ECAI 2024
U. Endriss et al. (Eds.)

doi:10.3233/FAIA240880

Interpretable Image Classification Through an Argumentative Dialog Between Encoders

Dao THAUVIN[a,c,*], Stéphane HERBIN[a,c], Wassila OUERDANE[b,c] and Céline HUDELOT[b,c]

[a]ONERA - DTIS
[b]CentraleSupélec - MICS
[c]Université Paris Saclay

Abstract. We address the problem of designing interpretable algorithms for image classification. Modern computer vision algorithms implement classification in two phases: feature extraction - the *encoding* - that relies on deep neural networks (DNN), followed by a task-oriented decision - the *decoding* - often also using a DNN. We propose to formulate this last phase as an argumentative DialoguE Between two agents relying on visual ATtributEs and Similarity to prototypes (DEBATES). DEBATES represents the combination of information provided by two encoders in a transparent and interpretable way. It relies on a dual process that combines similarity to prototypes and visual attributes, each extracted from an encoder. DEBATES makes explicit the agreements and conflicts between the two encoders managed by the two agents, reveals the causes of unintended behaviors, and helps identify potential corrective actions to improve performance. The approach is demonstrated on two problems of fine-grained image classification.

1 Introduction

Over the past decade, deep learning techniques have become indispensable for implementing computer vision functions, giving rise to algorithms that are sometimes considered to surpass human skills [37, 14].

The work presented in this paper aims to introduce a higher level of transparency in decision algorithms that involve deep neural network components. Transparency, a desirable property of explainable Artificial Intelligence [4], refers to the ability to understand how algorithms arrive at their decisions.

Most modern computer vision algorithm architectures follow an encoder/decoder pattern. The encoder is frequently trained using a (very) large dataset, whereas the decoder is task-specific and sometimes just a linear decision function. Most of the design effort is concentrated on the encoder component, assuming it can extract useful information with minimal adaptation for various tasks. This is the spirit of the so-called foundation models [6].

Encoders based on foundation models are learned on large amounts of data to increase versatility and generalization. However, they can still exhibit unintended behaviors [25, 42] that can lead to critical hazards. These bad behaviors are challenging to predict and understand due to neural architectures' complexity, size, and opacity.

One key issue is, therefore, to enhance the transparency of algorithms that involve such models while maintaining their high performance. To address this question, we propose to focus on one of the simplest but fundamental tasks in computer vision, image classification, with two research directions: implementing a dual encoding process and constructing a decoder to handle this duality.

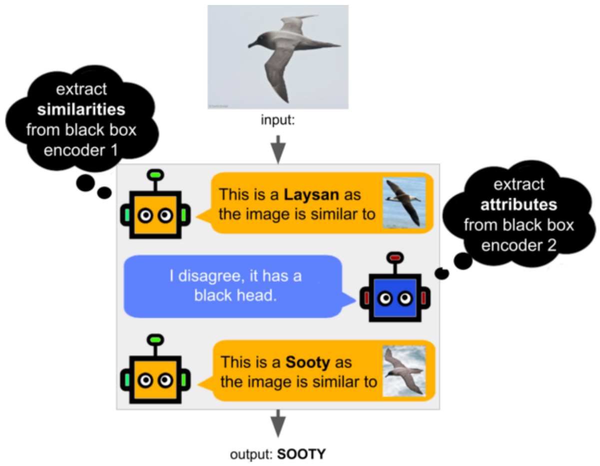

Figure 1: Simplified sketch of DEBATES for classifying an image of a bird. A first agent proposes a label based on a similarity to a prototype. The other agent counters using an attribute. This example shows an unintended behavior, as the first agent appears biased towards flying birds, thus confounding the `Laysan` and `Sooty` classes. A potential corrective action is to assess and correct the classes' conditional diversity concerning the bird's pose.

The first reason for using a dual encoding process is to enhance its reliability through redundancy and complementarity. The output of each encoder can be checked or reinforced by the other, leading to a more relevant encoding and a more trustworthy decision. The second reason is based on its analogy to human cognition. The concept of dual processes has a long history in psychological research [16] and was popularised by Daniel Kahneman's book [20]. In his book, he introduced to a broad audience the system 1 - intuitive / system 2 - deliberative duality for high-level cognitive functions, although the instantiation of this duality is still controversial [30, 15].

For image classification, the two encoders that will be used represent two levels of data analysis: a global similarity to image prototypes – which can be seen as implementing a fast system 1 process – and a set of local visual attribute detectors – system 2-like reasoning – both relying on pre-trained neural networks. These encoders are typically adapted to a fine-grained classification problem such as CUB-200 [41].

* Email: dao.thauvin@onera.fr

To represent how the decoder manages the dual encoding process, we use a formal argumentation-based dialog [5] between two agents. It provides a rigorous and standardised logic-based formalism to explicit potential conflicts and agreements when processing information by exchanging propositions and arguments. To the best of our knowledge, we present the first application of this formalism to a computer vision task with the purpose of explainability.

The role of our DEBATES method is to clarify the different steps of the image classification process and potentially help to identify the source of unintended behaviors or errors that can be corrected in a subsequent phase by modifying the algorithm features and the database. Figure 1, for example, illustrates a small dialog that reveals conflicts between attributes and similarities with prototypes, as well as possible biases and possible corrective actions.

The four main contributions of our work are to:

- propose a new transparent algorithm for image classification,
- able to handle a dual encoding strategy: similarity to prototypes and attribute detection,
- expressed as a formal argumentation-based dialog representing conflict resolution and agreements
- and demonstrate its ability to help identify unintended behaviors for further correction on CUB-200 [41] and Flowers 102 [31] datasets.

The paper is organised the following way: Section 2 discusses the related work, Section 3 describes the components needed to implement the dual encoding image classification, Section 4 shows on a detailed simulation how DEBATES can be analysed and how it explains the reasoning underlying the classification, Section 5 formalizes the structure of the dialog, Section 6 defines simple policies for choosing arguments and propositions and Section 7 demonstrates the capacity of DEBATES to suggest corrective actions on a problem of bird classification (CUB-200 [41]) and flower classification (Flowers 102 [31]).

2 Related work

Our work addresses two questions: the development of transparent algorithms and the representation of information processing as a dialog, both applied to image classification.

Transparent image classification Transparent algorithms are typically opposed to post-hoc explainability strategies such as feature attribution, counterfactual examples, attention heatmaps, or dissecting concepts [4].

