HOLMES: HOLonym-MEronym based Semantic inspection for Convolutional Image Classifiers

HOLMES: HOLonym-MEronym based Semantic inspection for Convolutional Image Classifiers

Francesco Dibitonto[1,2][0009−0006−7686−8860], Fabio Garcea[1][0000−0003−3460−5297], André Panisson[3][0000−0002−3336−0374], Alan Perotti[3][0000−0002−1690−6865], and Lia Morra[1][0000−0003−2122−7178]

[1] Department of Control and Computer Engineering, Politecnico di Torino, Italy
[2] EVS Embedded Vision Systems, Italy
[3] CENTAI Institute, Turin, Italy

Abstract. Convolutional Neural Networks (CNNs) are nowadays the model of choice in Computer Vision, thanks to their ability to automatize the feature extraction process in visual tasks. However, the knowledge acquired during training is fully sub-symbolic, and hence difficult to understand and explain to end users. In this paper, we propose a new technique called HOLMES (HOLonym-MEronym based Semantic inspection) that decomposes a label into a set of related concepts, and provides component-level explanations for an image classification model. Specifically, HOLMES leverages ontologies, web scraping and transfer learning to automatically construct *meronym* (parts)-based detectors for a given *holonym* (class). Then, it produces heatmaps at the meronym level and finally, by probing the holonym CNN with occluded images, it highlights the importance of each part on the classification output. Compared to state-of-the-art saliency methods, HOLMES takes a step further and provides information about both *where* and *what* the holonym CNN is looking at. It achieves so without relying on densely annotated datasets and without forcing concepts to be associated to single computational units. Extensive experimental evaluation on different categories of objects (animals, tools and vehicles) shows the feasibility of our approach. On average, HOLMES explanations include at least two meronyms, and the ablation of a single meronym roughly halves the holonym model confidence. The resulting heatmaps were quantitatively evaluated using the deletion/insertion/preservation curves. All metrics were comparable to those achieved by GradCAM, while offering the advantage of further decomposing the heatmap in human-understandable concepts. In addition, results were largely above chance level, thus highlighting both the relevance of meronyms to object classification, as well as HOLMES ability to capture it. The code is available at `https://github.com/FrancesCOde/HOLMES`.

Keywords: deep learning, machine learning, XAI, explainability, convolutional neural networks, computer vision.

1 Introduction

In recent years, the application of Machine Learning (ML) models has impacted the most disparate fields of application. In particular, Deep Learning (DL) models called Convolutional Neural Networks (CNNs) have become the de-facto standard approach to tackle Computer Vision (CV) problems, spanning from autonomous driving to image-based medical diagnosis, from satellite observation to advertisement-driven social media analysis [28].

Unfortunately, DL models are black-boxes, as their fully sub-symbolic internal knowledge representation makes it impossible for developers and users to understand the rationale behind the model decision process. This widely recognized limitation has multiple negative implications: (*i*) difficulty of adoption from domain experts [14], (*ii*) GDPR non-compliance [12], (*iii*) inability to detect learned spurious correlations [34], and (*iv*) risk of deploying biased models [22].

Due due this plethora of issues, the field of eXplainable Artificial Intelligence (XAI) has flourished in an attempt to make these black-box models more understandable from human developers and users [13].

In the specific case of CV tasks and CNN models, most XAI approaches are based on saliency and produce heatmaps [29], quantifying explanations in the form of *this image depicts a cat because of the highlighted region*. On the one hand, this approach can be sufficient to spot wrong correlations when the heatmap focuses on the wrong portion of the image, such as the background. On the other hand, a reasonably-placed heatmap is not a sufficient guarantee that the DL model is in fact implementing the desired task, and we argue that these shallow explanations are not enough for a human user to fully trust the algorithmic decision, nor for a developer to sufficiently debug a model in order to assess its learning progress. These approaches provide context-less label-level heatmaps: ironically, they pair deep models with shallow explanations. Conversely, when asked to justify an image-classification task, humans typically rely on the holonym-meronym (whole-part) relationship and produce part-based explanations, e.g. *this image depicts a cat (holonym), because there are pointy ears up there and a tail there, etc. (meronyms)*.

There is evidence that CNNs are capable of learning human-interpretable concepts that, although not explicitly labelled in the training set, are useful to detect classes for which labels are provided; for instance, scenes classification networks learn to detect objects present in scenes, and individual units may even emerge as objects or texture detectors [3]. At the same time, CNNs were shown to take shortcuts, relying on contextual or unwanted features for their final classification [8]; other works found CNNs being over-reliant on texture, rather than shape, for their final classification [9]. In this work, we tackle the important issue of how, and to what extent, post-hoc explanations can be linked to underlying, human-interpretable concepts implicitly learned by a network, with minimal effort in terms of annotation and supervision.

Our Research Question is therefore the following: ***can we decompose the given label (holonym) into a set of related con-***

cepts (meronyms), and provide component-level explanations for an image classification DL model?

In this paper we propose HOLMES (HOLonym-MEronym based Semantic inspection), a novel XAI technique that can provide explanations at a low-granularity level.

Given an input image of a given class, its parts (meronyms) are extracted from a Knowledge Base through the holonym-meronym (whole-part) relationship. Images depicting each part are either extracted from a densely annotated dataset or collected through Web scraping, and then used to train a meronym model through transfer learning. The resulting model is therefore a part dectector for the component of the image. The application of XAI techniques on the meronym model can thus produce part-based explanations. HOLMES can therefore highlight the occurrence and locations in the image of both labelled objects and their parts. We evaluated our approach through insertion/deletion/preservation metrics, showing how the parts highlighted by our approach are crucial for the predictions.

The rest of the paper is organized as follows. In Section 2 we connect the proposed technique with other existing approaches in the XAI literature. In Section 3 we go through the core concepts behind HOLMES. In Sections 4 and 5, we report experimental validation of the HOLMES pipeline. Finally, in Sections 6 and 7 we discuss advantages and limitations of the proposed approach, as well as future studies we plan to conduct to enhance HOLMES capabilities.

2 Related work

2.1 Feature Extraction and Transfer Learning

Deep Convolutional Neural Networks (CNNs) have been the de-facto standard models for computer vision in the last years [28]. These models typically encompass a number of convolutional layers, which act as feature extractors, followed by dense layers used for classification. The major drawback of these models is that, due to the large amount of parameters, training from scratch requires a vast amount of data and computational resources [33]. A common technique exploited to circumvent this problem is *transfer learning* [36], in which a model developed for a task is reused as the starting point for a model on a second task. The typical approach for CV tasks is to select a CNN that was pre-trained on the standard dataset of Imagenet [7], and reset and re-train the last dense layers on the new task. The underlying intuition of this approach is that CNNs learn a hierarchy of features in convolutional layers at different depths, starting from Gabor filters in the first layers to complex shapes in the last ones [36].

2.2 Interpretable and Explainable Machine Learning

The eXplainable Artificial Intelligence (XAI) research field tackles the problem of making modern ML models more human-understandable. XAI approaches

typically belong to one of two paradigms, namely, interpretability and post-hoc explainability[1,13]. Interpretable ML models are designed and trained in order to be, to some degree, passively transparent - that is, so that comprehensible information about the inner logic of the model is available without the application of other algorithms. Instead, explainability typically is performed *a posteriori* - it is a process that takes place after the ML model has been trained, and possibly even deployed. Explainability techniques apply external algorithms to the ML model in order to extract human-understandable information about the decision process that was produced by the training process.

Explainability methods can be further classified according to two orthogonal binary attributes: local/global and model agnostic/aware. Local methods provide an explanation for a single data point, while global methods aim to explain the behavior of the model as a whole, e.g., providing a joint explanation for all data points in the dataset. Model-agnostic methods can explain indifferently any type of black-box model, regardless of their typology or architecture, accessing input and outputs only. For instance, they could be applied even if the source code of the ML model is obfuscated or can be only accessed through APIs, provided that those can be invoked at will. Conversely, model-aware (also called model-specific) models exploit (and require access to) internal details of the black-box, such as gradients, and are therefore developed for specific kinds of ML models.

