Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 130 (2024) 250–263

www.elsevier.com/locate/procedia

57th CIRP Conference on Manufacturing Systems 2024 (CMS 2024)

Industrial Language-Image Dataset (ILID): Adapting Vision Foundation Models for Industrial Settings

Keno Moenck*,[a], Duc Trung Thieu[a], Julian Koch[a], Thorsten Schüppstuhl[a]

[a]*Hamburg University of Technology, Institute of Aircraft Production Technology, Denickestraße 17, 21073 Hamburg, Germany*

* Corresponding author. Tel.: +49-40-42878-3341; fax: +49-40-42731-4551. *E-mail address:* keno.moenck@tuhh.de

Abstract

In recent years, the upstream of Large Language Models (LLM) has also encouraged the computer vision community to work on substantial multimodal datasets and train models on a scale in a self-/semi-supervised manner, resulting in Vision Foundation Models (VFM), as, e.g., Contrastive Language–Image Pre-training (CLIP). The models generalize well and perform outstandingly on everyday objects or scenes, even on downstream tasks, tasks the model has not been trained on, while the application in specialized domains, as in an industrial context, is still an open research question. Here, fine-tuning the models or transfer learning on domain-specific data is unavoidable when objecting to adequate performance. In this work, we, on the one hand, introduce a pipeline to generate the Industrial Language-Image Dataset (ILID) based on web-crawled data; on the other hand, we demonstrate effective self-supervised transfer learning and discussing downstream tasks after training on the cheaply acquired ILID, which does not necessitate human labeling or intervention. With the proposed approach, we contribute by transferring approaches from state-of-the-art research around foundation models, transfer learning strategies, and applications to the industrial domain.

© 2024 The Authors. Published by Elsevier B.V.
This is an open access article under the CC BY-NC-ND license (https://creativecommons.org/licenses/by-nc-nd/4.0)
Peer-review under responsibility of the scientific committee of the 57th CIRP Conference on Manufacturing Systems 2024 (CMS 2024)

Keywords: industrial dataset; self-supervised; CLIP; vision foundation model

1. Introduction

Machine vision technologies facilitated by deep learning usually outperform traditional methods, especially in dynamic and open settings. In the scope of training deep models, industrial contexts[1] lack everyday objects and scenes, typically covered by publicly available datasets, which is why applications in these specialized domains here demand custom datasets, e.g., synthetically generated [1, 2, 3, 4], which model the specific object and sensor domain.

The availability of curated, publicly accessible datasets specific to industrial needs is exceedingly sparse, e.g., the MVTec [5, 6, 7, 8], VISION [9], or tool recognition [10] datasets encapsulate only a limited spectrum of objects and support only a handful of trainable tasks based on the provided ground truth annotations. Besides the need for training data, fine-tuning, domain adaptation, or transfer learning, transferring a model from a source to a related target, e.g., object/scene/sensor domain, is ineluctable, which can reduce the necessary samples per conceptual class to only a few shots during training. The model's pre-training is the critical point here, where training data size, variability, and model size directly relate to the overall performance [11].

Large-scale pre-trained foundation models represent a paradigm shift in Artificial Intelligence (AI), characterized by extensive self-supervised training [12]. These models, e.g., BERT [13], the well-known GPT-n series [14, 15, 16], or Llama [17, 18, 19], learn rich knowledge representations capable of transcending to various downstream tasks. The shift in AI drives single tasks and single-modalities learners to a paradigm encompassing diverse tasks and multimodalities, which more closely mimics human perception and cognitive processes. Following Large Language Models (LLM), Vision Foundation Models (VFM) have been upstreamed in the last few years, capable of supporting not only 2D or even 3D modalities but also language [20]. Data for training at scale is typically web-crawled from the vast resources of the Internet, which then demands sophisticated post-processing, posing a variety of chal-

[1] We define the industrial domain as follows: industrial activities serve to produce consumable goods or a capital asset, which includes production as the superordinate term involving all processes around it, including activities from manufacturing, assembly, logistics, or finance. In addition, tasks in the later lifecycle of a product, like Maintenance, Repair, and Overhaul (MRO), also belong to industrial activities. Vision applications are typically closer to the shopfloor than the topfloor.

2212-8271 © 2024 The Authors. Published by Elsevier B.V.
This is an open access article under the CC BY-NC-ND license (https://creativecommons.org/licenses/by-nc-nd/4.0)
Peer-review under responsibility of the scientific committee of the 57th CIRP Conference on Manufacturing Systems 2024 (CMS 2024)
10.1016/j.procir.2024.10.084

lenges [21, 22]. Besides, given such large, partially unstructured datasets, only self-supervised or unsupervised methods are able to learn from the data effectively.

A self-supervised approach capable of learning from text and image modalities is contrastive learning, in which a model learns to distinguish between positive and negative combinations of samples, firstly, nearly concurrently, presented by CLIP [23] and ALIGN [24] at a large scale. Contrastive learning by contrasting positive and negative samples in a batch, in the case of vision and language, is based on a text and image encoder. The idea is that the encoders are trained to output embeddings for the image and text, increasing the similarities of positive samples by decreasing the distance in the joint embedding space and increasing the distance of negative samples. Employing a text encoder allows for natural language supervision, relaxing the necessity of fixed classes as in the case of training traditional deep learning models like a ResNet [25]. This fact makes assembling a dataset at a scale less laborious since assigning an image to a fixed class omits, enabling learning from unstructured data. Different language-image datasets of scale have emerged, ranging from 12M [22] to 5B [21] samples. Since they are based on web-available data, not all cleaned, post-processed, and curated datasets are published, as in the case of CLIP.

Fig. 1: CLIP on the task of classification after (a) transfer learning on the Industrial Language-Image Dataset (ILID) and (b) the zero-shot baseline results.

VFMs exhibit rich knowledge representations, are adaptable to various downstream tasks, and generalize better than conventional models, but only to a certain extent in novel and out-of-distribution domains, necessitating fine-tuning or transfer learning. As demonstrated in Fig. 1, the zero-shot model CLIP, given a highly out-of-distribution image, does not predict nor even close to the ground truth. As already outlined, in the industrial domain, we face non-everyday objects and scenes, which is why we can not rely on commonly available datasets for fine-tuning or transfer learning, which also inhibits the use of VFM here. In this work, we try to make a step in the direction of utilizing VFM capabilities in specialized industrial domains by contributing three-folded:

- We propose a method to generate the Industrial Language-Image Dataset (ILID) from web-crawled data and release a version that covers objects from different industrial-related domains[2]. We publish the pipeline to generate the ILID at github.com/kenomo/ilid.
- We effectively demonstrate transfer learning to CLIP with the given dataset, which outperforms CLIP's zero-shot capabilities.
- We elaborate on different tasks that serve industrial domain-related vision applications. We publish the training- and evaluation-related code here github.com/kenomo/industrial-clip.

This work focuses on utilizing CLIP rather than other vision-language models due to the significant established usage and fine-tuning/transfer learning strategies. Besides, comparing only one established model on the data increases the focus, clarity, and depth of the findings in the scope of this work. Nevertheless, we encourage the reuse of ILID with other strategies or also employ further fine-tuning and transfer learning strategies.

The rest of this work is structured as follows: First, we outline in Sec. 2 existing applications of VFMs in industrial applications, introduce Contrastive Image-Language Pre-training (CLIP) and current existing fine-tuning/transfer-learning approaches. In Sec. 3, we present our overall method of generating the dataset as well as our training procedure. We elaborate on our extensive experiments in Sec. 4. We conclude and discuss this work in Sec. 5.

2. Related Works

2.1. VFMs in industrial applications

Code recognition, object or position recognition, completeness, shape/dimension check, or quantitive or qualitative inspection are typical vision applications in manufacturing [26]. While in manufacturing, these are often suited toward narrow fields of view and close to the object; in the neighboring domain, intralogistics, tasks are besides close ones, like inspecting load carriers for trash, contamination, and damage or documentation, verification, assistance, and automation, perceiving the environment is often of interest, which results in, e.g., foreign debris detection or tracking objects [27, 28]. The first step in the perception pipeline of these applications is typically a fundamental vision task, e.g., in the 2D domain, giving each

[2] Since the data from the web do not belong to us, we are not allowed to publish the images and texts, but we provide the final post-processed metadata, which can be used to reassemble the dataset. Please contact the corresponding author.

