Neurosymbolic Visual Reasoning for Ambiguity Resolution

Justin Brody
Franklin and Marshall College
Lancaster, PA 17604
United States
justin.brody@fandm.edu

Abstract

While modern computer vision systems have notched tremendous successes there are still obstacles to deploying such systems in real-world scenarios. Two salient obstacles are the lack of flexibility that comes with systems trained on fixed categories of data and a lack of stability. Indeed, it is not uncommon when using a pre-trained classifier on a video stream to find it will correctly identify an object as a tree (for example) in one frame only to fail to identify the same object as a tree in the next. In this paper we propose a system that can ameliorate both of these issues by introducing reason into the process of object detection. In particular, we will introduce a hybrid computer-vision / logical reasoning system that observes the world, reasons about what it sees, and can change its judgement as a result of that reasoning.

We test our system by addressing the challenging problem of distinguishing between real and artificial objects in two datasets, showing improved performance over our base computer vision model in both cases.

1 Introduction

Our goal in this paper is to develop a hybrid perceptual/reasoning system which can reason about and resolve ambiguities in perception. This has the potential to ultimately improve the stability of computer vision algorithms which are deployed in the wild. While computer vision has matured tremendously in the past decade (e.g. (Bochkovskiy, Wang, and Liao 2020), (Wang, Bochkovskiy, and Liao 2022), (Dosovitskiy et al. 2020)), even the basic problem of general image classification is not fully solved. Indeed, it is not uncommon when using a pre-trained classifier on a video stream to find it will correctly identify an object as a tree (for example) in one frame only to fail to identify the same object as a tree in the next. In this paper we propose a system that can ameliorate this issue by "closing the loop" between vision and reason. In particular, we will introduce a hybrid computer-vision / logical reasoning agent that observes the world, reasons about what it sees, and can change its judgement as a result of that reasoning.

Our approach to ambiguity resolution relies on a two-part architecture as detailed in 3. The vision system, based on OpenAI's CLIP, produces estimates that each of a set of labels applies to a given image. If the top two estimates are sufficiently close, reasoning engines are instantiated. The first reasons about what properties should be present in the image if the top category is correct, while the second engine performs similar reasoning under the assumption that the second category is correct. Taken together, these two engines thus give a measure of relative consistency for each category; these scores are then combined to adjust the confidences for the top categories, as detailed in Section 3. In Section 4 we describe a feature selection algorithm which measures the performance of features described in a given knowledge base an greedily selects a performant subset of those features.

As a system for ambiguity resolution, the system offers some unique features. As discussed in Section 2, several contemporary systems combine the capacities of deep vision-language models with some kind of reasoning. To our knowledge, ours is the only system that uses reasoning specifically in combination with additional queries of the underlying vision model specifically to resolve ambiguities (as opposed, for example, to engaging in question answering).

One of the great advantages of our system is that is allows for a CLIP-based reasoner to be deployed in a wide range of environments with improved performance that does not rely on fine-tuning the underlying model. The reasoning of the symbolic system is also transparent and thus explainable. In situations where an in-distribution dataset is available, our fine-tuning mechanism allows for greater performance enhancement.

Using our system does rely on the manual creation of a knowledge base, but creating a reasonable set of axioms is usually fairly straightforward. If in-distribution data is available, it can be used for feature selection on the knowledge base.

While our goal with this system is not to provide a general purpose commonsense reasoning or visual question answering system, the system as a whole an be used as an improved-accuracy visual object detector which can ground such systems in a number of architectures.

We test our system on a dataset which involves distinguishing real trains from model trains. We summarize our primary contributions as follows:

1. We develop a hybrid vision reasoning system that can

employ reason to correct mistakes in its perception and demonstrate its effectiveness.

2. We develop a feature selection algorithm to choose a performant subset of a given knowledge base and demonstrate its effectiveness.
3. We present a new dataset consisting of images of real trains and model trains, and demonstrate that our model outperforms basic CLIP in distinguishing the two.

Although our target is object detection, for most of the datasets we have in mind we cannot compare against standard object classifiers like ResNet (He et al. 2016) because the former does not have categories that distinguish between real and artificial objects. Thus our use of a language-vision model like CLIP is essential.

We also note that, given a particular knowledge base, while we do not explicitly measure the individual performance of each rule in the knowledge base, this measurement is done implicitly by the feature selection algorithm when it is used.

2 Related Work

This work fits broadly into the recent literature on using *neurosymbolic* methods (Sarker et al. 2021) to apply to the broad category of research on *visual reasoning*. As its name indicates, the former refers to a fusion of neural and symoblic approaches to artificial intelligence, while the latter refers to the uses of reasoning in visual processing. In recent years, two primary subtasks of visual reasoning have seen significant progress: visual question answering (VQA) (Antol et al. 2015), (Kafle and Kanan 2017), (Wang et al. 2022), (Bao et al. 2021), (Zeng et al. 2022) and visual commonsense reasoning (VCR) (Park et al. 2020), (Zellers et al. 2019). Of particular relevance is (Amizadeh et al. 2020), which approaches visual question answering neurosymbolically and disentangles visual processing from reasoning in a way which mirrors our own approach. Their approach differs, however in some fundamental ways. Their notion of inference is statistical rather than employing classical logical inference, and their primary concern is in visual question answering. While they develop a mechanism for learning from context, it is based on updating Bayesian priors rather than being based on a logical knowledge base. Finally, their goal is to answer questions accurately in the absence of good perceptual information rather then attempting to rectify perceptual errors per se.