2.3 XAI for Computer Vision

Arguably the two most famous XAI approaches are LIME [24] and SHAP [17], both being local and model-agnostic. An important counterpoint in the field is the concept of global and model-specific approaches, as exemplified by TCAV [15]. This methodology allows for global interpretability, focusing on understanding high-level concepts used by the model across a broad set of inputs. However, for the specific task of computer vision, most approaches are model-aware and based on saliency.

When explaining image classification models, saliency methods compute pixel-level relevance scores for the model final output. These scores can be visualized as heat-maps, overlaid on the classified images, in order to be visually inspected by humans. One of these approaches is the Gradient-weighted Class Activation Mapping (Grad-CAM) [29], a model-aware, local, post-hoc XAI technique. Grad-CAM uses the gradient information flowing into the last convolutional layer of the CNN to assign importance values to each neuron for a particular decision of interest, such as a target concept like *dog*. By visualizing the positive influences on the class of interest (e.g., *dog*) through a global heatmap, Grad-CAM provides insight into which regions of the input image are 'seen' as most important for the final decision of the model. By overlaying this heatmap onto the input image, Grad-CAM facilitates a deeper understanding of the correlation between specific image features and the final decision.

Saliency maps methods such as Grad-CAM ask *where* a network looks when it makes a decision; the network dissection approach takes a step further and

asks *what* a network is looking for. In [3], the authors find that a trained network contains units that correspond to high-level visual concepts that were not explicitly labeled in the training data. For example, when trained to classify or generate natural scene images, both types of networks learn individual units that match the visual concept of a *tree* even though the network was never taught the tree concept during training. The authors investigate this phenomenon by first identifying which individual components strongly correlate with given concepts (taken from a labelled segmentation dataset), and then turn off each component in order to measure its impact on the overall classification task. Following this line of investigation, [41] seeks to distill the information present in the whole activation feature vector of a neural network's penultimate layer. It achieves this by dissecting this vector into interpretable components, each shedding light on different aspects of the final prediction. Our work differs from the network dissection literature in the following ways: (i) we allow for representations of concepts that are scattered across neurons, without forcing them to be represented by a single computational unit; (ii) we do not require additional, domain-specific ground truth sources, relying instead on web scraping and general purpose-ontologies and (iii) we do not focus on the specific scene recognition task, embracing instead the part-of relationships of labels in the more general image classification task.

2.4 Ontologies and Image Recognition

Ontologies, and structured representation of knowledge in general, are typically ignored in most DL for image processing papers [28]. However, there are notable exceptions where efforts have been made to merge sub-symbolic ML models together with ontologies.

In [10], the authors leverage the fact that ImageNet labels are WordNet nodes in order to introduce quantitative and ontology-based techniques and metrics to enrich and compare different explanations and XAI algorithms. For instance, the concept of semantic distance between actual and predicted labels for an image classification task allows to differentiate a labrador VS husky misclassification as milder with respect to a labrador VS airplane case.

In [25], the authors introduced a hybrid learning system designed to learn both symbolic and deep representations, together with an explainability metric to assess the level of alignment of machine and human expert explanations. The ultimate objective is to fuse DL representations with expert domain knowledge during the learning process so it serves as a sound basis for explainability.

Among the global methods for explainability, TREPAN [6] is able to distill decision trees from a trained neural network. By pairing an ontology to the feature space, the authors use the ontological depth of features as a heuristic to guide the selection of splitting nodes in the construction of the decision tree, preferring to split over more general concepts.

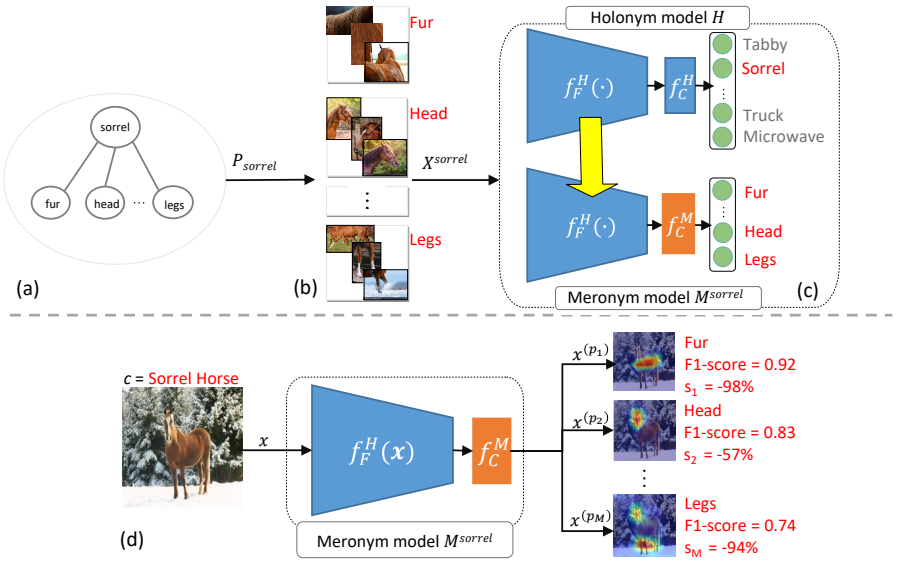


Fig. 1. HOLMES pipeline. Given an input image of class c, its parts (meronyms) are extracted from a Knowledge Base (a). Images depicting each part are either extracted from a densely annotated dataset or collected through Web scraping (b), and then used to train a meronym model by exploiting, through transfer learning, the implicit knowledge embedded in the original holonym model (c). The meronym model then produces part-based explanations, highlighting the most relevant parts for the class prediction (d).

3 Methodology

The proposed method, Holonym-Meronym based Semantic inspection (HOLMES), is a post-hoc approach that aims to explain the classification given by a CNN image classifier to an image in terms of its parts. It is indeed a model-dependent method, specifically tailored for CNNs. Hence, HOLMES takes as input the image whose class has been predicted, recovers its meronyms, and provides an explanation in terms of its parts.

Problem Formulation

Let us define the image classifier as a function $\mathcal{H} : \mathbf{x} \in \mathbb{R}^{h \times w \times ch} \mapsto c \in \mathcal{C}$, where $\mathbf{x}$ is an input image with dimensions $h \times w \times ch$ and $\mathcal{C}$ is the set of image classes. Lets assume that $\mathcal{H}$ is a CNN that can be expressed as a combination of two functions, a feature extractor $f_F^{\mathcal{H}} : \mathbf{x} \mapsto \mathbf{f}$ and a feed forward classifier $f_C^{\mathcal{H}} : \mathbf{f} \mapsto c$, where $\mathbf{f}$ is a feature vector. Let us define a holonym-meronym relationship mapping HolMe : $c \in \mathcal{C} \mapsto \{p_1, ..., p_n\}$ meaning that a whole object of class c, i.e. a holonym, is made of its parts, or meronyms $P_c = \{p_1, ..., p_n\}$. The goal of HOLMES is to explain the classification $\mathcal{H}(\mathbf{x}) = c$ through the meronyms

P_c, by highlighting which of the meronyms are important for the classification result. The explanation takes the form of a function $\xi : \mathbf{x} \mapsto \{(\mathbf{x}^{(p_i)}, s_i),\ p_i \in P_c\}$, where $\mathbf{x}^{(p_i)}$ is a saliency map that highlights the meronym p_i in the original image $\mathbf{x}$, and s_i is an explanation score associated to $\mathbf{x}^{(p_i)}$.

HOLMES solves this problem by training a new meronyms classifier $\mathcal{M}^c : \mathbf{x} \mapsto p_i \in P_c$ as a combination of the same feature extractor $f_F^{\mathcal{H}}$ that is part of the image classifier $\mathcal{H}$ and a new feed forward classifier $f_{P_c}^{\mathcal{M}} : \mathbf{f} \mapsto p_i \in P_c$. The meronyms classifier is then used to determine which parts p_i are present in the input image $\mathbf{x}$ and to create saliency maps that correspond to these parts. Finally, each saliency map $\mathbf{x}^{(p_i)}$ is used to create a mask on $\mathbf{x}$, which is then classified by $\mathcal{H}$, and the drop of the classifier confidence in the class c is used to determine the importance of the selected parts.

