Combining Vision-Language Models and Weak Supervision for Nuanced Vision Classification Tasks

Seyed Mohamad Ali Tousi[†*], Jacket Demby's[†*] , Ramy Farag[†*], Gbenga Omotara[†*], G. N. DeSouza[†]
[†]Vision Guided and Intelligent Robotics Laboratory (ViGIR), EECS Dept.
University of Missouri Columbia - MO - US
(stousi, udembys, rmf3mc, goowdf, desouzag)@missouri.edu

Abstract

Nuanced-concept image classification tasks often require substantial labeled data. The labeling process for such problems is time-consuming and labor-intensive. While zero-shot methods like Modeling Collaborator have shown promising results, they generally lack a versatile open source pipeline for domain-independent, multi-class fine-grained classification. We are proposing a classification pipeline consisting of weak supervision and open-source Vision Language Models (VLMs) to be employed in both binary and multi-class nuanced classification problems. Our proposed pipeline is domain-independent as it uses knowledge embedded in the pre-training of VLMs. This eliminates the need for additional fine-tuning for specific contexts, as required by methods such as AdaptCLIPZS. In our proposed pipeline, VLMs serve as weak labelers in the classification tasks, while a Weak Supervision (WS) model aggregates those labels and produce a set of pseudo labels (pseudo ground-truth) to train an end classifier. We have conducted multiple experiments to demonstrate the validity of the pipeline in both binary and multi-class classification tasks. The experimental results have shown that our proposed pipeline is capable of producing superior results in both binary and multi-class problems compared to the state-of-the-art zero-shot classification methods.

1. Introduction

Obtaining annotated datasets is the bottleneck of a majority of supervised learning tasks. Traditionally, this crucial step of generating labels has been recognized as labor-intensive, time-consuming, and costly, often requiring the hiring of experts or employing *crowdsourcing* methods. These problems are particularly worse for large-scale datasets used in objective tasks – e.g. labeling a "zebra", as in [3, 26, 36]. Recently, research efforts have increasingly focused on developing models that can handle subjective or nuanced classification tasks [16, 35, 41]. Labeling data for these tasks is even more challenging when it involves individual/subjective judgment or interpretation – e.g., deciding whether "a meme is hateful or not" [16]. This often leads to variability among annotators, making the process complex even for experts. In that sense, the research community has been working actively to minimize human involvement in the labeling process of subjective datasets [34, 35, 41]. Labeling and training frameworks like Agile Modeling (AM) [35] and Modeling Collaborator (MC) [41] appear to have successfully achieved this goal by substantially reducing the human effort involved in labeling to approximately 2000 and just 100 required labels, respectively. However, they still require a certain amount of human intervention, which is not ideal for subjective classification tasks.

Weak Supervision (WS) has been successful in addressing data labeling challenges by systematically generating training labels from potentially imperfect, noisy, or imprecise sources [45]. These sources, used as labeling functions (LFs), include heuristics, knowledge-based, and pre-trained models among others. Collectively, they can produce a large set of training labels. However, LFs frequently produce unreliable labels due to their inherent inability to generate ground-truth labels, while they may also exhibit dependencies among each other – which can cause additional problems. To address some of these issues, Weak Supervision employs label models that can aggregate the noisy predictions from LFs into refined, fused training labels, which are then used to train end models for downstream tasks [30–33, 45]. In this regard, Snorkel [30, 31, 33] was introduced as a data programming framework that provides a comprehensive system for creating training data in two-stages, using WS. In the first stage, Snorkel uses a label model to combine the imperfect labels generated by user-defined LFs. Then, in the second stage, these aggregated labels are utilized to train an end model in a supervised manner. Notably, LFs are a fundamental component of WS frameworks.

In the meantime, Vision-Language Models (VLMs) have

[*]These authors contributed to the paper equally.

emerged as a class of multimodal learners to bridge the gap between language and image, i.e. between Natural Language Processing (NLP) and Computer Vision (CV). VLMs can interpret, reason, and generate unimodal or multimodal outputs based on the inputs, which include visual (e.g., images, videos) and linguistic (text) data. In a recent survey [12], VLMs have been categorized into three main groups based on their capabilities: (1) models dedicated to vision-language understanding [17, 28, 29, 37], (2) models focused on unimodal output generation (e.g., text) derived from multimodal inputs, [2, 18–21] and (3) models that generate multimodal outputs from multimodal inputs [4, 23, 39, 40]. Several VLMs have been explored by the computer vision community, including Flamingo [2, 28], Llama-vid [18], among others [4, 17, 19–21, 23, 37, 39, 40]. Among VLMs, CLIP [29] certainly stands out for its ability to perform zero-shot image classification without requiring task-specific training datasets. By leveraging their zero-shot classification capabilities across various domain tasks, VLMs emerge as promising candidates for use as LFs to generate weak labels.

In this work, we exploit the use of VLMs as LFs in a Weak Supervision (WS) framework for subjective classification tasks. The main contributions of our work are the following:

1. We propose Labeling Collaborator (LC), a unified WS domain-independent pipeline that employs open-source Vision-Language Models (VLMs) as labeling functions (LFs) for unlabeled image classification problems.
   - Unlike the Modeling Collaborator's pipeline (MC) [41], which utilizes the combined strengths of VLMs, Large Language Models (LLMs), Visual Question-Answering (VQAs), and Image Captioning (IC) models, our pipeline only relies on generic, open-source VLMs – i.e. which do not need to be state-of-the-art models nor fine-tuned on specific downstream tasks.
   - Unlike Agile Modeling (AM) [35], which relies on user input, our pipeline is end-to-end and does not require any user input or machine learning expertise.
   - Unlike AdaptCLIPZS [34], which uses labeled datasets for fine-tuning CLIP [29], our method automatically generates labels for a given dataset by leveraging WS and open-source VLMs as labeling functions without requiring prior labeled datasets.
   - Also, because our pipeline uses open-source VLMs, we do not require knowledge distillation as MC [41] did; making our pipeline simpler, faster, and easier for non-experts to use;
2. We demonstrate that by using several VLMs with weak prediction accuracy as labeling functions (LFs) on two challenging datasets (Hateful Memes [16], and EuroSAT [14]), we can train a lightweight end classifier with a prediction accuracy higher than the accuracies of the weak LFs themselves. Furthermore, we show that our label model produces superior/comparable performances compared to other closed/open source methods in 14 different nuanced concept binary datasets introduced by Agile Modeling [35];
3. While Agile Modeling [35] and Modeling Collaborator [41] are binary classification-based frameworks, ours is multi-class; and
4. To the best of our knowledge, our research is the first to propose a unified framework for fine-grained and subjective concept classification.

