# COSE474-2024F: Final Project Report
# "A Lightweight CLIP Model for IoT Environments: An Accuracy and Latency Optimized Approach"

**2022100124 Seoeun Kwon**

## 1. Introduction

Recently, leveraging the CLIP model in edge environments has become a significant trend. These edge environments typically require limited computational resources and low power consumption. Under such constrained conditions, optimizing and lightweighting the CLIP model to enable its use on edge devices and mobile platforms is essential.

In this project, I analyzed the basic CLIP model and apply quantization techniques, which are commonly used for model lightweighting. Furthermore, I propose methods to improve the performance of the quantized CLIP model. The focus is on achieving higher accuracy and lower latency for the CLIP model in edge environments.

### 1.1. Motivation

The increasing demand for efficient AI models in edge environments highlights the limitations of existing resource-intensive architectures like CLIP. While CLIP offers state-of-the-art performance in aligning visual and textual data, its deployment on edge devices remains a challenge due to high computational and memory requirements. This project is motivated by the need to make CLIP accessible to edge devices. By improving the performance of the quantized CLIP model, I aim to unlock its potential for real-time edge applications, ultimately contributing to the broader adoption of AI in resource-constrained settings.

### 1.2. Problem definition

CLIP models, renowned for their ability to align visual and textual data, are inherently complex and resource-intensive, making them unsuitable for direct deployment on edge devices. These devices, typically CPU-based and lacking GPU acceleration, face challenges in running such models due to limited computational power, memory, and energy efficiency. Moreover, while GPUs are optimal for the parallel processing required by CLIP, the prevalence of CPU-only edge devices exacerbates the performance gap. Therefore, there is a critical need to address the model's size, optimize it for CPU environments, and strike a balance between computational efficiency and accuracy, ensuring its applicability in resource-constrained edge environments.

### 1.3. Contribution

Using the knowledge distillation technique, I tried to transfer knowledge from a large-scale CLIP teacher model to a smaller student model, reducing the model size while maintaining its performance. This approach ensures that the lightweight model preserves CLIP's ability to learn text-image associations. Additionally, I optimized the lightweight CLIP model for CPU-based edge devices, enabling efficient operation in resource-constrained environments without requiring GPU acceleration.

## 2. Methods

### 2.1. Quantization

Quantization is a pivotal step in our approach to lightweighting the CLIP model for deployment on resource-constrained edge devices. By reducing the precision of model parameters, I significantly decrease memory usage and computational complexity while maintaining an acceptable level of accuracy. The method addresses two critical challenges in edge environments: limited computational power and the absence of GPU acceleration.

Algorithm 1 illustrates the process of loading the student model with an optional quantization mechanism. The pre-trained CLIP model serves as the base, and if quantization is enabled, dynamic quantization is applied to specific layers, such as linear and convolutional layers, using the QNNPACK backend. This choice ensures compatibility with CPU-based edge devices and enhances inference efficiency. For instance, parameters and operations are transformed into an 8-bit integer representation, which drastically reduces the model size and improves latency without compromising the model's ability to understand text-image relationships.

Unlike general-purpose quantization methods, our approach ensures that the unique characteristics of CLIP, such as multimodal alignment, are preserved. Additionally, while prior work often assumes the availability of GPUs, I optimize the model explicitly for CPU environments, bridging a critical

gap in edge AI deployment.

**Algorithm 1** Load the Student Model

**Input:** Quantization flag $quantized$ (default: True)
**Output:** Loaded student model
Load the base CLIP model:
$model$
$\leftarrow$ LoadPretrainedModel("openai/clip-vit-base-patch32")
**if** $quantized$ **then**
  Set quantization backend to QNNPACK
  Apply dynamic quantization to the model for size reduction and efficiency:
  $model \leftarrow$ ApplyDynamicQuantization(
  model, target_layers={LinearLayer,
  ConvolutionalLayer},
  data_type=QuantizedInt8)
  Print "Quantized student model loaded."
**else**
  Print "Non-quantized student model loaded."
**return** $model$

### 2.2. Knowledge Distillation

Figure 1 visually represents the Knowledge Distillation process employed in this study to transfer the representational power of a large-scale CLIP teacher model to a lightweight, quantized student model. Knowledge Distillation enables the student model to mimic the semantic understanding of the teacher model while maintaining computational efficiency. This technique is particularly useful in resource-constrained edge environments, where GPU acceleration is not feasible.

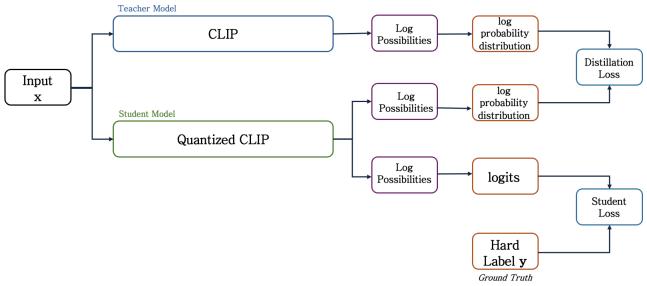


*Figure 1.* Overall process for knowledge distillation.

