Final Year Project Term 1 End Report:
Title: Leverage CLIP to Impart the Capability to Understand Text Features to Lightweight Vision Models

1. Introduction:

Contemporary visual recognition pipelines excel when category vocabularies are fixed and exhaustively annotated, yet real-world deployments are rapidly shifting toward open-vocabulary and long-tail regimes where models must respond to evolving taxonomies, unseen concepts, and text-driven queries. At the same time, edge-facing scenarios—smart cameras, mobile phones, and IoT nodes—impose stringent latency, memory, and power constraints, forcing practitioners to balance broad semantic coverage with lightweight computation [4][5]. This dual mandate exposes a widening gap between the expressive capabilities of large multimodal models and the practical requirements of resource-constrained platforms.

Vision–language pre-training, typified by CLIP and its successors, has shown that contrastive learning over massive image–text corpora can embed heterogeneous modalities into a shared semantic space, enabling zero-shot classification, text-to-image retrieval, and phrase grounding by replacing closed-set classifiers with prompt-conditioned similarity scoring [1][2][3]. However, these models rely on heavy backbones (e.g., ViT-B/16) whose memory footprints and inference latency remain prohibitive for edge hardware. Conversely, highly efficient detectors such as the YOLO family achieve real-time throughput but remain bound to closed-label heads and lack robust linguistic grounding, limiting their utility for open-ended, language-guided perception [4][5].

Bridging this discount motivates the present project:to compress the semantic alignment learned by large vision-language models into lightweight architectures without sacrificing throughput.By leveraging a frozen CLIP text encoder as a stable semantic reference, pairing it with a streamlined YOLO-based visual backbone, and introducing targeted distillation plus text-conditioned modulation, the project aims to deliver a deployable pipeline that supports zero-shot classification, cross-modal retrieval, and eventually open-vocabulary object detection at practical inference speeds. Such a solution would democratize advanced multimodal understanding for edge environments, enabling “visual search by text” experiences on everyday devices while charting a pathway toward efficient, open-world perception.

To summarize, we make the following contributions: (i) We design a dual-stream model that marries a frozen CLIP text encoder with a YOLO-based visual backbone augmented by a learnable projection head and Feature-wise Linear Modulation (FiLM), enabling text-conditioned feature extraction without invoking the heavy CLIP image tower at inference time.(2) We introduce a feature alignment pre-training

regime that couples cosine-regression distillation against CLIP image embeddings with a contrastive InfoNCE objective, followed by task-specific fine-tuning heads for zero-shot classification, bidirectional image–text retrieval, and open-vocabulary detection, all reusing the shared embedding space.(iii) We provide a PyTorch codebase with modular APIs, pretrained weights, and scripts for the three downstream tasks, and we benchmark the model on standard datasets while reporting accuracy, retrieval metrics, detection quality, and efficiency indicators (parameters, FLOPs, FPS) on both GPU and edge-oriented hardware targets.

2.Related Works:

2.1. Vision-language model

Vision-language models (VLMs) seek to ground visual perception in natural language by learning joint representations that capture semantic correspondences between images and text. Formally, a typical VLM comprises an image encoder and a text encoder that map inputs into a shared embedding space. Once aligned, simple operations—such as cosine similarity—enable zero-shot recognition, bidirectional retrieval, visual question answering, and captioning without task-specific classifiers. The field has progressed from early dual-encoder systems trained on curated datasets to large-scale pre-training on web-scale corpora, paired with increasingly expressive Transformer-based backbones.

2.1.1. Contrastive pre-training at scale:

CLIP catalyzed the modern wave of VLM research by demonstrating that contrastive pre-training on 400M web-harvested image–text pairs can yield representations with broad semantic coverage [1]. CLIP couples a Vision Transformer (ViT) or ResNet image tower with a Transformer text tower, normalizes their outputs, and optimizes a symmetric InfoNCE objective with a learnable temperature to tighten positive pairs and repel negatives within a minibatch. Crucially, CLIP’s design decouples classification from fixed label vocabularies: downstream zero-shot classification is achieved by prompting the text encoder with natural-language templates (e.g., “a photo of a {class}”), generating textual “class embeddings,” and ranking them by cosine similarity against the image embedding. Follow-up works push on data scale, model capacity, and optimization. ALIGN scales contrastive pre-training to 1.8B noisy image–alt-text pairs scraped from the web, showing that data quantity can overcome weak supervision when paired with efficient filtering and large-scale TPU training [2].LiT introduces Locked-Image Tuning, freezing a pre-trained image encoder (from supervised or self-supervised regimes) and fine-tuning only the text

encoder plus a projection head on image–text pairs, improving sample efficiency and simplifying adaptation to new image towers [3].

