in object detection are due to the use of deep learning techniques, namely convolutional neural networks (CNN), that have already been proved to be the best-known models to perform this task, as they are capable of automatically learning objects' inherent features and correctly identify their intrinsic concepts.

The object detection field has been mainly dominated by two different approaches: one stage and two-stage detectors.

Two-stage detectors achieve great accuracy but lower speed, because they require firstly to refine proposals to obtain the features needed to classify the objects. This type of approach is marked by the R-CNN [14] architecture followed by its variants Fast R-CNN [15], Faster-RCNN [16] and by Mask R-CNN [17].

One-stage detectors significantly reduce computational cost, achieving real-time performance while still maintaining a high accuracy. The most relevant detectors of this kind of approach are the YOLO [18] and its following variants YOLOv2 [19], YOLOv3 [20], YOLOv4 [21] and SSD [22].

The model that we have used to perform object detection is the YOLOv4. It is composed of three parts:

- A backbone that is pre-trained on ImageNet [6], and in this case, the authors chose the CSPDarknet53 [23].
- A neck which is used to collect feature maps from different stages, they used SPP [24] and PAN [25].

- Lastly, a head is used to predict classes and the object's bounding boxes. In this model, they used the previous detector, YOLOv3 [20].

III. Approach

To achieve our goal of verifying whether classification classifies images according to the objects contained in them, we developed a classification network with a gradient-based method, Grad-CAM [11], which highlights the parts of the images that contribute the most to a prediction. Then we developed an improved classification network that takes as inputs features extracted from an image as well as data about the objects contained in an image, obtained through an object detection network. Both these networks were developed using Keras [26], with Tensorflow backend [27].

In this section, we will detail the process of preparing the dataset used and the approaches taken for the development of the baseline experiment with an image classifier and Grad-CAM, and of the final experiment with the improved image classification network.

A. Data preparation

We built a dataset composed by indoor images, extracted from Open Images [28]. This database contains annotations for object detection for many different objects. During the data collection process, we selected a set of house divisions: WC, Bedroom and Kitchen, and a set of objects that are usually found in each of these divisions: toilet, bathtub, window, sink, bed, oven and refrigerator. As can be seen in Figure 1, there are objects that can be found in multiple divisions, such as window and sink.

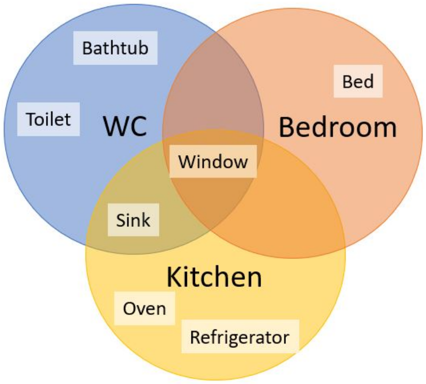


Fig. 1: Organization of objects per house division.

To extract images that possess these objects and the respective annotations from the Open Images Dataset we used OIDv4 ToolKit [29], which is a tool developed for the purpose of extracting images that belong to specific classes of objects from the Open Images Dataset. After obtaining the images, we went through a manual process of identifying whether they were taken from one of the specified house divisions, removing any image that wasn't, and we separated the images into three classes according to the respective house divisions. At the end of this process, we had a small dataset with indoor images that contained two ground truths: the labels corresponding to the respective house divisions and the coordinates of the bounding boxes of the objects contained in the images.

After collecting and cleaning the data, we have analyzed it to verify whether it's balanced and to visualize other characteristics. The dataset is composed by 535 images. Regarding its classes, the dataset is well balanced with small differences between the number of images per class, as can be seen in Figure 2.