The process begins by feeding the input data $\boldsymbol{x}$ simultaneously into the teacher model (CLIP) and the quantized student model. Both models compute log probabilities of the input data and generate probability distributions. The probability distribution from the teacher model serves as a "soft target" for the student model. To align the outputs of the two models, the Distillation Loss, based on the Kullback-Leibler (KL) Divergence, is minimized. This ensures that the student model learns effectively from the teacher model's probability distribution.

Simultaneously, the student model directly minimizes a Student Loss, which is computed using Cross-Entropy Loss, based on the "hard label" $\boldsymbol{y}$. This additional objective ensures that the student model maintains the ability to generalize to new data and correctly predict ground truth labels. By jointly optimizing these two loss functions—Distillation Loss and Student Loss—the student model inherits the teacher model's knowledge while also being capable of performing well on real-world data. This dual optimization process is key to balancing model compression with robust performance. The final loss is computed as a weighted combination of the two losses:

$$\text{final\_loss} = \alpha \cdot \text{distillation\_loss} + (1 - \alpha) \cdot \text{student\_loss},$$

where $\alpha$ is a hyperparameter that balances the influence of the distillation loss and the student loss.

Through this process, the goal is to preserve the multimodal alignment capabilities of the CLIP model while significantly reducing its size and computational demands. Unlike existing research that focuses solely on performance improvement, this study is uniquely designed to enable the practical deployment of the CLIP model in CPU-based edge environments, enhancing its applicability in real-world settings. This distinction highlights the study's significance, as it provides a solution tailored to resource-constrained scenarios while maintaining strong performance.

## 3. Experiments

### 3.1. Datasets

In this study, I utilized the "mscoco_15k dataset", an open-source collection of image-caption pairs specifically designed for multimodal learning tasks. The dataset was accessed through the Hugging Face library, enabling seamless integration and efficient preprocessing. Due to the computational limitations of the CPU environment, a subset of 1,000 samples was selected from the original dataset for the experiments. This approach allowed the study to maintain computational feasibility while ensuring the representativeness of the data for the intended tasks. The selected subset was further processed to create training and validation datasets, with 80% of the samples allocated to training and 20% to validation.

For preprocessing, the images were resized to 224×224 dimensions, normalized using a mean of [0.5,0.5,0.5] and a standard deviation of [0.5,0.5,0.5], and converted into PyTorch tensor format. Captions were also tokenized and paired with their respective images to enable effective mul-

timodal learning. To accelerate preprocessing, parallel processing utilizing 4 CPU cores was employed.

### 3.2. Experimental Setting

All experiments in this study were conducted in a local CPU environment, carefully configured to account for limited computational resources. The computing resources used in this study are as follows: A local processor with multi-core support utilizing 4 cores was used as the CPU. The operating system was macOS 14.6 (BuildVersion: 23G80). The framework employed for the experiments was PyTorch 1.12.1.

### 3.3. Quantitative Results

| Model | I2T Recall@1 (%) | I2T Recall@5 (%) | I2T Latency (ms) |
|---|---|---|---|
| TEACHER | 76.0 | 99.0 | 667.9 |
| STUDENT | 78.7 | 98.2 | 1671.65 |
| DISTILLED | 51.8 | 85.3 | 1655.13 |

*Table 1.* Model performance comparison for Image-to-Text (I2T) tasks.

| Model | Model Size (MB) |
|---|---|
| TEACHER | 577.2 |
| STUDENT | 224.46 |
| DISTILLED | 224.46 |

*Table 2.* Model performance comparison for Text-to-Image (T2I) tasks.

Overall, in the I2T task, the Teacher model (original CLIP) demonstrated higher performance than the Student model (quantized CLIP). However, the model size of the Teacher model was approximately twice as large as that of the Student model. This indicates that model quantization has the advantage of reducing model size but tends to compromise performance, particularly for complex tasks that require high precision.

Moreover, the Teacher model exhibited shorter latency compared to the quantized Student model(the optimal $\alpha$ value was 0.3.), showing greater efficiency. This suggests that the quantization process may introduce additional computational overhead, leading to increased latency.

When comparing the Student model with the Distilled model, which underwent a knowledge distillation process, a performance decline was observed in both I2T and T2I tasks. However, latency was slightly reduced, indicating potential for speed improvement, though the reduction was not significant enough to be considered meaningful.