Our system introduces a hybrid neurosymbolic reasoning framework designed specifically for ambiguity resolution in computer vision tasks. We build on neurosymbolic approaches like (Amizadeh et al. 2020), but with a focus on refining and correcting errors in perceptual understanding rather than answering visual questions or addressing commonsense reasoning. Unlike the purely probabilistic inference methods employed by previous work, we employ a paraconsistent logic system for handling contradictory information, drawing specifically on active logic's ability to both draw inferences and defuse contradictions (Elgot-Drapkin et al. 1999), (Anderson et al. 2008), (Purang 2001), (Goldberg 2022). This provides a novel approach to combining high-dimensional neural inference with traditional symbolic reasoning, particularly for tasks that require distinguishing fine-grained differences (for example, distinguishing real from artificial objects).

While reasoning and visual models have been integrated before our contribution is unique in that it uses logical reasoning to not only supplement but correct perceptual errors by issuing queries back to the underlying vision model. This active feedback loop allows the system to dynamically adjust classifications, improving accuracy without needing to retrain the neural components.

Central to our system is the open vocabulary of CLIP, which allows for reasoning with arbitrary predicates. This open vocabulary has been exploited in a number of neurosymbolic system; one prominent example is ConceptGraphs (Gu et al. 2024) which use vision language models to dynamically build a scene-graph of an unknown environment, thereby grounding symbolic reasoning in neural pattern recognition.

3 Architecture

Overall Architecture

Our overall architecture is summarized in Figure 2. It consists primarily of a perceptual component and a reasoning component. The perceptual system (based on OpenAI's CLIP) takes a set of images and a set of potential labels as input and outputs a score for the likelihood that a given image corresponds to each of the given labels.

Specifically, given a set of categories $\mathbf{C} = [C_1 \quad \dots \quad C_n]$ and an image I, the perceptual system will produce a distribution $\mathbf{P} = [p_1 \quad \dots \quad p_n]$ where p_i is the estimated confidence that I is an instance of category C_i. If the distribution $\mathbf{P}$ exhibits sufficient ambiguity about the results (defined as the difference between the top two scores being less than some ambiguity threshold θ), then the reasoning system is deployed. This systems instantiates two logical reasoning engines, e_1 and e_2 as follows. A common knowledge base $\mathcal{K}$ is used which describes general features of the categories under consideration. If the top-scoring feature is C_t, then the engine e_1 is run with its initial knowledge base set to $\{ C_t \} \cup \mathcal{K}$. Similarly if the second top-scoring feature is C_s, the engine e_2 has its initial knowledge base set to $\{ C_s \} \cup \mathcal{K}$ (see Figure 2).

As an example, consider an instance of our system that is trying to distinguish images of real trains from those of model trains. Suppose that CLIP classifies an image as a real train with confidence $p_t = 0.5$ and a model train with confidence $p_s = 0.4$. If the system is using an ambiguity threshold of $\theta = 0.25$, it will consider this difference small enough to require further analysis. Thus it instantiates two different ALMA engines as in Figure 2: one under the assumption that the image is real train and the other under the assumption that it is a model train.

In the knowledge base shown, the presence of smoke or plastic are given as indicative of whether the image is of a train or model train. The system then queries CLIP to determine whether or not these categories are present in the image, with the result that plastic is detected while smoke is

not. The system thus derives a contradiction for the hypothesis that it is a train and uses the mechanism described in Section 3 to recalculate p_t and p_s.

CLIP

The OpenAI CLIP model (Radford et al. 2021) is a text-vision model that takes as input an arbitrary set of textual prompts and a set of images and outputs a contrastive score which indicates how well each textual prompt matches each image. Because it can work with arbitrary text prompts, it gives a way of performing image classification on arbitrary categories. To our knowledge, this was not previously possible in computer vision where systems worked on a fixed set of categories. This capacity is essential for the system we propose, since it is designed to work with arbitrary knowledge bases. It is especially helpful for the task we focus on (distinguishing model trains from real trains) since the categories are sufficiently similar that an off-the-shelf computer vision algorithm is unlikely to have both among its fixed categories.

We also note that previous work has indicated that textual transformers often perform poorly on basic reasoning tasks (Helwe, Clavel, and Suchanek 2021). We found that CLIP was a reasonable, if highly imperfect, basis for simple reasoning and conjecture that the grounding of language in vision inherent in CLIP may improve the reasoning capacity of the model over pure text transformers.

Active Logic

Our reasoning system is based on the active logic formalism developed by Perlis at al , (Elgot-Drapkin et al. 1999), (Purang 2001), (Anderson et al. 2008). In particular, we use ALMA 2.0, Matthew Goldberg's implementation of the formalism as a reasoning engine (Goldberg 2022), (Goldberg 2019). Active logic is designed to be an internal, time-situated and paraconsistent reasoning system which is specifically designed for agents which reason in real-world situations (as opposed to abstract disembodied agents) (Perlis et al. 2017). For the experiments conducted in this paper, the ability of the system to detect contradictions without introducing new ones is the essential element. In particular, active logic gives a way of defining the number of contradictions that are entailed by a knowledge base. Crucially, this is very different from classical reasoning – in traditional first order logic, if a knowledge base K entails both σ and $\neg\sigma$ for some sentence σ, then K entails both σ' and $\neg\sigma'$ for **every** sentence σ'. Thus in a traditional reasoning system the number of contradictions entailed by a knowledge base is either 0 (if the knowledge base is consistent) or ∞ (otherwise). Active logic avoids this by developing logical consequences over time, so that as soon as a contradiction is derived it can be recognized and defused (along with its sources). In particular, active logic is a *step-logic* (Elgot-Drapkin et al. 1999) – it does not assume logical omniscience but rather proceeds by instantiating all possible single applications of a deduction rule applied to the current knowledge base.