2.1.2. Generative and instruction-aligned models:
While pure contrastive alignment excels at retrieval and zero-shot categorization, it provides limited fine-grained grounding and lacks generative capability. BLIP addresses this by combining three objectives—image-text contrastive learning, image-conditioned text generation, and text-conditioned image feature regression—within a unified framework trained on curated and web-scale data [4]. The retrieval-augmented pre-training pipeline iteratively mines high-quality image–caption pairs, mitigating noise and enhancing alignment. BLIP-2 further decouples visual and linguistic expertise by freezing a pre-trained vision encoder (e.g., ViT-G/14) and a large language model (e.g., FlanT5, OPT), then learning a lightweight “Q-Former” that interfaces between them, effectively bootstrapping multimodal capabilities while capping training cost [5].PaLI advances multilingual multimodal modeling by jointly pre-training on billions of image–caption pairs across over 100 languages, enabling cross-lingual captioning and VQA [6]. Kosmos-1 and related instruction-tuned systems align multimodal encoders with large language models via vision-conditioned prompting, facilitating open-ended reasoning, instruction following, and dialogue grounded in visual inputs [7]. These models widen the application spectrum from simple retrieval to conversational agents capable of multimodal inference.

2.1.3. Knowledge distillation:
Despite their prowess, state-of-the-art VLMs are computationally intensive: ViT-L/14 backbones with hundreds of millions of parameters, high-resolution inputs, and expensive cross-attention layers render them impractical for latency-sensitive or resource-constrained environments. This tension motivates research on distilling semantic alignment into lightweight students. Several lines of work investigate regressing image embeddings from CLIP into compact convolutional backbones [10], performing contrastive distillation with curated positives/negatives [11], or injecting textual guidance via Feature-wise Linear Modulation (FiLM) and cross-attention modules within efficient detectors [12]. Such strategies treat large VLMs as “semantic teachers,” whose frozen text (and sometimes image) towers provide stable targets while the student learns to project visual features into the aligned space. The distillation perspective is particularly relevant to edge deployment scenarios, where the goal is to maintain zero-shot flexibility and compositional semantics without incurring the inference cost of full-fledged VLMs.

In summary, vision-language modeling has matured from dual-encoder contrastive systems to versatile multimodal frameworks capable of understanding and generating rich semantics.

2.2.Open-Vocabulary Object Detection

Open-vocabulary object detection (OVOD) extends traditional detection beyond a fixed label set by allowing models to recognize and localize categories specified at inference time, including concepts unseen during training. Formally, given an image I and an arbitrary textual prompt t, an OVOD system must predict bounding boxes B = {b_i} and associated semantic scores s_i(t) without being constrained to a closed taxonomy. Achieving this requires combining the localization proficiency of detectors with the semantic richness of vision-language models, while mitigating the distribution shift between training labels and open-world concepts.

2.2.1.Flexible category vocabularies:

Open-vocabulary object detection (OVOD) requires detectors to adapt to user-specified labels or descriptive phrases at inference time—phrases such as “cat wearing a red hat” or “Ferris wheel”—instead of relying on a fixed classifier trained on a closed set of categories. Architectures therefore replace conventional softmax classifiers with text-conditioned scoring heads, often operating in a shared vision–language embedding space so that arbitrary prompts can be compared against regional visual features. Models like OWL-ViT and Grounding DINO exemplify this design by accepting free-form textual queries and matching them to image regions via cross-modal similarity, enabling detectors to scale their vocabularies far beyond the supervised labels seen during training[12][17].