The remaining of this paper is structured as follows: Section 2 provides a brief survey of WS and VLMs, including early attempts to leverage VLMs for data labeling. Section 3 describes and details the components of our proposed LC framework for developing subjective classification models in the presence of unlabeled data. Section 4 summarizes and discusses our experimental results. Finally, Section 6 concludes the paper and outlines potential future research directions based on our work.

2. Background and Related Work

Weak Supervision (WS): WS approximates supervision by using various forms of “weak" labels to achieve strong performance [30, 33, 45]. Ratner et al. [33] initially introduced data programming as a WS framework for programmatically generating labels for unlabeled training datasets using various sources, significantly reducing the need for manual labeling. Building on this concept, Ratner et al. [30] developed Snorkel, an enhanced data programming framework that serves as a comprehensive two-stage system for creating training data using WS. First, Snorkel allows users to define labeling functions that encode heuristics, patterns, or leverage external knowledge bases. These functions, which may be noisy and have overlaps, create a matrix of probabilistic labels without requiring ground-truth data. The framework employs a generative model to estimate the accuracies and correlations among these labeling functions, synthesizing them into probabilistic labels. These probabilistic labels are then used to train a discriminative model, such as a deep neural network, which generalizes beyond the labeling functions and achieves improved predictive performance on unseen data.

To tackle the challenge of combining multiple weak supervision (WS) sources into a reliable training signal, Ratner et al. [31, 32] introduced **MeTaL**, a framework for multi-task WS that models the accuracies and dependencies of noisy sources using a matrix completion approach. This improves label quality and enhances downstream model performance. Building on this, Sala et al. [42] proposed **Dugong**, a framework for sequential data (e.g., videos) that fuses multi-resolution WS signals—such as frame-, window-, and scene-level labels—using a probabilistic graphical model. Without relying on ground truth, Dugong estimates source reliability to generate high-quality probabilistic labels, achieving strong results on video and sensor tasks.

Inspired by the work mentioned above, our proposed LC framework employs multiple VLMs as labeling functions to generate weak labels for unlabeled data across diverse domain knowledge, and leverages WS to generate fused probabilistic labels, which are then used to train an end model for downstream tasks.

Vision-Language Models (VLMs): Radford et al. introduced CLIP (Contrastive Language-Image Pre-training) [29] as a method to learn visual representations based on natural language guidance. This approach leverages a large dataset of image-text pairs collected from the internet and employs a contrastive learning objective to train image and text encoders jointly. CLIP [29] and its variants [17, 28, 37] focus on aligning image and text embeddings in a shared multimodal space, enabling robust zero/few-shot transfer capabilities to downstream vision tasks without additional training. Alayrac et al. [2] introduced Flamingo, a framework for few-shot learning that combines pre-trained vision and language models to handle sequences of text and images/videos and produce text outputs. The model converts visual data into fixed-length representations and uses gated cross-attention layers within a frozen LLM to take into account visual context during text generation. This design helps Flamingo adapt to different multimodal tasks with minimal data, achieving top performance in few-shot scenarios across various benchmarks.

Liu et al. [19, 20] introduced Visual Instruction Tuning through their LLaVA (Large Language and Vision Assistant) model, aiming to build a general-purpose vision-language assistant. The model uses a combination of a CLIP-based visual encoder and a language model (Vicuna [10, 27]) trained with multimodal instruction-following data generated using GPT-4 [1]. This framework enables LLaVA to handle diverse vision-language tasks by following specific instructions, allowing it to generalize across various multimodal applications. Mu et al. [24] introduced SLIP, a multi-task learning framework that combines self-supervised learning and language-image pre-training to enhance visual representation learning. By integrating CLIP [29] pretraining, (SimCLR) [5, 6], and and self-supervised learning; SLIP leverages the strengths of both methods. This approach improves performance across various evaluation tasks, including zero-shot transfer, linear classification, and fine-tuning. A new class of VLMs has taken a step toward performing any-to-any input/output modalities[4, 23, 39, 40]. Notably, Mizrahi et al [23] proposed 4M, as a framework for training a single unified Transformer encore-decoder using a masked modeling objective across a variety of input/output modalities including text, images, and videos. This approach results in foundational models that train for any-to-any modalities, learn rich representations, and provide reliable performance across diverse vision tasks. All the VLMs discussed above rely on large-scale labeled datasets with aligned image-text pairs tailored to their training domains. Moreover, they often struggle with fine-grained categorization, where obtaining sufficient labeled image-text pairs is particularly challenging.

VLM-based Subjective Data Labeling: Stretcu et al. [35] introduced Agile Modeling (AM), a human-in-the-loop model training framework for subjective visual concepts. Users start by defining a concept using text descriptions, which are used to select images from large unlabeled datasets using VLMs models like CLIP [29]. Then, the users label these images, which are used to train a classifier. Through an active learning process, the system interactively selects uncertain images for the users to label, and refines the classifier via one or more rounds. This framework allows non-machine learning experts to rapidly build customized classifiers for nuanced concepts with minimal effort and has shown improved performance for fine-grained classification tasks.

