Visual Model Distillation

Yidan Qin
New York University
yq2421@nyu.edu

Ann Xiao
New York University
yx3493@nyu.edu

Abstract

Modern computer vision tasks have achieved remarkable success using pre-trained models; however, their large size and computational demands pose significant challenges for deployment in resource-constrained environments. This project explores knowledge distillation as a solution to compress these models into smaller, efficient student models while retaining competitive performance. We evaluate a range of distillation techniques, including simple projection, ResNet and DenseNet integration, and attention-based mechanisms, applied to both vision-only models (DINO, BEiT) and vision-language models (CLIP, BLIP).

Our results highlight the effectiveness of lightweight methods like simple projection for simpler datasets such as Tiny ImageNet, while advanced techniques like CBAM-based model deliver better accuracy on more complex datasets like COCO. The findings emphasize the importance of aligning distillation strategies with dataset complexity and teacher model characteristics. While resource constraints limited the scope of this study, our work provides valuable insights into the trade-offs between model size, accuracy, and efficiency, serving as a foundation for optimizing distillation techniques for real-world applications link for the code.

1. Introduction

Vision Language Models (VLMs) excel at integrating visual and textual information for tasks like image captioning and visual reasoning, while vision-only models specialize in tasks such as object detection and image classification. This dual focus enables a comprehensive evaluation of knowledge distillation techniques across diverse model architectures and application scenarios. Despite their success, the large size and computational cost of these models limit their deployment in resource-constrained environments, such as mobile devices, edge computing, and real-time systems.

Model distillation offers a promising solution by transferring knowledge from large, pre-trained teacher models to smaller student models. This approach achieves significant compression while retaining much of the teacher's performance. However, several questions remain unanswered: How effective are different distillation techniques across diverse datasets and tasks? What trade-offs exist between model size, accuracy, and computational efficiency?

In this project, we train student models distilled from both Vision Language Models (e.g., CLIP, BLIP) and vision-only models (e.g., DINO, BEiT) using diverse algorithms. Specifically, we focus on evaluating how different distillation approaches perform on datasets like Tiny ImageNet and MS COCO. By testing various techniques such as logit matching, attention transfer loss, and combined loss, we aim to identify the strengths and limitations of these strategies under distinct conditions.

This project investigates:

- How VLMs like CLIP and BLIP manage the integration of visual and textual information during distillation.

- How vision-only models such as DINO and BEiT specialize in visual tasks and adapt through distillation.
- The comparative performance of student models distilled using different techniques across datasets.

Rather than providing definitive solutions, this study serves as a stepping stone in exploring the complexities of distillation research, offering insights to guide further investigation and optimization of both VLMs and vision-only models.

2. Related Work

2.1. Vision Language Models (VLMs)

Recent advancements in VLMs have explored methods for integrating visual and textual modalities, enabling tasks such as image captioning, visual reasoning, and open-vocabulary object detection. Radford et al. [1] introduced CLIP, a contrastive learning framework for aligning vision and language representations, while Li et al. [2] proposed BLIP, a state-of-the-art model optimized for visual and language reasoning. Fang et al. [3] introduced a knowledge distillation method for transformer-based VLMs, addressing challenges in aligning hidden representations.

2.2. Vision-Only Models

In contrast to VLMs, vision-only models focus exclusively on tasks that require detailed visual representation learning, such as object detection and image classification. Models like DINO [4] and BEiT [5] have shown strong performance in these areas. DINO employs self-supervised learning to learn robust visual representations, while BEiT employs masked image modeling to pre-train a transformer for image understanding. Despite their success, there is limited research on distillation techniques specifically tailored to vision-only models, highlighting a gap in exploring their optimization.

2.3. Model Compression Techniques

Several strategies have been proposed to reduce model size while maintaining performance. Ye et al. [6] introduced VoCo-LLaMA, which compresses vision tokens using large language models. Chen et al. [7] presented LLaVolta, a stage-wise visual context compression framework, and Li et al. [8] demonstrated preference distillation to enhance the faithfulness and helpfulness of large VLMs.