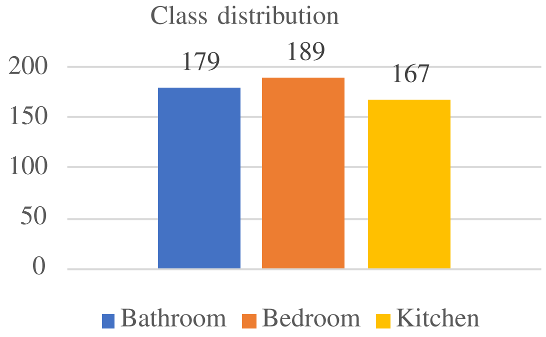


Fig. 2: Class distribution.

Regarding object distribution in the images, the dataset is not balanced, as it contains a lot of images with beds and few images with windows and bathtubs, as can be seen in Figure 3. Each image has at least one object and can have a maximum of 5 objects.

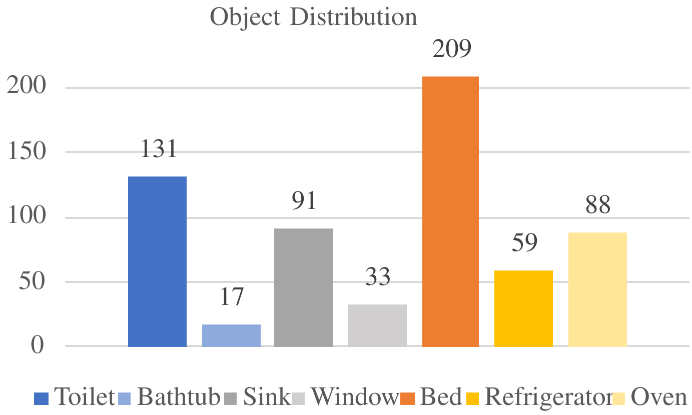


Fig. 3: Object distribution.

We split the data in 70% for training, 10% for validation and 20% for testing.

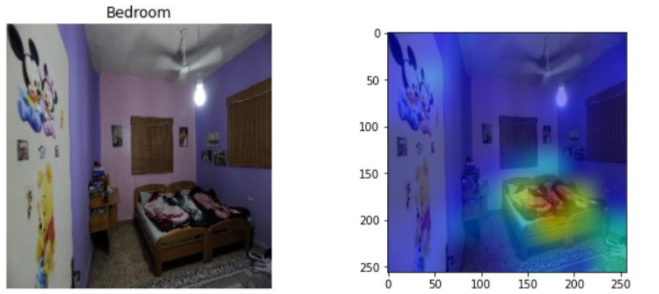
B. Baseline Experiment: Basic classification with Grad-CAM

We have developed an image classification network based on the VGG-16 network [3]. The network is composed by the VGG-16 network, responsible for extracting features from the images, followed by two fully-connected layers that use softmax as the activation function, responsible for the classification. During training, we started by freezing the layers of the VGG-16 network so that the respective weights are not updated during training, with the goal to train only the two fully-connected layers. Then, after 30 epochs, we have unfreezed the top layers of the VGG-16 network and resumed training for another 30 epochs, in a process of "fine-tuning" that aims to improve the features obtained in this network to be more relevant for this specific classification task. To avoid overfitting, we have used early stopping so that the network stops learning after a few epochs without significant improvements, and we added a dropout layer between the fully connected layers. Since the dataset is very small, we used data augmentation to increase the amount of data by performing slight alterations to the existing data. After training we have achieved very positive results with an accuracy on the testing set of 91,46%, together with the results in Table I, from which we can infer that the classifier is better at classifying bathrooms and bedrooms than kitchens, as can be seen by the respective F1 measures.

TABLE I: Results for baseline classification network.

Class	Precision	Recall	F1-Score
Bathroom	1.00	0.92	0.96
Bedroom	1.00	0.86	0.93
Kitchen	0.74	1.00	0.85

We applied a gradient-based method for interpretability to this network, in order to evaluate whether a normal image classification network uses the objects contained in the images for its prediction. The method used was the Grad-CAM method, which creates a heatmap that, when overimposed on the image, highlights the parts of the image that contribute the most to the its classification, as can be seen in Figure 4, whose prediction was "Bedroom".