Toubal et al [41] aimed at reducing the human effort required for subjective or nuanced data labeling by introducing, MC (Modeling Collaborator) with significantly reduced human effort compared to AM [35]. MC reduces the typical requirement of manually annotating thousands of images to just 100 labeled samples plus some natural language interactions. The framework integrates components like LLMs, a VQA model, and a captioning VLM to generate and process questions, perform reasoning, and produce reliable image annotations for training purposes. MC demonstrates good performance in classifying subjective visual concepts compared to existing zero-shot and AM methods. To automate the process of labeling vast amounts of unlabeled data without any human intervention, Saha et al [34] proposed AdaptCLIPZS, a framework that enhances zero-shot classification performance of VLMs in fine-grained domains. AdaptCLIPZS leverages LLMs to generate detailed category-specific descriptions (e.g., visual attributes). These generated descriptions are combined with existing image datasets and employed in context-rich prompts to fine-tune VLMs. This approach demonstrated consistent improvements in zero-shot classification accuracy across multiple benchmarks.

Our work draws inspiration from the recent advances in WS and VLMs mentioned above to propose a unified WS-based framework that employs multiple open-source VLMs as weak labeling functions to address subjective binary and multi-class image classification problems relying on unlabeled datasets. Furthermore, it provides an improved approach compared to the recently proposed AM [35] and MC

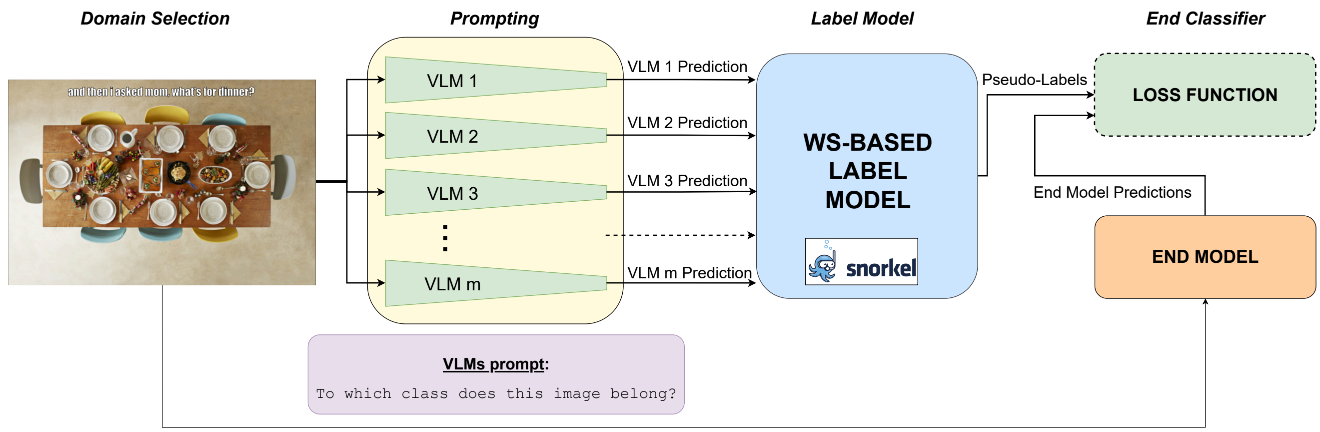


Figure 1. Overview of our proposed Labeling Collaborator (LC) training pipeline. Our approach utilizes a Weak Supervision (WS) framework that leverages multiple open-source Vision-Language Models (VLMs) as labeling functions to generate pseudo-labels for unlabeled input images. Then, these pseudo-labels are used to train the end model in a supervised learning setup.

[41] labeling frameworks with absolutely no human labeling effort involved.

3. Methodology

As shown in Figure 1, the proposed Labeling Collaborator (LC) pipeline consists of four main components: Domain Selection; Prompting, Label Modeling, and End-Classifier.

3.1. Domain Selection

Consider a dataset $\mathscr{D}_1 = \{x_i\}_{i=1}^n$ consisting of unlabeled images x_i defining the domain for which we would like to train a classification model. We refer to this first step of our LC framework as domain selection where the user gathers a subjective dataset of interest without the need for the corresponding label.

3.2. Prompting

LC leverages *weak* LFs to annotate unlabeled training data. In other words, instead of designing LFs coined from specific heuristics, LC prompts open-source VLMs to generate *weak* labels for the selected images. For instance, when working with the Hateful Memes dataset [16], one could prompt multiple VLMs with a generic query such as: "is this a hateful meme or not?". We refer to these labels as *noisy* or *weak* because VLMs do not always produce perfect annotations. At this point, a user has created a dataset $\mathscr{D}_2 = \{(x_i, (\Lambda_{i,1}, \ldots, \Lambda_{i,m}))\}_{i=1}^n$ where $(\Lambda_{i,1}, \ldots, \Lambda_{i,m})$ are the m generated labels from m employed VLMs for each x_i of the unlabeled training images. Subsequently, LC will combine the labels originating from multiple *weak* labeling sources to generate more accurate aggregated pseudo-labels. The same pseudo-labels are then used as ground truths to train an end model for downstream tasks in a supervised learning manner.

3.3. Label Modeling

We employ Snorkel [30] as our WS-based labeling model to fuse multiple *weak* labels into single, probabilistic pseudo-labels. Snorkel uses the generative model $p_w(\Lambda, Y)$ in Equation (1) to combine these *weak* labels:

$$p_w(\Lambda, Y) = Z_w^{-1} exp\left(\sum_{i=1}^{m} w^T \phi_i(\Lambda, y_i)\right) \quad (1)$$

where Z_w is a normalizing constant, w is a vector of weight parameters, Λ is the label matrix where each entry $\Lambda_{i,j} = \lambda_j(x_i)$ is obtained by applying a labeling function λ_j (VLMs) to a training image x_i, Y is the ground truth labels, and ϕ_i represents the concatenation of three different factors given by the following equations:

$$\phi_{i,j}^{\text{Lab}}(\Lambda, Y) = \mathbf{1}\{\Lambda_{i,j} \neq \emptyset\}$$
$$\phi_{i,j}^{\text{Acc}}(\Lambda, Y) = \mathbf{1}\{\Lambda_{i,j} = y_i\}$$
$$\phi_{i,j,k}^{\text{Corr}}(\Lambda, Y) = \mathbf{1}\{\Lambda_{i,j} = \Lambda_{i,k}\} \quad (j,k) \in C$$

It is important to note that users do not need to know those ground truths as Snorkel uses a generative approach to compute the weight for each labeler (VLM).