The HOLMES pipeline comprises the following steps:

A) **Meronyms Extraction**: given an image $\mathbf{x}$ and its predicted class c, this first step consists of retrieving the list of the object parts P_c (Fig 1(a)).
B) **Meronyms Image Data Collection**: once the object parts list P_c is available, a distinct dataset for each part shall be created in this second step (Fig 1(b)).
C) **Meronyms model Training**: in this third step, the auxiliary meronyms models $\mathcal{M}^c$ are trained to recognize the object parts P_c by exploiting the knowledge (about the parts) embedded in the original CNN $\mathcal{H}$ (Fig 1(c)).
D) **Explanations**: in this last step, a set of part-based explanations is produced, highlighting those parts which are most relevant for the class prediction (Fig 1(d)).

3.1 Meronyms Extraction

The first step of the pipeline consists of constructing the holonym-meronym relationship mapping HolMe : $c \in \mathcal{C} \mapsto P_c$ by retrieving the visible parts P_c associated to the holonym concept c. Hence, HOLMES relies on external Knowledge Bases (KBs) which include part-of relationships, i.e. containing class concepts (e.g., camel, horse, etc.) and their respective list of parts (e.g., head, legs, etc.), along with information about their visibility. Thus, for obtaining the parts of an holonym concept, HOLMES queries one selected knowledge base for the desired holonym concept and results its associated visible meronyms. Concepts that are not present in the chosen reference KB are mapped to the respective WordNet[19] ontology concepts, and the *hypernym/hyponym relationship* is exploited: the WordNet semantical hierarchy is climbed back up to the first hypernym (i.e., a broader class, like *bird* for *seagull*) which occurs in the reference KB, and its associated (more generic) parts are then assigned to the initial holonym concept.

The meronyms extracted in the previous step are then divided in two categories: *hyper-meronyms* and *hypo-meronyms*. Given a generic list of meronyms, $P = \{p_1, p_2, \ldots, p_n\}$, hypo-meronyms are parts whose visual space is completely

within any other part in P. The other parts whose visual space is not completely within any other part in P are the hyper-meronyms. For instance, for the holonym concept *cat*, the hyper-meronyms would be *head, legs, feet, tail*, given that none of them is visually contained in any other part, but rather, can only contain hypo-meronyms (e.g., *mouth, whiskers*, etc.).

For the final list of meronyms, only the hyper-meronyms are retained while hypo-meronyms are discarded. The final list of parts is thus defined as $P_c = \{p_1, p_2, \ldots, p_n\}$.

3.2 Meronyms Image Data Collection

Once the part set P_c for the target class c is available, the next step is to create a dataset $X^c = \{(\mathbf{x}_0, y_0), ..., (\mathbf{x}_n, y_n)\}$ where $\mathbf{x}_0, ..., \mathbf{x}_n$ are images corresponding to parts $y_0, ..., y_n \in P_c$. HOLMES can rely on a pre-existing labelled dataset, or it can exploit web image scraping to incrementally build a dataset for each meronym. In this scenario, HOLMES queries different web search engines for each part, by prefixing the holoynm first (e.g. *sorrel fur*, *sorrel head*, etc.) and downloads the associated images from each of those engines.

Due to the limited reliability of search engine results (discussed in Section 4), some obtained images could be still extraneous to the desired part concept. Moreover, duplicates could be present in the scraped parts' datasets. For these reasons, HOLMES integrates two additional sub-steps:

1. **deduplication**: duplicates are detected by means of the pHash[26] hash-based deduplication method, then they are removed from each meronym dataset.
2. **outlier removal**: meronyms images are mapped to a feature vector representation (e.g. using the output of the feature extractor or the activations of one of the feedforward layers of the classifier). The feature vectors are then fed to an outlier detection algorithm. The detected outliers are then removed from the meronym dataset.

3.3 Meronyms model Training

The training phase is the core of the HOLMES method. In this step the concept parts are visually learned, so that they can later be provided as explanations. This is achieved by training an auxiliary CNN model $\mathcal{M}^c$, trained and evaluated on the collected meronym dataset X^c (training and evaluation are performed in disjoint sets).

Let us recall that the goal of HOLMES is to explain the target holonym CNN $\mathcal{H}(\mathbf{x}) = \hat{y}$, where $\mathbf{x}$ is an holonym image of class c and $\hat{y}$ its predicted class. Let us also recall that the CNN can be expressed as a combination of two functions $\mathcal{H}(\mathbf{x}) = f_C^{\mathcal{H}}(f_F^{\mathcal{H}}(\mathbf{x}))$, where $f_F^{\mathcal{H}}(\cdot)$ is a feature extractor, and $f_C^{\mathcal{H}}(\cdot)$ is a feedforward classifier. Previous works already demonstrated that the units contained in the last convolutional layers of a CNN tend to embed objects, and more specifically, objects parts as well [11,3], and HOLMES leverages on this

fact to learn the parts by defining $\mathcal{M}^c(\mathbf{x}) = f^{\mathcal{M}}_{P_c}(f^{\mathcal{H}}_F(\mathbf{x}))$, where the feature extraction $f^{\mathcal{H}}_F(\cdot)$ is shared among the holonym $\mathcal{H}$ and meronym $\mathcal{M}^c$ models, whereas a feedforward classifier $f^{\mathcal{M}}_{P_c}(\cdot)$ is trained anew for each class c and each part list P_c.

The idea is to learn the parts concepts by using the same features learned by the original reference CNN model $\mathcal{H}$, such that the base knowledge for learning both the concept parts and the concepts themselves would be the same: effectively, HOLMES relies on transfer learning[37] for learning objects parts. Under the reasonable assumption that characteristic object parts, and consequently their associated features, are useful for the classification of the whole object itself, the same units which activate in the presence of the parts will also activate in presence of the object. For instance, a unit activating in the presence of a wheel, will also be likely to activate in the presence of a wheeled vehicle like a car. Hence, training $\mathcal{M}$ by keeping the feature extractor $f^{\mathcal{H}}_F$ intact will later allow us to understand if the knowledge about the parts was already available and embedded in the original model $\mathcal{H}$. Specifically, the feature maps obtained in the presence of the individual parts will be useful to create a visual explanation for the (holonym) predictions of the original model.

A held-out test set is used to calculate the per-part calibrated F1-score [31] to determine the degree to which each part was learned and distinguished by the others. The F1-score is calibrated to be invariant to the class prior, enabling the comparison of models trained on different numbers of meronyms.

3.4 Explanations

At the end of the previous step a trained meronyms CNN model $\mathcal{M}^c$ is obtained. For any input holonym image $\mathbf{x}$, this model outputs a set of prediction scores $Y_p = \{y_{p_1}, \ldots, y_{p_n}\}$, where n is the the number of parts the model was trained on, and $y_{p_1}, \ldots, y_{p_n}$ are the scores produced for each different part. Hence, by feeding the network with an holonym sample (such as a car image), a score about each of its parts (e.g., wheel, bumper, etc.) will be produced. Intuitively, the output scores reflect *how much* of each part the network sees in the input holonym image. Exploiting the fact that the network can 'see' the part concepts within an holonym image sample, we can look *where* the network exactly sees the parts, i.e., in which portion of the input image.

Specifically, the visualization of each part in the holonym image is obtained trough the state-of-the-art saliency method Grad-CAM[29]. After obtaining a saliency map $\mathbf{x}^{(p_i)}$ related to each part that the network can recognize, each saliency map $x^{(p_i)}$ is thresholded into a binary segmentation mask $m^{(p_i)} \equiv (x^{(p_i)} \geq T^{(p_i)})$, where $T^{(p_i)}$ is set to the q^{th} percentile of the corresponding saliency map pixel distribution. We later feed the same input holonym image into the original CNN model, and verify whether each part is fundamental for the original network prediction, by ablating one part from the image at a time based on the meronyms masks $m^{(p_i)}$. By observing the score drop for the original predicted holonym class label (calculated in percentage, with respect to the original holonym score), we can determine how much the removed meronym was

important in order to predict that class label: the more consistent the drop, the more significant the visual presence in the image of the part would be for the original model.