(a) Original image.

(b) Image with highlighted feature importance.

Fig. 4: Example of application of Grad-CAM.

Using Grad-CAM, we have evaluated the extent to which the developed classification network uses the objects contained in the image to classify it. For this, we have developed two metrics:

- Percentage of important pixels that are inside a bounding box that belongs to an object.
- Percentage of the pixels inside a bounding box which are important.

We have considered important pixels to be those whose magnitude in the heatmap obtained with Grad-CAM, which is overimposed on the image, is higher than a threshold. The values of the heatmap vary between 0 and 1. As such, we have defined the threshold to be "0.3". Using these metrics we have analyzed how each individual object is used in classification. On average, an object's bounding box occupies around 30% of an image, which means that if the percentage of important pixels that are inside a bounding box is significantly higher that 30%, then we can consider that the respective object contributes to the classifier's decision.

In Figure 5, we have developed a bar graph which represents the defined metrics relative to each object. The only object that does not contribute at all to the prediction is ”Window”, which is the only object that is common to all classes and, therefore, has low discriminative power. The object ”Refrigerator” possesses the highest discriminative power as 86% of the important pixels are inside its bounding box. In general, for most of the objects, the images possess more than half percent of their important pixels inside bounding boxes, which leads to the conclusion that these objects are implicitly used by the classification network in its predictions.

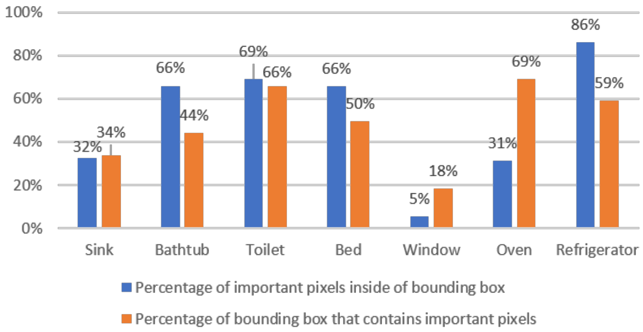


Fig. 5: Evaluation of object importance in classification.

We have also analyzed the same metrics grouped according to each class. As can be seen in Figure 6, on average, around 50% of the important pixels in an image are contained in a bounding box of the objects represented, with the class ”Kitchen” possessing lower percentage of pixels inside bounding boxes. This class is also the one that provided the worst classification results, characterized by the lowest F1 measure on Table I, which points to the conclusion that the network's inability to implicitly recognize objects normally found in kitchens negatively affects its performance. Since the average value of the important pixels inside bounding boxes is higher than 30%, respective to the percentage of the image occupied by bounding boxes, we can assert that objects are used implicitly in the classification, with "Kitchen" being the class where the network is worse at identifying the objects contained in the image. From this bar graph, we can also verify that, on average, around half of the bounding boxes are covered by important pixels.

With this baseline experiment we have achieved very good

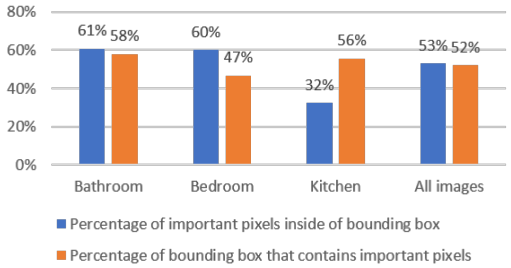


Fig. 6: Evaluation of object importance regarding each class.

results, with an accuracy that surpasses 90%, in a well-balanced dataset, whose baseline accuracy, equivalent to always predicting one class, would be around 33%. Through the analysis of the parts of the images that contribute the most to a decision, we can conclude that, even though objects are not explicitly given as input to the classifier, it is still able to recognize these objects through the process of feature extraction and use them to make a prediction.