In image classification, most transparent architectures adhere to a three-step bottleneck pattern [7, 21, 26]. This pattern involves encoding the image with a single DNN, mapping it to a representation space (referred to as the "bottleneck") that is expected to be interpretable, and finally decoding it to make the prediction, which is also expected to be interpretable. The primary distinctions between these approaches lie in the nature of the representation space (symbol, prototype image, text, clusters of samples), how it is calculated (supervised or unsupervised learning, utilization of a generative model, etc.), and how it is employed for prediction (linear function, decision tree, nearest neighbor or prototype, etc.). Readers are encouraged to refer to [32] for an extensive recent review.

Our work draws inspiration from two classical approaches for the representation space: a symbolic description of the image using a fixed vocabulary of visual attributes as in the Concept Bottleneck Model (CBM) [21] and a description by similarity to global prototypes as proposed in [24]. Note that in DEBATES, we do not use prototypes of local patches as in ProtoPNet [9] but transfer the idea of local geometric patterns to the detection of visual attributes.

In contrast to the aforementioned approaches, DEBATES employs two distinct levels of representation. The first motivation is to facilitate the detection of unintended behaviors by identifying inconsistencies and biases in the representations. The second motivation is to benefit from a global/local data representation for fine-grained or subordinate classification. This can be postulated as a potential mechanism underlying human cognition, given the evidence indicating that subordinate-level categorization occurs after basic-level categorization [35] and takes longer to learn [29, 34, 2].

Dialog for computer vision A few studies have introduced the idea of a dialog between agents either to inquire sequentially about the content of an image [12], to solve image retrieval tasks ("Guess What?" [13]) or object identification ("Guess Which?" [11]). In [1], image classification is implemented as a process where an agent sequentially queries the value of a visual attribute. In all these studies, the role of each agent is fixed (active questioner or reactive answerer), and neither collaborates nor argues.

Explainability through argumentation for computer vision tasks. Argumentation has been proposed as a promising direction to achieve explainability in AI [43]. Very few studies in this area have addressed computer vision applications: [3] proposes ProtoArgNet, which relies on a bottleneck architecture, for which the final multilayer perceptron generating the prediction can be interpreted as implementing argumentation [33]. [23] describes a post-hoc explanation resulting from a contradictory debate.

Our proposed methodology utilizes an argumentation-based dialog to select and evaluate the most reliable representation elements. The final decision is delegated to an external mechanism that will rely on the results of the exchanges between the two agents. These exchanges will convey arguments and counterarguments for the classification, thereby providing a more comprehensive framework for understanding the decision.

Transparent algorithms mainly fall into four leading families [4]: single decision trees, where the branching process is expected to represent a sequential reasoning process; additive processes such as linear decision models, where the weighted combination of features reveals their individual contribution to the decision; the nearest neighbor algorithm, which instantiates reasoning as similarity ranking and voting; and logical rules, where the decision process follows propositional or predicate calculus and is expected to be semantically interpretable. Our work synthesizes these strategies to achieve transparency in image classification. The dialog is a sequential exploration of hypotheses and arguments expressed in the framework of propositional logic. The decoder consists of a dialog between two agents based on attribute detection and similarity ranking. The final prediction is a voting scheme.

3 Dialog-based image classification

In this section, we describe the essential components of DEBATES. The classification is based on a dialog between two agents that share a common database of prototypes, i.e. a set of images that sample each class. However, the type of information available – attribute values of the prototypes, attribute values on the image to be classified, and similarity between the prototypes and the image – is specific to each agent. The role of the dialog is to reveal the relevant information that will be used for the final prediction.

3.1 Task formalisation

Given an image $x \in \mathbb{R}^d$, the objective is to correctly predict its label from a list of C classes $\{1, \ldots, C\}$.

To achieve this, the global predictor has access to images representing the different classes, which are referred to as prototypes. Each prototype is annotated with a class and the presence or absence of different binary visual attributes. In the CUB dataset, the classes are the bird species, and the attributes include the color of the different bird's parts (e.g. black head present or absent).

The binary visual attributes are drawn from a vocabulary $V = \{t^n\}_{n=1}^N$. The vocabulary may originate from two distinct sources. It may be derived from existing data, as exemplified by the CUB dataset, which includes 312 attributes. Alternatively, it may be generated using a large language model, as described in [28].

We denote the prototype dataset as $D = \{(p_i, \{a_i^n\}_{n=0}^N, y_i)\}_{i=1}^M$ where $p_i \in \mathbb{R}^d$ is the i-th image prototype, $y_i \in \{1, \ldots, C\}$ its label among C classes and $\{a_i^n\}_{n=0}^N$ its attribute values where a_i^n is 1 if the attribute t^n is present in p_i and 0 if not.

The role of the dialog is not to predict the label of the image but to expose the statements (propositions, arguments, counterarguments) that reveal the relevant information for classification. As explained later (see Section 3.3), the final prediction relies on analyzing in a posterior phase the statements exchanged between the two agents hosting the encoders.

The content of the information exchanged by the agents relies on two *encoders* used to extract relevant information from images: f_{proto} and f_{att}. The first encoder, f_{proto}, selects the prototypes that best represent the input data x using a similarity distance in the f_{proto} embedding space. We denote the similar prototypes as $\{p_{\sigma(k)}\}_{k=1}^K$ where K is the number of prototypes considered similar by the agent, and σ is the similarity order on prototypes from most to least similar according to a similarity distance. The second encoder, f_{att}, represents data by detecting the presence of meaningful attributes. It relies on a detector for each attribute t^n that can be applied to the input data. The attribute value detected on x is denoted $\hat{a}^n$ and equals 1 if the attribute is detected in the image, 0 if not. The technical details of the implementation of the two encoders are presented in Section 7.1.

3.2 *Agent features*

In this section, we discuss the characteristics of agents: goals, roles, and the information they manipulate according to their role in the dialog.

In our dialog, the agents collaborate by exchanging their respective information about the image: a first agent uses similarities predicted from f_{proto} and considers that an image of a given class should be similar to prototypes of the same class. In contrast, the other agent uses the attributes predicted by f_{att} and considers that an image of a class should have attributes identical to the prototypes of the class. Agents handle each proposed label separately in different dialog branches that give arguments for and against the label proposition.

We will refer to these two agents as Prototype and Attribute agents, respectively, simplified for brevity as P and A agents. Next, we introduce their roles and the information they have access to, summarised in Table.1.