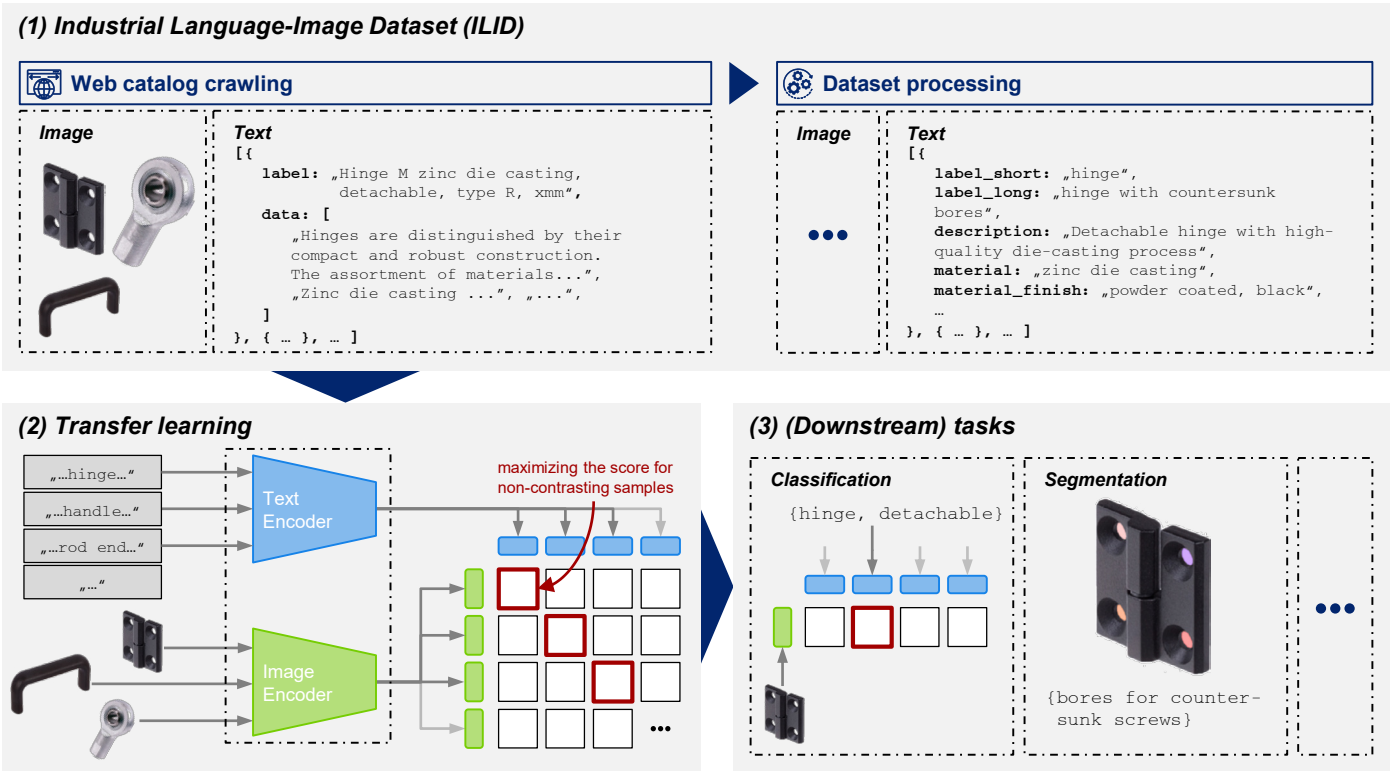

Fig. 2: Overview of this work's method: (1) generation of the Industrial Language-Image Dataset (ILID), (2) transfer learning using the ILID, and (3) evaluating the performance in different tasks.

pixel semantic and clustering pixel to semantically meaningful regions. Then follows additional enhancing the output with further semantics and finally forming the application-specific decision used in, e.g., part of a production system.

Up to this date, there exists only a small set of publications on the topic of utilizing VFMs in one or more steps of such vision pipelines, which we give a small excerpt in the following: On a broader scale [29] explore use cases for deploying VFMs in the industrial context without designing or elaborating on specific architectures and how to train, fine-tune, or do transfer learning. [28] discusses the abilities of the Segment Anything Model (SAM), a class-agnostic segmentation model that outstandingly generalizes to unseen objects and scenes, in the context of vision applications in the aircraft industry, including manufacturing, intralogistics, and MRO. [30] name two use cases in PCB defect inspection and industrial human action recognition.

Current literature throws up ideas on utilizing LLMs, e.g. [31], or VFMs, e.g., [28, 29, 30, 32], in the industrial domain; little is known about how to enable VFM to perform effectively in specific use cases. Besides having suitable datasets, training with the data demands specific strategies. We will elaborate on the aspects in the following sections.

2.2. Contrastive Language-Image Pre-training (CLIP)

CLIP learns rich image-text representations from natural language supervision utilizing natural language as a prediction space to reach higher performance in generalization and transfer. It is not an entirely novel approach; however, the origin of the idea of learning from perceptions in natural language is not exactly dated to specific research. In 1999, [33] explored retrieving words for unknown images based on statistical learning to predict nouns and adjectives. In 2007, [34] demonstrated learning image representations using manifold learning to predict words for images. Recent approaches that emerged before CLIP and learn visual representations from text are Visual representations from Textual annotations (VirTex) [35], Image-Conditioned Masked Language Modeling (ICMLM) [36], and Contrastive Visual Representation Learning from Text (ConVIRT) [37].

2.2.1. Contrastive learning

A contrastive learning model consists of two main components: (1) an encoder for all input modalities and (2) a loss function measuring the similarity between positive and negative pairs. The encoder can be reused from other models and training, e.g., demonstrated by OpenScene [38], which employs a frozen text and 2D image encoder while training a 3D point cloud encoder for language-based 2D/3D scene understanding. The encoder models are trained to complement and comprehend each other fully by encoding similar concepts of images and text in similar vectors. That is, a text representing "photo of a hinge" would output a similar vector as the image counterpart and be further away from images that are not connected, as shown in Fig. 3. Besides prompting for the object's name, a

sufficiently trained text encoder would encode, e.g., conceptual close activities near the object's name embedding (s. Fig. 3).

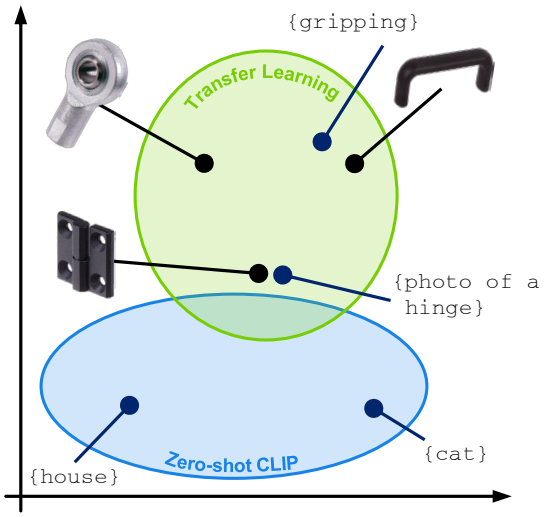

Fig. 3: Joint embedding space of text and image representations: conceptually similar texts and images are encoded close to each other, dissimilar pairings do not share similar positions.

The self-supervised pre-training of CLIP followed: Given N pairs of image and text, CLIP estimates the similarity for all the possible $N \times N$ pairings. With each text and image pair in the multimodal vector space, the models inside CLIP are jointly trained to maximize the similarity of each positive pairing and, at the same time, minimize the similarity of $N \times N - N$ antagonistic pairs (s. Fig. 2). The embedding similarities between pairs are represented by the cosine similarity metric, which is used to optimize the cross-entropy loss in order to build the most optimized versions of both the image and text encoder at the same time.

2.2.2. Performance

Zero-shot CLIP achieves similar performance or even outperforms conventional fully supervised class-wise models while preserving robustness through the ability to learn from a broader range of representations from natural language, especially on in-distribution or slightly out-of-distribution data. On the other hand, zero-shot CLIP weakly performs on datasets that are far out-of-distribution, such as satellite images (EuroSAT [39], NWPU-RESISC45 [40]) or tumors (PatchCamelyon [41]) [23]. When comparing CLIP and other large pre-trained n-shot models such as BiT-M [42] and SimCLRv2 [43], CLIP's authors depict that the zero-shot performance outperforms all other models on the metric of average accuracy score up to 4-shot linear classifiers trained on the same dataset [23]. The limitations are that scaling the model to learn from more data has steadily increased the performance, but computing power increases exponentially, which is currently barely economically reasonable.

2.2.3. Recent development

Meanwhile, much work exists on further development and adaptations of CLIP [44, 45, 46, 47, 48, 49, 50, 51]. The most notable works are SLIP [50], DeCLIP [44], ReCLIP [46], CoCa [47], and FILIP [49], aiming to improve efficiency in the training process. SLIP combines language supervision and image self-supervision to improve performance further. DeCLIP employs supervision across modalities as well as self-supervision within each modality, whereas ReCLIP at first learns pseudo labels and then applies cross-modality self-supervision. CoCA, on the other hand, skips cross-attention in some decoder layers to encode unimodal text representations and cross-attend the remaining layers with the image encoder. By using contrastive loss between unimodal image and text embeddings, along with captioning loss for multimodal decoder outputs, CoCa efficiently trains on a wider variety of data with minimal overhead. Improved fine-grained performance of CLIP is demonstrated in the works of FILIP [49], where instead of contrastive loss being calculated from global features of an entire image and text sequence, token-wise cross-modal interaction is modeled to take into account image patches and textual tokens more fine-grained.

Since this work focuses mainly on the training data, we will not evaluate all the individual strategies that aim to increase performance. Instead, we use the vanilla CLIP model and employ basic transfer learning methods that we can employ with limited hardware resources, which also demonstrate the effectiveness in the scope of lower-cost applications.