2.4. Task-Specific Applications of Pre-Trained Models

Several studies highlight the adaptability of pre-trained models for specialized tasks. For VLMs, Zhou et al. [5] proposed AnomalyCLIP, adapting CLIP for zero-shot anomaly detection. Wu et al. [9] introduced CLIPSelf, enhancing CLIP's capabilities for open-vocabulary dense prediction. For vision-only models, tasks such as fine-grained object recognition and self-supervised learning demonstrate the potential of DINO and BEiT in domains requiring high-resolution visual understanding.

3. Methods

This section shows the datasets, teacher models, distillation techniques, and evaluation metrics used in our project to evaluate the performance-efficiency trade-offs for all the pre-trained models we chose.

3.1. Datasets

We utilize two datasets to evaluate the distillation methods:

- **Tiny ImageNet:** A reduced version of the ImageNet dataset with 200 classes and 64x64 resolution images. This dataset is ideal for evaluating the generated student models of the vision-only pre-trained teacher models on image classification tasks due to its smaller size and lower computational overhead.
- **MS COCO-2017:** A large-scale dataset that is well-suited for VLMs, it contains both images and corresponding textual annotations, we can easily use this dataset to train and validate our generated student models of the selected pre-trained VLMs.

3.2. Teacher Models

We chose 4 state-of-the-art pre-trained models as teacher models:

- **DINO:** A self-supervised vision transformer model designed for representation learning, serving as a robust teacher for vision-based tasks. It is not only widely used in computer vision research but has also been applied in many real-world applications.
- **BEiT:** A transformer-based model pre-trained using masked image modeling, which is highly effective for vision-language tasks. Its ability to reconstruct and interpret visual patterns also contributes to large performance improvements in real-world scenarios.
- **CLIP:** A model trained to align images and text, providing a strong baseline for tasks requiring both visual and textual understanding. We studied its architecture in our class; its contrastive loss is a critical mechanism that helps maximize the similarity between matching image-text pairs while minimizing the similarity between non-matching pairs.
- **BLIP:** Besides contrastive loss, BLIP also utilizes masked language modeling to learn rich semantic relationships between images and text. It is an excellent teacher model that can be used to distill smaller, more efficient student models.

3.3. Distillation Techniques

With the knowledge we studied in class, we designed and implemented six distinct approaches for transferring knowledge from teacher models to student models:

- **Simple projection:** The SimpleProjectionModel is a streamlined architecture for knowledge distillation, designed to map high-dimensional teacher outputs to compact embeddings using a linear vision encoder. The model normalizes the projected features to unit length and computes pairwise similarities, scaled by a trainable logit scale parameter. The resulting logits, representing feature similarities, are essential for aligning the student model's outputs with the teacher's distribution during distillation. This simplicity ensures efficiency and ease of integration into distillation tasks.

- **Distillation integrating Bottleneck (Basic Block):** This approach optimizes feature projection by employing a bottleneck-based architecture. The bottleneck block compresses high-dimensional outputs from the teacher model into efficient, low-dimensional representations through a series of layers. It uses a Linear layer to reduce the teacher's outputs to a bottleneck dimension, followed by a ReLU activation to enhance feature learning with non-linearity, and another Linear layer to project the bottlenecked representations to the desired dimensionality of the student model. This method prioritizes computational efficiency, significantly reducing model size while maintaining core feature representations. During training, logit matching loss is applied to align the predictions of the student model with those of the teacher, ensuring effective learning of high-level and fine-grained patterns.

- **Distillation integrating ResNet:** Instead of just using a Linear layer to map the high-dimensional features of the teacher models, this way integrates a ResNet to extract vision features from the teacher model, followed by a projection mapping these features to a lower-dimensional space to the student model. Both components reduce the complexity of the teacher model's outputs. To get a more reasonable training loss of the student model in VLM tasks, we applied a combined loss function: contrastive loss to align the vision and text embeddings of the output of the student model purely and KL divergence loss to minimize the difference between the student and teacher logits. The goal of the loss function is also to minimize the differences in predictions between the student models and teacher models, but since the vision features are not directly taken from the teacher model but are instead extracted using a ResNet backbone in the student model. This new loss function not only calculates the alignment loss between the vision and text modalities but also ensures feature learning from the teacher model.