Agent P. Its role is to propose possible labels for the image x with support from the most similar prototypes. If the image x is similar to a given prototype, the agent can assume that x has the same label as that prototype. To link prototypes to labels, it has access to them and their labels $\{y_i\}_{i=1}^M$. It has also access to their attribute values $\{a_i^n\}_{i=1,n=1}^{M,N}$. This information helps to counter attribute detection if this happens to be inconsistent with the presence or absence of attributes in similar prototypes.

Agent A. As similarities between images may not encode reliably fine-grained differences between images, the role of agent A is to refine label propositions. To do so, it can agree with propositions and disagree by stating a difference of attributes between x and prototypes of the proposed label. It has access to prototypes $\{p_i\}_{i=1}^M$, their attributes $\{a_i^n\}_{i=1,n=1}^{M,N}$ and detected attributes $\{\hat{a}^n\}_{n=1}^N$ in x.

Table 1: Information available to the agent P and A. The symbols ✓ and $\times$ indicate whether the information is available to the agent.

Available Information	P	A
$V = \{t^n\}_{n=1}^N$: visual attribute vocabulary	✓	✓
$\{p_i\}_{i=1}^M$: prototypes	✓	✓
$\{a_i^n\}_{i=1,n=1}^{M,N}$: attribute values of prototypes	✓	✓
$\{y_i\}_{i=1}^M$: labels of prototypes	✓	$\times$
$\{p_{\sigma(k)}\}_{k=1}^K$: prototypes ranked by similarity to x	✓	$\times$
$\{\hat{a}^n\}_{n=1}^N$: attribute values detected in x	$\times$	✓

3.3 *Classification mechanism*

The predicted label of the image is not selected by an agent but predicted according to a mechanism exterior to the dialog. It allows a transparent mechanism that uses only the human-interpretable information available in the dialog to make the decision.

To predict a class, we propose a simple argument counting strategy. For each proposed class, we consider a dialog branch with two sets of arguments. Arguments for the label proposition provided by agent P, and arguments against the label proposition provided by agent A. The predicted label is the label with the highest difference between the number of arguments for and against the label proposition. In the case of equality, we take the first proposed label.

Before examining the dialog formalization, the subsequent section will illustrate interactions between our two agents with a simulated dialog. The simulation will also demonstrate how attribute detection and similarities with prototypes are used to discuss the labeling of an input image. Furthermore, it will demonstrate how it is possible to identify unintended behaviors.

4 An illustrative example

To illustrate the transparency of DEBATES, we instantiate a simple dialog highlighting the benefits of relying on a dialog for detecting and understanding unexpected behaviors.

(a) Sooty Albatross (b) Laysan Albatross

Figure 2: Instances of the two classes considered in the example.

Simulation. *We consider the following setup example of image classification:*

- *An image x to be classified.*
- *Two classes* `Sooty Albatross` *and* `Laysan Albatros`*. For brevity, we denote them as* `Sooty` *and* `Laysan`*. Figure 2 shows instances of both classes.*
- *We limit our example to only two attributes* $V = \{t_1 : \texttt{black\_head}, t_2 : \texttt{orange\_beak}\}$.
- *We consider 6 prototypes, their attributes and labels*
$D = \{(p_1, \{\texttt{black\_head}\}, \texttt{Sooty}),$
$(p_2, \{\texttt{black\_head}\}, \texttt{Sooty}), (p_3, \{\texttt{black\_head}\}, \texttt{Sooty})$

, $(p_4, \{\}, \texttt{Laysan}), (p_5, \{\}, \texttt{Laysan}), (p_6, \{\}, \texttt{Laysan})\}$.
If an attribute is indicated for a prototype, it is present; otherwise, it is not. For example, in p_5 both attributes `black_head` *and* `orange_beak` *are absent.*

- *Similar prototypes to x are p_1, p_4, p_6 (ordered from most to least similar).*
- `orange_beak` *is detected in x and* `black_head` *is not.*

The dialog is described by the following script.

*(1) **P**: I propose the label* `Laysan` *for x.*
*(2) **A**: Why is x of class* `Laysan`*?*
*(3) **P**: Since p_4 is similar to x and p_4 is of class* `Laysan`*, so x is a* `Laysan`*.*
*(4) **A**: p_4 is of label* `Laysan` *and does not have the attribute* `orange_beak` *but x has the attribute* `orange_beak`*, so x is not a* `Laysan`*.*
*(5) **P**: I disagree that x doesn't have the attribute* `orange_beak`*, since x is similar to p_1 and p_1 has the attribute* `orange_beak`*.*
*(6) **A**: I concede that x is of label* `Laysan`*.*
*(7) **P**: Also p_6 is similar to x and p_6 is of class* `Laysan`*, so x is a* `Laysan`*.*
*(8) **A**: I concede that x is of label* `Laysan`*.*
*(9) **P**: I propose the label* `Sooty` *for x.*
*(10) **A**: Why is x of class* `Sooty`*?*
*(11) **P**: Since the prototype p_1 is similar to x and p_1 is of class* `Sooty`*, so x is a* `Sooty`*.*
*(12) **A**: The prototype p_1 is of label* `Sooty` *and has the attribute* `black_head` *but x doesn't have the attribute* `black_head`*, so x is not a* `Sooty`*.*
*(13) **P**: Ok, I'm not able to find another interesting prototype. x is not a* `Sooty`*.*

In this simulation, we consider a dialog example where both classes in (1) and (9) are discussed to show that agent P can hesitate between multiple labels. The set of exchanges shows how agents exchange information to converge to a class. In this example, the class `Laysan` has three arguments for it in (3), (5), and (7) and one against it in (4). The class `Sooty` has one argument for it in (11) and one against it in (12). The resulting predicted label is `Laysan`. The dialog reveals the reasons for the prediction, x is similar to p_4 and p_6.

The interactions also bring to the forefront elements that would have been difficult to detect in a classical classification model. Indeed, we can highlight some conflicts as disagreements between agents on a label proposition or on a detected attribute.

- In turn (1), agent P proposes the label `Laysan`. In turn (4), agent A disagrees with agent P on this label proposition as it detects attribute `orange_beak` in image x that is absent in the prototype p_4 of label `Laysan`.
- In turn (5), agent P disagrees on the attribute `orange_beak` detected by agent A in turn (4) as similar images do not have the mentioned attribute.
- In turn (12), agent A disagrees on the label `Sooty` because `black_head` differs in x and the prototype p_1 of label `Sooty`. It detects that it is absent in x while it is present in the prototype p_1.

These conflicts reveal two sources of uncertainty that can lead to unintended behavior. Firstly, agent P can't differentiate between the two classes. Secondly, agent A can't correctly predict the `orange_beak` attribute, possibly due to the beak of a `Laysan` being yellow, which is very similar to the orange color.