To find the parameters w essential to combine noisy labels into weighted probabilistic ones, $p_w(\Lambda, Y)$ is trained by minimizing the negative log marginal likelihood [30] in Equation (2):

$$\hat{w} = \underset{w}{\text{argmin}} - log \sum_{Y} p_w(\Lambda, Y) \quad (2)$$

At this stage, WS-based labeling modeling has created a pseudo labeled dataset $\mathscr{D}_3 = \{(x_i, \hat{y}_i)\}_{i=1}^n$ where $\hat{y}_i =$

$p_w(y_i|\Lambda)$ represents the final aggregated probabilistic training labels required to train the end model.

3.4. End Model

Once final annotations for the selected image domain are created, the proposed LC pipeline can employ $\mathscr{D}_3$ to train any classifier for downstream tasks in a supervised learning manner. Similar to other classification models [13, 15, 29, 38], our end classifier model has two parts: the image encoder and the projector. The image encoder extracts relevant features from an input image x_i to produce feature embeddings $f(x_i)$ that are provided to the projector to generate the final classification prediction $\hat{y}_i$. In our experiments, we investigated the use of ResNet [13] and CLIP [29] as image encoders and a simple lightweight Multilayer Perceptron (MLP) as a projector. We optimized our end model with the cross-entropy loss function [22].

4. Experimental Results

As explained earlier, the proposed LC relies on open-source, domain agnostic VLMs and Weak Supervision to deliver fine-grained, multi or binary class, and subjective classification of images using a lightweight end classifier. In order to demonstrate those claims, the following research questions must be addressed:

1. Can open-source VLMs provide weak (noisy) labels, which yet lead to aggregated labels strong enough for challenging subjective or nuanced (fine-grained) public datasets?
2. How many VLM labelers are required for such task?
3. How does the proposed framework perform with respect to binary and multi-class classification problems?
4. How does the LC with a lightweight end classification model perform in comparison to state-of-the-art models in subjective datasets?

To answer these questions and showcase the contributions of the proposed LC, we devised and executed a series of associated experiments. In this section we will first present the experimental setup for each of the experiments and then their individual results.

4.1. Experimental Setup

4.1.1 Datasets

For binary classification, we used two sets of subjective datasets; **Hateful Memes** [16] and the **Agile Modeling** [35] datasets. **Hateful Memes** [16] is a multimodal dataset created by Facebook AI to advance research on hate detection in memes. It consists of 10000 image-text pairs belonging to two categories “Not Hateful” or “Hateful”. Agile Modeling [35] dataset comprises 14 sub-datasets referred to as concepts, each with “Positive” and “Negative” classes. Table 1 shows the number of samples in train/test sets of each

	Concept	Train	Test
Easy concepts	arts-and-crafts	3001	707
	dance	4051	730
	emergency-service	6764	675
	hair-coloring	4108	645
	in-ear-headphones	3687	687
	pie-chart	2994	594
	single-sneaker	4216	556
Hard concepts	astronaut	3759	637
	block-tower	3942	669
	gourmet-tuna	3543	576
	hand-pointing	2811	832
	healthy-dish	8715	633
	home-fragrance	4050	716
	stop-sign	3758	704

Table 1. Number of train/test samples in each binary sub-dataset (concept) in the Agile Modeling dataset [35].

of the 14 sub-datasets used from Agile Modeling dataset [35]. For multiclass classification, we use **EuroSAT** [14] which is a dataset designed to address the challenge of land use and land cover classification. This dataset has 27000 images organized by ten classes (with their class index in parentheses): “Annual Crop” (0), “Forest” (1), “Herbaceous Vegetation” (2), “Highway” (3), “Industrial Buildings” (4), “Pasture” (5), “Permanent Crop” (6), “Residential Buildings” (7), “River” (8), and “Sea & Lake” (9).

4.1.2 Weak Labelers

We chose Ollama [25] as the framework to interact with various open-source VLMs to generate noisy labels in the proposed LC. In Ollama, a combination of different multi modal Large Language Models (or VLMs) were used and are listed below [25]:

- **Llava models** (7B, 13B, 34B) [20]: A novel end-to-end trained large multi modal model that combines a vision encoder and Vicuna for general-purpose visual and language understanding.
- **Llava-Phi3** [11]: A new small LLaVA model fine-tuned from Phi 3 Mini.
- **Llava-Llama3** [11]: A LLaVA model fine-tuned from Llama 3 Instruct with better scores in several benchmarks.
- **BakLlava** [20]: BakLLaVA is a multi modal model consisting of the Mistral 7B base model augmented with the LLaVA architecture.
- **MoonDream** [43]: moondream2 is a small vision language model designed to run efficiently on edge devices.
- **MiniCPM** [44]: A series of multi modal LLMs (MLLMs) designed for vision-language understanding.

4.1.3 End Classifier Model

As discussed in Section 3.4, we used a simple lightweight MLP as our projector model. The MLP took as input the features extracted by either CLIP [29] or ResNet[13] – i.e.