At this point, the input image $\mathbf{x}$ is associated to a set of saliency maps $\mathbf{x}^{(p_i)}$ for each part $p_i \in P_c$, and each saliency map is associated to a score drop $s_i \in S = \{s_1, s_2, ..., s_n\}$. Additionally, the per-part calibrated F1-score previously computed is used to measure the reliability of the part identification. We assume that a meronyms model which had difficulties to learn and distinguish a part, would have consequently achieved a low F1-score for that part. Hence, the parts whose holonym score drop s_i exceeds a threshold T_s and whose meronyms model are above a F1-score threshold T_{F1} are provided as part-based explanations for the original model prediction, as it would mean that those parts are both correctly detected by the meronym model and deemed relevant for the classification of the holonym.

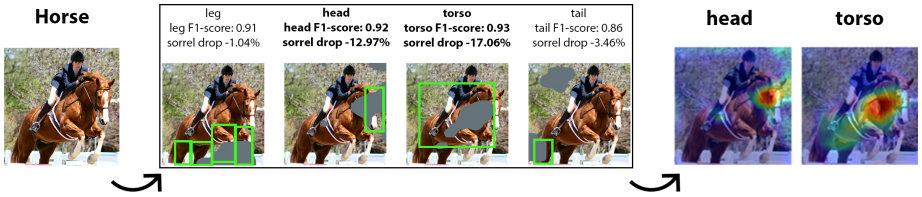
4 Experimental Settings

HOLMES can generate part-based explanations for any model that can be expressed as a feature extractor and a feed forward classifier. In our experiments, we explain the outputs of a VGG16 [33] image classifier pre-trained on the ImageNet [7] dataset. In this section, we describe the application of the HOLMES pipeline in two different experimental settings to explain the outputs of the VGG16 model. In the first experiment, we exploit bounding boxes for objects and their parts from the PASCAL-Part dataset to explain and validate the results. In the second one, we first build a part-based mapping for many ImageNet classes, then we use scraping to build a dataset for training the meronyms models, and finally we generate part-based explanations and evaluate the results with insertion, deletion and preservation curves. Examples of HOLMES explanation on both datasets are provided in Fig. 2.

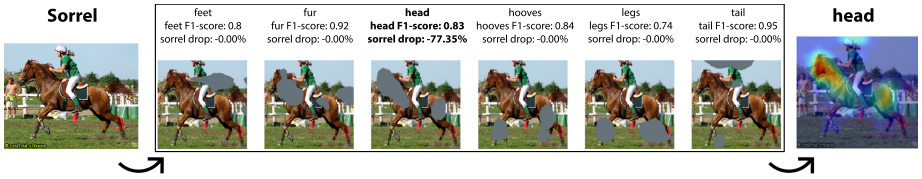
4.1 PASCAL-Part dataset

The PASCAL-Part [5] dataset contains additional annotations over the PASCAL VOC 2010 dataset, i.e., bounding boxes for objects and their parts, that can be used as the holonym-meronym relationship mapping (HolMe). We held out a set of 50 images for each of the selected holonyms and their corresponding cropped meronym images that are used exclusively to evaluate the HOLMES explainability and part localization performance. The other images were used in the meronym models $\mathcal{M}^c$ training.

Meronyms extraction settings: The twenty classes in the Pascal VOC 2010 dataset can be divided in four macro-classes: (1) person, (2) animals (bird, cat, cow, dog, horse, sheep), (3) vehicles (aeroplane, bicycle, boat, bus, car, motorbike, train) and (4) indoor objects (bottle, chair, dining table, potted plant, sofa, tvmonitor). Two of the classes (person and potted plant) have no corresponding class in the ImageNet 1000 classes and thus are discarded. Other five



(a) PASCAL-Part



(b) ImageNet

Fig. 2. HOLMES Explanation example for the *horse* class – PASCAL-Part (a) and *sorrel* class – ImageNet (b). For each part, the corresponding ablation mask (grey), the per-part calibrated F1-score and the holonym score drop are shown. For PASCAL-Part, the ablation masks are compared against the ground truth bounding boxes (green). The final heatmap(s) show the part-based explanations. Two and one part are included in the explanations for examples (a) and (b), respectively, as they exceed both the holonym score drop threshold T_s (0.1) and the calibrated F1-score threshold T_{F1} (0.7).

classes were discarded because they do not have part-based annotations (boat, chair, dining table, sofa, and tvmonitor). For each of the 13 remaining classes, the respective meronyms P_c were extracted from the PASCAL-Part parts list. For the six animal classes we performed hyper-meronym selection as in [39], and for the remaining classes we selected the hyper-meronyms by majority voting. The final HolMe mapping is thus:

- $P_{bird} = P_{cat} = P_{dog} = P_{horse} = \{\text{head, torso, leg, tail}\}$
- $P_{cow} = \{\text{head, torso, leg, horn}\}$
- $P_{sheep} = \{\text{head, torso, leg}\}$
- $P_{aeroplane} = \{\text{stern, wheel, artifact wing, body, engine}\}$
- $P_{bicycle} = \{\text{saddle, wheel, handlebar}\}$
- $P_{motorbike} = \{\text{saddle, wheel, handlebar, headlight}\}$
- $P_{car} = P_{bus} = \{\text{window, wheel, headlight, mirror, door, bodywork, license plate}\}$
- $P_{train} = \{\text{coach, locomotive, headlight}\}$
- $P_{bottle} = \{\text{body, cap}\}$

Meronyms image extraction settings: Once the holonym-meronym relationship mapping is defined, we extract the images to train the meronyms models. For each holonym image, we retrieved the bounding box coordinates associated with their parts P_c, and cropped the holonym image accordingly to produce a set of meronym images.

To obtain images compatible with the square VGG16 input, while preserving the aspect ratio and shape of each part, before cropping we extended the

bounding box in the x or y direction to obtain a square crop, with the constraint of not overlapping other bounding boxes in the same image. Then, if a 1:1 aspect ratio was not completely reached, we applied padding to get a final square crop.

Moreover, the number of crops extracted for each meronym might be very different, e.g., for a *horse* meronym, there are more *leg* parts with respect to just one *head*. To avoid high class unbalance, we used data augmentation to balance the number of samples in each meronyms class. Specifically, we applied both random rotation and random shear, and one among the gaussian blur, emboss, and gaussian noise transformations to each cropped image. Finally, for each holonym class, the extracted meronyms samples were split into training/validation/test folds with ratios of 0.81/0.09/0.1.

Training and Explanations settings: For each holonym, we built a separate meronym model $\mathcal{M}^c$ and we retrained a feed forward classifier $f_{P_c}^{\mathcal{M}}(\cdot)$ with the same structure of the original VGG16 classifier using Cross Entropy Loss. Common data augmentation techniques were employed: horizontal flipping, rotation, cropping, color jittering, and random gray scale. Each meronym model was trained for 100 epochs, with a batch size of 64 and learning rate of 0.001 (determined experimentally). Early stopping policy with patience set to 5 was used to avoid overfitting.

Regarding the explanations, the activations of the last convolutional layer of VGG16 were used to produce the Grad-CAM meronyms heatmaps, which are then binarized using a threshold $T^{(p_i)}$. We found $T^{(p_i)} = 83^{th}$ percentile by performing a grid search upon the $[75, 90]$ percentile values and by finding the best trade-off between different causal metrics performance (described in the Evaluation section) on the whole PASCAL-Part training set, comprising all training holonym image samples. The masked pixels were ablated by replacing with the gray RGB value (as the ImageNet mean pixel is gray) for retaining the natural image statistics [27]. Finally, the T_s and T_{F1} thresholds were set to 10 and 0.7, respectively.

Evaluation settings: The whole HOLMES pipeline was run and tested upon each validation image sample associated to each selected class. The meronyms localization performance was measured by computing the per-pixel AUC score of the HOLMES meronym heatmap versus the same meronym ground truth. To calculate this metric, each pixel was assigned the corresponding heatmap value as score: true positive pixels were those belonging to the actual part (i.e., falling within the bounding box), while the remaining pixels were labeled as false positives. The performance is compared to the per-pixel AUC score of the Grad-CAM holonym heatmap as baseline. Moreover, the faithfulness of HOLMES explanations was assessed by means of common causal metrics based on the deletion/insertion/preservation curves [23,16].