The supplementary material [39] presents several additional concrete examples obtained with our implementation.

Table 2: Possible replies in the dialog game.

Speech acts	Attacks	Surrenders
PROPOSE (x_is_y)	WHY-PROPOSE (x_is_y) ARGUE(Ψ, $\neg(x\_is\_y)$)	
WHY-PROPOSE (x_is_y)	ARGUE(Ψ, x_is_y)	DROP-PROPOSE(x_is_y)
ARGUE(Ψ,ϕ)	ARGUE(Ψ',ϕ') where $\phi' = \neg(\phi)$ or $\neg(\phi') \in \Psi$	CONCEDE(ϕ)
DROP-PROPOSE(x_is_y)		
CONCEDE(ϕ)		

5 Dialog formalisation

Automated dialog generation and analysis can be supported by a rigorous formalism: its design is one of the main contributions of the present work. The formalism defines how agents can communicate (Section 5.1), what they are allowed to say (Section 5.2), and how the elements of speech relate to the visual information extracted from encoders (Section 5.3 and Section 5.4).

5.1 Dialog moves

The communication between agents is instantiated as *moves*. Moves are defined by a locution and its parameters. Several types of locutions are available in the literature; we select the most relevant ones for our problem in this section. Assuming a class $y \in \{1, \dots, C\}$ and x the image to be classified, the available locutions for the agents are the following:

- PROPOSE (ϕ), which puts forward a proposition ϕ. In our case, it is always a proposition of a label y for the image x.
- WHY-PROPOSE (ϕ), which questions about the argument(s) that support ϕ.
- ARGUE(Ψ, ϕ), which explains the conclusion ϕ with premises Ψ. ϕ should be a logical consequence of Ψ [38]. Agents use it to justify a label proposition or to disagree with a label proposition or an attribute detection. However, these locations do not intend to explain why a prototype is similar to the image or why an attribute is present or absent.
- DROP-PROPOSE(ϕ): which abandons a proposition. Depending on the agent's policy (see Section 5.4), an agent may propose a label without justification, requiring the agent to drop its proposition.
- CONCEDE(ϕ): which concedes a proposition ϕ. In our dialog, an agent may concede a label proposition or its negation, i.e. the label of the image is not the proposed one. An agent concedes when it has no other arguments for or against the proposition. So conceding a label is different from predicting a label, it's a decision by default.

5.2 Dialog protocol

Communication between agents requires a protocol to avoid incoherent behaviors. Following previous works on deliberative dialog [22, 18], the dialog protocol corresponds to a reply structure that specifies the authorised moves according to the previous move. Table 2 summarises these moves. The reply structure is the same for both agents. However, some locutions are available to only one agent: PROPOSE and DROP-PROPOSE are only available to agent P and WHY-PROPOSE to agent A only.

5.3 Links between dialog and vision

In order to construct a coherent dialog for image classification, we need to link the information from the two encoders (i.e. visual attributes, similarities) and the logical reasoning of the two agents. To do so, we define the following statements and rules for our dialog. We note $\wedge$, $\rightarrow$, and $\neg$ as the logical conjunction, implication, and negation respectively.

Definition 1. *The dialog is based on different statements:*

- *An assignment statement of the form p_is_y where $p \in \{x\} \cup \{p_i\}_{i=1}^M$ and $y \in \{1, \dots, C\}$, meaning that the image or prototype p is of class y.*
- *A similarity statement of the form $x\_is\_sim\_to\_p$ where $p \in \{p_{\sigma(k)}\}_{k=1}^K$, meaning that the image x is similar to prototype p. In our case, we consider that the K nearest prototypes according to σ are similar to x.*
- *An attribute statement of the form $x_i\_has\_t$ or $\neg(x_i\_has\_t)$ where $x_i \in \{p_i\}_{i=1}^M \cup \{x\}$ and $t \in V$, meaning that the image x_i has or has not the attribute t respectively.*

The dialog also provides several rules to make the agent's reasoning explicit. On the one hand, the agent P considers that images of the same class are similar according to its encoder. Thus, it justifies a label proposition with a similar prototype.

Definition 2. *An assignment by similarity rule is of the form*

$$p\_is\_y \wedge p\_is\_sim\_to\_x \rightarrow x\_is\_y$$

where $y \in \{1, \dots, C\}$ and $p \in \{p_{\sigma(k)}\}_{k=1}^K$, meaning that if p is a prototype of label y and p is similar to x then x is of class y.

Example 1. *(Simulation cont.) Agent P uses an assignment by similarity rule at the turn (11): $p_1\_is\_\texttt{Sooty} \wedge x\_is\_sim\_to\_p_1 \rightarrow x\_is\_\texttt{Sooty}$. It also uses the rule in (3).*

On the other hand, agent A considers that images of the same class have similar attributes; it can thus disagree on a label proposition if a class prototype has different attributes than x.

Definition 3. *An assignment rejection by attribute rule is of one of the following forms*

$$p\_is\_y \wedge \neg(p\_has\_t) \wedge x\_has\_t \rightarrow \neg(x\_is\_y)$$

$$p\_is\_y \wedge p\_has\_t \wedge \neg(x\_has\_t) \rightarrow \neg(x\_is\_y)$$

where $y \in \{1, \dots, C\}$ and $p \in \{p_i\}_{i=1}^M$, meaning that if p is of label y and p does not have the attribute t and x has t, then x is not of class y. Alternatively, if p is of label y and has the attribute t and x does not have the attribute t, then x is not of class y.

Example 2. *(Simulation cont.) In (4), agent A uses an assignment rejection by attribute rule: $p_4\_is\_\texttt{Laysan} \wedge \neg(p_4\_has\_\texttt{orange\_beak}) \wedge x\_has\_\texttt{orange\_beak} \rightarrow \neg(x\_is\_\texttt{Laysan})$. It also uses the rule in (12).*

Finally, agent P may also disagree on a detected attribute in x if it is incoherent with its similar prototypes, i.e. the detected attribute in x differs from its similar prototype attributes.

Definition 4. *An attribute detection reject rule is of one of the following forms*

$$p\_is\_sim\_to\_x \wedge p\_has\_t \rightarrow x\_has\_t$$

$$p\_is\_sim\_to\_x \wedge \neg(p\_has\_t) \rightarrow \neg(x\_has\_t)$$

where $p \in \{p_{\sigma(k)}\}_{k=1}^K$ and $t \in V$, meaning that if p is similar to x and has the attribute t then x has t. Conversely, if p is similar to x and does not have the attribute t, then x does not have t.