2.2.2.Semantic transfer from training to inference:

Despite their open-ended goals, OVOD systems typically train on a limited collection of “base” classes for which bounding-box annotations exist, then generalize to “novel” classes that were unseen during training. Achieving this requires robust semantic transfer: the detector’s learned features must align with language representations so that the knowledge gained from base categories extrapolates to new concepts. Approaches such as ViLD and RegionCLIP distill semantics from large vision–language models into detection backbones, narrowing the gap between supervised base classes and open-world vocabularies by enforcing feature consistency between region proposals and textual embeddings [13][14]. Benchmarks like LVIS and ODinW further stress-test this transfer by evaluating performance separately on base and novel categories.

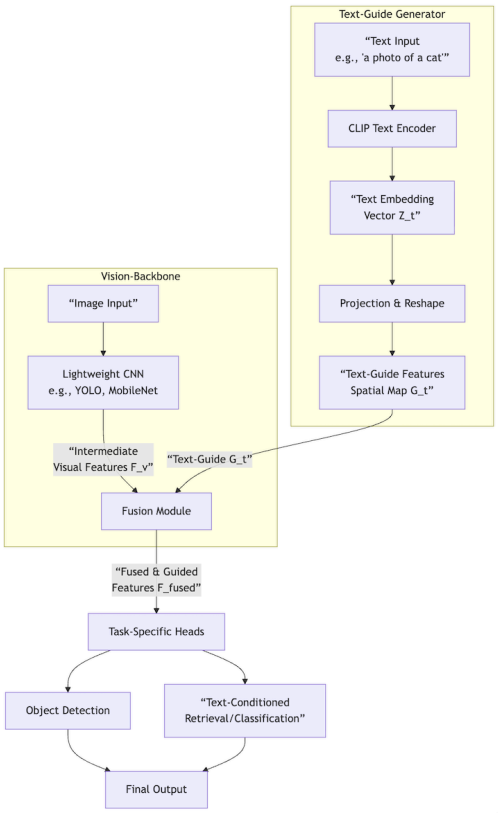
2.2.3.Coupling localization with linguistic semantics.

Unlike pure image–text retrieval, OVOD must pair accurate semantic understanding with precise localization. Detectors therefore integrate language conditioning into

proposal generation, classification, and bounding-box regression to answer both “what” and “where.” GLIP, Detic, and Grounding DINO demonstrate how joint optimization over large corpora of region–text pairs or pseudo-labeled boxes can align localization heads with language prompts, producing bounding boxes that respond to complex queries ranging from single nouns to descriptive phrases [15][16][17]. These systems highlight that effective OVOD hinges on uniting traditional detection modules—such as region proposal networks or transformer matching—with language-grounded supervision.

2.2.4.Leveraging vision–language pre-training:
Most state-of-the-art OVOD pipelines depend on powerful vision–language models, notably CLIP, to supply rich semantic priors. By freezing or lightly adapting these pre-trained encoders, detectors inherit broad linguistic knowledge while focusing fine-tuning on region-level alignment. Methods including PromptDet and F-VLM illustrate different transfer strategies: prompt tuning and adapter modules calibrate text embeddings to detection-specific distributions, whereas frozen encoders combined with lightweight heads drastically reduce training cost while preserving open-vocabulary recognition [18][19]. This synergy between large-scale pre-training and downstream detection continues to drive progress, enabling compact detectors to deliver competitive open-world performance through knowledge distillation, contrastive alignment, and prompt engineering.

3.Methodology:
The core objective of our approach is to distill the rich, cross-modal semantic understanding of a large pre-trained Vision-Language Model (VLM), specifically CLIP [20], into a lightweight vision model (YOLO) to enable efficient open-vocabulary vision tasks. Our methodology is centered on a two-stage pipeline: 1) Feature Alignment Pre-training and 2) Task-Specific Fine-Tuning. An overview of the proposed architecture is presented in the following figure.



3.1.Model Overview:

The proposed model is a dual-stream architecture designed for efficient inference, as it does not require the forward pass of the heavyweight CLIP image encoder during deployment.

3.1.1.Frozen CLIP Text Encoder:

This component serves as a static "semantic teacher". Given a set of text prompts (e.g., "a photo of a [class]"), it generates a set of L2-normalized text embeddings, $E_t \in R^{C*D}$,where C is the number of classes or concepts and D is the embedding dimension (e.g., 512). These embeddings define the target semantic space.

