Mini-CLIP: Contrastive Language-Image Pretraining for Multimodal Learning

ES-667 (Deep Learning) Project Report

Aditya Borate
IIT Gandhinagar
aditya.borate@iitgn.ac.in

Aryan Solanki
IIT Gandhinagar
aryan.solanki@iitgn.ac.in

Nishchay Bhutoria
IIT Gandhinagar
nishchay.bhutoria@iitgn.ac.in

Abstract

We present **Mini-CLIP**, a lightweight reimplementation of the **Contrastive Language-Image Pretraining (CLIP)** architecture originally introduced by OpenAI. Our project aims to reproduce the core capabilities of CLIP, specifically **zero-shot generalization** and **cross-modal retrieval** within a computationally constrained environment. We pair a **ViT-S/16** and a **ResNet-101** with a **RoBERTa text encoder** to learn joint visual–semantic embeddings.

We experiment with three distinct fine-tuning strategies: (1) **projection-head-only** training, (2) **full-model fine-tuning**, and (3) a **stabilized partial fine-tuning** approach.

We evaluate our models on the **Flickr30k** dataset, benchmarking against **zero-shot CLIP**. Our results demonstrate that while full fine-tuning achieves the highest **Recall@K** scores, the stabilized partial fine-tuning strategy provides a **strong balance between performance and training stability**.

1 Introduction

1.1 Motivation

For years, the dominant paradigm in computer vision has been supervised learning on datasets like ImageNet, where models are trained to classify images into a fixed set of categories (e.g., 1,000 specific classes). While effective, this approach has significant limitations. A standard classifier reduces a rich image full of context, including background textures, lighting, and interactions, into a single label, such as "King Charles Spaniel." If the model encounters a visual concept outside its training set, it fails completely. Furthermore, it often "forgets" or ignores other aspects of the image, such as the grass in the background or the weather conditions, because they are not part of the target label.

OpenAI's CLIP (Contrastive Language-Image Pretraining) proposes a shift toward **Natural Language Supervision**. Instead of predicting a single label, CLIP is trained to predict which caption goes with which image. This allows the model to learn from the raw, noisy text found on the internet. Because it learns to match images to arbitrary sequences of words, CLIP is not restricted to a fixed vocabulary. It can model the notion of a "photo," a "sketch," or a "satellite view," and can generalize to entirely new tasks such as Optical Character Recognition (OCR) or geo-localization without any task-specific training.

1.2 Project Objectives

The original CLIP model was trained on a massive dataset of 400 million image-text pairs, a scale that requires significant computational resources. The primary objective of this project is to implement a "Mini-CLIP" that reproduces the fundamental mechanics of this architecture on a smaller scale. We aim to map images and text into a unified embedding space to enable:

- **Image → Text Retrieval:** Retrieving the most relevant caption for a given image.
- **Text → Image Retrieval:** Retrieving the most relevant image for a given text query.
- **Zero-shot Classification:** Classifying images using natural language prompts (e.g., "A photo of a dog") without training on explicit class labels.

We explore the efficacy of different vision backbones (ViT vs. ResNet) and investigate the impact of different fine-tuning regimes on the model's ability to learn semantic alignment from the Flickr30k dataset.

2 Methodology

2.1 Model Architecture

The Mini-CLIP architecture utilizes a dual-encoder design to process visual and textual inputs independently before mapping them into a shared latent space. The complete pipeline is illustrated in Figure 1 and proceeds as follows:

(1) **Input Processing:** A batch of N image-text pairs is sampled. Images are augmented and normalized, while texts are tokenized.
(2) **Feature Encoding:**
   - The **Image Encoder** extracts a feature vector I_f from the image.
   - The **Text Encoder** extracts a feature vector T_f from the caption using `RoBERTa-base`.
(3) **Projection:** Both feature vectors are passed through linear projection heads (W_I and W_T) to map them to a joint embedding dimension d.
(4) **Normalization:** The projected embeddings are L2-normalized to lie on a hypersphere.
(5) **Similarity:** The dot product between the normalized embeddings is computed to measure semantic similarity.