As mentioned before, HOLMES produces a set of part-based explanations, which are obtained by computing a set of saliency maps $\mathbf{X}^{(P_c)} = \{\mathbf{x}^{(p_i)},\ p_i \in P_c\}$, each associated a specific part p_i. However, to assess the global quality of such explanations, all part-based saliency maps need to be merged into a unique heatmap, comprising all parts. Given the set of saliency maps $\mathbf{X}^{(P_c)}$, and the

corresponding score drops $S = \{s_1, s_2, ..., s_n\}$ associated with the ablation of each part, the HOLMES global heatmap is obtained through a weighted linear combination of the part-based saliency maps. First, normalized score drops $Z = \{z_1, ..., z_n\}$ are calculated by dividing each score drop by the L1-Norm of S. Then, the global heatmap is obtained by summing each weighted heatmap element-wise: $G = \sum_{i \in n} \mathbf{x}^{(p_i)} z_i$. This weighting scheme emphasizes parts whose ablation causes a significant holonym class score drop.

After obtaining the global heatmap G for an input image in this way, it is possible to use causal metrics such as the areas under the insertion [23], the deletion [23] and the preservation curves [16] to assess the overall quality of the part-based explanations, whose information is combined into G. These metrics were computed for all held out PASCAL-Part validation images. Notably, distinct from simply replicating Grad-CAM results, our global heatmap stresses the pivotal role of part-based explanations, serving as an integral instrument to appraise the global effectiveness of the part-based explanations.

4.2 ImageNet

HOLMES can be applied in scenarios where part-level annotated datasets are not available. In this case, we leverage ontologies and image scraping to construct the required meronym datasets. In particular, we exploit the connection between Imagenet[7] labels and WordNet[19] nodes in order to retrieve a list of parts of the object-label, relying on the holonym-meronym (whole-part) relationship.

Meronyms extraction settings: Across the ImageNet 1000 class concepts, 81 of them were selected and treated as holonym classes. The selected holonym classes belong to two main categories:

1. Medium- or large-size animals
2. Medium- or large-size man-made objects

The size constraint is necessary to obtain acceptable training sets. In fact, the smaller the holonym (e.g., bugs in the animals category), the more troublesome it becomes to retrieve images of distinct parts by querying web search engines. Specifically, when querying for such parts, the engines tend to return images of the whole holonym concept instead (e.g., the whole butterfly when querying for a butterfly head). This would consequently result in meronyms datasets very similar among each other and with a strong visual overlap, thus greatly hindering the associated meronym model performance.

Therefore, for each of the 81 classes, the respective meronyms were extracted from the Visual Attributes for Concepts (VISA)[32] dataset. Hyper-meronyms were further extracted by manual filtering: the meronyms obtained from the ontology were manually categorized into hyper-meronyms and their respective hypo-meronyms. In this way, for each occurrence of a hyper-meronym, the associated hypo-meronyms were automatically filtered out. The final HolMe mapping is available in the Supplementary Material, Section I.A.

Meronyms image scraping settings: The Google and Bing web search engines were selected and queried for downloading the images. The number of

downloads per part over all the engines was forcibly limited, since the pertinence of the images with respect to the desired part concept naturally decreases as more images are downloaded (e.g., after too many downloads for *sorrel head*, an engine would for instance start returning images depicting plants and flowers); a good rule-of-thumb is to limit the download to the first 100 items [20,18]. Finally, in order to further increase the dataset size for each part, the *Visually similar images* function of Google was exploited: for each downloaded image, the most visually similar ones are searched in this way and then added to the parts' samples. The download limit, i.e., the number of images to be downloaded for each part by each engine, was set to 40 for Google and 60 for Bing, since Bing showed to be slightly more reliable. The visually similar images download limit was instead set to 5. Duplicates and near-duplicates [21] are detected by means of the pHash[26] hash-based deduplication method. For outlier removal, meronyms images are mapped to a feature vector using the activations of the penultimate FC layer of VGG16[33], which are then given as input to the PCA outlier detection algorithm[30], with the outlier contamination rate hyper-parameter set to 0.15. The scraped data was split into training/validation/test folds with proportions of 0.81, 0.09 and 0.1 respectively.

Training and Explanations settings: The training and explanation steps are carried out with the same settings as detailed for the PASCAL-Part dataset (Section 4.1).

Evaluation settings: The global heatmap is evaluated using the insertion, deletion, and preservation curves as detailed for the PASCAL-Part dataset (Section 4.1).

5 Results

The HOLMES pipeline was quantitatively and qualitatively evaluated in all its steps. Experimental validation aimed at determining i) to what extent HOLMES is able to correctly identify and locate meronyms?, ii) to what extent the classification score can be attributed to individual meronyms and iii) how good are the explanations generated by HOLMES?

RQ1: How well can HOLMES classify and locate meronyms?

As introduced in Section 4.1, the PASCAL-Part ontology contains 13 classes with an average of ≈ 4 visible parts per class. Following the procedure described in the experimental setting, on average ~ 750 sample per meronym were collected (~ 1400 after data augmentation), for a total of 74,772 training samples. For ImageNet, 81 classes were selected, with an average of ≈ 7 visible parts per image. Thus, web scraping was performed for a total of 559 meronyms, yielding on average ~ 450, of which 18% were detected as duplicates and 11% as outliers, and hence, eliminated. The final average number of images per part is ~ 320.

First, we assess HOLMES ability to *classify different meronyms* by reporting the distribution of the calibrated F1-scores of the $\mathcal{M}^c$ models, trained upon each training set X_c for each of the selected classes, is reported in Fig. 3 for both PASCAL-Part and ImageNet dataset. The average F1-score was good in both

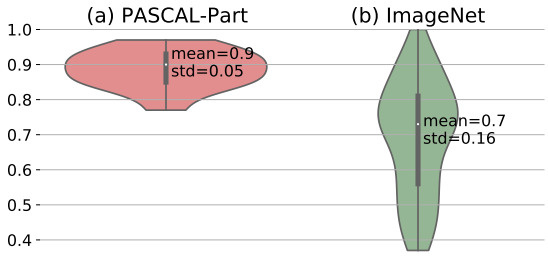


Fig. 3. Distribution (violin plot) of the average per-part calibrated F1-score.

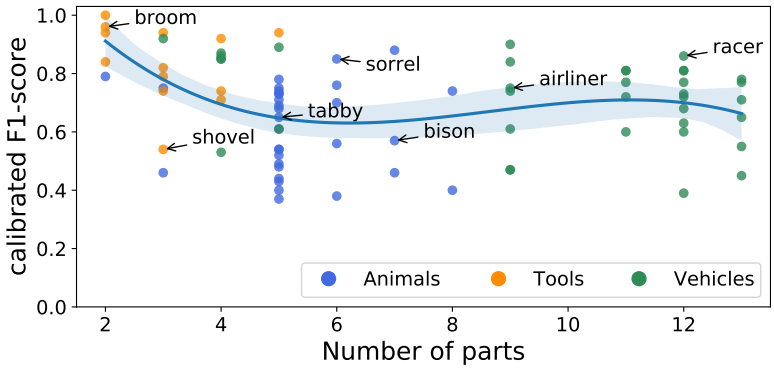


Fig. 4. Average per-part calibrated score as a function of the number of parts per holonym class (colored dots represent a holonym, blue line is the mean average per-part F1-score).

cases, but higher on PASCAL-Part (0.9 ± 0.05 vs. 0.7 ± 0.16). This difference can be attributed, at least partially, to the higher precision of the PASCAL-Part reference standard, for which bounding boxes are available.

On the other hand, it can be observed from Fig. 4 how the performance degrades with the number of parts per class. This is especially evident for the ImageNet dataset, since the ontology is richer with more classes and more parts per classes. As the number of parts increases, the likelihood of visual overlap between images belonging to different parts also increases, negatively impacting the performance of the trained $\mathcal{M}^c$ model. Additionally, different class categories tend to be associated with a lower/higher number of meronyms: for instance, tools tend to have between one and four parts, animals between three and eight, and vehicles more than eight. Thus, we cannot exclude that the category may also play a role either by influencing the quality of the scraping, or the differentiation of the meronyms themselves.