2.3. Fine-tuning and transfer learning

Depending on the application, CLIP has two different ways to adapt to a new distribution, i.e., new sets of data entirely outside the dataset on which CLIP was pre-trained. Fine-tuning and transfer learning are very similar ways to adapt CLIP, but they have different applications depending on the task at hand and different processes in modifying the architecture. Fine-tuning consists of training all layers or at least parts of the model. This process is usually more suitable for adapting to small sets of data that are closely related to the dataset CLIP was pre-trained on, such as everyday objects and general concepts. On the other hand, in tasks where the dataset is too specific, i.e., specialized knowledge, transfer-learning is better suited, as it freezes all the original layers of the pre-trained model and only adds or injects extra trainable layers or parameters. This way, the learned features of the zero-shot model are preserved and optimized for generalization to novel, previous out-of-distribution data. Usually, the fine-tuning process requires much more resources in terms of time, data, and computation as it modifies all the layers of the model compared to transfer learning. In the case of ILID, transfer learning proved to be a fitting solution, as the dataset is specialized specifically on industrial components, which are not presumably contained in CLIP's dataset used for pre-training.

Notable works in transfer-learning of CLIP are adapter-styled tuning, e.g., CLIPAdapter [52], and prompt learning, e.g., CoOp [53] and APEX [54]. CLIPAdapter (s. Fig. 5) adds dense down- and up-sampling layers on top of CLIP either to the image, text, or both encoders. Thereby, only the most prominent features

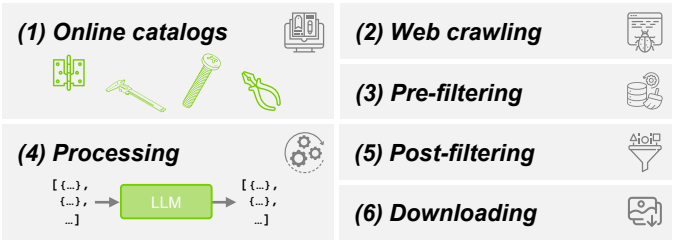

Fig. 4: Dataset generation pipeline resulting in the Industrial Language-Image Dataset (ILID).

are compressed into lower dimensions. From the latent space, the adapter then learns to reconstruct the essential ones. CoOp (s. Fig. 5) is the first to demonstrate continuous prompt learning as introduced by [55] for CLIP, which is learning continuous prompts by backpropagation for each label or one specific prompt template for all labels. Concretely, CoOp creates a set of learnable vectors, initialized by random values or given text embeddings, which, during training, the model adapts to. APEX is the most recent approach that also evaluates adding learnable tokens to the residual transformer blocks in the image encoder. Besides, APEX introduces a residual connection skipping the text adapter steered by an adaptive coefficient to perform better on a variety of out-of-distribution data.

3. Method

In this section, we first outline the generation of the ILID, including a thorough outline of the dataset acquisition, the criteria for data selection, web crawling to gather extensive sets of unlabeled data, and filtering (s. Sec. 3.1). Secondly, we elaborate on the decision for the model architecture and training procedure in Sec. 3.2.

3.1. Dataset generation pipeline

Following a typical data pipeline structure, including data selection, transforming, and pre-/post-filtering (s., e.g., [22]), we employed six steps (s. Fig. 4) to generate the Industrial Language-Image Dataset (ILID). Each of the steps results in a structured JSON document containing all the outputs. The next step always takes the respective document as input.

1. While searching for reasonably organized industrial-related data on the Internet, we found that **online catalogs** contain relevant language-image information. Typically, web stores have a page per product, sometimes imaging a set of product configurations, a precise, often standardized, title, description, information about the material, and further information about the product. These online stores are an adequate data source for the industrial domain. The first step was identifying a store set containing the necessary object-domain.
2. **Web crawling** data from online catalogs follows two basic steps: getting the sitemap from *robots.txt* and writing a crawler for the specific structure of the product pages. The top-level *robots.txt* file delineates the *Robots Exclusion Protocol*, which guides crawlers and other bots on which sections of the website they are permitted to access. Typically, this file also specifies the location of the sitemap, an XML-formatted document designed to provide crawlers with information about all pages on a website. Sitemaps can be hierarchically ordered; in the case of online catalogs, typically, there is one specific sitemap containing all products and their respective locations. We use Scrapy[3] as a Python-based web crawler that takes a sitemap as input and crawls through all the specified locations. Creating a specific spider for a web catalog requires manual intervention since one has to define which images and text blocks to yield. Besides a central label tag for each entry, we save an unstructured list-typed data object, which can contain all other available information about the product, like materials, finish, colors, etc. Using the sitemap as the initial crawling entry point is a common step in every online search engine.
3. In the **pre-filtering** step, we filter for duplicate entries, remove special characters, as well as diminish entries that do not have sufficient information. Besides, we filter the data for a set of trade names and remove these from all product information. Often, industrial product names include the manufacturer, which we do not want to use further or bias the data within the following information extraction.
4. In the central **processing** step, we use a small local-deployable LLM to extract our five target information from the unstructured data. We define these as (1) a long label describing the product, (2) a short label that is shorter than the long label, (3) a description of the product, (4) the material, (5) the finish or color of the product (s. also Fig. 2). In our study, we used Llama3-8B [19] in the fine-tuned instruct version (s. Appendix A for the respective prompt). We ask the LLM not to output any numbers or sizes; additionally, we remove them from the initial data since, on the one hand, we do not expect that a 2D image task can identify or recognize any dimensional quantities given different camera positions and varying intrinsics, on the other hand, we do not want to bias the dataset with it. After prompting for the desired information, we extract these from the response and save them for further processing. We discard the item from the dataset if the prompt does not return sufficient output.
5. In the **post-filtering** step, we again filter for any unwanted characters and do some further cleaning, like lowering words.
6. In the final **downloading** step, all images are downloaded, post-processed, and resized while also assembling the final JSON specifying the dataset's text and metadata.

With the given steps, we are able to extract a product's image and a structured set of five pieces of information. Besides, we observed that even a small model such as Llama3-8B in its instruct fine-tuned version is mostly able to extract the demanded

[3] Scrapy: A Fast and Powerful Scraping and Web Crawling Framework

information from the bunch of unstructured text. We show an excerpt of the dataset in Appendix B.

3.2. Transfer learning

3.2.1. Model architecture

As we already outlined in Sec. 2.3, we adopt a simple yet effective strategy for transfer learning from CLIP's in-distribution to our ILID dataset. Within CLIP's dual-encoder setup, we must utilize a strategy for the image and text stream. Fig. 5 depicts the used model architectures.

While we estimate that the images we want to learn but also infer from show similar characteristics as CLIP's in-distribution data compared to other fully out-of-distribution image data as in the case of, e.g., PatchCamelyon [41] (s. Sec. 2.2.2), we employ on the image stream only a simple trainable adapter as proposed by [52]. We tuned the mixing coefficient manually; we observed that a low α can vastly result in overfitting, while a high value does not necessarily increase the performance significantly during cross-validation. That is why we chose a balanced value of $\alpha = 0.5$. The adapters reduce the feature by 4 as proposed in the original paper [52]. We omitted testing prompt tuning on the image stream as introduced by APEX [54] since we estimate a relatively low distribution shift from the CLIP dataset to ILID regarding the images.

In contrast, prompt engineering is a crucial task for learning, as well as inference with textual, promptable models. In a preliminary study, we have already observed that vanilla CLIP performs differently, given different prompt templates like *"a photo of* {}." compared to *"a photo of* {}*, an industrial product."* The difference from the minor change results from the prompts CLIP was pre-trained with, which follow similar characteristics. Having not to discretely prompt-tune manually motivated us to utilize CoOp [53] as a continuous prompt learning method. Besides, we also evaluate in the experimental study (s. Sec. 4) the performance of adding an additional adapter to the text stream.

3.2.2. Training

During the pre-training of CLIP, a very large minibatch size of 32,768 was used, which took for the largest Vision Transformer (ViT) configuration (428M parameters) a total of 12 days on 256 V100 GPUs [23]. Compared to the pre-training, during transfer learning with CoOp, we have a total of $c_n \times 512$ trainable weights (c_n = number of context vectors), which can be managed on a single consumer GPU in a reasonable time. However, the batch size has to be chosen wisely from the memory point of view, as well as by looking at the dataset labels.

Given 32k samples per minibatch out of a total of 400M, the chance, utilizing random sampling, that non-contrastive samples are included in one minibatch is negligibly slight. In contrast, fine-tuning or transfer learning approaches typically contrast all possible class labels against a set of images [52, 54, 53, 56] during the benchmark studies on datasets like ImageNet [57], which is why non-contrasting samples are not possible as long as the classes are conceptually far away from each other. The assembled ILID dataset does not have any class concept, meaning that we, as a priori, do not know how two samples and their labels are semantically close to each other. Contrasting a set of images against all possible labels is infeasible memory-wise; that is why we can not follow this training method and only contrast the images and their labels inside one batch as done during pre-training. This change led us to employ a different optimizer from the one used in the original CoOp implementation since Stochastic Gradient Descent (SGD) would not converge given the smaller batch size. We changed from vanilla SGD to Adadelta [58], an SGD optimizer that adapts learning rates over time using only first-order information.