Dataset	Labeler VLM	Polarity	Coverage	Overlaps	Conflicts	Correct	Incorrect	Emp. Acc.
Hateful Memes	Llava-7B [19]	[0, 1]	1.00	1.00	0.752	298	202	0.596
	BakLlava [20]	[0, 1]	0.95	0.95	0.710	276	199	0.581
	Llava-13B [20]	[0, 1]	1.00	1.00	0.752	288	212	0.576
	Llava-Llama3 [11]	[0, 1]	1.00	1.00	0.752	278	222	0.556
	Llava-Phi3 [11]	[0, 1]	0.99	0.99	0.744	267	228	0.540
	Llava-34B [20]	[0, 1]	1.00	1.00	0.752	269	231	0.538
	MoonDream [43]	[0, 1]	1.00	1.00	0.752	246	254	0.492
EuroSAT	BakLlava [20]	[1, 3, 4, 5, 6, 7, 8, 9]	0.188	0.187	0.175	935	585	0.615
	Llava-34B [20]	[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]	0.901	0.901	0.766	4462	2840	0.611
	Llava-13B [20]	[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]	0.915	0.915	0.765	4360	3052	0.588
	Llava-7B [19]	[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]	0.974	0.966	0.814	4367	3520	0.554
	Llava-Llama3 [11]	[1, 3, 4, 5, 6, 7, 8, 9]	0.908	0.905	0.757	3016	4339	0.410
	Llava-Phi3 [11]	[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]	0.341	0.338	0.280	1085	1677	0.393
	MiniCPM [44]	[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]	0.772	0.768	0.646	2414	3841	0.386

Table 2. Combined performance metrics of the labeling VLMs with coverage, overlaps, conflicts (on the training sets), and empirical accuracy (on the validation sets) for Hateful Memes and EuroSAT datasets. The coverage, overlaps and conflicts are calculated based on [30]

	Concept	CLIP [29]			CuPL [19]			Ours (LC)		
		Pre	Rec	F1	Pre	Rec	F1	Pre	Rec	F1
Easy concepts	arts-and-crafts	0.68	0.86	0.76	0.68	0.90	**0.77**	0.79	0.68	0.73
	dance	0.51	0.95	0.66	0.52	0.89	0.66	0.73	0.67	**0.70**
	emergency-service	0.53	0.87	0.65	0.54	0.91	0.67	0.76	0.62	**0.68**
	hair-coloring	0.70	0.99	**0.82**	0.70	0.99	**0.82**	0.76	0.38	0.51
	in-ear-headphones	0.43	0.95	0.59	0.44	0.96	0.60	0.58	0.73	**0.64**
	pie-chart	0.52	0.80	0.63	0.50	0.92	0.65	0.70	0.87	**0.78**
	single-sneaker	0.51	0.99	0.67	0.51	1.00	0.67	0.68	0.85	**0.75**
Easy concepts avg.				0.68			0.69			0.69
Hard concepts	astronaut	0.40	0.95	0.56	0.42	0.95	0.58	0.61	0.83	**0.71**
	block-tower	0.38	0.99	0.55	0.37	0.98	0.54	0.43	0.92	**0.58**
	gourmet-tuna	0.29	1.00	0.45	0.29	1.00	0.45	0.42	0.81	**0.55**
	hand-pointing	0.39	0.87	0.54	0.39	0.94	0.55	0.45	0.87	**0.60**
	healthy-dish	0.37	0.99	0.54	0.38	1.00	0.55	0.51	0.66	**0.58**
	home-fragrance	0.40	0.95	0.56	0.40	0.96	**0.57**	0.52	0.54	0.53
	stop-sign	0.48	1.00	0.65	0.49	0.99	0.65	0.57	0.98	**0.72**
Hard concepts avg.				0.55			0.56			0.61
Overall avg.				0.61			0.62			**0.65**

Table 3. Comparison of the performance of our teacher model (LC) with SOTA open-source zero-shot models on nuanced binary easy and hard concepts classification task [35]. LC is outperforming the second best method (CuPL [19]) by **3%** on overall average F1 score.

two sets of pipelines were investigated: CLIP+MLP and ResNet+MLP. For the MLP architecture, we have used the same 128x128x128 structure proposed by [41] for comparison.

4.2. Weak Labelers's Individual Performance

To address research question *(1)* above, we generate weak labels using seven VLMs on all the datasets presented in Section 4.1.1. Table 2 presents the performance of each weak labeler (or VLM) individually on one of the binary (**Hateful Memes** [16]) and the multiclass (**EuroSAT** [14]) datasets. The *Polarity* column indicates the class numbers represented in the labels produced by each labeler (for information on class indices, check 4.1.1). It is important to mention here that for a multi-class dataset like EuroSAT, it is very possible that some labeling functions (VLMs) may not cover the entire range of possible classes, making the generated labels not only weak/noisy, but also incomplete.

4.3. Label Model Optimization

The proposed LC framework produces (aggregated) pseudo-labels by optimizing the negative log marginal likelihood, as presented in Section 3.3. So, regarding research question *(2)*, we investigated the validity of the LC framework for different numbers of labeling functions (VLMs). Figures 2a and 2b show the performance of the labeling model on the **Hateful Memes** and **EuroSAT** datasets af-

	Concept	CuPL	PaLI-X	MC	Ours (LC)
		Machine Annotators			
Easy concepts	arts-and-crafts	0.78	0.77	0.78	**0.81**
	dance	0.72	0.68	0.68	**0.82**
	emergency-service	0.59	0.66	0.72	**0.86**
	hair-coloring	0.77	0.58	**0.80**	0.65
	in-ear-headphones	0.65	0.73	0.72	**0.83**
	pie-chart	0.72	0.82	0.82	**0.89**
	single-sneaker	0.51	0.61	0.56	**0.86**
Easy concepts avg.		0.68	0.69	0.73	**0.82 (+9%)**
Hard concepts	astronaut	0.60	0.65	0.65	**0.89**
	block-tower	0.48	0.49	0.50	**0.56**
	gourmet-tuna	0.54	0.52	0.52	**0.60**
	hand-pointing	0.56	0.81	**0.81**	0.59
	healthy-dish	0.42	0.45	0.53	**0.83**
	home-fragrance	0.56	0.53	0.53	**0.68**
	stop-sign	0.62	0.51	0.64	**0.73**
Hard concepts avg.		0.54	0.57	0.60	**0.70 (+10%)**
Overall average		0.61	0.63	0.66	**0.76 (+10%)**

Table 4. Comparison of machine annotators across easy and hard concepts datasets (obtained from [35]) in terms of the auPR on the training set. Our method outperforms all the SOTA machine labelers by a margin of **10%**.