The HOLMES meronyms localization performance was measured by computing the per-pixel AUC score of each HOLMES meronym heatmap against their

Table 1. The HOLMES meronyms localization performance is measured by computing the average per-pixel AUC score of each HOLMES meronym heatmap (top row) against the holonym heatmap extracted the Grad-CAM (bottom row).

Method	horse	cat	bird	cow	dog	sheep	aeroplane	bicycle	bottle	bus	car	motorbike	train	Avg
HOLMES	0.77	0.74	0.8	0.77	0.74	0.75	0.74	0.76	0.67	0.75	0.68	0.74	0.71	**0.74**
Grad-CAM	0.68	0.68	0.71	0.66	0.71	0.62	0.76	0.63	0.6	0.65	0.62	0.66	0.62	0.66

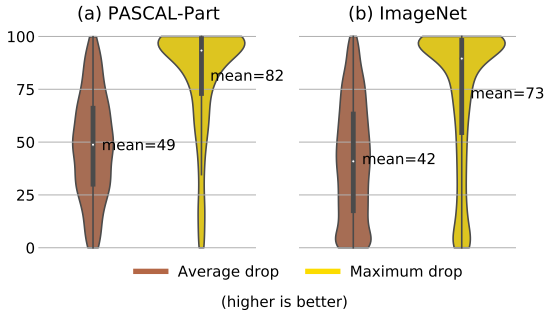


Fig. 5. Distribution (violin plots) of the average score drop and maximum score drop (in percentages) per image on the PASCAL-Part (a) and ImageNet (b) validation sets. The score drop is calculated for each image and meronym by ablating the corresponding mask; then, the average and maximum score drop are computed over all meronyms appearing in an image.

PASCAL-Part ground truth bounding boxes. As a baseline, we could assume that either the meronym could be randomly assigned to any region of the image (in this case, AUC=0.5), or we could focus on the actual holonym as extracted by the Grad-CAM algorithm and assume that the meronym is inside the region of the Grad-CAM holonym heatmap. This second choice offers a baseline that is harder to beat, but as reported in Table 1, the HOLMES meronyms explanations consistently localize the parts better and more precisely, compared to the whole Grad-CAM heatmaps which instead localize the entire object.

RQ2: To what extent the classification score can be attributed to individual meronyms?

Having established the ability to classify meronyms, the next step is to evaluate their impact on the holonym classifier $\mathcal{H}$, as exemplified in Fig. 2.

The distribution of the *per-meronym score drop*, i.e., the score drop observed for the holonym when the corresponding meronym is ablated, is reported in Fig. 5. The average score drop is 49% for PASCAL-Part and 42% for ImageNet, respectively, meaning that on average, the ablation of a single meronym roughly halves the holonym model confidence. When considering only the most significant part (i.e., the one associated with the highest score drop for each test image), the score drop increases to 82% on PASCAL-Part. Hence, in this dataset, individual meronyms have a substantial impact on the classifier output and, in most cases, the classification can almost be fully explained or attributed to a single meronym.

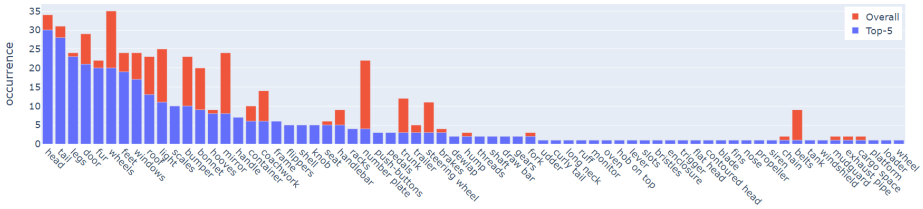


Fig. 6. Top-5 meronyms distribution. For each class, the top-5 meronyms are computed (i.e., the meronyms with the highest score drops on average). For each meronym, the total number of associated holonyms (blue) is compared with their frequency in the Top-5 meronym list (red). Meronyms are listed in descending order of frequency.

On the other side, on ImageNet the mean maximum drop is lower (73%). In fact, the mean ($\pm$ standard deviation) number of meronyms included in each explanation was 2.28 ± 2.46. Examples of explanations with one, two, three or more meronyms are provided in the Supplementary Material.

Conversely, when grouping images according to their holonym labels, the mean average drop and mean maximum drop were 50% and 80% for PASCAL-Part, and 46% and 73% for ImageNet, respectively. While the score distribution in Fig. 5 characterizes HOLMES behavior at the instance (image) level, the mean values provide insight into the general properties of the holonym classes. For instance, the mean maximum score drop on PASCAL-Part shows a bimodal distribution, with animal classes having a higher mean maximum drop than vehicles and man-made objects (78.5% vs. 62.1%). This entails that, on average, explanations for image belonging to animal classes are highly focused, with only one meronym almost explaining the entire holonym concept, whereas for other classes explanations require the combination of two or more meronyms. On ImageNet, we did not find a large difference between animals (70%), tools (73%), and vehicles (74%); however, we observed a wider variation between classes, with mean maximum drop ranging from 23% (zebra) to 95% (persian cat).

Different holonym classes may have an overlapping set of meronyms: this is especially evident for the richer ImageNet ontology. For instance, most of the animal classes have a head and a tail (although each class will be associated with its own training dataset and its own classifier $\mathbf{M}^c$). From this observation, we sought to understand if certain parts induced a consistent and substantial holonym score drop (Fig. 6). For instance, for classes belonging to the animal category, the meronyms head, tail, and legs frequently cause a consistent drop when ablated from the image. Similarly, for vehicles, the door, wheels and window meronyms have the highest impact on the holonym class prediction.

RQ3: How good are the explanations generated by HOLMES?

Finally, the overall quality of the generated explanations was evaluated. For the purpose of evaluating against Grad-CAM, which is not designed to provide part-based explanations, part-level heatmaps are linearly combined in a global heatmap, as detailed in Fig. 7. The AUC values under the insertion, deletion, and preservation curves are summarized in Table 2.

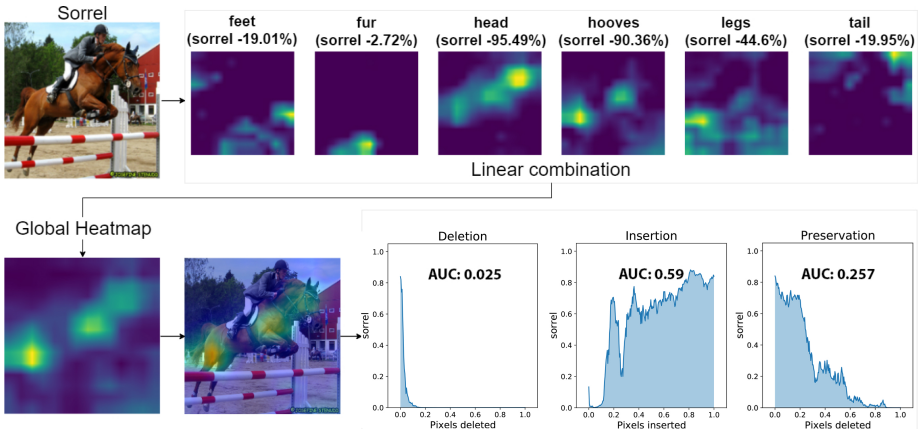


Fig. 7. HOLMES Global Explanation. Starting from an input image, the per-part heatmaps and the respective holonym score drops are obtained. Then, by a linear combination of the heatmaps, the global heatmap is obtained, and its quality is measured by means of the insertion/deletion/preservation metrics.

Table 2. Deletion/Insertion/Preservation AUCs for HOLMES and Grad-CAM.