Since reasoning in active logic occurs from timestep to timestep, at any particular timestep t let us denote the contents of the knowledge-base by K_t. Suppose that an initial knowledge base $\mathcal{K}_0$ consists of $\{\, p, \neg q \rightarrow \neg p, q \rightarrow r, r \rightarrow \neg q \,\}$. At any timestep t, K_t is updated to K_{t+1} by applying inference rules to the sentences in K_t. Thus $K_1 \supseteq \{\, p, \neg q \rightarrow \neg p, q \rightarrow r, r \rightarrow \neg q, \mathbf{q}, \neg\mathbf{q}, \neg\mathbf{p} \vee \neg\mathbf{r}, \mathbf{p} \rightarrow \mathbf{r} \,\}$, where the sentences in boldface have been derived from K_1 by applying forward chaining and resolution inference rules (see Chapter 9 of (Russell and Norvig 2020) for a discussion of these inference rules). In the next reasoning step, the active logic engine will note the presence of both q and $\neg q$, conclude that a contradiction has been derived, and mark the sentences as distrusted. This prevents any further reasoning from employing q or $\neg q$ and thus the derivation of arbitrary contradictions. Thus K_2 will replace $\{\, q, \neg q \,\}$ with the set $\{\, \mathrm{distrusted}(q), \mathrm{distrusted}(\neg q), \mathrm{contradiction}(q, \neg q) \,\}$.

Amiguity Resolution

If the perceptual system assigns categories with sufficient ambiguity, then two instances of ALMA, e_t and e_s are instantiated with a common knowledge base. The category with the highest confidence is also added to e_t and the category with the second highest confidence is added to e_s. Then tens step of reasoning are performed in each engine, and the number of contradictions in $e_t$is saved as χ_t, while the number of contradictions in e_s is saved as χ_s.

If p_t and p_s are the highest two confidences, we want to use the contradictions to redistribute the confdience between the two top categories in a way that preserves the total confidence $p_t + p_s$ and gradually assigns the confidence of p_t to p_s as $\chi_t - \chi_s$ gets larger.

To that end, a reduction factor ρ is computed according to

$$\begin{aligned}\rho &= [\rho_1 \quad \rho_2] = 1 - \mathrm{softmax}(\chi_t, \chi_s) \\ &= \left[\left(1 - \tfrac{\exp(\chi_t)}{\exp(\chi_t)+\exp(\chi_s)}\right) \quad \left(1 - \tfrac{\exp(\chi_s)}{\exp(\chi_t)+\exp(\chi_s)}\right)\right]\end{aligned}$$

Thus the reduction factor ρ_t is a real value in the interval $[0, 1]$ which will be used to compute how much of p_t's confidence will be given to p_s and vice versa: intuitively, if $\rho_t = 1$ then all of p_t is given to the second category while if $\rho_t = 0$ then all of p_s is given to the top category. The details of the computation are as follows.

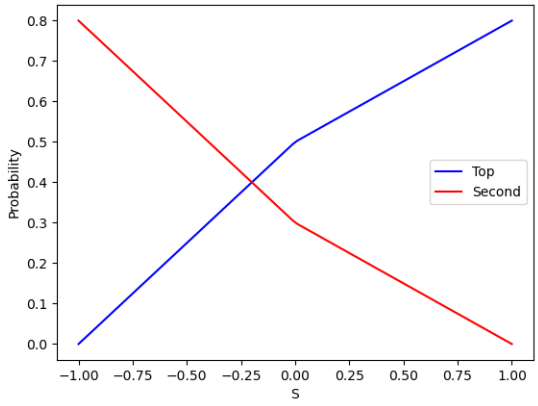
We define $\tau := (\rho_t - \rho_s)$; thus τ is a scaled difference of ρ_s from ρ_t. We then define $S := \tanh(\tau)$. We have $-1 < S < 1$ and S can be thought of as a bounded measure of the difference between ρ_t and ρ_s.

Finally, we define:

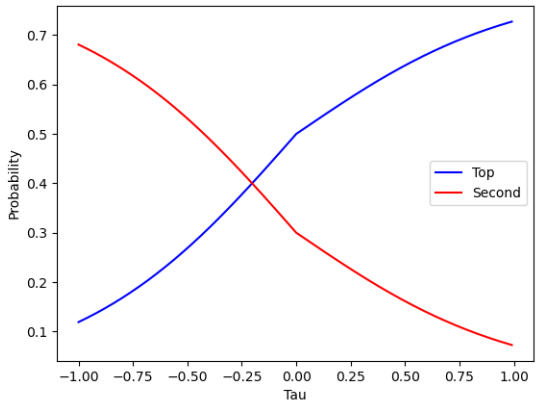
$$\hat{p}_t = \begin{cases} p_t + S p_s & \text{if } S > 0 \\ p_t + S p_t & \text{otherwise} \end{cases}$$
$$\hat{p}_s = (p_s + p_t) - \hat{p}_t$$

It is easy to check that:

1. As $S \rightarrow 1$, $\hat{p}_t \rightarrow p_t + p_s$ and $\hat{p}_s \rightarrow 0$. Thus when the top category generates significantly fewer contradictions, it will absorb most of the confidence of p_s. Similarly, as $S \rightarrow -1$, $\hat{p}_t \rightarrow 0$ and $\hat{p}_s \rightarrow p_t + p_s$. Thus when the second category generates significantly fewer contradictions, it will absorb most of the confidence of p_t

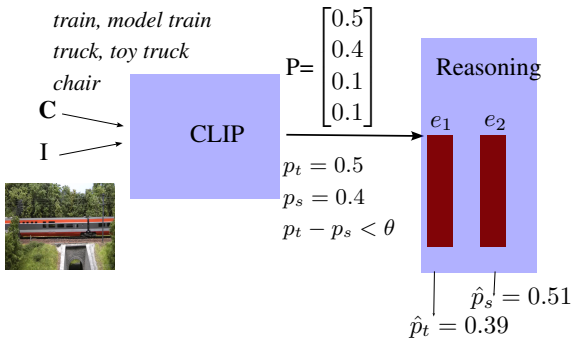


(a) Redistribution as a function of S



(b) Redistribution as a function of τ

Figure 1: Redistribution of Probability ($p_t = 0.5, p_s = 0.3$)



Step	*ALMA Instance 1*	*ALMA Instance 2*
0:	train	model_train
0:	train → not(model_train)	train → not(model_train)
0:	model_train → not(train)	model_train → not(train)
0:	smoke → train	smoke → train
0:	plastic → model_train	plastic → model_train
	CLIP queried for [`smoke, plastic` *]*	
1:	*plastic ∧ not(smoke)*	*plastic ∧ not(smoke)*
2:	model_train	model_train
3:	not(train)	
4:	**contra(train, not(train))**	

Figure 2: Main Architecture.
On being presented with an ambiguous image of a train, the top two confidences are close enough to warrant ambiguity resolution. Two logical reasoning engines are instantiated; Instance 1 operates under the assumption that the image is of a train while Instance 2 assumes that the image is of a model train. CLIP is queried for associated features and Instance 1 derives a contradiction, meaning that the image seems to be logically inconsistent with being a train. The ambiguity resolution algorithm then adjusts the confidences of the respective categories to output $\hat{p}_t$ and $\hat{p}_s$.

2. At $S = 0$, $\hat{p}_t = p_t$ and $\hat{p}_s = p_s$. Thus when the top category generates the same number of contradictions as the second category, the confidences are unchanged.
3. $\hat{p}_t + \hat{p}_s = p_t + p_s$. Thus the total confidence shared by the top two categories is unchanged.

We thus redistribute the total confidence $p_t + p_s$ according to a piecewise linear function of S. Graphs are shown in Figure 1.

In the example in Figure 2, we will have $\chi_t = 1, \chi_s = 0$ leading to $\rho = [0.27 \quad 0.73]$, $\tau = -0.23, S = -0.23, \hat{p}_t = 0.5 - .23(0.5) = 0.39, \hat{p}_s = 0.9 - 0.39 = 0.51$. Thus the reasoning process will adjust the confidences sufficiently to indicate to the system that the image is of a model train.

4 The Knowledge Base

The additional efficacy of our system relies on having a good knowledge base that allows the system to reason effectively in the presence of ambiguity. There are some immediate challenges that this presents. First, CLIP is not designed to be logically coherent, so obvious implications that hold in the real world (if something is a toy train, then it is a toy) may not be respected by CLIP. Second, the problem of distinguishing an artifical object from a real one is inherently asymmetric – while "tells" exist which give away that an object is artificial, there do not generally exist "anti-tells" which guarantee that an object is authentic. Finally, CLIP is not a uniformly accurate classifier, and it's performance on classifying different categories can be expected to vary and produce differing numbers of false positives and false negatives. Taken together, these considerations mean that it will not suffice to simply deploy a set of facts about the real-world differences between trains and model trains; we will need to calibrate our knowledge base with the perceptual module and the data we expect.

We note that the active logic engine can support several inferences processes, including forward chaining and resolution. Of these, resolution is the more general: for example from $p \wedge q \rightarrow r$ a forward chaining process will draw no infereces unless p and q are both established while a resolution process will note the equivlance of the implication with $\neg p \vee \neq q \vee r$ and conclude that $q \rightarrow r$ when p is established.

In general, forward chaining gives for a more controlled and easily analyzed process, and we will focus on forward chaining based inferences in this paper. An implication $p \to q$ that is to be used with forward chaining is written in ALMA as `fif(p, conclusion(q))`.

We note that given our reliance on forward chaining, we can expect tells to increase the accuracy scores for model trains (possibly while decreasing the score for real trains) and conversely for anti-tells and real trains. This observation will allow us some intuitive control over the performance of the algorithm in our knowledge base selection.

For these initial experiments, we chose a fairly simple knowledge base which consisted entirely of tells and anti-tells – that is simple object detection conditions which implied either that the image was of a real train or a model train. In future work we will explore more complicated conditions, where, e.g., the presence of smoke might indicate a real train if the smoke is black but a model train if the smoke is light. Future work will also explore the use of temporal features which span several frames. For example, even with single frames we found that looking for moving foliage was an accurate anti-tell, even though no actual movement will be detectable in the still image. Since CLIP is trained on still images, more effecitely looking for such features will involve a certain amount of reasoning across frames.

Our final knowledge base is given in the Supplementary Materials. We note that most of the tells and anti-tells listed are *ad hoc*.