Model	Labeler	Acc.	Rec.	Prec.	F1
Majority Vote	-	0.58	0.43	0.61	0.50
Ours (TM: Label Model)	-	0.58	0.52	**0.59**	0.55
Ours (SM: ResNet+MLP)	LC	0.54	0.48	0.55	0.51
Ours (SM: CLIP-B+MLP)	LC	0.53	**0.92**	0.51	**0.66**

Table 5. Performance of our proposed framework on the Hateful Memes [16] dataset and its comparison with doing a majority vote among the labelers. TM, SM, and CLIP-B respectively stand for Teacher Model, Student Model, and CLIP-Base model.

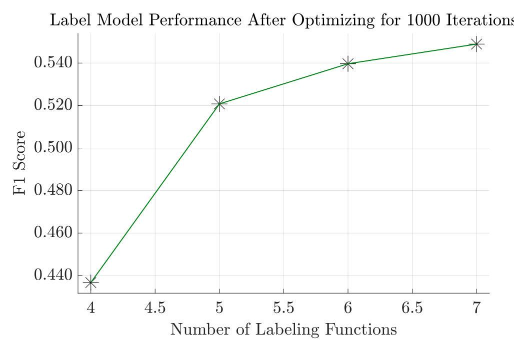
ter 1000 epochs, for 4 to 7 labeling functions. The specific labeling function (VLM) used in each sequence was based on their individual performance on the validation sets. As the figures demonstrate, even with a small number (4) of labeling functions, the labeling model still performed reasonably well on those challenging (subjective, binary and multi class) classification tasks.

4.4. Label Model Performance

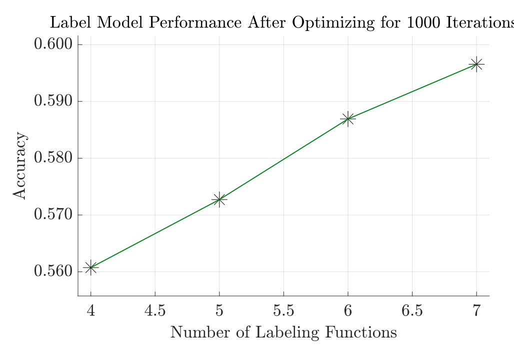
In order to address research question *(3)*, we have compared our label model performance on the **Agile Modeling** [35] datasets. Table 3 compares LC with two open-source state-of-the-art zero-shot methods (CLIP [29] and CuPL [19]) on nuanced binary easy and hard concepts described in Section 4.1.1. For easy concepts, LC outperformed other methods in five out of seven cases, while for hard concepts, it

Model	Labeler	Acc.	Rec.	Prec.	F1
Llava-7B [19]	-	0.59	0.61	0.59	0.60
BakLlava [20]	-	0.58	0.46	0.60	0.52
Llava-13B [20]	-	0.57	0.45	0.60	0.51
Llava-Llama3 [11]	-	0.55	0.70	0.54	0.61
Llava-Phi3 [11]	-	0.53	0.50	0.53	0.52
Llava-34B [20]	-	0.53	0.11	0.75	0.19
MoonDream [43]	-	0.49	0.03	0.40	0.05
CLIP [29]	-	0.53	0.65	0.51	0.57
CuPL [28]	-	0.64	0.87	0.50	0.64
PaLI-X [8]	-	0.61	0.42	0.66	0.51
MC (SM) [41]	MC	0.52	0.64	0.50	0.56
Ours (SM: CLIP-B+MLP)	LC	0.53	0.92	0.51	**0.66**
CLIP+MLP [41]	Human	0.60	0.38	0.65	0.48

Table 6. Performance of our best model compared to other state-of-the-arts in the literature and the labelers themselves on the Hateful Memes [16] dataset. MC, SM, and CLIP-B, respectively stand for Modeling Collaborator [41], Student Model, and CLIP-Base model.



(a) F1 Score on Hateful Memes [16]



(b) Accuracy on EuroSAT [14]

Figure 2. The performance of the Label Model using different numbers of labeling functions (VLMs) on both binary and multi-class classification tasks. The labeling functions have been incorporated into the labeling model in order based on their individual performance on each dataset.

achieved superior performance in six out of seven cases based on the F1 score with CuPL [19] coming second. In

Method	Model	Labeler	Acc.
CLIP-ResNet	RN50	-	0.41
	RN101	-	0.33
	RN50x4	-	0.35
	RN50x16	-	0.40
	RN50x64	-	0.59
CLIP-ViT	B/32	-	0.49
	B/16	-	0.54
	L/14	-	0.59
	L/14-336px	-	0.59
Ours	Majority Vote	-	0.60
	(TM: Label Model)	-	0.62
	(SM: RN18+MLP)	LC	0.60
	(SM: CLIP-B+MLP)	LC	**0.68**

Table 7. Comparison of the performance of our proposed framework and a majority vote of the proposed labelers on the EuroSAT dataset [14] with the state-of-the-art Zero-Shot method CLIP [29]. TM stands for Teacher Model.

Table 4, we compare the labeling capabilities of the two best-performing open-source methods from Table 3 against two closed (non open-source) methods — PaLI-X [7, 9] and MC [41] — in terms of Area Under the Precision-Recall Curve (auPR) on the training set. It can be observed that LC outperforms all other methods in five out of six easy concepts and six out of seven hard concepts, showcasing that LC is a better labeler. Furthermore, when averaging across all easy and hard concepts, **our LC model surpasses all other methods in each case by a margin of 10%**.

4.5. End Model Performance

Finally, to address research question *(4)*, we performed the end classification using an MLP. Table 5 shows the performance of the proposed LC framework for both the label model (teacher) and the end models (students), using the **Hateful Memes** [16] dataset and compares them with the majority vote performance across the weak labelers. In Table 6, we have also compared our best model's performance (SM: CLIP-B+MLP) with the state-of-the-art zero-shot methods on the same **Hateful Memes** [16] dataset. Table 5 also shows the performance of the weak labelers on the **Hateful Memes** [16] dataset. Our student model surpasses all the other methods in terms of F1 score performance including the weak labelers themselves, showcasing the value of using the WS framework to train an end classifier.