2.2 Backbone Models Used

To implement Mini-CLIP efficiently, we pair lightweight but expressive vision and language encoders:

Vision Encoders

ViT-S/16 (DINO) — A compact Vision Transformer using 16×16 patches. Embedding dim: 384. It captures global context via self-attention and performs strongly in multimodal alignment.

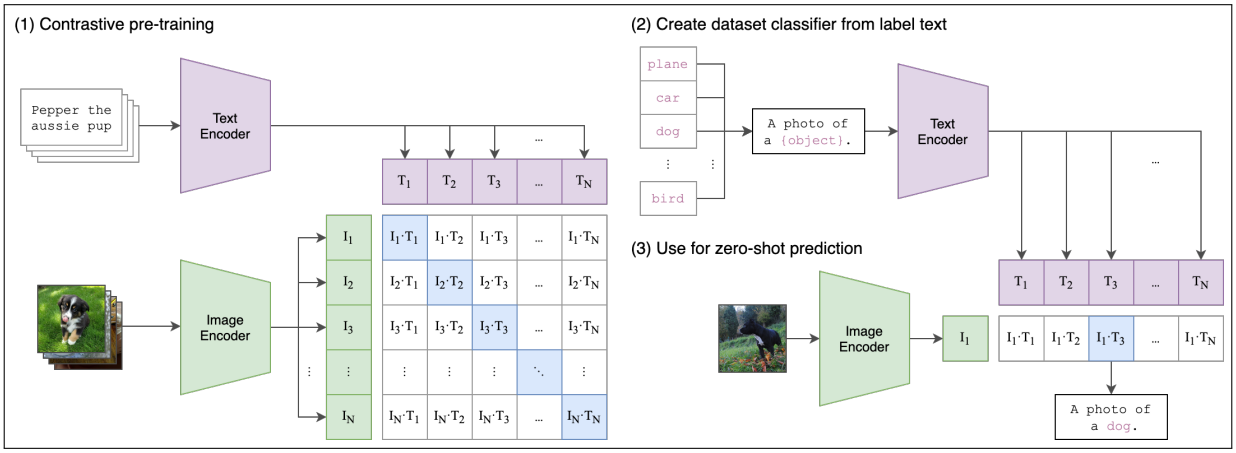


Figure 1: Overview of the Mini-CLIP architecture. Images and text are encoded separately and projected into a shared space where correct pairs (diagonal) are maximized and incorrect pairs (off-diagonal) are minimized.

ResNet-101 — A deep convolutional network with strong inductive biases such as translation invariance. Embedding dim: 2048 (projected). Used as a CNN baseline to study transformer vs. convolution performance.

Text Encoder

RoBERTa-base — A masked-language transformer encoder with rich contextual representations. Embedding dim: 768. This encoder maps raw captions into a semantic embedding compatible with image features.

These encoders are projected into a shared embedding space using learned linear layers.

2.3 Contrastive Learning Objective

To align the visual and textual representations, we employ the InfoNCE (Information Noise Contrastive Estimation) loss function. Given a batch of N pairs, the model maximizes the cosine similarity of the N correct pairs on the diagonal while minimizing the similarity of the $N^2 - N$ incorrect pairings.

Let $f_{img}(I)$ and $f_{text}(T)$ be the outputs of the projection heads. We compute normalized embeddings:

$$\tilde{z}_i^I = \frac{f_{img}(I_i)}{||f_{img}(I_i)||_2}, \quad \tilde{z}_i^T = \frac{f_{text}(T_i)}{||f_{text}(T_i)||_2} \tag{1}$$