4. Experiments

In this section, we present a series of studies utilizing ILID, designed to evaluate the effectiveness of the dataset and transfer learning approach for different tasks. We begin with the dataset properties (s. Sec 4), describe the experimental setup (s. Sec. 4.2), and present quantitative results on cross-validation (s. Sec. 4.3) as well as training and inference on a different label type (s. Sec. 4.4). Further, we present the results of a downstream task on segmentation (s. Sec. 4.5).

4.1. Dataset

For the presented ILID, as of now, we crawled five different online shops, resulting in 12,537 valid samples, including a diverse range of products ranging from standard elements small in size like hinges, linear motion elements, bearings, or clamps to larger ones, like scissor lifts, pallet trucks, etc. (an excerpt is depicted in Appendix B).

Fig. 6 depicts the top-40 word occurrences in label *label_short*, showing that typical concepts of industrial standard parts like *clamp*, *lever*, *handle*, *knob*, *hinge*, or *swivel* are pronounced represented but also material types (*steel*, *aluminum* / *aluminium*) and properties (*stainless*) as well.

Tab. 1 lists the number of unique labels per label category and hints at the dataset's diversity. Obviously, with increasing words (on average: *label_short* < *label_long* < *description*), the number of label-wise unique labels increases. So, nearly every sample has a unique *description*, but only two labels, on average, share the same *label_short*. Since we do not account for minor preposition words like *a*/*an*/*the* in the labels, the labels are slightly more equal on the semantically level. However, we estimate a good diversity in the dataset, and since we do not account for preposition words in the counting, at least three to four samples are included per semantical similar class, which should suffice for a tuned CLIP to outperform fully supervised models (s. Sec. 2.2.2). We use the presented version of ILID in the following experiments.

Table 1: Number of unique labels per label category.

label_short	*label_long*	*material*	*material_finish*	*description*
6785	8476	2899	3375	11452

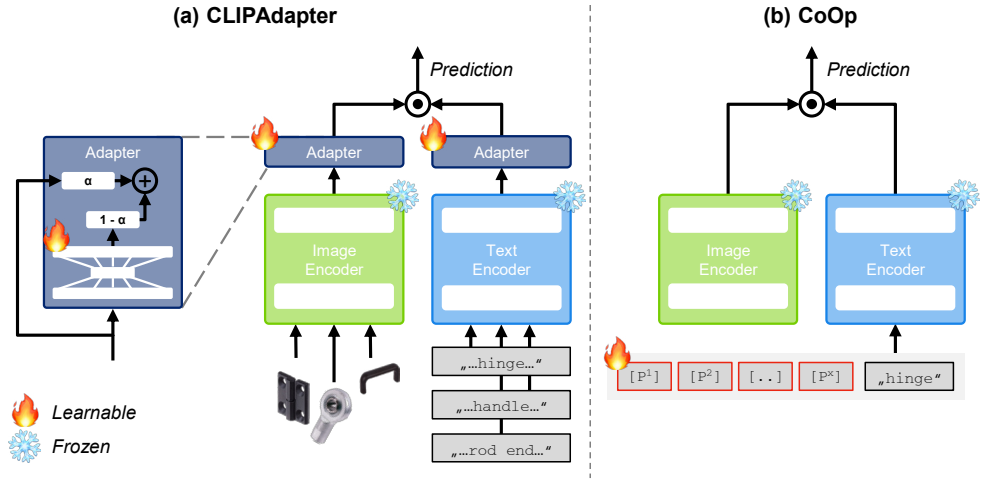

Fig. 5: The architectures used in this work: (a) CLIPAdapter [52] and (b) CoOp [53].

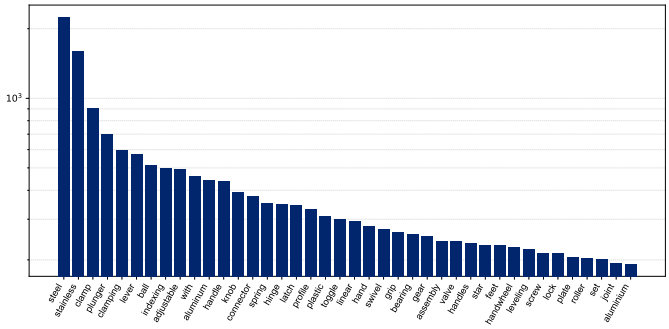

Fig. 6: Top-40 word occurrences in label *label_short*.

4.2. Setup

We build upon the code base of Dassl [59, 60] and trained on a single 4090 GPU. We chose random sampling, an image input size of 224×224, and CLIP's pre-trained ViT-B/16 as the image encoder instead of the ResNet version, as ViTs have much less image-specific inductive bias than CNNs. We initialized CoOp with a context length of $c_n = 10$ if not otherwise stated. We trained with a batch size of 64 while testing with only 32 samples. This increases the accuracy during validation/testing compared to training accuracy, but we feel this is more realistic in the case of real-world applications contrasting only 32 different conceptual object classes in one application. We applied common data augmentation techniques of randomly resizing and cropping as well as flipping edges during training. Besides, we normalized ILID's image data. We use Adadelta [58] with a learning rate of 0.15 and a cosine learning rate scheduler with a weight decay of $1e$-3. Besides, we used 3 warm-up epochs with a constant learning rate of $1e$-2 to prevent rapid parameter changes in the initial training stages, which can lead to early overfitting.

4.3. Quantitative results

Since we do not have a different real-world language-image dataset at hand, we used 6-fold cross-validation during the evaluation of the different model architectures. Fig. 7 depicts the validation results of training on the *label_short* and *label_long* with CoOp, CoOp + image adapter (CoOpIA) with $\alpha_i = 0.5$, CoOp + image $\alpha_i = 0.5$ and text adapter $\alpha_t = 0.2$ (CoOpIATA), image and text adapter only (CLIPAdapter) (same αs), and the zero-shot CLIP performance. All accuracies are derived from the top-1 predictions. Additionally, we listed the top-3 accuracies for zero-shot CLIP and CoOpIATA (dashed lines).

A first observation (▷ Obs. 1) is that all transfer learning approaches effectively outperform CLIP's zero-shot capabilities, even the top-3 accuracies after training for ≈ 20 epochs, highlighting that the ILID is out-of-distribution. Even training on the less information-rich *label_short* outperforms CLIP's zero-shot capabilities.

CLIP highly depends on the chosen prompt template (▷ Obs. 2). If we look at training on the *label_long*, the zero-shot (prompt template *"a photo of an industrial product {label_long}"*) accuracy is lower than all other trained methods initialized with random weights in the adapters and CoOp with *"X X X X a photo of an industrial product {label_long}"*, which tokens will be optimized except for the label. CLIP performs poorly if given a prompt that deviates much from the ones during pre-training. That especially accounts for combining the prompt template with multi-word product descriptions.

As expected, the more trainable weights we add, the better the model adapts to the data, while the overall domain generalization to the in-distribution data achieves in the case of *label_short* and *label_long* an accuracy of maximum 79.93% and 84.31%, respectively, an image adapter is crucial to effective transfer learning in this case (▷ Obs. 3).

Moreover, the top-performing model also depends on prompt learning, but adapter-styled tuning performs better than only prompt learning on the text stream (▷ Obs. 4). However,

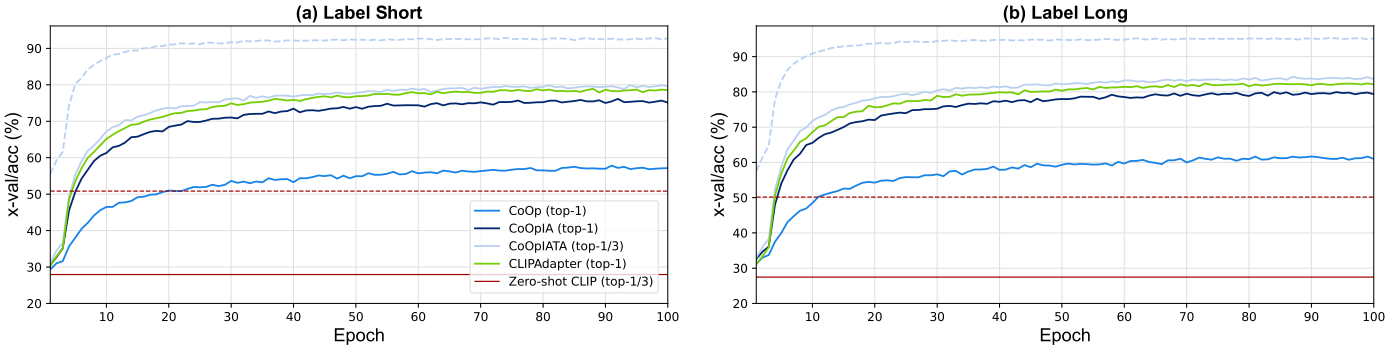

Fig. 7: Results of 6-fold cross-validation during transfer learning using different approaches on the ILID.

adapting to images will reduce the model's performance on slight to out-of-distribution data. That means inference on images that vastly differ from catalog-style ones will definitely have lower performance than the in-distribution images. However, the trained model will still outperform zero-shot CLIP, as we will see in the following sections.