Example 3. *(Simulation cont.) In (4), agent P uses an attribute detection reject rule: $p_1\_is\_sim\_to\_x \wedge \neg(p_1\_has\_\texttt{orange\_beak}) \rightarrow \neg(x\_has\_\texttt{orange\_beak})$.*

5.4 Agents beliefs

To construct their arguments with visual information, agents use their beliefs about the state of the world (in our case, the classification task), which correspond to information that is specific to each agent and not shared between them [40]. This section introduces the beliefs of our agents and how they evolve during the dialog according to the information communicated by the other agent.

We denote the beliefs of agents P and A by Σ_P and Σ_A, respectively. Using the term belief rather than knowledge recognizes that what an agent believes at a given step of the dialog may not necessarily be true (and may change in the future). Thus, we separate the notions of knowledge and input image beliefs by denoting $\Sigma_P = K_P \cup B_P$ where K_P corresponds to the knowledge of agent P and B_P corresponds to its input image beliefs.

On the one hand, according to Table.1, agent P has access at the beginning of the dialog to similar prototypes, prototype labels, and attributes. Thus, $K_P = \{\{p_i\_is\_y_i\}_{i=1}^M \cup \{p_i\_has\_t^n | a_i^n = 1\}_{i=1,n=1}^{M,N} \cup \{\neg(p_i\_has\_t^n) | a_i^n = 0\}_{i=1,n=1}^{M,N}\}$ and $B_P = \{x\_is\_sim\_to\_p_{\sigma(k)}\}_{k=1}^K$.

Example 4. *In the simulation, agent P has the following beliefs and knowledge at the beginning of the dialog.*

- $K_P = \{\{p_i\_has\_\texttt{black\_head}\}_{i \in \{1,2,3\}} \cup \{\neg(p_i\_has\_\texttt{black\_head})\}_{i \in \{4,5,6\}} \cup \{\neg(p_i\_has\_\texttt{orange\_beak})\}_{i \in \{1,2,3,4,5,6\}}\} \cup \{p_i\_is\_\texttt{Sooty}\}_{i \in \{1,2,3\}} \cup \{p_i\_is\_\texttt{Laysan}\}_{i \in \{4,5,6\}}\}$,
- $B_P = \{x\_is\_sim\_to\_p_i\}_{i \in \{1,4,6\}}$

Agent P adds to its beliefs the detected attributes communicated by agent A unless it disagrees. That is, if the last dialog move is an ARGUE(Ψ,ϕ) of agent A such that $x\_has\_t \in \Psi$ then $B_P = B_P \cup \{x\_has\_t\}$ unless agent P's next move is ARGUE(Ψ',$\neg(x\_has\_t)$). Vice versa for $\neg(x\_has\_t)$.

Example 5. *In the simulation, agent A communicates $\neg(x\_has\_\texttt{black\_head})$ in (12), agent P adds $\neg(x\_has\_\texttt{black\_head})$ to its beliefs as it doesn't disagree on the detected attribute in (13). Thus, $B_P = \{\{x\_is\_sim\_to\_p_i\}_{i \in \{1,4,6\}} \cup \neg(x\_has\_\texttt{black\_head})\}$.*

After (3) where agent A communicates $x\_has\_\texttt{orange\_beak}$, the detected attribute is not added to agent P beliefs as it disagrees to $x\_has\_\texttt{orange\_beak}$ in (4).

On the other hand, according to Table.1, agent A has access at the beginning of the dialog to prototype attributes and detected attributes in x. Thus, $K_A = \{\{p_i\_has\_t^n | a_i^n = 1\}_{i=1,n=1}^{M,N} \cup \{\neg(p_i\_has\_t^n) | a_i^n = 0\}_{i=1,n=1}^{M,N}\}$ and $B_A = \{\{x\_has\_t^n | \hat{a}^n = 1\}_{n=1}^N \cup \{\neg(x\_has\_t^n) | \hat{a}^n = 0\}_{n=1}^N\}$.

Example 6. *In the simulation, agent A has the following beliefs and knowledge at the beginning of the dialog.*

- $K_A = \{\{p_i\_has\_\texttt{black\_head}\}_{i \in \{1,2,3\}} \cup \{\neg(p_i\_has\_\texttt{black\_head})\}_{i \in \{4,5,6\}} \cup \{\neg(p_i\_has\_\texttt{orange\_beak})\}_{i \in \{1,2,3,4,5,6\}}\}$,
- $B_A = \{\neg(x\_has\_\texttt{black\_head}), x\_has\_\texttt{orange\_beak}\}$

The agent A adds to its knowledge the prototype labels communicated by agent P. Thus, if the last dialog move is an ARGUE(Ψ,ϕ) of agent P such that $p\_is\_y \in \Psi$ then $K_A = K_A \cup \{p\_is\_y\}$.

Example 7. *In the simulation, after (3) where agent P communicates $p_4\_is\_$`Laysan`, agent A adds to its knowledge $p_4\_is\_$`Laysan`. Thus $K_A = K_A \cup p_4\_is\_$`Laysan`. It uses this knowledge in its next move (4) to disagree with the label proposition.*

6 Agent policies

To operate the dialog, we propose to endow our agents with transparent move policies so they can select which move to make at the current dialog step.

6.1 Agent P policy

The role of the agent P is to PROPOSE labels, justify them in response to a WHY-PROPOSE with a similarity measure (with prototypes), and ARGUE to reject detected attributes if they do not correspond to similar prototypes. All these actions require a policy to choose the move parameters and when to carry them out. We define these in this section.

PROPOSE a label. Agent P proposes a label at the outset of the dialog. Subsequently, the agent may propose new labels when the argumentation about a label ends. The agent always proposes the most prevalent label in the set of labels of prototypes similar to x, which have not yet been proposed. This process will continue until all labels present in similar prototypes have been proposed.

Example 8. *In the simulation, p_1, p_4 and p_6 are considered similar to x for agent P. Since p_4 and p_6 are of class* `Laysan` *and p_1 is of class* `Sooty`, `Laysan` *is proposed first, then* `Sooty`.

Respond to WHY-PROPOSE. Agent P ARGUE about the label proposition with an assignment by similarity rule. It uses multiple similar prototypes to justify the label. This approach is preferable to relying on a single similarity, which may be erroneous, to justify the label.