The similarity between image i and text j is scaled by a learnable temperature τ:

$$logits_{ij} = (\tilde{z}_i^I)^\top (\tilde{z}_j^T) \cdot e^\tau \tag{2}$$

We compute the loss symmetrically for both modalities. The Image-to-Text loss $\mathcal{L}_{I2T}$ is the cross-entropy loss over the rows of the similarity matrix:

$$\mathcal{L}_{I2T} = -\frac{1}{N} \sum_{i=1}^{N} \log \frac{\exp(logits_{ii})}{\sum_{j=1}^{N} \exp(logits_{ij})} \tag{3}$$

Similarly, the Text-to-Image loss $\mathcal{L}_{T2I}$ is computed over the columns:

$$\mathcal{L}_{T2I} = -\frac{1}{N} \sum_{i=1}^{N} \log \frac{\exp(logits_{ii})}{\sum_{j=1}^{N} \exp(logits_{ji})} \tag{4}$$

The final training objective is the average of these two losses:

$$\mathcal{L}_{CLIP} = \frac{1}{2}(\mathcal{L}_{I2T} + \mathcal{L}_{T2I}) \tag{5}$$

2.4 Zero-Shot Classification Protocol

A key capability of CLIP is performing classification without specific training. To achieve this in our Mini-CLIP implementation:

(1) We define a set of target classes (e.g., “dog”, “car”, “plane”).
(2) We generate prompt templates for each class, such as “A photo of a {label}.”
(3) These prompts are encoded by the Text Encoder to produce embedding vectors T_{class}.
(4) The test image is encoded to I_{img}.
(5) We compute the cosine similarity between I_{img} and all T_{class} vectors. The class with the highest similarity score is selected as the prediction.

2.5 Fine-Tuning Strategies

We investigated three distinct fine-tuning strategies to balance plasticity and stability:

- **Setup 1 (Last Layer):** We freeze both backbones and train only the linear projection heads. This baseline tests the raw alignment capability of pretrained weights.
- **Setup 2 (Full Fine-tuning):** We unfreeze all parameters and fine-tune the entire network. This maximizes plasticity but risks catastrophic forgetting and overfitting.
- **Setup 3 (Stabilized Partial):** We perform full fine-tuning, followed by a stabilization phase where we freeze everything except the last two transformer blocks. This strategy aims to refine the high-level semantic features while preserving the robust low-level feature extraction filters.

3 Experiments

3.1 Dataset

We trained on the **Flickr30k** dataset, which contains 31,000 images and 150,000 captions. While significantly smaller than OpenAI's WebImageText dataset (400M pairs), it provides a sufficient density of high-quality descriptive captions for academic experimentation.

3.2 Implementation Details

Models were trained using the **AdamW** optimizer with a weight decay of 10^{-3} to regularize the weights. We utilized a **OneCycle learning rate scheduler** to cyclically adjust the learning rate, enabling faster convergence and improved generalization.

We optimized hyperparameters separately for the Full and Partial fine-tuning setups to ensure fair comparison. Table 1 details the specific configurations used.

Table 1: Hyperparameters for different fine-tuning setups.

Parameter	Full Fine-Tuning	Partial Fine-Tuning
Batch Size	16	24
Epochs	8	6
Learning Rate	5×10^{-5}	Head: 3×10^{-4} Backbone: 1×10^{-5}
Weight Decay	1×10^{-4}	1×10^{-4}
Patience	3	2
Projection Dim	256	256
Sequence Length	128	128

3.3 Compute Resources

To train Mini-CLIP efficiently, we utilized a combination of local and cloud compute resources:

- **Lightning AI GPU Studio:** Used for full-model training runs with access to **NVIDIA L40S** GPUs, enabling faster batching and large-scale fine-tuning.
- **Apple M4 Pro (MPS Backend):** Used for fine-tuning experiments through PyTorch's `mps` device support.
- **Local CPU Execution:** Used for embedding extraction and Streamlit demo inference.

3.4 Evaluation Metric: Recall@K

Recall@K measures the percentage of queries where the correct match appears in the top K results:

$$\text{Recall@K} = \frac{\text{Queries with match in Top K}}{\text{Total Queries}} \tag{6}$$

We report R@1, R@5, and R@10 for both directions. R@1 is the strictest metric, requiring the model to identify the exact correct match as its top prediction.