3.1.2.Lightweight Visual Backbone:

We replace the computationally expensive CLIP image encoder with a lightweight model, specifically a YOLOv8n [21] backbone and neck. This component extracts multi-scale visual feature maps, F_v, from an input image I. The inherent efficiency and object-level feature extraction capability of YOLO make it an ideal candidate for edge deployment.

3.1.3.Projection Network:

To align the visual features with the CLIP embedding space, we introduce a small, trainable projection network. This network, typically a Multi-Layer Perceptron (MLP), maps the global visual features to a L2-normalized image embedding, $E_i \in R^D$.

3.1.4.Cross-Modal Fusion Module:
A crucial innovation in our architecture is the integration of text-guided semantics directly into the visual backbone. We employ a Feature-wise Linear Modulation (FiLM) [22] layer to dynamically modulate the intermediate feature maps of the YOLO backbone using the text embedding. This conditions the visual feature extraction process on the semantic query, teaching the model to focus on relevant regions.

3.2.Feature Alignment Pretraining:
We trained the lightweight visual backbone and projection network to produce image embeddings that are semantically aligned with CLIP's joint embedding space. It is noted that the CLIP text and image encoders remain frozen. Plus, for training, we employ a composite loss function to enforce this alignment.

3.2.1.Distillation Loss ($\mathcal{L}_{distill}$):
We minimize the cosine distance between the projected image embedding E_i from our lightweight model and the image embedding E_i^{CLIP} obtained from the frozen CLIP image encoder for the same input image. This directly forces the student model to mimic the teacher's representation.
$\mathcal{L}_{distill} = 1 - cosine\_similarity(E_i, E_i^{CLIP})$

3.2.2.Contrastive Loss ($\mathcal{L}_{contrast}$):
To further strengthen the multi-modal alignment, we use a contrastive objective similar to CLIP but within our student model. For a batch of N image-text pairs, we compute the similarity matrix between all N image embeddings E_i and N text embeddings E_t. The loss is a symmetric cross-entropy loss aiming to maximize the similarity for the correct pairs.
$\mathcal{L}_{\text{contrast}} = \frac{1}{2}\big(\text{CE}(S, Y) + \text{CE}\big(S^T, Y\big)\big)$,
where S is the similarity matrix ($S_{i,j} = E_i \cdot E_t^T$), and Y are the ground-truth labels.

The total loss for feature alignment fine tuning is a weighted sum:
$\mathcal{L}_{total} = \alpha \cdot \mathcal{L}_{distill} + \beta \cdot \mathcal{L}_{contrast}$
where α, β are hyper-parameters.

3.3.Task-Specific Fine-Tuning:
Upon completion of the feature alignment pre-training, our model has learned a unified vision-language embedding space where the representations from the lightweight image encoder and the CLIP text encoder are semantically aligned. We

then leverage this foundational capability by attaching small, task-specific heads (scripts on GitHub)to perform the final objectives. Crucially, the heavyweight CLIP image encoder is completely discarded at this stage, and all inference is performed efficiently using only the lightweight image encoder and the frozen text encoder.

The following sections detail the fine-tuning strategies and inference mechanisms for each of our three core deliverables.

3.3.1.Cross-modal Retrieval:
A pivotal objective of our approach is not only to achieve the efficiency of lightweight models but also to preserve the powerful cross-modal alignment capabilities inherent in the original CLIP model. This task is explicitly designed to demonstrate that our distilled, lightweight visual model, in conjunction with the frozen CLIP text encoder, can maintain CLIP's proficiency in bidirectional image-text retrieval. The goal is to show that our efficiency gains do not come at the cost of losing the teacher model's core strength in understanding the semantic relationship between visual and textual content. Plus, this task evaluates the model's ability to perform bidirectional retrieval between images and text. We distinguish between two sub-tasks: Text-to-Image Retrieval and Image-to-Text Retrieval.