- **Distillation integrating DenseNet:** When discussing ResNet, DenseNet often comes to our mind as the next logical progression in network architecture due to its dense connectivity pattern. DenseNet builds on the foundation of ResNet by connecting each layer to every subsequent layer, ensuring efficient feature reuse and gradient flow. There are also some works integrating ResNet and DenseNet as building blocks to create robust architectures. We also incorporated the DenseNet substructure into our student model, replacing its classifier layer with a projection layer mapping high-dimensional visual embeddings to a lower-dimensional space. By combining the dense connectivity benefits of DenseNet with our projection and encoding layers, similar to the ResNet structure above, we also utilize the combined loss for VLM student model generations.

- **Distillation integrating MultiAttention Block:** We have studied the structure of multihead attention in class, realizing its critical role in modern computer vision architectures. Multihead attention blocks are foundational components in many large-scale models. It can focus on different parts of the input simultaneously without generating the features sequentially, significantly improving the training efficiency of the large models. In this way, rather than directly projecting the high-dimensional features of the teacher model into lower-dimensional features to create the student model, we integrated attention blocks into the model architecture. Specifically, we employed multihead attention mechanisms for both vision and text modalities. In our experiment, the attention block processes high-dimensional features correspondingly, allowing the student model to selectively focus on relevant aspects of the input. These attention outputs are then projected

into lower-dimensional spaces using linear layers. The goal of this design is to help the student model get a better representation of the input features. Additionally, to train the student models, we applied the attention transfer loss for the VLM student model generation. Since we integrate the attention block into the student generation, we want to align the attention maps of the student model with those of the teacher model, ensuring that the student not only learns from the output distributions but also captures the focus of the teacher model.

- **Distillation integrating CBAM block:** The Convolutional Block Attention Module (CBAM) is another attention mechanism designed to refine feature representations by focusing on the most informative regions of the input data. Unlike standard attention mechanisms, CBAM incorporates both channel attention and spatial attention to sequentially enhance feature representations. We applied CBAM only to the vision part of the student model. The ability of CBAM block can enhance visual representations by emphasizing critical regions in the feature maps. Stacking multiple CBAM blocks in the vision encoder of the student model to mimic the heads for the traditional attention blocks. Each CBAM block refines the feature representations iteratively. This allows the model to try to focus more and more on the most important visual features during the 2 steps of attention in the CBAM chain. These enhanced features are then passed through projection layers to reduce the dimensions. Due to the utilization of the attention mechanism, we also applied attention loss when we trained the student models for VLM student model generations.

3.4. Evaluation Metrics

We assess the distilled student models using the following metrics:

- **Model Size:** The total storage requirement (in MB) of the student model, an essential metric for deployment in resource-constrained environments.
- **Accuracy:** Performance on classification and detection tasks, evaluated on Tiny ImageNet and MS COCO, respectively.
- **Training Efficiency:** Computational efficiency, measured in terms of training time for each epoch.

4. Experiments

This section details the experimental setup, baselines, and the distillation process used in this study.

4.1. Setup

The experiments were conducted in a controlled environment with the following configurations:

- We created several Jupyter notebooks writing with PyTorch for model implementation and training. These notebooks were designed to streamline the experimental process, including defining student models, implementing distillation methods, and training pipelines.