4 Results

We evaluated our models on a held-out subset of 1,000 Flickr30k test images. Table 2 summarizes the performance.

The results indicate that the transformer-based vision encoder (ViT) significantly outperforms the ResNet-101 baseline. Notably, our **Setup 3** (Stabilized Partial Fine-tuning) achieved the best results, outperforming even the zero-shot baseline of the original OpenAI CLIP model on this specific dataset partition.

5 Streamlit Demo Interface

We built an interactive retrieval demo using Streamlit to evaluate Mini-CLIP qualitatively. The app supports both Image→Text and Text→Image retrieval with real-time model inference.

Below are screenshots from the application (see next page):

6 Project Resources

All associated artifacts for Mini-CLIP are publicly available:

- **GitHub Repository:** https://github.com/Aryan-IIT/miniCLIP_DL
- **Presentation Slides:** https://github.com/Aryan-IIT/miniCLIP_DL/blob/main/report_and_slides/miniclip_slides.pdf
- **Original Project Proposal:** https://github.com/Aryan-IIT/miniCLIP_DL/blob/main/report_and_slides/DL_Project_Proposal.pdf

These resources include code, training logs, experiment configurations, and demo materials.

7 Released MiniCLIP Checkpoints

We publicly release the trained Mini-CLIP checkpoints to enable reproduction and downstream experimentation. All models are hosted on Hugging Face.

Table 3: Released MiniCLIP Checkpoints

Model	HuggingFace Link
ViT Small + RoBERTa	`HexAryan/vitsmall_roberta`
ResNet101 + RoBERTa	`HexAryan/resnet101_roberta`

Notes:

- **ViT Small + RoBERTa:** Full fine-tuning followed by last-two-layer stabilization (Setup 3).
- **ResNet101 + RoBERTa:** Full-model fine-tuning on Flickr30k (Setup 2).

8 Discussion

8.1 Transformers vs. CNNs

Our experiments strongly favor the Vision Transformer (ViT) over ResNet-101. While ResNets have strong inductive biases that help with small data, the global attention mechanism of the ViT appears better suited for alignment with the global context of text descriptions. This aligns with recent trends in the field suggesting that Transformers scale better with multimodal tasks.

8.2 Compute vs. Data Efficiency

A critical insight from our work and the broader literature is the trade-off between computational and data efficiency. Transformers