For the **EuroSAT** [14] dataset, Table 7 also shows the comparison between the performances of the proposed LC framework and one of the state-of-the-art zero-shot methods (CLIP [29]) using different feature extraction backbones. All of our teacher and student models showed superior performances compared to CLIP with different feature extraction backbones. In fact, **our best model presented a 9% improvement over the best CLIP performance**.

5. Limitations

Our proposed framework relies solely on open-source VLMs to generate pseudo-labels for subjective classification tasks. As a result, it may be limited by the VLM-based labeling functions employed to address the subjective task of interest. A potential issue that may arise is related to the complexity of the prompting framework, particularly for multiclass classification tasks. We observed that as the number of classes increases, the VLM prompts become more complex, which can lead to a higher likelihood of incorrect VLM predictions. For example, the VLMs were prompted with a simple question for the binary Hateful Memes [16] dataset:

> Is this image a hateful meme?

In contrast, for the the 10-class EuroSAT [14] dataset, the prompt specify each class explicitly, resulting in a more complex sentence:

> Classify the satellite image into one of the following categories: SeaLake, PermanentCrop, River, Residential, Pasture, Industrial, Highway, HerbaceousVegetation, Forest, or AnnualCrop. Provide only the class name as your answer.

That is, the prompting framework can become a challenge for multiclass classification problems as the number of classes grows, leading to increased prompt complexity and potentially impacting the prediction accuracy of VLM-based labeling functions and subsequently the performance of the WS-based label model.

6. Conclusion

In this paper, we introduced Labeling Collaborator: a novel Weak Supervision (WS)-based framework designed to eliminate human involvement in the labeling process of nuanced or subjective classification datasets. LC exploits open-source Vision-Language Models (VLMs) as labeling functions (LFs) to generate pseudo-labels for unlabeled data. We demonstrated the effectiveness of our proposed pipeline for both binary and multi-class classification tasks involving subjective concepts. In the end, a lightweight end classifier model trained with the WS-generated labels outperformed other SOTA methods by up to 10% in most cases, and in the other cases it achieved accuracies comparable to the same SOTA models under zero-shot classification scenarios. For future work, we aim to extend the proposed LC framework to more complex, nuanced multiclass classification problems (e.g. with more than 10 classes).

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida,

Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 3

[2] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736, 2022. 2, 3

[3] Görkem Algan and Ilkay Ulusoy. Image classification with deep learning in the presence of noisy labels: A survey. *Knowledge-Based Systems*, 215:106771, 2021. 1

[4] Roman Bachmann*, Oğuzhan Fatih Kar*, David Mizrahi*, Ali Garjani, Mingfei Gao, David Griffiths, Jiaming Hu, Afshin Dehghan, and Amir Zamir. 4m-21: An any-to-any vision model for tens of tasks and modalities. In *NeurIPS*, 2024. 2, 3

[5] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. 3

[6] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E Hinton. Big self-supervised models are strong semi-supervised learners. *Advances in neural information processing systems*, 33:22243–22255, 2020. 3

[7] Xi Chen, Josip Djolonga, Piotr Padlewski, Basil Mustafa, Soravit Changpinyo, Jialin Wu, Carlos Riquelme Ruiz, Sebastian Goodman, Xiao Wang, Yi Tay, et al. Pali-x: On scaling up a multilingual vision and language model. *arXiv preprint arXiv:2305.18565*, 2023. 8

[8] Xi Chen, Josip Djolonga, Piotr Padlewski, Basil Mustafa, Soravit Changpinyo, Jialin Wu, Carlos Riquelme Ruiz, Sebastian Goodman, Xiao Wang, Yi Tay, Siamak Shakeri, Mostafa Dehghani, Daniel Salz, Mario Lucic, Michael Tschannen, Arsha Nagrani, Hexiang Hu, Mandar Joshi, Bo Pang, Ceslee Montgomery, Paulina Pietrzyk, Marvin Ritter, AJ Piergiovanni, Matthias Minderer, Filip Pavetic, Austin Waters, Gang Li, Ibrahim Alabdulmohsin, Lucas Beyer, Julien Amelot, Kenton Lee, Andreas Peter Steiner, Yang Li, Daniel Keysers, Anurag Arnab, Yuanzhong Xu, Keran Rong, Alexander Kolesnikov, Mojtaba Seyedhosseini, Anelia Angelova, Xiaohua Zhai, Neil Houlsby, and Radu Soricut. On scaling up a multilingual vision and language model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14432–14444, 2024. 7

[9] Xi Chen, Josip Djolonga, Piotr Padlewski, Basil Mustafa, Soravit Changpinyo, Jialin Wu, Carlos Riquelme Ruiz, Sebastian Goodman, Xiao Wang, Yi Tay, et al. On scaling up a multilingual vision and language model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14432–14444, 2024. 8

[10] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, 2023. 3

[11] XTuner Contributors. Xtuner: A toolkit for efficiently fine-tuning llm. `https://github.com/InternLM/xtuner`. 5, 6, 7

[12] Akash Ghosh, Arkadeep Acharya, Sriparna Saha, Vinija Jain, and Aman Chadha. Exploring the frontier of vision-language models: A survey of current methodologies and future directions. *arXiv preprint arXiv:2404.07214*, 2024. 2

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 5

[14] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(7):2217–2226, 2019. 2, 5, 6, 7, 8

[15] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. 5

[16] Douwe Kiela, Hamed Firooz, Aravind Mohan, Vedanuj Goswami, Amanpreet Singh, Pratik Ringshia, and Davide Testuggine. The hateful memes challenge: Detecting hate speech in multimodal memes. *Advances in neural information processing systems*, 33:2611–2624, 2020. 1, 2, 4, 5, 6, 7, 8