This is mitigated by the quantitative analysis of each feature that is implicit in the feature selection. Given this, it is perhaps unsurprising that, as we will see, a large set of ad hoc features degrades overall performance while a knowledge base calibrated through feature selection improves it.

Each of these logical terms was given an equivalent natural language term, as listed in the appendix. For example, the logical literal `wall_mounted_tv` would be translated into the prompt "wall mounted television" which was used as the input to CLIP.

One of the ways in which the lack of logical coherence in CLIP manifested was that certain exclusive categories ended up not being desirable. For example, demanding that a scene was either indoor or outdoor let to a degradation in performance because CLIP would often give low scores to both `indoor` and `outdoor`.

We note that in the reasoning algorithm when CLIP is being queried for the presence or absence of certain features, this is done by presenting CLIP with the image along with two prompts, one for the feature and one for its negation. For example, in querying for smoke we would present CLIP with the prompts $\{$ `smoke`, `notsmoke` $\}$.

Because of the possibility of CLIP making mistakes, each of the implications in our knowledge will be of varying utility – indeed for each there is some chance of false positives or false negatives. The situation is somewhat akin to employing a number of noisy sensors to an object detection task, and indeed we found that using the entire knowledge base gives poor results (see Section 6). We therefore employ a feature selection algorithm to attempt to measure the efficacy of each individual implication and choose an optimal subset.

Feature Selection

Our feature selection algorithm is roughly modelled on a number of mutual feature selection algorithms which greedily grow a set of features by locally maximizing mutual information (see, for example, Joint Mutual Information and others, as surveyed in (Brown et al. 2012)). Specifically, for each feature in the knowledge base (e.g. *moving foliage*) we measure the number of true positive, true negatives, false positives and false negatives of that feature (counting a determination that an image is of a model train as a positive). We then initialize our feature set S to the feature with the highest estimated F-score. We then recursively grow S, naively estimating the F score that would be obtained by adding each individual feature to S and adding the feature that would increase this measure the most if such a feature exists.

Recall that our aim to distinguish amongst a set of fixed categories $\mathbf{C} = [C_1 \ \ldots \ C_n]$. Throughout this discussion, let $\mathcal{K}_B$ denote a supporting knowledge base which indicates that the categories under consideration are mutually exclusive. For example, when distinguishing trains from model trains we would have $\mathcal{K}_B = \{$

```
fif(train, conclusion(not(model_train))),
fif(model_train, conclusion(not(train)))
```

$\}$

Let $\mathcal{K}$ denote any consistent knowledge base which contains $\mathcal{K}_B$ (intuitively $\mathcal{K}$ will be the large knowledge base which we will select from).

The complete set of features F will consist of the predicates which occur as literals in $\mathcal{K}$, with the exceptions of `train` and `model_train`.

Definition 1. For any feature $f \in F$ and $\mathcal{K}$ containing $\mathcal{K}_B$, we define $\mathcal{K}(f)$, the knowledge-base associated with f, as the minimal subset of $\mathcal{K}$ which satisfies:

1. $\mathcal{K}_B \subseteq \mathcal{K}(f)$
2. $\mathcal{K}(f) \cup \{ f \} \vdash \bigvee_{1 \leq i \leq n} C_i$

Intuitively, $\mathcal{K}(f)$ is the smallest subset of $\mathcal{K}$ which allows us to make a category determination when f is true.

For example,

$$\mathcal{K}(\texttt{ceiling}) = \{ \texttt{fif}(\texttt{ceiling}, \texttt{conclusion}(\texttt{indoor})), \texttt{fif}(\texttt{indoor}, \texttt{conclusion}(\texttt{model\_train})) \}$$

. We will insist that in general $\mathcal{K}$ is chosen so that $\mathcal{K}(f)$ is always well-defined and $\mathcal{K}(f) \cup \{ f \}$ is consistent; this is a fairly modest requirement.

Definition 2. Let $S \subseteq F$, let i be an image. We will say that S *is estimated as positive* or *negative on* i based on the following.

- We define the *knowledge base associated with* S as $\mathcal{K}(S) := \bigcup_{f \in S} \mathcal{K}(f)$.

- Let $\chi_j^*(i, S)$ be the number of contradictions derived when the reasoning step is performed on image i with initial knowledge base $\mathcal{K}(S) \cup \{ \mathtt{C_j} \}$ for category C_j. We estimate $\chi_j^*(i, S)$ by

$$\chi_j(i, S) = \sum_{f \in S} \chi_j^*(i, \{ f \})$$

  . This allows for a naive but inexpensive estimate of $\chi_j(i, S)$ by using measures of the number of contradictions for each individual feature.
- Let us say that i *is estimated positive at* S if C_j is the top category from the base vision system and $\chi_j(i, S)$ is minimal amongst $\chi_k(i, S)$ for $1 \leq k \leq n$.
- Similarly we say that i *is estimated negative at* S if the visual reasoning system gives the highest confidence to some category besides C_j with $\chi_j(i, S)$ minimal amongst $\chi_k(i, S)$.

Note that the naivete in this estimate comes from two sources:

1. Assuming that a difference in the number of contradictions will be enough to determine the assigned classification without considering the details of the ambiguity resolution algorithm.
2. Using the naive estimates of χ_j^*.

The intuition is that the minimal $\chi_j(i, S)$ indicates which category engenders the fewest contradictions; i being estimated positive or negative indicates that the vision model agrees or disagrees with this indication.