To underscore that this capability stems directly from the aligned feature space, we employ a simple cosine similarity calculator as the retrieval mechanism. The absence of a complex, task-specific head demonstrates that the retrieval performance is a direct consequence of successful feature distillation in the alignment pretraining.

We fine-tune the entire lightweight image encoder and projection network on image-text caption pair datasets (e.g., Flickr30k). The objective is to specialize the model for the ranking task, ensuring that the image embeddings produced by our efficient encoder can be effectively matched with the frozen, high-quality text embeddings from the CLIP text encoder. We use a contrastive loss (e.g., InfoNCE) that aims to maximize the similarity for positive image-text pairs and minimize it for negative pairs within a batch.

The retrieval process is a direct and efficient similarity search, leveraging the frozen components and precomputed embeddings.

- Text-to-Image Retrieval: Given a text query, its embedding E_t is computed using the frozen CLIP text encoder. For every image in a database, its embedding E_i^k is precomputed using the lightweight image encoder. The database images are then ranked by the cosine similarity between the text query embedding and all image embeddings.

- Image-to-Text Retrieval: Given a query image, its embedding E_i is computed using the lightweight image encoder. This embedding is then compared via cosine similarity against a database of text captions, whose embeddings E_t^k have been precomputed using the frozen CLIP text encoder.

This experimental setup allows for a direct and fair comparison to the original CLIP model's retrieval performance, validating that our distillation process successfully transfers the crucial cross-modal understanding from the teacher to our efficient student architecture.

3.3.2.Zero-shot Image Classification:
This task directly utilizes the aligned embedding space created in feature alignment pretraining without any additional task-specific parameters, serving as a direct validation of our pre-training success. Thus, there is no trainable head. The task is performed using a simple cosine similarity calculator between the global image embedding and all text embeddings. In some experiments, we may apply a slight fine-tuning of the entire model on a classification dataset to further adapt the features, but the core capability is established in the feature alignment pretraining.

In this task, for a set of C candidate classes, we first use the frozen CLIP text encoder to generate their text embeddings, $E_t \in R^{C*D}$, by feeding in prompts (e.g., “a photo of a [class]”. The lightweight image encoder then processes the input image I to produce a single, global image embedding $E_i \in R^D$. The probability distribution over the classes is computed via a softmax over the cosine similarities:

$$p(y=c \mid I)=\frac{\exp(\tau \cdot \text{cosine\_similarity}\ (E_i, E_t^c))}{\sum_{j=1}^{C} \exp\left(\tau \cdot \text{cosine\_similarity}\ \left(E_i, E_t^j\right)\right)}$$

where τ is a learnable or fixed temperature parameter. The final prediction is the class with the highest probability.

3.3.3.Open-Vocabulary Object Detection:
Open-vocabulary object detection extends zero-shot capabilities from global image classification to localized object instance, enabling the detection of categories not seen during pretraining.

To achieve this, we retain the powerful feature extraction backbone of the lightweight YOLO model but replace its standard classification head with a custom, lightweight OVOD head. This new head is designed to output semantically meaningful embeddings for each proposed region rather than a fixed set of class logits. It consists of two parallel branches:

1. A localization branch that performs regression to output the precise bounding box coordinates (center x, y, width, height) for each region of interest. This branch is functionally analogous to the bounding box regression component in a standard detection head.
2. A region embedding branch that projects the visual features of each proposed region into the shared D-dimensional semantic space established during feature alignment pretraining. This is implemented via a small, trainable Multi-Layer Perceptron (MLP) that outputs a normalized region-specific embedding, $E_{region} \in \mathbb{R}^D$.

The entire model (YOLO backbone, neck, and the new OVOD head) is fine-tuned on a detection dataset with bounding box annotations (e.g., COCO). The loss function is a composite of two objectives: $\mathcal{L}_{OVDO} = \mathcal{L}_{reg} + \gamma \cdot \mathcal{L}_{embed}$. Here, $\mathcal{L}_{reg}$ is the standard detection loss, combining objectness, bounding box regression (e.g., GIoU loss), and possibly a distillation loss for the localization quality. The critical component $\mathcal{L}_{embed}$ is a region-embedding alignment loss. For each ground-truth bounding box, we compute a distillation loss (e.g., cosine embedding loss) between its predicted E_{region} and the text embedding of the corresponding class name generated by the frozen CLIP text encoder. This forces the model to learn region-level visual representations that are semantically aligned with the language concepts defined by the text encoder.