Dataset	Method	Deletion ↓	Insertion ↑	Preservation ↑
PASCAL-Part	HOLMES	0.050±0.053	0.487±0.269	0.392±0.255
	GradCAM	0.052±0.060	0.505±0.277	0.381±0.264
ImageNet	HOLMES	0.112±0.113	0.660±0.252	0.538±0.257
	GradCAM	0.111±0.107	0.684±0.242	0.539±0.261

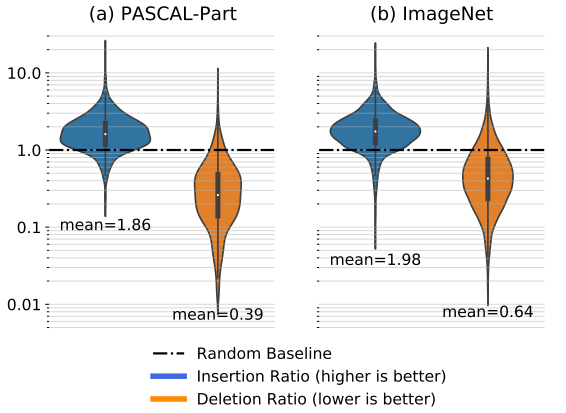


Fig. 8. Insertion/Deletion Ratio distribution (violin plot) for the PASCAL-Part (a) and ImageNet (b) datasets. The average insertion ratio (left) and the average deletion ratio (right) are calculated with respect to the random baseline (dotted black line).

On PASCAL-Part, the HOLMES insertion AUC is, on average, 0.96 times the GradCAM insertion AUC, while the average deletion AUC is 0.95 and the average preservation AUC is 1.03 times the respective GradCAM score. Analogously, on ImageNet, the average insertion, deletion, and preservation AUCs are 0.96, 1.01, and 0.99 times the corresponding GradCAM scores.

Additionally, we compared HOLMES against a random baseline obtained by dividing the images in super-pixels, which are then erased in random order. The random baseline is designed to account for the object scale: in fact, a good heatmap for a small object will yield a lower deletion AUC than an equally good heatmap for a larger object. As shown in Fig. 8, HOLMES metrics are substantially higher than the random baseline, with average insertion AUC 0.58 lower and average insertion AUC 1.77 higher than the baseline.

6 Discussion

Unlike previous methods [3], HOLMES does not require a densely annotated dataset with pixel-level annotations. Instead, it can be trained using weak annotations either in the form of bounding boxes, such as those available in the PASCAL-Part dataset [5], or relying on the potentiality of web scraping, which drastically reduces the annotation effort, whilst forgoing the limiting closed-world assumption intrinsic to traditional labelled datasets. The effectiveness of web scraping for object recognition has been established in previous works [18,35], which HOLMES capitalizes on and extends, using deduplication and outlier removal to reduce noise and increase variety in the training dataset. At the same time, retrieving high quality images for meronyms, as opposed to holonyms, introduces additional challenges which may impact dataset quality and, thus, the meronym models. Qualitatively, we observed that the pertinence of the retrieved images is generally good, but may decrease depending on the popularity of the meronym as a search term. Another specific challenge is the visual overlap, as it is difficult to find images that precisely isolate one and only one meronym.

Overall, quantitative evaluation on PASCAL-Part allowed us to conclude that the meronym models are capable of detecting object parts and locating their position within the image. This is achieved by exploiting, without any retraining or fine-tuning, the features learned by the holonym model, thus further supporting the conclusion that knowledge about object parts is implicitly embedded in deep neural networks [40,2,3].

Partly due to the imperfect background, the ablated mask does not always provide a perfect segmentation of the part itself, as shown in Fig. 2. In some cases, the ablated masks for different parts could be very similar, especially for meronyms that are physically next to each other. An example is provided by the meronyms *legs* and *hooves* for the holonym *sorrel*, as shown again in Fig. 2. Less frequently, it may occur that the ablated part may include a portion of the background; for instance, in the case of legs, some terrain or grass may be included. This may have an impact on the score drop observed when the corresponding part is deleted, and the resulting metrics.

The average F1-score for most holonyms ranges roughly between 0.6 and 1.0 for ImageNet, and between 0.8 and 1.0 for PASCAL-Part, spanning different categories of objects (animals, tools and vehicles), and up to 14 visible parts per class. The F1-score depends on the quality of the ground truth, but also on the number of meronyms that compose each object. Increasing the quality of scraping, and thus the meronym model, could allow HOLMES to recognize and include in the provided explanations an even larger pool of meronyms. In addition, HOLMES provides intrinsic safeguards against this type of noise, as only meronyms with sufficient F1-score are included in the explanations, and the user can readily inspect the heatmaps associated to each individual meronym.

Quantitative causal metrics based on the deletioninsertionpreservation curves confirm that the part-based explanations provided by HOLMES are effective in identifying those parts that are most relevant to the final classification, achieving results comparable to the state-of-the-art GradCAM method, and substantially above chance level. However, unlike GradCAM, HOLMES provides an articulated set of heatmaps, associated to human-interpretable concepts, and allows exploration of the impact of individual meronyms on the holonym classification, at both instance and class level.

HOLMES was evaluated on two distinct datasets, PASCAL-Part and ImageNet. Since both pipelines evaluate the same classifier, we attribute absolute differences in the insertion/deletion/preservation curves to the dataset themselves (e.g., how the images were sourced), and possibly to the domain shift between ImageNet and PASCAL-Part (given that the holonym classifier $\mathcal{H}$ was trained on ImageNet). However, the relative performance with respect to both baselines shows similar behavior, despite wide differences in how the meronym datasets X^c were sourced. HOLMES performs slightly better on PASCAL-Part, especially in terms of the deletion and preservation curves. Also, explanations on PASCAL-Part appear to be concentrated, on average, on fewer parts than ImageNet. Beyond the meronym datasets X^c, other factors could account for these differences: on the one hand, PASCAL-Part includes fewer and more distinct classes than ImageNet, thus potentially it includes images that are 'easier' to classify. On the other hand, the KB derived from PASCAL-Part annotations is simpler, with fewer meronyms (≈ 4 vs. ≈ 7 parts per class), and less visual overlap. Overall, HOLMES shows to be robust to the choice of experimental settings, and performs well even when exploiting more cost-effective annotations sourced through general purpose KBs and web scraping.

7 Conclusions and Future Work

In this paper we introduced HOLMES, an eXplainable Artificial Intelligence technique able to enrich image classification tasks with part-level explanations. Our approach allows to take a further step with respect to the standard label-level heatmaps which represent the state of the art in XAI for image classification. It proves valuable in integrating image classification models into decision support systems, as it provides more detailed explanations. These explanations

can help both the model developer, aiding in debugging the classifier before deployment, and also the end user, assisting in assessing the level of trust in the classifier's predictions for previously unseen data.

Furthermore, HOLMES sheds light on how holonyms are learned and stored within a CNN during and after the training phase. Other recent research works proposed relevant contributions, such as [3,4], but with additional requirements (focus on scene recognition, need for a segmented ground truth) and without the connection of a DL model with a symbolic knowledge base such as an ontology.

We adopt a strategy that avoids confining concepts to a single computational unit. We contend that this approach aligns better with the robust learning capabilities of DL models while also facilitating greater expressive power in the symbolic domain. As shown in [2], models with equivalent discriminative abilities can exhibit varying degrees of interpretability, influenced by factors such as the architecture, regularization, and learning task.

Given the novelty of our proposed pipeline, there is room for exploration concerning alternatives for many components. First, a more refined scraping method could be employed to both increase the training sample size and its quality, for instance using more complex semantic expansion techniques, by using more robust outlier detection algorithms such as Robust and Kernel PCA, or by incorporating novel data purification techniques to obtain cleaner data [38].

Second, it could be useful to study the effects of using the activations of units belonging to different convolutional layers, or even sets of such layers, for producing HOLMES explanations; units belonging to different convolutional layers could better match some specific (part) concepts, and, accordingly, by means of their activations, a better explanation could be hence generated for those concepts. Also, more model architectures can be tested to see how this method results change according to the model which is used. For instance, shallower (e.g., VGG13) or deeper (e.g., VGG19) model architectures, or even different types of networks (e.g., Deep Residual Networks) could be inspected.

Third, alternative perturbation techniques can be tried for removing the relevant pixels of the parts; it was observed that substituting pixels with just constant values introduces contiguous shapes in the image, thus biasing, if even minimally, the prediction towards certain types of objects having a similar shape. Moreover, other types of semantic relationship can be studied for both retrieving the desired (visible) parts of a specific concept (either alternative or complementary to the proposed holonym-meronym relationship), and for mapping different concepts between different knowledge bases (in alternative to the proposed hypernym-hyponym relationship).