To gain an understanding of how transfer learning affects the embeddings further, we derived the image and text embeddings after training on the full ILID given the label *label_short* for 100 epochs. Fig. 8 visualizes the high-dimensional embeddings of the same 100 samples. With each transfer learning method, adding more trainable weights, the text and image embeddings more jointly share the same embedding space. Further, multiple text embeddings get so close that they are hard to distinguish in the t-SNE diagram at all, which we estimate follows that the transfer learning approaches learn to group semantically close concepts, while in the zero-shot case, these are still more widely clustered. Moreover, image and text embeddings are much more pronounced after transfer learning than in the zero-shot case.

4.4. Prompting for materials

Besides training and testing on the *label_short* and *label_long*, we additionally trained CoOpIATA for 100 epochs on the *material* label with the initial prompt *"X X X X a photo of an industrial product with material {}"*. We then evaluated the zero-shot and CoOpIATA performance on the images depicted in Fig. 9 while choosing for the zero-shot test a prompt template similar to for training CoOpIATA. The results are listed in Tab. 2.

Surprisingly, CLIP's zero-shot performance shows 2 out of 5 true positives, while the transfer learning result in 5 out of 5. Further, looking at the scores, we see that our proposed transfer learning method produces much higher confidence in every case, which follows that the different concepts of materials are not in-distribution in the zero-shot case. Interestingly, a prompt including {*aluminum*} results in lower scores than using the word {*aluminium*}, which points out that the subtleties or discrepancies of the language used in an industrial context are not mapped after the transfer learning nor in the zero-shot case. That is why we added both words in the prompts. Further,

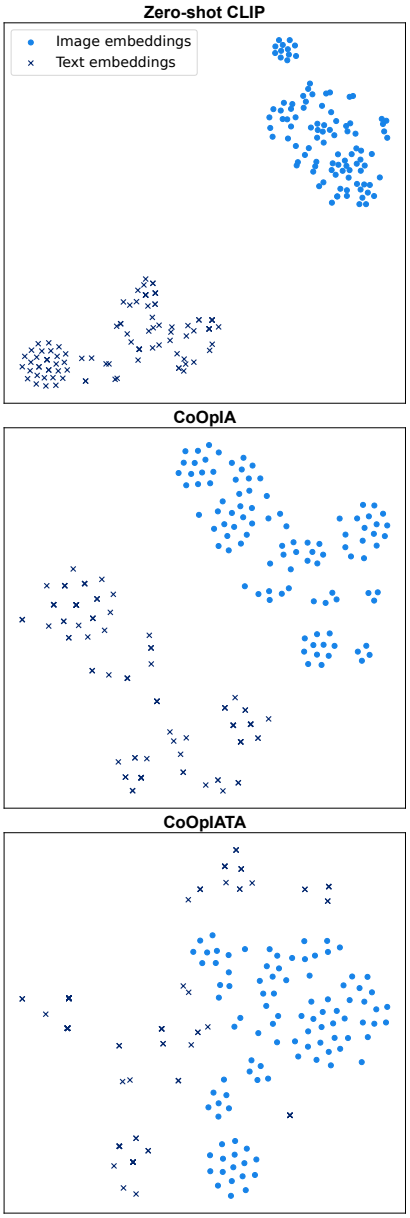

Fig. 8: t-SNE diagrams from the same randomly selected 100 samples (CoOpIA and CoOpIATA were trained for 100 epochs on the full ILID given the label *label_short*).

after transfer learning, judging based on the scores, there is still slight confusion between the material concepts of aluminum and polyamide as well as polyamide and brass. We estimate that the transfer learning introduced a specific object-material-awareness but is still heavily influenced by other image characteristics, like, in our case, the yellow taint.

The given task might not serve a real-world industrial vision use case at this stage, but it shows how ILID can serve different tasks at hand by combining images with different (broad) language information during training. These results again underline a natural language supervised VFM's rich multimodal capabilities.

Table 2: Scores on predicting the object's material properties in the images from Fig. 9 (bold indicates the highest scores; underlined values correspond to the ground truth).

	(a)	(b)	(c)	(d)	(e)
Zero-shot CLIP					
"steel"	0.024	0.113	**0.330**	0.168	0.059
"polyamide"	0.149	0.196	0.062	0.107	0.208
"thermoplastic"	0.245	0.141	0.050	0.034	0.097
"aluminum or aluminium"	0.043	0.143	0.166	0.238	0.094
"anodized aluminum or aluminium"	0.030	0.143	0.070	0.064	0.023
"plastic"	**0.352**	**0.244**	0.099	0.107	**0.280**
"brass"	0.156	0.020	0.223	**0.282**	0.240
CoOpIATA trained on the *material* label					
"steel"	0.007	0.033	**0.950**	**0.829**	0.137
"polyamide"	0.135	0.368	0.004	0.008	**0.361**
"thermoplastic"	0.010	0.004	0.002	0.001	0.160
"aluminum or aluminium"	0.009	0.085	0.020	0.011	0.001
"anodized aluminum or aluminium"	0.007	**0.374**	0.003	0.007	0.001
"plastic"	**0.694**	0.135	0.008	0.041	0.077
"brass"	0.139	0.000	0.012	0.104	0.264

4.5. Language-guided segmentation

A typical downstream task is a language-guided segmentation utilizing the Segment Anything Model (SAM) [61]. SAM is a class-agnostic point promptable image segmentation model that outputs hierarchical masks and predicted Intersection over Unions (IoU). Without the need for manual intervention, an automatic mask generation pipeline can sample a point grid and subsequently use Non-Maximum Suppression (NMS) to diminish through merging a large set of masks to form more precise proposals. In the simplest form, language-guided image segmentation based on SAM and CLIP can be employed by applying CLIP onto all generated masks, which we cut out with a particular delineation factor. CLIP's softmaxed logits can then be thresholded to get the final per-mask class-wise predictions. We only contrasted the object to prompt against an empty class label. Contrasting only two prompts is challenging since the model's overconfidence in one of them is the most pronounced. We chose to do so to avoid any bias by introducing hard negative prompts.

For the language-guided segmentation, we used a CoOpIATA model trained on the complete ILID dataset given the *label_long* for 40 epochs. For completeness, it should be mentioned that we did not compare it against the other approaches, e.g., CLIPAdapter.

Fig. 10 depicts the segmentation results in a challenging scene composed of multiple collets stacked on a trolley. The zero-shot results do have many true positives, but overall, we are not able to observe any further prediction patterns. In contrast, the transfer learning approach can effectively distinguish between a mask containing a collet and a mask that does not. With only 17 word occurrences of *"collet"* in ILID's *label_long* labels, the resulting model's confidence compared to zero-shot CLIP effectively demonstrates the proposed method. Additionally, the images relating to the labels do not contain collets of the same shapes and sizes, which emphasizes CLIP's learned rich representations. We discuss two further examples in Appendix C.

5. Conclusion and Outlook

Using VFMs as a building block in an industrial vision application is a promising and transforming technique, improving systems' accuracy, speed, and reliability, e.g., involved in inspection, robotic control, parts identification, and process control, leading to enhanced operational efficiencies and product quality. As we outlined in Sec. 2.1, up to this date, literature only has a limited number of use case ideas regarding using VFMs in industrial applications, which we want to motivate further.

This work strived to make a step towards enabling employing VFM in industrial machine vision applications by introducing the Industrial Language-Image Dataset (ILID) to bring industrial context into CLIP and evaluating effective self-supervised transfer learning from the dataset. We demonstrated this by evaluating downstream tasks from prompting for material properties to language-guided segmentation. With only a limited dataset size of $\approx 12k$ samples, the results show promising opportunities in machine vision applications when increasing the dataset size or further restricting it to more specific domains.

One can argue that the bigger digital giants like OpenAI or Meta can also incorporate industrial data during the training of their models; however, the overall proposed method from dataset curation to fine-tuning CLIP also suits, e.g., companies with intellectual property constraints or limitations in available computing resources in employing VFMs. Nevertheless, fine-tuning expert models for specific tasks is a common step in creating an AI application, which we, e.g., showcased, given the transfer learning from material properties. Future work must also elaborate on training with ILID's other labels, like *description*, to further discuss opportunities for other applications.

The current limitations we observed on the text stream are especially the limited learned language subtleties and discrepancies as they occur in industrial contexts. The confusion between the same concept put differently termed in American

(a) (b) (c) (d) (e)

Fig. 9: Five different real-world images used for prompting material properties.