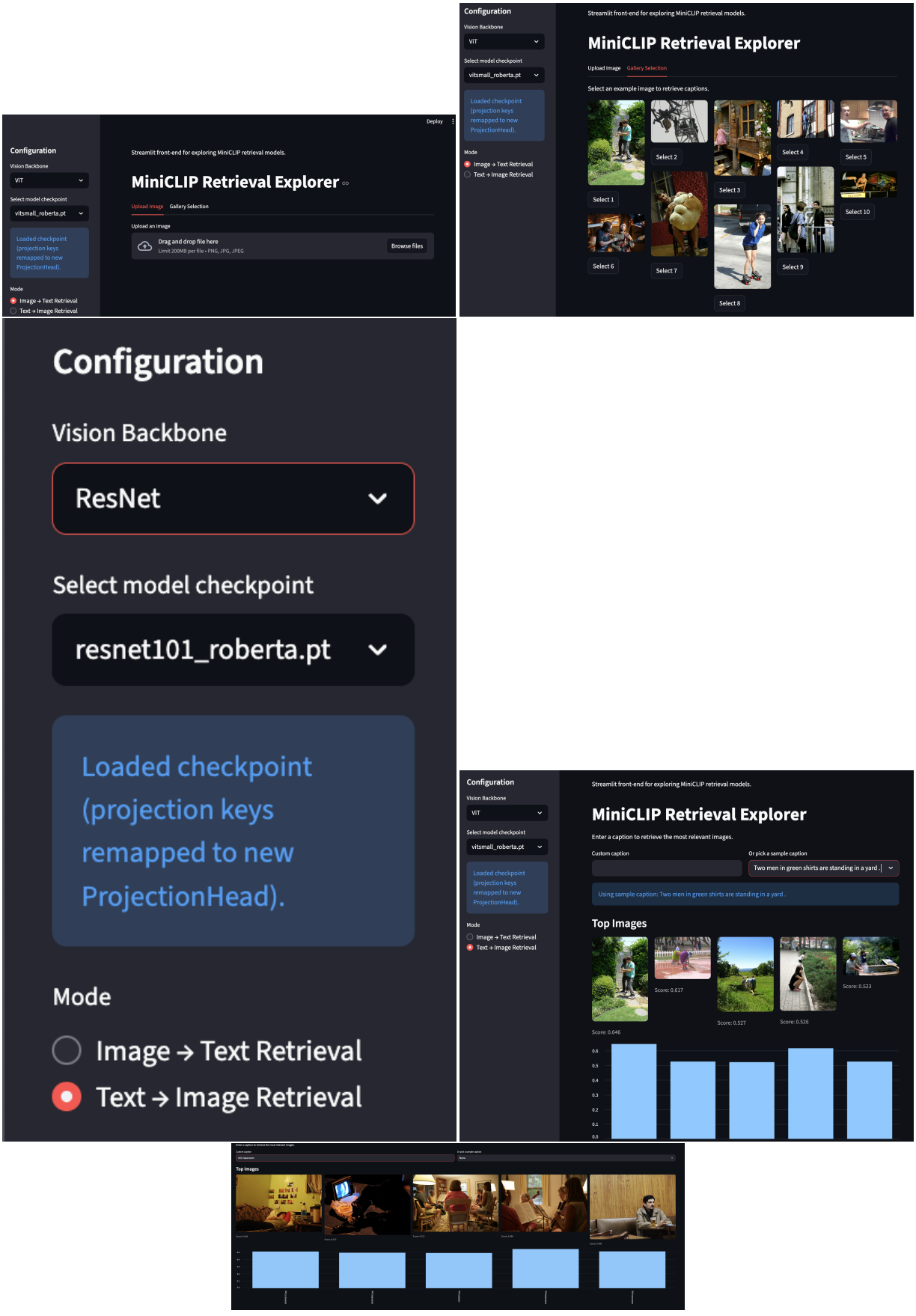
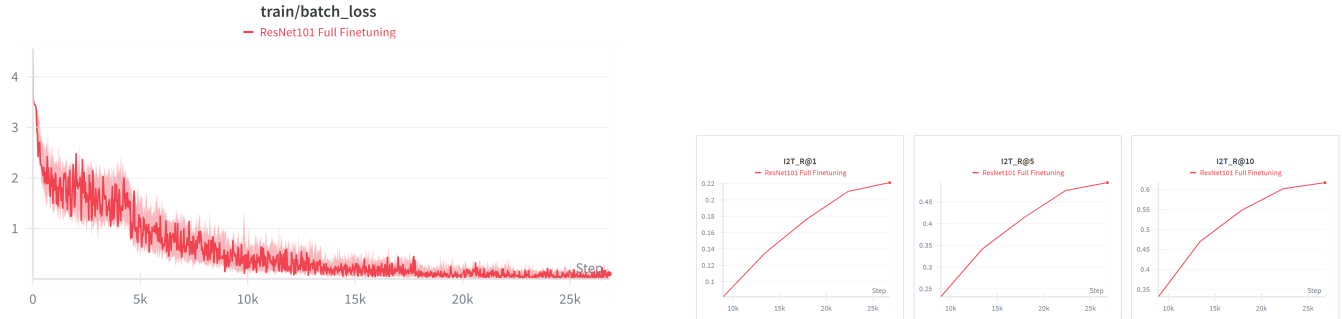


Figure 2: Screenshots from the Mini-CLIP Streamlit Retrieval App.