We now define an analogy to the F_1-score which forms the basis of our feature-selection.

Definition 3. Let $S \subseteq F$ and let D be a set of images. Then $\mathrm{ETP}(S, D)$ is the number of images in D which are estimated positive at S and which are of the ascribed category. Similar definitions hold for $\mathrm{EFP}(S, D)$, $\mathrm{ETN}(S, D)$ and $\mathrm{EFN}(S, D)$ – the number of estimated false positives, true negatives and false negatives. Then

$$\mathrm{EF}(S, D) := \frac{2\,\mathrm{ETP}(S, D)}{2\,\mathrm{ETP}(S, D) + \mathrm{EFP}(S, D) + \mathrm{EFN}(S, D)}$$

is the estimated F-score at S.

Our approach in feature selection is to greedily build a subset $S \subseteq F$ that maximizes the estimated F-score for a fixed sample of images D. To that end, for any set of remaining features R let $\Psi(S, R) := \arg\max_{f \in R} EF(S \cup \{ f \})$. Then we recursively define $\Phi_F := \Phi'(\emptyset, F)$, where $\Phi'(S, R)$ is defined by Algorithm 1.

The final output of the algorithm is then $\mathcal{K}(S)$ for the optimal subset $S = \Phi_F$.

Algorithm 1: Feature Selection Algorithm

function $\Phi'(S, R)$
  $f \leftarrow \Psi(S, R)$
  if $\mathrm{EF}(S \cup \{ f \}) \leq \mathrm{EF}(S)$ **or** $R = \emptyset$ **then**
    $M \leftarrow S$
  else
    $S' \leftarrow S \cup \{ f \}$
    $R' \leftarrow R \setminus \{ f \}$
    $M \leftarrow \Phi'(S', R')$
  end if
  return M
end function

5 Experimental Setup

Our experimental setup will worked with a dataset focused on distinguishing model trains from real trains. This was created from working with four different videos, two of model trains and two of real trains. The table below gives the names and access URLs for the videos we used (with permission), along with a code that will be used to refer to each video in what follow

To this end, we work with two videos of model trains (Television 2014), (Television 2022) along with two videos of real trains (Armstrong 2017), (Armstrong 2019) – these are referred to in what follows as F_1, F_2, R_1, R_2 respectively.

Each video was rendered into a series of still images at a rate of 60 frames per second[1]. Before being used, the datasets were also cleaned so that each image showed either a train or model train – in practice this meant removing parts of the opening and closing credits as well as scene transitions (the latter tended to use some kind of transformation from one scene to the next so that the images were not clearly images from a single scene).

We then ran the images through our system, using the following following parameters:

- The categories passed into clip for each image were: `'train', 'model train', 'truck', 'toy truck', 'chair'`
- The ambiguity threshold was 0.25; that is an image would be considered ambiguous and go through the ambiguity resoltuion process if the difference between the top two confidences was less than 0.25.
- We used one of several knowledge bases, starting with that given in Figure 3. The other knowledge bases are derived from this and described in Section 6

6 Results

Full Knowledge Base for Trains

Taken over all four videos, we processed 180788 images, of which 40210 were found to be ambiguous. Of the ambiguous images, 2.7% were properly corrected from the raw CLIP classification, while 28.5% were improperly corrected, 19% were properly not corrected and 49.7% were improperly not corrected. By itself, CLIP made the correct classification for 75.3% of the images while incorporating the reasoning component brought the accuracy down to 69.5%. We report data for all the train images, all he model train images, and each individual dataset in Table 1.

[1]This was accomplished using `ffmpeg -i INPUT_FILE -vf fps=60 %04d.jpg`

```
fif(train, conclusion(not(toy_train))).
fif(model_train, conclusion(not(train))).
fif(train, conclusion(not(plastic))).
fif(train, conclusion(metal)).
fif(train, conclusion(outdoor)).
fif(plastic, conclusion(model_train)).
fif(toy_people, conclusion(model_train)).
fif(indoor, conclusion(model_train)).
fif(toy, conclusion(model_train)).
fif(ceiling_fan, conclusion(ceiling)).
fif(ceiling, conclusion(indoor)).
fif(ceiling_light, conclusion(ceiling)).
fif(wall_mounted_tv, conclusion(indoor)).
fif(exit_sign, conclusion(indoor)).
fif(painted_sky, conclusion(indoor)).
fif(desk, conclusion(indoor)).
fif(telephone, conclusion(indoor)).
fif(fake_foliage,
    conclusion(model_train)).
fif(giant_phone,
    conclusion(model_train)).
fif(small_scale,
    conclusion(model_train)).
fif(miniature, conclusion(model_train)).
fif(detailed, conclusion(model_train)).
fif(intricate, conclusion(model_train)).
fif(plastic, conclusion(model_train)).
fif(indoor_lighting,
    conclusion(model_train)).
fif(led_lights,
    conclusion(model_train)).
fif(small_lights,
    conclusion(model_train)).
fif(bright_lights,
    conclusion(model_train)).
fif(not(toy), conclusion(train)).
fif(outdoor, conclusion(train)).
fif(outdoor_lighting,
    conclusion(train)).
fif(sunset_lighting, conclusion(train)).
fif(real_person, conclusion(train)).
fif(real_smoke, conclusion(train)).
fif(moving_trees, conclusion(train)).
fif(clouds, conclusion(train)).
fif(moving_water, conclusion(train)).
fif(moving_foliage, conclusion(train)).
fif(moving_plants, conclusion(train)).
fif(black_smoke, conclusion(train)).
fif(side_steam, conclusion(train)).
fif(large_scale, conclusion(train)).
```

Figure 3: Trains Knowledge Base.