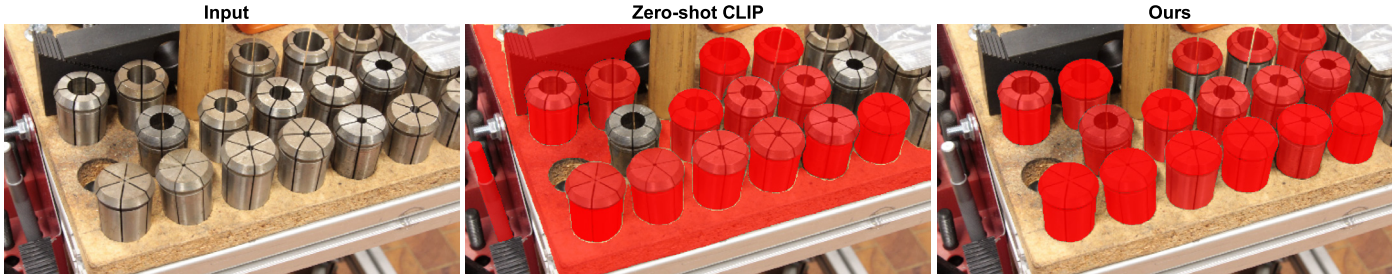

Fig. 10: Language-guided segmentation results given prompt *"collet"* compared to zero-shot CLIP under the same settings (segmentation properties and thresholds).

(aluminium) and British (aluminum) English shows that there is a need for pre-training of the text encoder with broader natural language, e.g., even with extended context, which would enable not only training on shorter image labels. Further, on the image stream, we observed that the model generalizes well to a variety of an object's different views but does less perform well when contrasting between finer-grained different object types. Here, a custom expert model is probably more suited than transfer learning from a dataset that includes many different object concepts. The most limiting characteristic is including or inferencing with dimensional quantities, which can hardly be solved when training on images captured with different cameras and their individual intrinsics.

With this work, we hope to encourage the industrial community to employ and work on using VFM in the industrial domain more and more. Therefore, we publicly provide ILID and the code used during training. In the future, we plan to continue increasing the dataset size by incorporating more web catalogs.

Acknowledgments

This work is part of the research project *Intelligent Digital Cabin Twin (InDiCaT)* under the grant number 20D1902C, supported by the *Federal Ministry for Economic Affairs and Climate Action (BMWK)* as part of the *Federal Aeronautical Research Programme LuFo VI-1*.

Supported by:

Federal Ministry for Economic Affairs and Climate Action

on the basis of a decision by the German Bundestag

We thank MÄDLER GmbH for granting us the rights to use some of their product images (included in Fig. 2, 3, 5, B.11, and B.12) in this publication.

CRediT author statement

K. Moenck: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Resources, Data Curation, Writing – original draft, Writing - review & editing, Visualization, Supervision, Project administration; D.T. Thieu: Conceptualization, Methodology, Software, Formal analysis, Investigation, Data Curation, Writing – original draft; J. Koch: Writing - review & editing; T. Schüppstuhl: Supervision, Project administration, Funding acquisition, Writing - review & editing.

References

[1] D. Schoepflin, D. Holst, M. Gomse, T. Schüppstuhl, Synthetic training data generation for visual object identification on load carriers, Procedia CIRP 104 (2021) 1257–1262. doi:10.1016/j.procir.2021.11.211.

[2] D. Schoepflin, K. Iyer, M. Gomse, T. Schüppstuhl, Towards synthetic ai training data for image classification in intralogistic settings, in: Schüppstuhl (Ed.) 2022 – Annals of Scientific Society, pp. 325–336. doi:10.1007/978-3-030-74032-0_27.

[3] D. Holst, D. Schoepflin, T. Schüppstuhl, Generation of synthetic ai training data for robotic grasp-candidate identification and evaluation in intralogistics bin-picking scenarios, in: K.-Y. Kim (Ed.), Flexible Automation and Intelligent Manufacturing, Lecture Notes in Mechanical Engineering Ser, Springer International Publishing AG, Cham, 2022, pp. 284–292. doi:10.1007/978-3-031-18326-3_28.

[4] O. Schmedemann, M. Baaß, D. Schoepflin, T. Schüppstuhl, Procedural synthetic training data generation for ai-based defect detection in industrial surface inspection, Procedia CIRP 107 (2022) 1101–1106. doi:10.1016/j.procir.2022.05.115.

[5] B. Drost, M. Ulrich, P. Bergmann, P. Hartinger, C. Steger, Introducing mvtec itodd — a dataset for 3d object recognition in industry, in: 2017 IEEE International Conference on Computer Vision Workshops (ICCVW), IEEE, 2017, pp. 2200–2208. doi:10.1109/ICCVW.2017.257.

[6] P. Bergmann, M. Fauser, D. Sattlegger, C. Steger, Mvtec ad — a comprehensive real-world dataset for unsupervised anomaly detection, in:

2019 IEEECVF Conference on Computer Vision and Pattern Recognition, IEEE, Piscataway, NJ, 2019, pp. 9584–9592. doi:10.1109/CVPR.2019.00982.

[7] P. Bergmann, K. Batzner, M. Fauser, D. Sattlegger, C. Steger, The mvtec anomaly detection dataset: A comprehensive real-world dataset for unsupervised anomaly detection, International Journal of Computer Vision 129 (4) (2021) 1038–1059. doi:10.1007/s11263-020-01400-4.

[8] P. Bergmann, K. Batzner, M. Fauser, D. Sattlegger, C. Steger, Beyond dents and scratches: Logical constraints in unsupervised anomaly detection and localization, International Journal of Computer Vision 130 (4) (2022) 947–969. doi:10.1007/s11263-022-01578-9.

[9] H. Bai, S. Mou, T. Likhomanenko, R. G. Cinbis, O. Tuzel, P. Huang, J. Shan, J. Shi, M. Cao, Vision datasets: A benchmark for vision-based industrial inspection. doi:10.48550/arXiv.2306.07890.

[10] L. Büsch, J. Koch, D. Schoepflin, M. Schulze, T. Schüppstuhl, Towards recognition of human actions in collaborative tasks with robots: Extending action recognition with tool recognition methods, Sensors (Basel, Switzerland) 23 (12) (2023). doi:10.3390/s23125718.

[11] J. Zhang, J. Huang, S. Jin, S. Lu, Vision-language models for vision tasks: A survey, IEEE transactions on pattern analysis and machine intelligence PP (2024). doi:10.1109/TPAMI.2024.3369699.

[12] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill, et al., On the opportunities and risks of foundation models. doi:10.48550/arXiv.2108.07258.

[13] J. Devlin, M.-w. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding. doi:10.48550/arXiv.1810.04805.

[14] P. Budzianowski, I. Vulić, Hello, it's gpt-2 – how can i help you? towards the use of pretrained language models for task-oriented dialogue systems. doi:10.48550/arXiv.1907.05774.

[15] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al., Language models are few-shot learners. doi:10.48550/arXiv.2005.14165.

[16] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, et al., Gpt-4 technical report. doi:10.48550/arXiv.2303.08774.

[17] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, et al., Llama: Open and efficient foundation language models. doi:10.48550/arXiv.2302.13971.

[18] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, et al., Llama 2: Open foundation and fine-tuned chat models. doi:10.48550/arXiv.2307.09288.

[19] M. AI, Introducing meta llama 3: The most capable openly available llm to date (26.05.2024).
URL https://ai.meta.com/blog/meta-llama-3/

[20] M. Awais, M. Naseer, S. Khan, R. M. Anwer, H. Cholakkal, M. Shah, M.-H. Yang, F. S. Khan, Foundational models defining a new era in vision: A survey and outlook. doi:10.48550/arXiv.2307.13721.

[21] C. Schuhmann, R. Beaumont, R. Vencu, C. Gordon, R. Wightman, M. Cherti, T. Coombes, A. Katta, C. Mullis, M. Wortsman, et al., Laion-5b: An open large-scale dataset for training next generation image-text models. doi:10.48550/arXiv.2210.08402.

[22] S. Changpinyo, P. Sharma, N. Ding, R. Soricut, Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. doi:10.48550/arXiv.2102.08981.

[23] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al., Learning transferable visual models from natural language supervision. doi:10.48550/arXiv.2103.00020.

[24] C. Jia, Y. Yang, Y. Xia, Y.-T. Chen, Z. Parekh, H. Pham, Q. Le V, Y. Sung, Z. Li, T. Duerig, Scaling up visual and vision-language representation learning with noisy text supervision, International Conference on Machine Learningdoi:10.48550/arXiv.2102.05918.

[25] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition. doi:10.48550/arXiv.1512.03385.

[26] A. Hornberg, Handbook of machine and computer vision: The guide for developers and users, second, revised and updated edition Edition, Wiley-VCH, Weinheim, 2017. doi:10.1002/9783527413409.

[27] A. Naumann, F. Hertlein, L. Dörr, S. Thoma, K. Furmans, Literature review: Computer vision applications in transportation logistics and warehousing. doi:10.48550/arXiv.2304.06009.

[28] K. Moenck, A. Wendt, P. Prünte, J. Koch, A. Sahrhage, J. Gierecker, O. Schmedemann, F. Kähler, D. Holst, M. Gomse, et al., Industrial segment anything – a case study in aircraft manufacturing, intralogistics, maintenance, repair, and overhaul. doi:10.48550/arXiv.2307.12674.