- Training pre-trained Visual Language Models (VLMs) like CLIP or BLIP, even for distillation, demands substantial GPU memory and compute time. With limited resources, we had to carefully balance the number of experiments and training configurations we could feasibly explore. Thus, the experiments were conducted with two types of epochs, allowing us to systematically analyze how different training durations and methods impacted the performance of the generated student models.
- To avoid retraining the models for the same epochs when the epoch count increases, we integrated a checkpoint mechanism into our code. This system stores all required parameters, including the model state, optimizer state, and loss, at the end of each epoch. These checkpoints are saved in the corresponding folder, allowing us to retrieve and resume training from any specific epoch seamlessly. This approach not only saves computational resources but also provides flexibility for fine-tuning and analyzing model performance at different training stages.
- All the pre-trained models for DINO, BEiT, CLIP, and BLIP are from publicly available repositories.

4.2. Distillation Process

The distillation process involves the following steps:

1. **Student Initialization:** Student models are initialized with a smaller architecture (they are initialized in the previous mentioned ways).
2. **Distillation Training:**
    - Implement each distillation technique (logit matching, attention transfer, combined loss are applied based on the student generation methods).
    - Train the student models using the loss signals from the corresponding teacher models.
3. **Evaluation:** Measure the performance (accuracy) and efficiency (size, inference time) of student models on the respective datasets.

4.3. Team Effort

Our team collaborated closely throughout the project, discussing all details together to ensure alignment on goals and methodologies. We held meetings twice a week to share progress, address challenges, and brainstorm ideas during the exploration phase.
After selecting the teacher models, we discussed together to brainstorm which distillation method we could apply and which corresponding loss function should be used. Since we have 2 sets of experiments: Yidan Qin conducted experiments on the COCO dataset, while Ann Xiao focused on running experiments on the Tiny ImageNet dataset. While running the experiments, we synced every day about the results. This collaborative approach allowed us to efficiently analyze results and draw comprehensive conclusions across both datasets.

5. Results and Discussion

The experiments are designed to analyze several aspects of knowledge distillation.

5.1. Results Analysis Across Methods

The results for the model size and the accuracy of each teacher model are summarized in Figure 1 and 2.

1. **CLIP:** Training each epoch for the CLIP-based student model takes around 15 - 18 minutes. The experiment results indicate that the methods that simply project from the teacher model and integrate the CBAM block can generate the student model with consistent accuracy while maintaining a relevant small model size of the student model among all methods. Similarly, the Multihead Attention integration achieves comparable accuracy, but the resulting student models are over 7 times as large as the previous method. The accuracy difference between training with 5 epochs and 10 epochs remains under 1% for the above 3 methods. However, with the integration of ResNet or DenseNet, more accuracy is gained from more training epochs. This is because these methods use raw image data to train the vision decoders instead of directly fetching pre-trained features from the teacher model, which adds complexity to the student model(their model size is dozens of times larger than the first 2 methods). As a result, they may need more epochs to converge or get more accurate results.

2. **BLIP:** Overall, the time required for training each epoch for BLIP is over 50 minutes, indicating the resource requirement is over 2 times as much as the CLIP-based model. The results shown in the Figure 1 highlight the method integrating the CBAM block produces student models with the highest accuracy while keeping the model size small. The methods utilizing ResNet and DenseNet to capture features from the raw data can also generate student models with comparable accuracy when the epochs reach 10, but the model size of the student model is dozens of times larger than the CBAM-based model. We observe increments in accuracy for these two methods as the training epochs increase, suggesting that the student models continue to benefit from both raw data and the teacher model, aligning with the findings in the CLIP student model experiments. Surprisingly, the methods integrating the Multihead-attention block deliver the highest accuracy levels when the epochs equals 5, but the accuracy then drops a lot when the number of epochs reaches 10. This behavior suggests that while the MultiHeadAttention-based method is effective at quickly distilling knowledge from the BLIP teacher model in the early stages of training, they may struggle to maintain the performance of BLIP. The bottleneck-based method, while achieving a compact model size of 0.75 MB, follows a similar trend to the Multihead-attention block. These contrast with ResNet and DenseNet-based methods, where accuracy consistently improves with additional training. Thus, for teacher models like BLIP, we may not simply rely on the teacher model for generating effective student models. The same trend can also be observed in the simple projection method, which delivers the worst accuracy results. When we purely rely on the teacher model, the student models struggle to generalize effectively, especially when the teacher model's representations are not directly aligned with the task or dataset used during distillation.