### 3.4. Qualitative Results

The test image can be found in the appendix for reference. The Teacher model accurately described the person in the image, demonstrating its ability to capture details of the scene. However, the description was not perfectly describing the scene. The Student model attempted to describe the broader context, but its output was less specific, indicating a loss of detail in the learned embeddings. The Distilled model failed to generate a meaningful description, suggesting that the distillation process introduced semantic drift, potentially due to reduced representational capacity.

| **Model** | **Output** |
|---|---|
| Teacher Model | A person that is dressed up very nicely. |
| Student Model | People in a large room, use multiple computers. |
| Distilled Model | There is an image of an outdoor area. |

*Table 3.* Comparison of Model I2T Outputs for the Test Image

### 3.5. Analysis

While the distilled model was expected to outperform the quantized student model in terms of performance, the results revealed that the quantized student model achieved higher accuracy in both I2T and T2I tasks, despite both models having the same size.

To analyze this, it is essential to consider the nature of knowledge distillation. Traditional knowledge distillation primarily focuses on improving classification tasks by utilizing the teacher model's softened logits via the softmax function. However, in the case of the CLIP model, which is designed for cross-modal generation tasks rather than strict classification, this approach might not have been as effective.

The CLIP model relies heavily on learning nuanced relationships between visual and textual modalities. The distilled model might have struggled to retain these intricate cross-modal relationships, resulting in a loss of performance. On the other hand, the quantized student model could maintain the essential structure and behavior of the original model more effectively due to its direct inheritance of pre-trained weights and tasks without altering the learning dynamics.

This discrepancy underscores the limitations of applying traditional knowledge distillation methods to tasks that involve multi-modal generation and highlights the need for distillation techniques tailored to such models.

## 4. Future Direction

It is necessary to develop customized knowledge distillation techniques tailored to multimodal models like CLIP. For instance, moving beyond the traditional softmax-based logit transfer approach, methods that preserve multimodal relationships, such as feature map-based distillation or enhanced cross-modal learning techniques, could be designed.

# A. Supplementary Materials

| Model | T2I Recall@1 (%) | T2I Recall@5 (%) | T2I Latency (ms) |
|---|---|---|---|
| TEACHER | 88.4 | 99.7 | 657.11 |
| STUDENT | 79.0 | 98.5 | 1654.01 |
| DISTILLED | 38.5 | 71.5 | 1652.73 |

*Table 4.* Model performance comparison for Text-to-Image (T2I) tasks.

The supplementary materials section provides additional information and references to support the main content of the report. Specifically, Table 4 highlights the performance comparison for Text-to-Image (T2I) tasks across the TEACHER, STUDENT, and DISTILLED models. The metrics include Recall@1, Recall@5, and latency, showcasing the relative strengths and weaknesses of each model in this task.

*Figure 2.* Test image for I2T test.

Figure 2 displays a sample test image used for the I2T evaluation. This image demonstrates a real-world scenario commonly encountered in such tasks, helping to contextualize the performance metrics reported in the main results. For further clarification or insights, please refer to this appendix.

## References

Wenliang Dai, Lu Hou, Lifeng Shang, Xin Jiang, Qun Liu, Pascale Fung, *Enabling Multimodal Generation on CLIP via Vision-Language Knowledge Distillation*, arXiv, 2022. [Online]. Available: https://arxiv.org/abs/2203.06386

Haoyu Song, Li Dong, Wei-Nan Zhang, Ting Liu, Furu Wei, *CLIP Models are Few-shot Learners: Empirical Studies on VQA and Visual Entailment*, arXiv, 2022. [Online]. Available: https://arxiv.org/abs/2203.07190

Ying Nie and Wei He and Kai Han and Yehui Tang and Tianyu Guo and Fanyi Du and Yunhe Wang, *LightCLIP: Learning Multi-Level Interaction for Lightweight Vision-Language Models*, arXiv, 2023. [Online]. Available: https://arxiv.org/abs/2312.00674

Gustavo Adolfo Vargas Hakim, David Osowiechi, Mehrdad Noori, Milad Cheraghalikhani, Ali Bahri, Moslem Yazdanpanah, Ismail Ben Ayed, Christian Desrosiers, *CLIPArTT: Adaptation of CLIP to New Domains at Test Time*, arXiv, 2024. [Online]. Available: https://arxiv.org/abs/2405.00754

Sanghyeok Lee, Joonmyung Choi, Hyunwoo J. Kim, *Multi-criteria Token Fusion with One-step-ahead Attention for Efficient Vision Transformers*, arXiv, 2024. [Online]. Available: https://arxiv.org/abs/2403.10030

Shuai Shen, Wanhua Li, Xiaobing Wang, Dafeng Zhang, Zhezhu Jin, Jie Zhou, Jiwen Lu, *CLIP-Cluster: CLIP-Guided Attribute Hallucination for Face Clustering*, Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), October 2023, pp. 20786-20795.