Finally, it emerges from the final results that the method performs better when considering for an object a small number of parts, preferably spaced enough to minimize visual overlap. Hence, a new strategy for selecting and filtering the meronyms of an object can be also studied.

References

1. Arrieta, A.B., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., García, S., Gil-López, S., Molina, D., Benjamins, R., et al.: Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. Information Fusion **58**, 82–115 (2020)
2. Bau, D., Zhou, B., Khosla, A., Oliva, A., Torralba, A.: Network dissection: Quantifying interpretability of deep visual representations. In: Computer Vision and Pattern Recognition (2017)
3. Bau, D., Zhu, J.Y., Strobelt, H., Lapedriza, A., Zhou, B., Torralba, A.: Understanding the role of individual units in a deep neural network. Proceedings of the National Academy of Sciences **117**(48), 30071–30078 (2020)
4. Chen, C., Li, O., Barnett, A., Su, J., Rudin, C.: This looks like that: deep learning for interpretable image recognition. In: Advances in neural information processing systems (2019)
5. Chen, X., Mottaghi, R., Liu, X., Fidler, S., Urtasun, R., Yuille, A.L.: Detect what you can: Detecting and representing objects using holistic models and body parts. 2014 IEEE Conference on Computer Vision and Pattern Recognition pp. 1979–1986 (2014)
6. Confalonieri, R., del Prado, F.M., Agramunt, S., Malagarriga, D., Faggion, D., Weyde, T., Besold, T.R.: An ontology-based approach to explaining artificial neural networks. CoRR **abs/1906.08362** (2019)
7. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. pp. 248–255 (2009)
8. Geirhos, R., Jacobsen, J.H., Michaelis, C., Zemel, R., Brendel, W., Bethge, M., Wichmann, F.A.: Shortcut learning in deep neural networks. Nature Machine Intelligence **2**(11), 665–673 (2020)
9. Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F.A., Brendel, W.: Imagenet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In: International Conference on Learning Representations (2019)
10. Ghidini, V., Perotti, A., Schifanella, R.: Quantitative and ontology-based comparison of explanations for image classification. In: International Conference on Machine Learning, Optimization, and Data Science. pp. 58–70. Springer (2019)
11. Gonzalez-Garcia, A., Modolo, D., Ferrari, V.: Do semantic parts emerge in convolutional neural networks? International Journal of Computer Vision **126** (2018)
12. Goodman, B., Flaxman, S.: EU regulations on algorithmic decision-making and a "Right to explanation". AI Magazine **38** (June 2016)
13. Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., Pedreschi, D.: A survey of methods for explaining black box models. ACM computing surveys **51**(5), 1–42 (2018)
14. Jiang, H., Kim, B., Guan, M., Gupta, M.: To trust or not to trust a classifier. In: Advances in Neural Information Processing Systems. pp. 5541–5552 (2018)
15. Kim, B., Wattenberg, M., Gilmer, J., Cai, C., Wexler, J., Viegas, F., et al.: Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In: International conference on machine learning. pp. 2668–2677. PMLR (2018)
16. Lim, D., Lee, H., Kim, S.: Building reliable explanations of unreliable neural networks: Locally smoothing perspective of model interpretation. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition pp. 6464–6473 (2021)

17. Lundberg, S.M., Lee, S.I.: A unified approach to interpreting model predictions. In: Advances in neural information processing systems. pp. 4765–4774 (2017)
18. Massouh, N., Babiloni, F., Tommasi, T., Young, J., Hawes, N., Caputo, B.: Learning deep visual object models from noisy web data: How to make it work. In: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 5564–5571 (2017)
19. Miller, G.: Wordnet: a lexical database for english. Communications of the ACM **38**(11), 39–41 (1995)
20. Molinari, D., Pasquale, G., Natale, L., Caputo, B.: Automatic creation of large scale object databases from web resources: A case study in robot vision. In: International Conference on Image Analysis and Processing (2019)
21. Morra, L., Lamberti, F.: Benchmarking unsupervised near-duplicate image detection. Expert Systems with Applications **135**, 313–326 (2019)
22. Obermeyer, Z., Powers, B., Vogeli, C., Mullainathan, S.: Dissecting racial bias in an algorithm used to manage the health of populations. Science **366**(6464), 447–453 (2019)
23. Petsiuk, V., Das, A., Saenko, K.: Rise: Randomized input sampling for explanation of black-box models. In: British Machine Vision Conference (2018)
24. Ribeiro, M.T., Singh, S., Guestrin, C.: 'Why should I trust you?' Explaining the predictions of any classifier. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. pp. 1135–1144 (2016)
25. Rodríguez, N.D., Lamas, A., Sanchez, J., Franchi, G., Donadello, I., Tabik, S., Filliat, D., Cruz, P., Montes, R., Herrera, F.: Explainable neural-symbolic learning (*X-NeSyL*) methodology to fuse deep learning representations with expert knowledge graphs: The monumai cultural heritage use case. Information Fusion **79**, 58–83 (2022)
26. Samanta, P., Jain, S.: Analysis of perceptual hashing algorithms in image manipulation detection. Procedia Computer Science **185**, 203–212 (2021). https://doi.org/https://doi.org/10.1016/j.procs.2021.05.021
27. Samek, W., Binder, A., Montavon, G., Lapuschkin, S., Müller, K.R.: Evaluating the visualization of what a deep neural network has learned. IEEE Transactions on Neural Networks and Learning Systems **28**(11), 2660–2673 (2017)
28. Sarraf, A., Azhdari, M., Sarraf, S.: A comprehensive review of deep learning architectures for computer vision applications. American Scientific Research Journal for Engineering, Technology, and Sciences **77**, 1–29 (03 2021)
29. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-CAM: Visual explanations from deep networks via gradient-based localization. International Journal of Computer Vision **128**(2), 336–359 (2019)
30. Shyu, M.L., Chen, S.C., Sarinnapakorn, K., Chang, L.: A novel anomaly detection scheme based on principal component classifier. In: Proceedings of International Conference on Data Mining (01 2003)
31. Siblini, W., Fréry, J., He-Guelton, L., Oblé, F., Wang, Y.Q.: Master your metrics with calibration. Advances in Intelligent Data Analysis XVIII p. 457–469 (2020)
32. Silberer, C., Ferrari, V., Lapata, M.: Visually grounded meaning representations. IEEE Transactions on Pattern Analysis and Machine Intelligence **39**(11), 2284–2297 (2017)
33. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: International Conference on Learning Representations (2015)
34. Steging, C., Schomaker, L., Verheij, B.: The XAI paradox: Systems that perform well for the wrong reasons. In: Proceedings of the 31st Benelux Conference on A.I. and the 28th Belgian Dutch Conference on Machine Learning (2019)

35. Yao, Y., Shen, F., Xie, G., Liu, L., Zhu, F., Zhang, J., Shen, H.T.: Exploiting web images for multi-output classification: From category to subcategories. IEEE transactions on neural networks and learning systems **31**(7), 2348–2360 (2020)
36. Yosinski, J., Clune, J., Bengio, Y., Lipson, H.: How transferable are features in deep neural networks? In: Advances in Neural Information Processing Systems. vol. 27, pp. 3320–3328 (2014)
37. Yosinski, J., Clune, J., Bengio, Y., Lipson, H.: How transferable are features in deep neural networks? In: Advances in Neural Information Processing Systems. vol. 27, pp. 3320–3328 (2014)
38. Zhang, C., Wang, Q., Xie, G., Wu, Q., Shen, F., Tang, Z.: Robust learning from noisy web images via data purification for fine-grained recognition. IEEE Transactions on Multimedia (2021)
39. Zhang, Q., Wu, Y.N., Zhu, S.C.: Interpretable convolutional neural networks. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition pp. 8827–8836 (2018)
40. Zhou, B., Khosla, A., Lapedriza, À., Oliva, A., Torralba, A.: Object detectors emerge in deep scene cnns. International Conference on Learning Representations **abs/1412.6856** (2015)
41. Zhou, B., Sun, Y., Bau, D., Torralba, A.: Interpretable basis decomposition for visual explanation. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 119–134 (2018)