3. **DINO:** Training each epoch for DINO-based student models requires around 6–10 minutes, making it more efficient compared to VLMs like BLIP and CLIP. The results reveal that the simple projection method consistently delivers high accuracy for DINO across Tiny ImageNet with the smallest model size (0.75 MB). Methods integrating ResNet or DenseNet result in much larger student models (43.03 MB and 27.16 MB, respectively) without any improvement in accuracy. This suggests that DINO's pre-trained features

are highly effective and can be distilled efficiently with lightweight methods. Attention-based methods, while offering competitive accuracy, do not outperform simple projection in terms of accuracy or model size for this dataset. Additionally, the integration of ResNet faces memory constraints, highlighting the resource demands of this method.

4. **BEiT:** Training each epoch for BEiT-based student models takes in between 11 and 30 minutes, slightly longer than DINO but still more efficient than VLMs like BLIP. BEiT's pre-trained vision representations enable effective knowledge distillation with compact student models using the simple projection method, achieving high accuracy with a model size of 0.75 MB. However, DenseNet and ResNet methods, while producing larger student models (27.16 MB and 43.03 MB, respectively), do not achieve better accuracy for Tiny ImageNet. Attention-based methods (MultiAttention and CBAM) achieve a balance between size and accuracy, offering competitive accuracy (up to 0.8450) with moderate model sizes (9.76 MB). These findings suggest that while BEiT's pre-trained features allow for efficient distillation with simple methods, additional structures may be necessary for more complex tasks or datasets.

5.2. Dataset-Specific Insights

The characteristics of the datasets played a crucial role in the performance of the student models.

- **Tiny ImageNet:**
  Models trained on Tiny ImageNet consistently achieved higher accuracy compared to COCO, with the highest accuracy of 1.0 achieved using the simple projection technique for both DINO and BEiT teachers. While there is a possibility of overfitting, the relatively simpler nature of Tiny ImageNet tasks likely contributed to the smaller performance gaps between techniques. Advanced methods like ResNet and DenseNet offered minimal additional gains over simpler approaches such as projection, further emphasizing the reduced complexity of the dataset.

- **COCO Dataset:**
  The COCO dataset's diversity and complexity led to larger variations in accuracy across techniques. CBAM-based method showed the highest performance gains, particularly for BLIP and CLIP. However, simpler techniques like simple projection struggled for distilling the knowledge on some teacher models on COCO, especially for BLIP, as they were less capable of leveraging the teacher models' rich representations for this more complex dataset.

5.3. Teacher-Student Model Comparisons

We compared the performance of DINO, BEiT, CLIP, and BLIP as teacher models and their corresponding student models.

- **Teacher Model Effectiveness:**
  Models distilled from DINO consistently achieved the highest accuracy on Tiny ImageNet, even with the simplest projection technique, underscoring its strong pre-trained visual representations. For more complex dataset for VLM model, like COCO, it is necessary to choose a suitable teacher model for knowledge distillation.

- **Student Model Limitations:**
  Student models showed significant accuracy drops for COCO when using simpler methods on BLIP, such as projection. This highlights the importance of integrating more advanced techniques especially when the teacher model's representations are not directly aligned with the task or dataset used during distillation, such as CBAM blocks or DenseNet, to enhance feature extraction and task-specific alignment.

5.4. Limitations and Future Directions

Our experiments highlighted several limitations and potential areas for improvement:

- **Limited Interpretability:** Pre-trained teacher models acted as black boxes, making it difficult to explore how knowledge was transferred. Future work should focus on analyzing their internal mechanisms for better insights.
- **Resource Constraints:** Under resource limitations, we trained the student models for a maximum of 10 epochs. While this allowed us to observe early-stage learning trends, it may not fully capture the potential performance of the student models. Further training to convergence could provide more accurate results.
- **Exploration of Additional Methods:** Currently, we utilize six ways to generate the student models. Exploring more efficient strategies to improve the trade-off between model size and performance is also important for our work.