Table 2: Performance Comparison on Flickr30k Test Set (1000 Images)

		Image-to-Text (I2T)			Text-to-Image (T2I)		
Encoder Models	**Fine-tune Method**	**R@1**	**R@5**	**R@10**	**R@1**	**R@5**	**R@10**
ViT(Small) + RoBERTa	Setup 1 (Head Only)	21.6	53.6	65.2	22.4	50.4	64.2
ViT(Small) + RoBERTa	Setup 2 (Full)	28.2	68.2	75.4	28.4	62.7	74.6
ViT(Small) + RoBERTa	**Setup 3 (Stabilized)**	**36.6**	**72.7**	**83.8**	**33.9**	**71.9**	**84.0**
ResNet101 + RoBERTa	Setup 1 (Head Only)	5.7	6.9	15.2	5.5	6.3	14.9
ResNet101 + RoBERTa	Setup 2 (Full)	22.1	49.4	61.7	20.7	48.3	61.0
CLIP (OpenAI)	*Zero-shot Baseline*	*32.3*	*53.7*	*62.9*	*27.6*	*49.7*	*59.0*



(a) Training Loss Convergence

(b) Validation Recall Metrics

Figure 3: Training dynamics showing loss minimization (left) and retrieval metric improvement (right) over epochs.

are computationally efficient because they are highly parallelizable (unlike RNNs/LSTMs). However, they lack the inductive biases of CNNs (like translation invariance), making them **data inefficient**. This explains why our ViT model required careful fine-tuning (Setup 3) to perform well on the relatively small Flickr30k dataset. It needed to adapt its massive capacity without overfitting to the limited data samples.

8.3 Limitations

While CLIP's zero-shot capabilities are impressive, our research highlights several limitations. One major hurdle is fine-grained classification; the model struggles with tasks requiring distinction between very similar subclasses, such as specific models of cars, variants of aircraft, or species of flowers. It excels at broad categorization but lacks the specific feature discrimination needed for these fine-grained tasks. Furthermore, the model encounters difficulty with systematic and abstract tasks, such as counting the exact number of objects in an image, suggesting a limitation in spatial reasoning and numeracy. Another critical issue is out-of-distribution generalization. Although CLIP generalizes well to natural images, it fails when the data distribution differs significantly from its training set, as seen with the MNIST dataset where it cannot generalize from digitally rendered text to handwritten digits. Finally, data efficiency remains a challenge, particularly for "Mini-CLIP" implementations. CLIP does not solve the problem of learning from small data; rather, it compensates for the lack of inductive biases in Transformers by scaling to hundreds of millions of examples. Our implementation, trained on only 31k images, naturally faces challenges in learning robust, generalized features compared to the 400M-image scale of the original model.

8.4 Future Work

Several avenues exist to improve Mini-CLIP further. One of the most promising directions is the use of larger batch sizes or implementing gradeint accumulation to simulate the same. Contrastive learning benefits immensely from larger batches because they provide a higher number of "negative" samples per update, making the classification task harder and the resulting gradients more informative. Additionally, implementing Hard Negative Mining could significantly boost performance. Instead of relying solely on random negatives, selecting "hard" negatives incorrect images that semantically or visually resemble the correct one would force the model to learn more discriminative features. Finally, with sufficient computational resources, fine-tuning on larger and more diverse datasets could help mitigate overfitting and enhance the model's robustness.

9 Conclusion

In this project, we successfully developed Mini-CLIP, demonstrating that it is possible to learn a meaningful joint vision-language embedding space using limited resources. Our findings confirm the superiority of Vision Transformers over CNNs for this task.

Code Availability

The complete implementation of Mini-CLIP, including training scripts and evaluation code, is available at: https://github.com/Aryan-IIT/miniCLIP_DL.

References

(1) Radford, A., et al. (2021). Learning Transferable Visual Models From Natural Language Supervision. *arXiv preprint arXiv:2103.00020*.

(2) Dosovitskiy, A., et al. (2020). An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *ICLR*.

(3) Liu, Y., et al. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv preprint arXiv:1907.11692*.