[29] J. Wang, Y. Tian, Y. Wang, J. Yang, X. Wang, S. Wang, O. Kwan, A framework and operational procedures for metaverses-based industrial foundation models, IEEE Transactions on Systems, Man, and Cybernetics: Systems 53 (4) (2023) 2037–2046. doi:10.1109/TSMC.2022.3226755.

[30] H. Zhang, S. S. Dereck, Z. Wang, X. Lv, K. Xu, L. Wu, Y. Jia, J. Wu, Z. Long, W. Liang, et al., Large scale foundation models for intelligent manufacturing applications: A survey. doi:10.48550/arXiv.2312.06718.

[31] L. Makatura, M. Foshey, B. Wang, F. HähnLein, P. Ma, B. Deng, M. Tjandrasuwita, A. Spielberg, C. E. Owens, P. Y. Chen, et al., How can large language models help humans in design and manufacturing? doi:10.48550/arXiv.2307.14377.

[32] C. Picard, K. M. Edwards, A. C. Doris, B. Man, G. Giannone, M. F. Alam, F. Ahmed, From concept to manufacturing: Evaluating vision-language models for engineering design. doi:10.48550/arXiv.2311.12668.

[33] Y. Mori, H. Takahashi, R. Oka, Image-to-word transformation based on dividing and vector quantizing images with words, in: First international workshop on multimedia intelligent storage and retrieval management, Vol. 2, 1999.

[34] A. Quattoni, M. Collins, T. Darrell, Learning visual representations using images with captions, in: IEEE Conference on Computer Vision and Pattern Recognition, 2007, IEEE Computer Society, Los Alamitos, Calif., 2007, pp. 1–8. doi:10.1109/CVPR.2007.383173.

[35] K. Desai, J. Johnson, Virtex: Learning visual representations from textual annotations. doi:10.48550/arXiv.2006.06666.

[36] M. B. Sariyildiz, J. Perez, D. Larlus, Learning visual representations with caption annotations. doi:10.48550/arXiv.2008.01392.

[37] Y. Zhang, H. Jiang, Y. Miura, C. D. Manning, C. P. Langlotz, Contrastive learning of medical visual representations from paired images and text. doi:10.48550/arXiv.2010.00747.

[38] S. Peng, K. Genova, C. Jiang, A. Tagliasacchi, M. Pollefeys, T. Funkhouser, Openscene: 3d scene understanding with open vocabularies. doi:10.48550/arXiv.2211.15654.

[39] P. Helber, B. Bischke, A. Dengel, D. Borth, Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. doi:10.48550/arXiv.1709.00029.

[40] G. Cheng, J. Han, X. Lu, Remote sensing image scene classification: Benchmark and state of the art, Proceedings of the IEEE 105 (10) (2017) 1865–1883. doi:10.1109/JPROC.2017.2675998.

[41] B. S. Veeling, J. Linmans, J. Winkens, T. Cohen, M. Welling, Rotation equivariant cnns for digital pathology. doi:10.48550/arXiv.1806.03962.

[42] A. Kolesnikov, L. Beyer, X. Zhai, J. Puigcerver, J. Yung, S. Gelly, N. Houlsby, Big transfer (bit): General visual representation learning. doi:10.48550/arXiv.1912.11370.

[43] T. Chen, S. Kornblith, K. Swersky, M. Norouzi, G. Hinton, Big self-supervised models are strong semi-supervised learners. doi:10.48550/arXiv.2006.10029.

[44] Y. Li, F. Liang, L. Zhao, Y. Cui, W. Ouyang, J. Shao, F. Yu, J. Yan, Supervision exists everywhere: A data efficient contrastive language-image pre-training paradigm. doi:10.48550/arXiv.2110.05208.

[45] S. Goel, H. Bansal, S. Bhatia, R. A. Rossi, V. Vinay, A. Grover, Cyclip: Cyclic contrastive language-image pretraining. doi:10.48550/arXiv.2205.14459.

[46] X. Hu, K. Zhang, L. Xia, A. Chen, J. Luo, Y. Sun, K. Wang, N. Qiao, X. Zeng, M. Sun, et al., Reclip: Refine contrastive language image pre-training with source free domain adaptation. doi:10.48550/arXiv.2308.03793.

[47] J. Yu, Z. Wang, V. Vasudevan, L. Yeung, M. Seyedhosseini, Y. Wu, Coca: Contrastive captioners are image-text foundation models. doi:10.48550/

arXiv.2205.01917.
[48] Y. Rao, W. Zhao, G. Chen, Y. Tang, Z. Zhu, G. Huang, J. Zhou, J. Lu, Denseclip: Language-guided dense prediction with context-aware prompting. doi:10.48550/arXiv.2112.01518.
[49] L. Yao, R. Huang, L. Hou, G. Lu, M. Niu, H. Xu, X. Liang, Z. Li, X. Jiang, C. Xu, Filip: Fine-grained interactive language-image pre-training. doi:10.48550/arXiv.2111.07783.
[50] N. Mu, A. Kirillov, D. Wagner, S. Xie, Slip: Self-supervision meets language-image pre-training. doi:10.48550/arXiv.2112.12750.
[51] Q. Sun, Y. Fang, L. Wu, X. Wang, Y. Cao, Eva-clip: Improved training techniques for clip at scale. doi:10.48550/arXiv.2303.15389.
[52] P. Gao, S. Geng, R. Zhang, T. Ma, R. Fang, Y. Zhang, H. Li, Y. Qiao, Clip-adapter: Better vision-language models with feature adapters. doi:10.48550/arXiv.2110.04544.
[53] K. Zhou, J. Yang, C. C. Loy, Z. Liu, Learning to prompt for vision-language models. doi:10.1007/s11263-022-01653-1.
[54] Y. Yang, J. Ko, S.-Y. Yun, Improving adaptability and generalizability of efficient transfer learning for vision-language models. doi:10.48550/arXiv.2311.15569v1.
[55] B. Lester, R. Al-Rfou, N. Constant, The power of scale for parameter-efficient prompt tuning. doi:10.48550/arXiv.2104.08691.
[56] K. Zhou, J. Yang, C. C. Loy, Z. Liu, Conditional prompt learning for vision-language models. doi:10.48550/arXiv.2203.05557.
[57] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: IEEE Conference on Computer Vision and Pattern Recognition, 2009, IEEE, Piscataway, NJ, 2009, pp. 248–255. doi:10.1109/CVPR.2009.5206848.
[58] M. D. Zeiler, Adadelta: An adaptive learning rate method. doi:10.48550/arXiv.1212.5701.
[59] K. Zhou, Y. Yang, Y. Qiao, T. Xiang, Domain adaptive ensemble learning. doi:10.1109/TIP.2021.3112012.
[60] K. Zhou, Z. Liu, Y. Qiao, T. Xiang, C. C. Loy, Domain generalization: A survey. doi:10.1109/TPAMI.2022.3195549.
[61] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, et al., Segment anything. doi:10.48550/arXiv.2304.02643.
[62] H. Wang, P. K. A. Vasu, F. Faghri, R. Vemulapalli, M. Farajtabar, S. Mehta, M. Rastegari, O. Tuzel, H. Pouransari, Sam-clip: Merging vision foundation models towards semantic and spatial understanding. doi:10.48550/arXiv.2310.15308.

Appendix A. Llama-3 prompt

We followed basic prompt assembly as described for Llama-2 [18] because up to the date of this publication, there has still been an in-depth explanation of Llama-3 missing. The Llama-2 chat version was trained with a variety of system prompts following patterns like *"You are a helpful, respectful and honest assistant. Always answer as helpfully as possible, while being safe."*, we also included a similar one but tried to include the targeted domain. The brackets {{}} point out where we insert the data.

Listing 1: System prompt used in the ILID generation pipeline's text transformation step.

```
You are a helpful assistant for a company that sells
industrial products.\n
Do not ask for further details or state additional
questions.\n
Do not add additional information or details that are not
given by the user.\n
```

Listing 2: User prompt used in the ILID generation pipeline's text transformation step.

```
Summarize 'Label: {{}} Text: {{}}'\n
returning the following information: \n
(1) a long label or name of the product without ids,
numbers, codes, or sizes
(2) a short label or name of the product with a maximum of
4 words and shorter than the long label
(3) description of the product with a maximum of 20 words
without ids, numbers, codes, or sizes
(4) material with a maximum of 5 words
(5) material finish/color with a maximum of 5 words
```

Appendix B. Excerpt from the dataset

Fig. B.11 and Fig. B.12 depict each two samples from the ILID given the keywords *"hinge"* and *"locking assembly"*. Based on the language label, we can observe that the LLM performs differently in extracting the relevant information. As an example, *material* and *material_finish* confusion occurs when the product page states more than one exact product configuration.

Appendix C. Additional language-guided segmentation results

Fig. C.13 and C.14 show supplementary results on language-guided image segmentation. In Fig. C.13, we prompted for *"socket"*, whereas zero-shot CLIP does not predict any mask as positive, while our approach segments all sockets.