COCO dataset												
	SimpleProjection		Bottleneck		MultiheadAttention		CBAM		ResNet-based		Dense-based	
epoch = 5	CLIP	BLIP	CLIP	BLIP	CLIP	BLIP	CLIP	BLIP	CLIP	BLIP	CLIP	BLIP
Model Size	1 MB		0.75 MB		9.02 MB		1.25 MB		43.39 MB		27.53 MB	
Accuracy	72.08	29.76	72.3 MB	48.3	72	57.12	72.42	48.64	59.06	33.9	54.12	41.04
epoch = 10	CLIP	BLIP	CLIP	BLIP	CLIP	BLIP	CLIP	BLIP	CLIP	BLIP	CLIP	BLIP
Model Size	1 MB		0.75 MB		9.02 MB		1.25 MB		43.39 MB		27.53 MB	
Accuracy	72.42	28.2	72.72	42.42	72	34.8	72.34	66.34	64.62	51.24	56.98	56.82

Figure 1: Results on MS COCO-2017

Tiny ImageNet										
	SimpleProjection		Bottleneck		MultiheadAttention		ResNet-based		DenseNet-based	
epoch = 5	DINO	BEiT	DINO	BEiT	DINO	BEiT	DINO	BEiT	DINO	BEiT
Model Size	0.75 MB		0.50 MB		9.76 MB		43.03 MB		27.16 MB	
Accuracy	1.00	1.00	1.00	1.00	0.2004	0.84	Out of Memory	0.7150	Out of Memory	0.7470
epoch = 10	DINO	BEiT	DINO	BEiT	DINO	BEiT	DINO	BEiT	DINO	BEiT
Model Size	0.75 MB		0.50 MB		9.76 MB		43.03 MB		27.16 MB	
Accuracy	1.00	1.00	1.00	1.00	0.2004	0.8450	Out of Memory	0.7011	Out of Memory	0.7402

Figure 2: Results on Tiny ImageNet

6. Conclusion

In this project, we explored distillation techniques to create compact student models from pre-trained teacher models, including vision-only (DINO, BEiT) and vision-language (CLIP, BLIP) architectures. Effective distillation reduced model size while maintaining accuracy for simpler datasets like Tiny ImageNet. For complex datasets like COCO, advanced methods such as ResNet and DenseNet were needed to boost accuracy but required larger models and longer training times. DINO excelled on Tiny ImageNet, achieving high accuracy with simple methods, while BLIP and CLIP performed better on COCO with advanced architectures.

References

[1] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, and Ilya Sutskever. Learning transferable visual models from natural language supervision. *Proceedings of the International Conference on Machine Learning (ICML)*, 2021.

[2] Junnan Li, Dongxu Liu, Quan Yu, Steven Xie, and Yue Wang. Blip: Bootstrapped language-image pretraining for unified vision-language understanding and generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[3] Y. et al. Fang. Compressing visual-linguistic model via knowledge distillation. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[4] Q. Li. *Delay Characterization and Performance Control of Wide-area Networks.* PhD thesis, Univ. of Delaware, Newark, May 2000.

[5] Qihang Zhou, Guansong Pang, Yu Tian, Shibo He, and Jiming Chen. Anomalyclip: Object-agnostic prompt learning for zero-shot anomaly detection. *CVPR*, 2024.

[6] X. et al. Ye. Voco-llama: Towards vision compression with large language models. *arXiv preprint arXiv:2406.12275*, 2024.

[7] J. et al. Chen. Llavolta: Efficient multi-modal models via stage-wise visual context compression. *arXiv preprint arXiv:2406.20092*, 2024.

[8] L. et al. Li. Silkie: Preference distillation for large visual language models. *arXiv preprint arXiv:2312.10665*, 2024.

[9] Size Wu, Wenwei Zhang, Sheng Jin, Xiangtai Li, Wentao Liu, and Chen Loy, Change. Clipself: Vision transformer distills itself for open-vocabulary dense prediction. *ICLR*, 2024.