More precisely, the agent uses prototypes similar to x of the corresponding label in descending order of similarity. Then, it considers the attributes detected by agent A to select the prototype. The prototype attributes must match the attributes of x that were added to B_P during the dialog. In other words, if $x\_has\_t \in B_P$, then t should be present in the prototype. Conversely, if $\neg(x\_has\_t) \in B_P$, then t should not be present in the prototype. Without a similar prototype of the corresponding label following these conditions, it DROP-PROPOSE.

Example 9. *In the simulation, to justify the label* `Laysan`, *agent P first uses p_4 because it is the most similar prototype of this label, then p_6. Also, after (12), prototypes used to justify labels should not have the attribute* `black_head` *as agent P believes that x does not have the attribute* `black_head`. *It does not impact our dialog here as p_2 and p_3 are not similar to x.*

Respond to ARGUE with an attribute detection reject rule. Agent P can ARGUE to deny a detected attribute. It does so if all prototypes similar to x have no such presence or absence of the attribute. To achieve this, it uses the closest prototype (excluding the prototype used in the answered ARGUE). Otherwise, it CONCEDE that x is not of the proposed label.

Example 10. *In (5), since none of similar prototypes p_1, p_4 and p_6 have the attribute* `orange_beak`, *agent P argues that x doesn't have the* `orange_beak` *attribute.*

6.2 Agent A policy

Agent A can ARGUE to disagree with a label proposition by pointing out differences in attributes between a prototype of the label and the image to be classified. The question is when and with which prototype and attribute to disagree. This is related to how we detect the presence or absence of attributes, which is discussed below.

Attribute detection. Vision-language models, such as CLIP [36], measure distances between texts and images. The distance between a text t and an image x is noted $d_{att}(x, t)$. We use these distances between images and text attributes to calibrate attribute detection. For each attribute $t \in V$, we define two thresholds $\gamma_0(t)$ and $\gamma_1(t)$. An attribute t is detected present in x if $d_{att}(x, t) > \gamma_1(t)$ and absent if $d_{att}(x, t) < \gamma_0(t)$. Otherwise, the attribute is omitted. Each threshold is defined to have a certain percentage of false positives and false negatives on the prototypes specified in Section 7.1.

When to ARGUE. Agent A may disagree with a label proposition. To do so, it must select a prototype p and an attribute t, as described in the next paragraphs. Following this selection, multiple attributes may be possible. The agent then uses all these attributes to disagree, one after the other, to avoid relying on only one attribute detection, which might be erroneous. If no prototype or attribute is selected, it CONCEDE.

Which prototype to ARGUE. All prototypes of the proposed label are eligible. It can be noted that, for agent A to know whether a prototype is of a particular label, this information must be exchanged by agent P.

Which attribute to ARGUE. The agent selects an attribute not already used within the dialog. Furthermore, the attribute should be detected as present in x and absent in p, or vice versa.

The agent's goal is to accurately predict the label, which depends on its detection capacity. To ensure accurate attribute detection, the detected attribute on p should also correspond to its ground truth. As p is similar to x, this increases the probability of correctly detecting the attribute in x.

If multiple attributes are possible, the agent first selects the attribute t minimizing $min(|d_{att}(t, x) - max_p d_{att}(t, p)|, |d_{att}(t, x) - min_p d_{att}(t, p)|)$.

Example 11. *In (4), since the attribute* `orange_beak` *is detected in x and is absent in p_4, agent A disagrees with the label proposition in (5) using the attribute. To do this, we assume that agent A detects the attribute absent in p_4. Since this is the only difference between the detected attributes of x and the attributes of p_4, no other attributes are used to disagree with the label proposition.*

7 Demonstration

We apply our DEBATES method to the CUB-200 bird dataset [41] and to the Flowers 102 [31] dataset. CUB-200 contains 200 classes of birds and 5,994 train and test images. Each image is annotated with a class and 312 binary attributes. Binary attributes describe the color and shape of different birds parts. The Flowers 102 dataset contains 102 classes of flowers, 1,030 train images, and 6,129 test images. Each image is annotated with a class. The dataset doesn't contain any attribute. To apply DEBATES, we propose to use LLM-generated attributes from [17]. For each class, it provides attributes that characterise the class, for a total of 1118 attributes.

We chose these datasets as they are challenging, with fine-grained differences between labels that make the task difficult for both agents, and results in rich dialogs.

7.1 Implementation

We test DEBATES with two different encoders f_{proto}: DINO (ViT-S/8) [8] and DINOv2 (ViT-B/14) [27]. For the attribute encoder, we use a CLIP [36] encoder pre-trained from [10] (Swin-L & CLIP text) on CUB-200 and a vanilla CLIP (ViT-B/32) on Flowers 102. We use the train images as prototypes for a total of 5,994 prototypes on CUB and 1,030 prototypes on Flowers 102.

In our experiments, agent P ranks the prototypes by their cosine similarity with $f_{proto}(x)$. Agent P considers the 5 closest images to x as similar ($K = 5$). On Flowers, we set the thresholds γ_1 and γ_0 to detect attributes present and absent with 0% of false positives and negatives respectively. On CUB, we set the thresholds γ_1 and γ_0 to 9% of false positives/negatives. For each class, we also remove attributes with a frequency smaller than 5% to reduce noise in attribute annotations.

7.2 Impact on performance

In computer vision, numerous transparent methods observe a performance loss when incorporated into a model. This study demonstrates that DEBATES improves transparency without compromising performance. We compare DEBATES to applying a K-NN with prototypes encoded using f_{proto} (DINO or DINOv2). The value of K is set to 5, similar to our method. Table 3 reports the performances. On CUB, DEBATES improves DINO accuracy by 1.57% and DINOv2 accuracy by 0.04%. On the Flowers dataset, we observe an improvement of 2.75% with DINO and the same accuracy with DINOv2. The results of our experiments indicate that the greater the difference in efficiency between encoders, the more challenging it is to improve their accuracy. When one encoder is more efficient, the impact of the other encoder is reduced. Nevertheless, we observe interesting dialogs described in the supplementary [39].

We also compare DEBATES to existing transparent image classification methods: ProtoPNet, Concept Bottleneck Models, and CLIP with our attributes generation method [17]. DEBATES with DINOv2 demonstrates superior performance on CUB.

Table 3: Accuracy of our DEBATES method compared to a KNN and classical image classification transparent methods.