C. Final Experiment: Improved Classification Network

We developed a classification network which takes as input both an image and the quantity of each object that it contains, in the form of tabular data. This process, as can be seen in Figure 7, was developed in two stages:

- **Stage 1**: Object detection to obtain the objects contained in an image.
- **Stage 2**: Classification using both image data and tabular data.

1) Object Detection: For the detection process, we used the implementation of the YOLOv4 model [21] provided by the authors in the Darknet Framework [30]. Table II contains an overview of the results obtained with this detector. We obtained an average intersection over union (IoU) of around 60%, which is fairly good since it means that the predicted bounding boxes overlap with more than 60% of the true bounding boxes, but it has room for improvement. The results of mean average precision are not satisfactory. The average precision results obtained for each object can be seen in Figure 8. In this graph, we can see that the detector has worse results for the objects ”Window” and ”Bathtub”, which correspond to the objects that are less represented in the database. The remaining objects present relatively good results, with average precision ranging between 78% and 90%. Some examples of results achieved with this detector can be seen in Figure 9.

Metric	Obtained Value (%)
Precision	81%
Recall	85%
F1 measure	83%
Average IoU	62.23%
Mean Average Precision (mAP@0.5)	71.16%

TABLE II: Results for object detection.

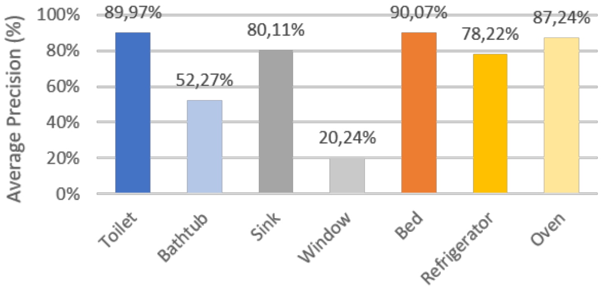
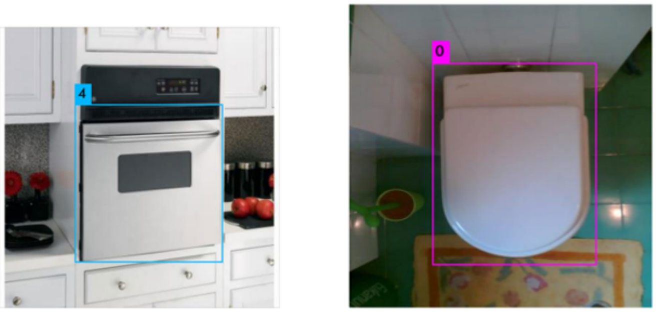


Fig. 8: Results of mean average precision for each object.



(a) Example of oven detected by object detector

(b) Example of toilet detected by object detector

Fig. 9: Example of results from object detection network.

2) Classification: For the classification process, we used a CNN, namely VGG-16 [3], to extract features from image data, similarly to the network developed in the baseline experiment. To process tabular data, we developed a Multi-Layer Perceptron (MLP) composed by two dense layers. We concatenated the results from both these networks which were then fed to a final dense layer, responsible for the classification.

At first, while the object detection network was still in development, we used the annotations of the objects in the images to build the tabular data. Using the annotations, which would be equivalent to having a perfect object detection network, the network achieved accuracy of 93,90%, along with the results expressed in Table III. These results are slightly better than the ones seen in the baseline classifier, whose accuracy was 91%, which proves that, with a perfect object detector, we can improve the results of an image classification network by providing the objects contained in the images as inputs.

TABLE III: Results for mixed-data classification network using real annotations.