In Fig. C.14, the results of our most challenging scene are depicted, in which we prompt for *"bracket for construction profile"*. The brackets are imaged far differently than the ones from catalog images, and sometimes they are barely visible.


```
{ "id": "...", "image": "...",
  "label_short": "clevis mounting hinge",
  "label_long": "bracket hinge for clevis
                 mounting",
  "description": "Rigid hinge for clevis
                  mounting applications",
  "material": "rigid metal",
  "material_finish": "black oxide finish" }
```



```
{ "id": "...", "image": "...",
  "label_short": "adjustable hinge mechanism",
  "label_long": "hinge with adjustable friction
                 mechanism",
  "description": "A hinge with adjustable
                  friction for smooth motion
                  and secure locking",
  "material": "zinc die casting",
  "material_finish": "silver plastic coating" }
```


Fig. B.11: Two samples from the ILID given the keyword *"hinge"*.


```
{ "id": "...", "image": "...",
  "label_short": "locking qpq assembly",
  "label_long": "locking assembly with qpq
                 coating",
  "description": "High corrosion resistance and
                  improved fatigue strength for
                  food safe applications",
  "material": "steel",
  "material_finish": "qpq-coated" }
```



```
{ "id": "...", "image": "...",
  "label_short": "stainless steel locking bar",
  "label_long": "locking assembly stainless steel
                 bar",
  "description": "A locking assembly designed
                  for industrial use made from
                  high-quality stainless steel
                  for durability and resistance",
  "material": "stainless steel aisi",
  "material_finish": "stainless steel finish" }
```


Fig. B.12: Two samples from the ILID given the keyword *"locking assembly"*.

At first sight, the results do not show good performance, especially since we have a few non-detected brackets and a few false positive predictions. We explain the false positive on the top with the cropping strategy, while we have no explanation for the false predictions on the lower right. The false positives can result from the axis-preserving cropping strategy of the used method, in which a cropped segment includes parts of the surroundings. A lot of the false positive segments contain parts of brackets. Employing more sophisticated language-image segmentation methods, like [62], based on CLIP and SAM that do not rely on such a straightforward cropping strategy could prevent such wrongful predictions. In contrast, we observed less performance during the segment classification with CLIP when the background was not included in the segments.

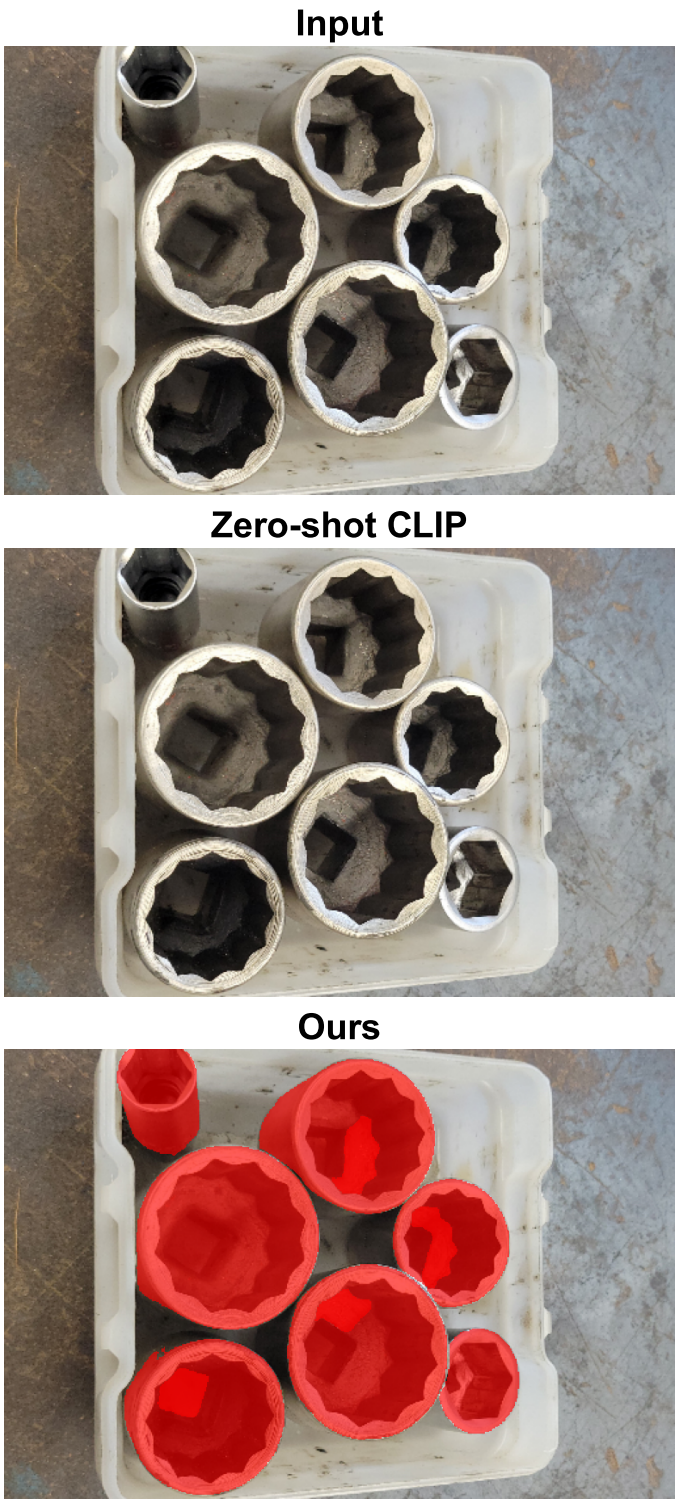

Fig. C.13: Language-guided segmentation results given the prompt *"socket"* compared to zero-shot CLIP under the same settings.

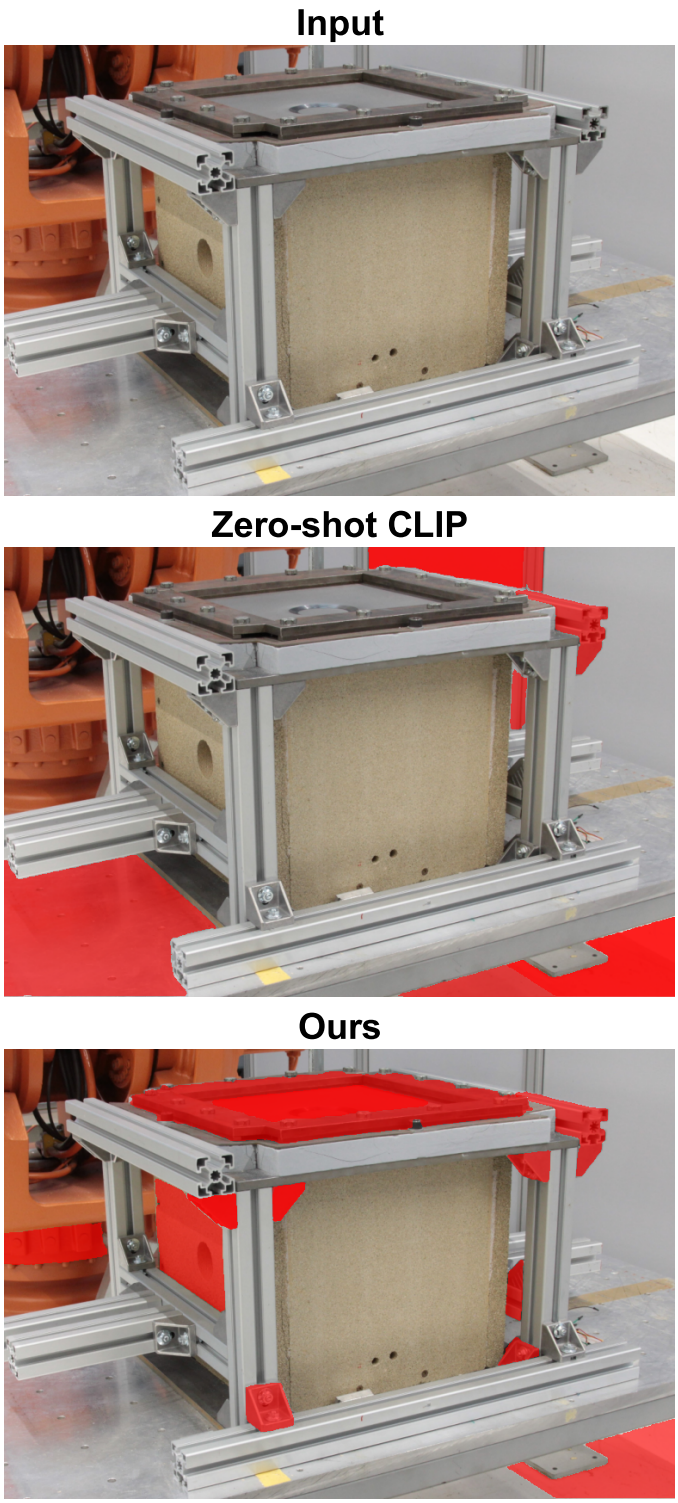

Fig. C.14: Language-guided segmentation results given the prompt *"bracket for construction profile"* compared to zero-shot CLIP under the same settings.