Method	Accuracy CUB	Accuracy Flowers 102
K-NN (DINO)	68.72%	80.92%
DEBATES (CLIP+DINO)	**70.29**%	**83.67**%
K-NN (DINOv2)	86.65%	**99.67**%
DEBATES (CLIP+DINOv2)	**86.69**%	**99.67**%
CLIP with LLM attributes [17]	56.13%	72.19%
Concept Bottleneck Models [21]	80.1%	/
ProtoPNet [9]	80.2%	/

7.3 Debugging models

DEBATES aims to reveal unintended behaviors and why they occur to correct them. In this section, we show how a model can be fixed in 3 steps: identifying unintended behaviors, interpreting the source of these behaviors, and acting to correct the model. DEBATES enables an easier identification of unintended behaviors and a wider range of corrections than existing methods. We analyse several dialogs (presented in the supplementary [39]) and describe the discovered unintended behaviors and ways to fix them here.

Identify unintended behaviors. Unintended behaviors refer to agent features that mislead classification. Typically, if agents disagree on the label to be predicted, at least one agent is misleading the classification. It is, therefore, possible to identify unintended behavior by looking for conflicts between agents.

Understand the source of these behaviors Having identified unintended behaviors, it is possible to identify their origin by interpreting the dialog. We observe different types of unintended behaviors:

- Similarities between images reveal biases in the encoder f_{proto}. We found several biases of DINO and DINOv2. The DINO encoder considers all images of a bird held in one hand similar. It also finds all images of a flower with a bee similar. DINOv2 and DINO also consider all images of a bird with a feeder similar.
- The encoder f_{proto} can confuse different classes.
- The encoder f_{att} can also be subject to errors and hallucinate attributes. The dialog helps to identify attributes that are difficult for the f_{att} encoder to detect.
- Some conflicts also reveal annotation problems. For example, in CUB, some images of the class Yellow Bellied Flycatcher don't have the attribute yellow belly color.

Propose potential corrective actions. Understanding these sources of unintended behavior makes it possible to propose corrective actions depending on the type of unintended behavior. We propose possible corrective actions for the different types of unintended behavior discussed before. These actions not only correct the inference but also apply to future inferences.

- It is possible to mitigate f_{proto} biases by removing prototypes with the corresponding bias. For example, we remove prototypes until birds held in one hand no longer appear in a dialog anymore. This process results in the removal of 73 prototypes, which eliminates 13 errors on our test set.
- f_{proto} confusions are usually cleared by the other agent. However, this may not always be the case. One potential solution is to introduce additional prototypes of the classes, which would assist in distinguishing between classes more effectively.
- In general, attribute hallucinations are corrected by the f_{proto} encoder. However, this does not exclude the possibility of error. One potential solution is to turn off the attribute detector for the attribute when it is not required. Alternatively, the calibration could be improved to prevent hallucinations.
- The annotation errors can be resolved by rectifying the annotations in question. However, f_{atts} may learn these false annotations and create new errors. In such a case, it may be necessary to train the encoder again with fixed annotations or refrain from utilizing these attributes.

8 Conclusion and future work

We have proposed and formalised a dialog for transparent image classification. We demonstrate its efficiency on the CUB-200 and Flowers datasets. Our study indicates that our DEBATES method can assist developers in identifying, understanding, and correcting unintended behaviors without compromising performance. We consider two possible directions for future work.

Exploiting the flexibility and expressiveness of the dialog [19] for other datasets and other vision tasks, such as object detection.

Another avenue for exploration is the implementation of a transparent automatic correction, which would allow a developer to analyse and verify corrections. The method should assist in identifying unintended behaviors, interpret them, and correct the decision. DEBATES already proposes a way to identify unintended behaviors and we show that humans can interpret and correct them. The remaining step is to allow the model to correct the decision by analyzing the encoders' behavior more deeply, not just the impact on their output.

Acknowledgements

The work is supported by a PhD scholarship to the first author from ONERA and the French Agence Innovation Défense.

References

[1] S. Alaniz, D. Marcos, B. Schiele, and Z. Akata. Learning Decision Trees Recurrently Through Communication. pages 13518–13527, 2021.

[2] M. N. Ashtiani, S. R. Kheradpisheh, T. Masquelier, and M. Ganjtabesh. Object Categorization in Finer Levels Relies More on Higher Spatial Frequencies and Takes Longer. *Frontiers in Psychology*, 8, July 2017. ISSN 1664-1078. doi: 10.3389/fpsyg.2017.01261.

[3] H. Ayoobi, N. Potyka, and F. Toni. ProtoArgNet: Interpretable Image Classification with Super-Prototypes and Argumentation [Technical Report], Nov. 2023.

[4] A. Barredo Arrieta, N. Díaz-Rodríguez, et al. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion*, 58:82–115, June 2020. ISSN 1566-2535. doi: 10.1016/j.inffus.2019.12.012.

[5] E. Black, N. Maudet, and S. Parsons. Argumentation-based Dialogue. In D. Gabbay, M. Giacomin, G. R. Simari, and M. Thimm, editors, *Handbook of Formal Argumentation, Volume 2*. College Publications, 2021.

[6] R. Bommasani, D. A. Hudson, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

[7] M. Bucher, S. Herbin, and F. Jurie. Semantic bottleneck for computer vision tasks. In *Asian Conference on Computer Vision (ACCV)*, 2018.

[8] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin. Emerging Properties in Self-Supervised Vision Transformers. pages 9650–9660, 2021.

[9] C. Chen, O. Li, C. Tao, A. J. Barnett, J. Su, and C. Rudin. This looks like that: deep learning for interpretable image recognition. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, number 801, pages 8930–8941. Curran Associates Inc., Red Hook, NY, USA, Dec. 2019.

[10] D. Coquenet, C. Rambour, E. Dalsasso, and N. Thome. Leveraging Vision-Language Foundation Models for Fine-Grained Downstream Tasks, July 2023. arXiv:2307.06795 [cs].

[11] A. Das, S. Kottur, J. M. F. Moura, S. Lee, and D. Batra. Learning Cooperative Visual Dialog Agents With Deep Reinforcement Learning. pages 2951–2960, 2017.

[12] A. Das, S. Kottur, et al. Visual Dialog, Aug. 2017. arXiv:1611.08669 [cs].

[13] H. de Vries, F. Strub, S. Chandar, O. Pietquin, H. Larochelle, and A. Courville. GuessWhat?! Visual object discovery through multi-modal dialogue. *arXiv:1611.08481 [cs]*, Feb. 2017. arXiv: 1611.08481.

[14] A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, and S. Thrun. Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542(7639):115–118, Feb. 2017. ISSN 1476-4687. doi: 10.1038/nature21056.

[15] J. S. B. Evans and K. E. Stanovich. Dual-process theories of higher cognition: Advancing the debate. *Perspectives on psychological science*, 8 (3):223–241, 2013.