Class	Precision	Recall	F1-Score
Bathroom	0.89	1.00	0.94
Bedroom	0.97	0.95	0.96
Kitchen	0.94	0.85	0.89

Using the developed object detection network, we achieved an accuracy of 91,46%, together with the results present on

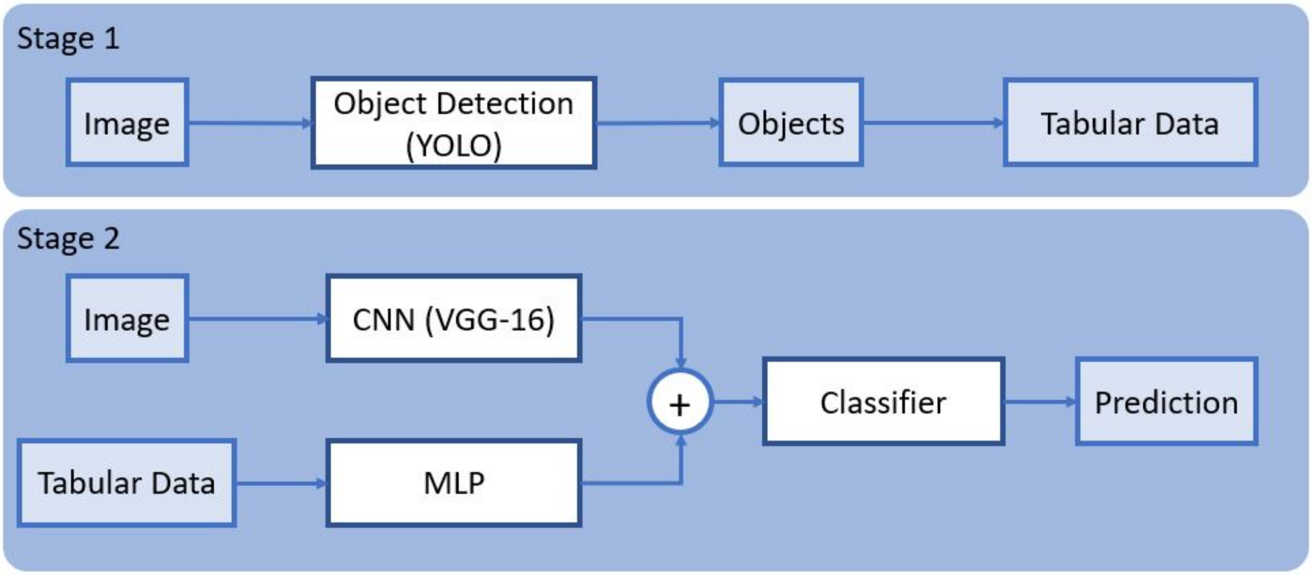


Fig. 7: Overview of the developed networks.

Table IV, which are similar to the results obtained in the baseline experiment.

TABLE IV: Results for mixed-data classification network using annotations obtained from object detection network.

Class	Precision	Recall	F1-Score
Bathroom	0.92	0.92	0.92
Bedroom	0.97	0.92	0.94
Kitchen	0.82	0.90	0.86

IV. Discussion

In the baseline experiment we have obtained satisfying results in terms of accuracy and performance of the network and we also arrived at the conclusion that the network does learn to identify some objects, which are used to achieve a decision, presenting a high percentage of important pixels inside an object's bounding box. Regarding the improved classifier, we have concluded that, with a perfect object detector, the image classification process can be improved.

Since the objects contained in an image are implicitly available in the images provided to the image classifier, the fact that providing them explicitly improves the results means that the feature extraction process of the baseline image classifier has room for improvement. Rather than facilitating the classification process by explicitly providing information that is implicitly available, forcing the network to learn to properly extract this information might have more benefits towards the improvement of deep learning algorithms.

This work has a limitation with respect to the dataset, given its small size and limited number of images for some of the objects, such as ”Bathtub” or ”Window”, whose images amount only to 17 and 33, respectively. This limitation has significantly affected the performance of the object detector, which has not achieved a satisfying level, specially regarding mean average precision, with very low average precision values for both the objects mentioned.