[17] Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, et al. Grounded language-image pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10965–10975, 2022. 2, 3

[18] Yanwei Li, Chengyao Wang, and Jiaya Jia. Llama-vid: An image is worth 2 tokens in large language models. 2024. 2

[19] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. 2, 3, 6, 7

[20] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26296–26306, 2024. 3, 5, 6, 7

[21] Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. Video-chatgpt: Towards detailed video understanding via large vision and language models. *arXiv preprint arXiv:2306.05424*, 2023. 2

[22] Anqi Mao, Mehryar Mohri, and Yutao Zhong. Cross-entropy loss functions: Theoretical analysis and applications. In *International conference on Machine learning*, pages 23803–23828. PMLR, 2023. 5

[23] David Mizrahi, Roman Bachmann, Oguzhan Kar, Teresa Yeo, Mingfei Gao, Afshin Dehghan, and Amir Zamir. 4m: Massively multimodal masked modeling. *Advances in Neural Information Processing Systems*, 36, 2024. 2, 3

[24] Norman Mu, Alexander Kirillov, David Wagner, and Saining Xie. Slip: Self-supervision meets language-image pre-training. In *European conference on computer vision*, pages 529–544. Springer, 2022. 3

[25] Ollama. Ollama github repository. https://github.com/ollama/ollama. 5

[26] Amandalynne Paullada, Inioluwa Deborah Raji, Emily M Bender, Emily Denton, and Alex Hanna. Data and its (dis) contents: A survey of dataset development and use in machine learning research. *Patterns*, 2(11), 2021. 1

[27] Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. Instruction tuning with gpt-4. *arXiv preprint arXiv:2304.03277*, 2023. 3

[28] Sarah Pratt, Ian Covert, Rosanne Liu, and Ali Farhadi. What does a platypus look like? generating customized prompts for zero-shot image classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15691–15701, 2023. 2, 3, 7

[29] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 2, 3, 5, 6, 7, 8

[30] Alexander Ratner, Stephen H Bach, Henry Ehrenberg, Jason Fries, Sen Wu, and Christopher Ré. Snorkel: Rapid training data creation with weak supervision. In *Proceedings of the VLDB endowment. International conference on very large data bases*, page 269. NIH Public Access, 2017. 1, 2, 4, 6

[31] Alex Ratner, Braden Hancock, Jared Dunnmon, Roger Goldman, and Christopher Ré. Snorkel metal: Weak supervision for multi-task learning. In *Proceedings of the Second Workshop on Data Management for End-To-End Machine Learning*, pages 1–4, 2018. 1, 2

[32] Alexander Ratner, Braden Hancock, Jared Dunnmon, Frederic Sala, Shreyash Pandey, and Christopher Ré. Training complex models with multi-task weak supervision. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 4763–4771, 2019. 2

[33] Alexander J Ratner, Christopher M De Sa, Sen Wu, Daniel Selsam, and Christopher Ré. Data programming: Creating large training sets, quickly. *Advances in Neural Information Processing Systems (NeurIPS)*, 29, 2016. 1, 2

[34] Oindrila Saha, Grant Van Horn, and Subhransu Maji. Improved zero-shot classification by adapting vlms with text descriptions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17542–17552, 2024. 1, 2, 3

[35] Otilia Stretcu, Edward Vendrow, Kenji Hata, Krishnamurthy Viswanathan, Vittorio Ferrari, Sasan Tavakkol, Wenlei Zhou, Aditya Avinash, Emming Luo, Neil Gordon Alldrin, et al. Agile modeling: From concept to classifier in minutes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22323–22334, 2023. 1, 2, 3, 5, 6, 7

[36] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE international conference on computer vision*, pages 843–852, 2017. 1

[37] Zeyi Sun, Ye Fang, Tong Wu, Pan Zhang, Yuhang Zang, Shu Kong, Yuanjun Xiong, Dahua Lin, and Jiaqi Wang. Alpha-clip: A clip model focusing on wherever you want. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13019–13029, 2024. 2, 3

[38] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019. 5

[39] Zineng Tang, Ziyi Yang, Mahmoud Khademi, Yang Liu, Chenguang Zhu, and Mohit Bansal. Codi-2: In-context interleaved and interactive any-to-any generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 27425–27434, 2024. 2, 3

[40] Zineng Tang, Ziyi Yang, Chenguang Zhu, Michael Zeng, and Mohit Bansal. Any-to-any generation via composable diffusion. *Advances in Neural Information Processing Systems*, 36, 2024. 2, 3

[41] Imad Eddine Toubal, Aditya Avinash, Neil Gordon Alldrin, Jan Dlabal, Wenlei Zhou, Enming Luo, Otilia Stretcu, Hao Xiong, Chun-Ta Lu, Howard Zhou, et al. Modeling collaborator: Enabling subjective vision classification with minimal human effort via llm tool-use. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17553–17563, 2024. 1, 2, 3, 4, 6, 7, 8

[42] Paroma Varma, Frederic Sala, Shiori Sagawa, Jason Fries, Daniel Fu, Saelig Khattar, Ashwini Ramamoorthy, Ke Xiao, Kayvon Fatahalian, James Priest, et al. Multi-resolution weak supervision for sequential data. *Advances in Neural Information Processing Systems*, 32, 2019. 2

[43] Vikhyatk. Moondream1. https://huggingface.co/vikhyatk. 5, 6, 7

[44] Yuan Yao, Tianyu Yu, Ao Zhang, Chongyi Wang, Junbo Cui, Hongji Zhu, Tianchi Cai, Haoyu Li, Weilin Zhao, Zhihui He, et al. Minicpm-v: A gpt-4v level mllm on your phone. *arXiv preprint arXiv:2408.01800*, 2024. 5, 6

[45] Jieyu Zhang, Cheng-Yu Hsieh, Yue Yu, Chao Zhang, and Alexander Ratner. A survey on programmatic weak supervision. *arXiv preprint arXiv:2202.05433*, 2022. 1, 2