Dataset	Number of Images	Raw Acc (%)	Reasoned Acc (%)
Overall	180788	**75.2**	69.5
Model Trains	85565	97.3	**98.5**
Real Trains	95223	**55.5**	43.6
F_1	7375	90.2	**91.7**
F_2	78190	98.0	**99.1**
R_1	51861	**59.5**	47.2
R_2	43362	**50.7**	39.2

Dataset	Num Amb	Good Corr	Bad Corr	Good Non-Corr	Bad Non-Corr
Overall	40210	1098	11443	7671	19998
Model	3220	1019	4	1784	413
Real	36990	79	11439	5887	19585
F_1	948	117	0	618	213
F_2	2272	902	4	1166	200
R_1	20212	36	6398	2454	11324
R_2	16778	43	5041	3433	8261

Table 1: Accuracy of Corrections Using Full Knowledge Base. Here "Num Amb" refers to the number of ambiguous images, "Good Corr" refers to the number of those images that were corrected to align with the ground truth, "Bad Corr" refers to the number that were corrected in a way that contradicted the ground truth, and similarly the "Non-Corr" columns refer to the ambiguous images which were not corrected

Analyzing these results with the full knowledge base we note first that by itself, CLIP is able to correctly identify 75.2 % of the images. Its strengths are asymmetric though – while it correctly identified 97.3% of model trains it was only correct 55.5% of the time on real trains. Thus CLIP has an apparent (surprising) bias toward classifying trains as model trains.

Reasoning seemed to exacerbate this asymmetry – using the full knowledge base the accuracy for model trains went up while the accuracy for real trains went down. Overall accuracy went down to 69.5%. Thus we say that the tells in the knowledge base were on the whole more efficacious than the anti-tells. It is worth noting that in the original knowledge base the number of tells greatly outnumbers the number of anti-tells.

The more detailed data reveals that on the datasets with real trains, the reasoning made many bad corrections but also made more non-corrections – that is failed to make corrections in cases where a correction was the right thing to do.

Feature Selection for Trains

After running the feature selection algorithm described above (using 80% of each dataset), we obtained the following reduced knowledge base:

```
fif(train, conclusion(not(toy_train))).
```

Dataset	Number of Images	Raw Acc (%)	Reasoned Acc (%)
Overall	180788	75.2	**82.8**
Model Trains	85565	**97.3**	96.6
Real Trains	95223	55.5	**70.4**
F_1	7375	**90.2**	85.1
F_2	78190	**98.0**	97.7
R_1	51861	58.9	**75.5**
R_2	43362	50.0	**64.4**

Dataset	Num Amb	Good Corr	Bad Corr	Good Non-Corr	Bad Non-Corr
Overall	40210	15350	627	17417	6816
Model	3220	32	625	1166	1397
Real	36990	15318	2	16251	5419
F_1	948	0	375	247	326
F_2	2272	32	250	919	1071
R_1	20212	8626	1	8547	3038
R_2	16778	6692	1	7704	2381

Table 2: Accuracy of Corrections Using Feature-Selection Based Knowledge Base

```
fif(toy_train, conclusion(not(train))).
fif(moving_foliage, conclusion(train)).
fif(side_steam, conclusion(train)).
fif(train, conclusion(not(plastic))).
fif(plastic, conclusion(toy_train)).
fif(desk, conclusion(indoor)).
fif(indoor, conclusion(toy_train)).
fif(real_smoke, conclusion(train)).
fif(small_lights, conclusion(toy_train)).
fif(indoor_lighting, conclusion(toy_train)).
fif(real_person, conclusion(train)).
fif(large_scale, conclusion(train)).
fif(sunset_lighting, conclusion(train)).
fif(ceiling, conclusion(indoor)).
```

We note that 6 features are tells (`plastic`, `desk`, `indoor`, `small_lights`, `indoor_lighting`, `ceiling`) and 6 are anti-tells (`moving_foliage`, `side_steam`, `real_smoke`, `real_person`, `large_scale`, `sunset_lighting`) so we have a much more balanced set in terms of what is being looked at.

The results are presented in Table 2. We note that compared to using the full knowledge base, overall accuracy and the accuracy on the real trains are significantly improved, although this comes at the cost of a slight decrease in performance on the model trains. At some level this is not too surprising, since the feature selection algorithm was designed to improve the F-score of the overall dataset, and the easiest path to that was by improving the performance on real trains.

Additional Feature Selection for Trains

Finally, we tried removing some anti-tells from the reduced knowledge base in the previous section to try to transfer some of the extra accuracy from the real trains to the model trains. We specifically worked with removing the anti-tells with the top scores: `moving_foliage` and `side_steam`. The results of this are below.

Dataset	Number of Images	Raw Acc (%)	Reasoned Acc (%)
Overall	180788	74.5	**75.5**
Model Trains	85565	97.8	**98.0**
Real Trains	95223	53.7	**55.3**
F_1	7375	91.5	**91.6**
F_2	78190	98.3	**98.5**
R_1	51861	58.2	**59.1.5**
R_2	43362	48.2	**50.1**

Table 3: Accuracy of Corrections Using Modified Feature-Selection Based Knowledge Base

Thus we find that we get very modest improvements in each of the datasets at significant cost to the overall accuracy

7 Conclusion

We have demonstrated the adding a reasoning module has the capacity to improve the raw classification results given by CLIP. The real-world consequence of this is that in situations where a classifier needs to adapt quickly, our system enables this much more quickly then would be possible by retraining a purely neural system, even if sufficient data were at hand. We thus allow for dynamically configurable classifiers which can handle even somewhat subtle classification tasks.