The quality of the object detector limits the improvement that a network can achieve by being provided with additional data. The fact that the results did not worsen even with an object detector with a low performance rate, may point towards two possible conclusions: the object detector provided with acceptable results that did not hinder the classification, or the classification network realizes during training that the image data is more reliable than the tabular data towards achieving accurate decisions, resulting in features extracted from image data weighing more in the final decision. The second conclusion would mean that the results of the classifier would not decrease beyond the results obtained with using only image data, independently of the object detector's performance. Further studies with object detectors of different capacity levels could be done in order to investigate which conclusion can be taken here.

V. Conclusions

In this work, we developed a basic image classifier and applied interpretability methods to understand whether its decision-process resembles that of a human, by taking into consideration the objects present in an image to classify it. We also explicitly provided the mentioned objects, obtained through an object detector, to a classification network to verify whether we could improve its results when compared to the basic image classifier.

We arrived at the conclusions that the basic image classification network implicitly learns to recognize objects, through the process of feature extraction, and uses them to achieve a prediction. By explicitly providing the objects contained in the image to a classification network, we can improve its results, which also means that the feature extraction process in the basic image classification can be improved.

As future work, the limitations of this work with respect to the dataset should be addressed, to provide better and more trustworthy results. Also, to understand if the results of the improved classifier don't drop beyond those of the baseline classifier, different object detectors trained at different levels, varying the number of epochs used, could be studied.

References

[1] S. McKinney, M. Sieniek, and V. e. a. Godbole, "International evaluation of an ai system for breast cancer screening," *Nature*, vol. 577, pp. 89–94, 2020. [Online]. Available: https://doi.org/10.1038/s41586-019-1799-6

[2] F. Doshi-Velez and B. Kim, "Towards a rigorous science of interpretable machine learning," 2017.

[3] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2015.

[4] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," 2016.

[5] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *arXiv preprint arXiv:1512.03385*, 2015.

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR09*, 2009.

[7] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Computer Vision – ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds., 2014, pp. 818–833.

[8] V. Petsiuk, A. Das, and K. Saenko, "Rise: Randomized input sampling for explanation of black-box models," 2018.

[9] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," 2014.

[10] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," 2015.

[11] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 618–626.

[12] D. Smilkov, N. Thorat, B. Kim, F. Viégas, and M. Wattenberg, "Smoothgrad: removing noise by adding noise," 2017.

[13] S. Srinivas and F. Fleuret, "Full-gradient representation for neural network visualization," 2019.

[14] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," 2014.

[15] R. Girshick, "Fast r-cnn," 2015.

[16] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," 2016.

[17] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," 2018.

[18] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," 2016.

[19] J. Redmon and A. Farhadi, "Yolo9000: Better, faster, stronger," 2016.

[20] ——, "Yolov3: An incremental improvement," 2018.

[21] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," 2020.

[22] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9905 LNCS, 2016.

[23] C.-Y. Wang, H.-Y. M. Liao, I.-H. Yeh, Y.-H. Wu, P.-Y. Chen, and J.-W. Hsieh, "Cspnet: A new backbone that can enhance learning capability of cnn," 2019.

[24] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *Lecture Notes in Computer Science*, p. 346–361, 2014. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-10578-9_23

[25] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," 2018.

[26] F. Chollet *et al.* (2015) Keras. [Online]. Available: https://github.com/fchollet/keras

[27] M. Abadi *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: http://tensorflow.org/

[28] A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Malloci, A. Kolesnikov, T. Duerig, and V. Ferrari, "The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale," *IJCV*, 2020.

[29] A. Vittorio, "Toolkit to download and visualize single or multiple classes from the huge open images v4 dataset," 2018. [Online]. Available: https://github.com/EscVM/OIDv4_ToolKit

[30] J. Redmon, "Darknet: Open source neural networks in c," http://pjreddie.com/darknet/, 2013–2016.