[16] JSBT. Evans. Dual process theories of deductive reasoning: Facts and fallacies. *The Oxford handbook of thinking and reasoning*, pages 115–133, 2012.

[17] S. Han, L. Zhuo, Y. Liao, and S. Liu. LLMs as Visual Explainers: Advancing Image Classification with Evolving Visual Descriptions, Nov. 2023. URL http://arxiv.org/abs/2311.11904. arXiv:2311.11904 [cs].

[18] D. Hitchcock, P. McBurney, and S. Parsons. A Framework for Deliberation Dialogues. 2001.

[19] G. Irving, P. Christiano, and D. Amodei. AI safety via debate, Oct. 2018. arXiv:1805.00899 [cs, stat].

[20] D. Kahneman. *Thinking Fast and Slow*. Farrar, Straus and Giroux, 2011.

[21] P. W. Koh, T. Nguyen, Y. S. Tang, S. Mussmann, E. Pierson, B. Kim, and P. Liang. Concept Bottleneck Models. In *Proceedings of the 37th International Conference on Machine Learning*, pages 5338–5348. PMLR, Nov. 2020. ISSN: 2640-3498.

[22] E. M. Kok, J.-J. C. Meyer, H. Prakken, and G. A. W. Vreeswijk. A Formal Argumentation Framework for Deliberation Dialogues. In P. McBurney, I. Rahwan, and S. Parsons, editors, *Argumentation in Multi-Agent Systems*, volume 6614, pages 31–48. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011. ISBN 978-3-642-21939-9 978-3-642-21940-5. doi: 10.1007/978-3-642-21940-5_3. Series Title: Lecture Notes in Computer Science.

[23] A. Kori, B. Glocker, and F. Toni. Explaining Image Classification with Visual Debates, May 2023. arXiv:2210.09015 [cs].

[24] O. Li, H. Liu, C. Chen, and C. Rudin. Deep Learning for Case-Based Reasoning Through Prototypes: A Neural Network That Explains Its Predictions. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), Apr. 2018. ISSN 2374-3468. doi: 10.1609/aaai.v32i1.11771.

[25] Y. Li, Y. Du, K. Zhou, J. Wang, X. Zhao, and J.-R. Wen. Evaluating Object Hallucination in Large Vision-Language Models. In H. Bouamor, J. Pino, and K. Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 292–305, Singapore, Dec. 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.20.

[26] M. Losch, M. Fritz, and B. Schiele. Semantic Bottlenecks: Quantifying and Improving Inspectability of Deep Representations. *International Journal of Computer Vision*, 129(11):3136–3153, Nov. 2021. ISSN 0920-5691, 1573-1405. doi: 10.1007/s11263-021-01498-0.

[27] Maxime Oquab, T. Darcet, T. Moutakanni, Vo, et al. DINOv2: Learning Robust Visual Features without Supervision, Apr. 2023. arXiv:2304.07193 [cs].

[28] S. Menon and C. Vondrick. Visual Classification via Description from Large Language Models. Sept. 2022.

[29] G. L. Murphy and H. H. Brownell. Category differentiation in object recognition: Typicality constraints on the basic category advantage. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 11(1):70–84, 1985. ISSN 1939-1285. doi: 10.1037/0278-7393.11.1.70.

[30] W. D. Neys. Advancing theorizing about fast-and-slow thinking. *Behavioral and Brain Sciences*, 46:e111, Jan. 2023. ISSN 0140-525X, 1469-1825. doi: 10.1017/S0140525X2200142X.

[31] M.-E. Nilsback and A. Zisserman. Automated Flower Classification over a Large Number of Classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pages 722–729, Bhubaneswar, India, Dec. 2008. IEEE. doi: 10.1109/ICVGIP.2008.47.

[32] E. Poeta, G. Ciravegna, et al. Concept-based Explainable Artificial Intelligence: A Survey, Dec. 2023. arXiv:2312.12936 [cs].

[33] N. Potyka. Interpreting Neural Networks as Quantitative Argumentation Frameworks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(7):6463–6470, May 2021. ISSN 2374-3468. doi: 10.1609/aaai.v35i7.16801.

[34] M. Praß, C. Grimsen, M. König, and M. Fahle. Ultra Rapid Object Categorization: Effects of Level, Animacy and Context. *PLOS ONE*, 8 (6):e68051, June 2013. ISSN 1932-6203. doi: 10.1371/journal.pone.0068051.

[35] P. C. Quinn. Development of Subordinate-Level Categorization in 3- to 7-Month-Old Infants. *Child Development*, 75(3):886–899, 2004. ISSN 1467-8624. doi: 10.1111/j.1467-8624.2004.00712.x.

[36] A. Radford, J. W. Kim, et al. Learning Transferable Visual Models From Natural Language Supervision. 2021.

[37] P. Rajpurkar, J. Irvin, et al. CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning, Dec. 2017.

[38] E. Sklar and S. Parsons. Towards the Application of Argumentation-Based Dialogues for Education. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 3*, AAMAS '04, pages 1420–1421, USA, July 2004. IEEE Computer Society. ISBN 978-1-58113-864-1.

[39] D. Thauvin, S. Herbin, W. Ouerdane, and C. Hudelot. Interpretable image classification through an argumentative dialog between encoders. In *27TH EUROPEAN CONFERENCE ON ARTIFICIAL INLIGENCE*, Saint-Jacques de Compostelle, Spain, Oct. 2024. URL https://hal.science/hal-04673794.

[40] W. van der Hoek and M. Wooldridge. Chapter 24 Multi-Agent Systems. In F. van Harmelen, V. Lifschitz, and B. Porter, editors, *Foundations of Artificial Intelligence*, volume 3 of *Handbook of Knowledge Representation*, pages 887–928. Elsevier, Jan. 2008. doi: 10.1016/S1574-6526(07)03024-6.

[41] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. *The Caltech-UCSD Birds-200-2011 Dataset*. juill 2011.

[42] P. Xu, W. Shao, K. Zhang, P. Gao, S. Liu, M. Lei, F. Meng, S. Huang, Y. Qiao, and P. Luo. LVLM-eHub: A Comprehensive Evaluation Benchmark for Large Vision-Language Models, June 2023.

[43] K. Čyras, A. Rago, et al. Argumentative XAI: A Survey. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, pages 4392–4399, Montreal, Canada, Aug. 2021. International Joint Conferences on Artificial Intelligence Organization. ISBN 978-0-9992411-9-6. doi: 10.24963/ijcai.2021/600.