After fine-tuning, the model can perform detection for an open set of categories. The inference process is as follows:

The image is processed by the lightweight YOLO backbone and OVOD head to produce a set of region proposals, each with a bounding box and its corresponding region embedding E_{region}.

A set of target class names (e.g., "cardboard box", "plastic bottle", "person") is converted into text embeddings $\{E_t^1, E_t^2, \ldots, E_t^C\}$ using the frozen CLIP text encoder.

For each proposed region, its embedding is compared to all text embeddings via cosine similarity. The class with the highest similarity score above a predetermined threshold is assigned to the bounding box.

This approach directly leverages the aligned vision-language space. The model's ability to detect a "cardboard box" stems from its training to make region embeddings

semantically close to the text embedding for "cardboard box", thereby enabling it to localize and recognize novel objects based solely on their textual descriptions.

4. Reference

[1] Radford, A., et al. “Learning Transferable Visual Models From Natural Language Supervision.” ICML, 2021.
[2] Jia, C., et al. “Scaling Up Visual and Vision-Language Representation Learning With Noisy Text Supervision.” ICML, 2021.
[3] Zhai, X., et al. “LiT: Zero-Shot Transfer with Locked-Image Tuning.” CVPR, 2022.
[4] Li, J., et al. “BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation.” ICML, 2022.
[5] Li, J., et al. “BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models.” arXiv:2301.12597, 2023.
[6] Chen, X., et al. “PaLI: A Jointly-Scaled Multilingual Language-Image Model.” ICLR, 2023.
[7] Huang, X., et al. “Language Is Not All You Need: Aligning Perception with Language Models.” ICLR, 2023.
[8] Li, Y., et al. “Supervision Exists Everywhere: A Data Efficient Contrastive Language-Image Pre-training Paradigm.” ICLR, 2022.
[9] Yuan, L., et al. “Florence: A New Foundation Model for Computer Vision.” CVPR, 2021 (Workshop on Computer Vision Foundation Models).
[10] Gao, M., et al. “Clip-Adapter: Better Vision-Language Models with Feature Adapters.” arXiv:2110.04544, 2021.
[11] Zhang, B., et al. “PointCLIP: Point Cloud Understanding by CLIP.” CVPR, 2022.
[12] Minderer, M., et al. “Simple Open-Vocabulary Object Detection with Vision Transformers.” ECCV, 2022.
[13] Gu, X., et al. “Open-Vocabulary Object Detection via Vision and Language Knowledge Distillation.” ICLR, 2022 (ViLD).
[14] Zhong, Z., et al. “RegionCLIP: Region-based Language-Image Pretraining.” CVPR, 2022.
[15] Li, X., et al. “Grounded Language-Image Pre-training.” CVPR, 2022 (GLIP).
[16] Zhou, X., et al. “Detecting Twenty-thousand Classes Using Image-level Supervision.” ECCV, 2022 (Detic).
[17] Liu, S., et al. “Grounding DINO: Marrying DINO with Grounded Pre-training for Open-Set Object Detection.” ICCV, 2023.
[18] Zhang, Z., et al. “PromptDet: Towards Open-vocabulary Detection using Uncurated Images.” CVPR, 2023.
[19] Dai, Z., et al. “F-VLM: Open-Vocabulary Object Detection upon Frozen Vision and Language Models.” NeurIPS, 2022.
[20] A. Radford *et al.*, “Learning Transferable Visual Models From Natural Language Supervision,” 2021. Available: https://arxiv.org/pdf/2103.00020

[21] G. Jocher, A. Chaurasia, and J. Qiu, “YOLOv8 by Ultralytics,” *GitHub*, 2023. https://github.com/ultralytics/ultralytics
[22] E. Perez, F. Strub, Harm de Vries, V. Dumoulin, and A. Courville, “FiLM: Visual Reasoning with a General Conditioning Layer,” *Proceedings of the ... AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, Sep. 2017, doi: https://doi.org/10.1609/aaai.v32i1.11671.