References

Amizadeh, S.; Palangi, H.; Polozov, A.; Huang, Y.; and Koishida, K. 2020. Neuro-symbolic visual reasoning: Disentangling. In *International Conference on Machine Learning*, 279–290. PMLR.

Anderson, M. L.; Gomaa, W.; Grant, J.; and Perlis, D. 2008. Active logic semantics for a single agent in a static world. *Artificial Intelligence*, 172(8-9): 1045–1063.

Antol, S.; Agrawal, A.; Lu, J.; Mitchell, M.; Batra, D.; Zitnick, C. L.; and Parikh, D. 2015. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, 2425–2433.

Armstrong, M. 2017. CF2105 Best Train Video Clips. *YouTube*. https://www.youtube.com/watch?v=1XG0QoXbshU.

Armstrong, M. 2019. Steam Trains Galore 7! *YouTube*. https://www.youtube.com/watch?v=5yVqfuPE7_8.

Bao, H.; Wang, W.; Dong, L.; Liu, Q.; Mohammed, O. K.; Aggarwal, K.; Som, S.; and Wei, F. 2021. Vlmo: Unified vision-language pre-training with mixture-of-modality-experts. *arXiv preprint arXiv:2111.02358*.

Bochkovskiy, A.; Wang, C.-Y.; and Liao, H.-Y. M. 2020. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*.

Brown, G.; Pocock, A.; Zhao, M.-J.; and Luján, M. 2012. Conditional likelihood maximisation: a unifying framework for information theoretic feature selection. *The journal of machine learning research*, 13: 27–66.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Elgot-Drapkin, J.; Kraus, S.; Miller, M.; Nirkhe, M.; and Perlis, D. 1999. Active logics: A unified formal approach to episodic reasoning. Technical report.

Goldberg, M. D. 2019. alma-2.0. *GitHub*. https://github.com/mclumd/alma-2.0.

Goldberg, M. D. 2022. *Time-Situated Metacognitive Agency and Other Aspects of Commonsense Reasoning*. Ph.D. thesis.

Gu, Q.; Kuwajerwala, A.; Morin, S.; Jatavallabhula, K. M.; Sen, B.; Agarwal, A.; Rivera, C.; Paul, W.; Ellis, K.; Chellappa, R.; et al. 2024. Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 5021–5028. IEEE.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Helwe, C.; Clavel, C.; and Suchanek, F. M. 2021. Reasoning with transformer-based models: Deep learning, but shallow reasoning. In *3rd Conference on Automated Knowledge Base Construction*.

Kafle, K.; and Kanan, C. 2017. Visual question answering: Datasets, algorithms, and future challenges. *Computer Vision and Image Understanding*, 163: 3–20.

Park, J. S.; Bhagavatula, C.; Mottaghi, R.; Farhadi, A.; and Choi, Y. 2020. Visualcomet: Reasoning about the dynamic context of a still image. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16*, 508–524. Springer.

Perlis, D.; Brody, J.; Kraus, S.; and Miller, M. J. 2017. The Internal Reasoning of Robots. In *COMMONSENSE*.

Purang, K. 2001. Alma/carne: implementation of a time-situated meta-reasoner. In *Proceedings 13th IEEE International Conference on Tools with Artificial Intelligence. ICTAI 2001*, 103–110. IEEE.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; Krueger, G.; and Sutskever, I. 2021. Learning Transferable Visual Models From Natural Language Supervision. In Meila, M.; and Zhang, T., eds., *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, 8748–8763. PMLR.

Russell, S.; and Norvig, P. 2020. *Artificial intelligence: a modern approach*. Prentice Hall.

Sarker, M. K.; Zhou, L.; Eberhart, A.; and Hitzler, P. 2021. Neuro-symbolic artificial intelligence: Current trends. *arXiv preprint arXiv:2105.05330*.

Television, P. 2014. Nearly realistic model train layout from France. *YouTube*. https://www.youtube.com/watch?v=wMd2zyD2Ncc.

Television, P. 2022. Northern Virginia Model Railroaders - One of the Largest Model Railway Layouts in the United States. *YouTube*. https://www.youtube.com/watch?v=1fwdwr_VC04.

Wang, C.-Y.; Bochkovskiy, A.; and Liao, H.-Y. M. 2022. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv preprint arXiv:2207.02696*.

Wang, W.; Bao, H.; Dong, L.; Bjorck, J.; Peng, Z.; Liu, Q.; Aggarwal, K.; Mohammed, O. K.; Singhal, S.; Som, S.; et al. 2022. Image as a foreign language: Beit pretraining for all vision and vision-language tasks. *arXiv preprint arXiv:2208.10442*.

Zellers, R.; Bisk, Y.; Farhadi, A.; and Choi, Y. 2019. From recognition to cognition: Visual commonsense reasoning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 6720–6731.

Zeng, Y.; Zhang, X.; Li, H.; Wang, J.; Zhang, J.; and Zhou, W. 2022. X2-VLM: All-In-One Pre-trained Model For Vision-Language Tasks. *arXiv preprint arXiv:2211.12402*.