
Graphormer: A Hybrid CNN, GNN & Vision Transformer Image Classifier

Om Anavekar, Pranav Venkatram, Siddhartha Vanjari
Carnegie Mellon University
Pittsburgh, PA 15213
oanaveka, pvenkatr, svanjari

Abstract

We present **Graphormer**, a novel hybrid architecture that augments the Vision Graph Neural Network (ViG) by interleaving Vision Transformer (ViT) blocks within its backbone to enhance global feature extraction. While ViG effectively captures local and relational structures through graph convolutions, it lacks explicit mechanisms for long-range dependency modeling. To address this, we introduce transformer layers after each feedforward block in the ViG architecture, enabling the model to integrate global self-attention with graph-based reasoning. Our work provides a flexible blueprint for fusing ViT-style attention with graph-based representations in visual tasks.

1 Overview and Context

1.1 Motivation

Accurate image classification is a key component of computer vision with applications such as medical diagnostics, autonomous driving, and robotics. Although CNNs have made significant progress in object detection over the past decade, they often struggle to capture the complex relationships between objects within an image.

To address this gap, hybrid models that combine CNNs with Graph Neural Networks (GNNs) and Vision Transformers (ViTs) with GNNs have emerged. These models have the potential to better understand spatial hierarchies and contextual dependencies, enabling more accurate classification.

This research aims to integrate the three components into one model, with the goal of improving classification performance and expanding the capabilities of image recognition systems across various domains.

1.2 Objective

We propose an integrated hybrid architecture ViG Grapher Layers interspersed with ViT attention layers followed by a final MLP classification head. Such a system would progressively capture local and global features in an alternating manner aiming to capitalize on the unique strengths of each component in a progressive manner.

2 Related work & Background

2.1 Literature Review

Table 1: Comparison of Core Architectures

Feature	CNN	ViG	ViT
Input Representation	Pixel grid	Graph of image patches (nodes and edges)	Sequence of image patches (tokens)
Core Mechanism	Convolutional filters, pooling	Graph convolutions, message passing	Self-attention mechanism
Captures	Local spatial hierarchies, translation invariance	Non-local relationships, complex object structures	Global context, long-range dependencies
Inductive Bias	Strong (locality, translation invariance)	Moderate (depends on graph construction)	Weak
Data Efficiency	Generally good	Varies; can be good with effective graph construction	Requires large datasets
Computational Cost	Can be efficient with optimized architectures	Graph construction (e.g., KNN) can be expensive; graph processing varies	Quadratic complexity with respect to sequence length
Strengths	Hierarchical features; widely studied	Flexible representation; models relationships	Global context; highly parallelizable
Limitations	Difficulty with long-range dependencies	Graph construction cost; scalability can be an issue	Requires large data; high computational cost for large images

2.1.1 Vision GNN (ViG): Treating Images as Graphs for Recognition and Detection

In a departure from traditional grid-based or sequence-based representations of images, Han et al. proposed the Vision GNN (ViG) architecture, which fundamentally treats an image as a graph structure. This innovative approach divides the input image into a number of patches, which are then considered as nodes in a graph. Connections, or edges, are established between these nodes based on their proximity, typically by connecting each node to its nearest neighbors.

The ViG architecture itself comprises two primary modules: the Grapher module and the Feed-Forward Network (FFN) module. The Grapher module, built upon graph convolution operations, is responsible for aggregating and updating information across the graph, allowing each node to learn from its neighbors. The FFN module, consisting of two linear layers, is then used to transform the features of each node. By iteratively applying these Grapher and FFN modules, the ViG model can effectively learn graph-level features from the image representation. The authors explored both isotropic and pyramid architectures of ViG, with varying model sizes, and demonstrated its superior performance on both image recognition and object detection tasks. This success highlights the potential of using GNNs as a core component in mainstream computer vision tasks, which have historically been dominated by CNNs and, more recently, by Transformers. The ability of ViG to achieve strong results in these fundamental tasks suggests that GNNs are not limited to inherently

graph-structured data but can be effectively adapted for image analysis by appropriately defining the graph structure.

The approach of representing images as graphs offers several advantages. Firstly, a graph is a generalized data structure, encompassing grids and sequences as specific instances within its broader framework. This inherent flexibility allows for a more natural representation of objects with irregular shapes, which are often challenging for rigid grid-based structures like those used in CNNs. Secondly, by explicitly modeling the relationships between different parts of an image through the graph structure, ViG can potentially capture more complex interactions compared to models that primarily rely on local receptive fields or sequential processing.

Building upon the success of ViG, subsequent research has explored its limitations and proposed extensions. One notable development is the Vision Hypergraph Neural Network (ViHGNN), introduced by Han et al. While ViG relies on pairwise connections between nodes, ViHGNN addresses the limitation of simple graphs in capturing higher-order relationships within images. Hypergraphs, which can connect any number of nodes with a single hyperedge, offer a more powerful way to model complex interactions between multiple image patches simultaneously. The authors demonstrated the efficacy of ViHGNN across various visual tasks, including image classification and object detection, achieving competitive results.

Another important area of research related to ViG is the interpretability of these graph-based vision models. Standard ViG models have been found to yield large and overlapping receptive fields for the graph nodes, making it difficult to attribute model predictions to specific regions in the input image. To address this challenge, researchers have proposed interpretable variants, such as i-WiViG (Interpretable Window Vision GNN), which aims to provide explanations by automatically identifying relevant subgraphs for the model's prediction. This is achieved through window-based image graph processing, which restricts the node's receptive field to a local image region, and by using a self-interpretable graph bottleneck to rank the importance of long-range relations between image regions. The ongoing efforts to improve the interpretability of GNN-based vision models are crucial for building trust and understanding in these complex systems, particularly for critical applications. The broader landscape of research also reveals a growing interest in leveraging GNNs for a wide range of computer vision problems beyond classification and object detection, including tasks like semantic segmentation, image registration, and even industrial defect detection. This expanding scope underscores the increasing importance of graph-based methods in the computer vision toolkit.

2.1.2 LeViT: a Vision Transformer in ConvNet's Clothing for Faster Inference [1]

The authors propose a CNN-ViT hybrid that outperforms traditional CNNs and ViTs with respect to the accuracy-throughput trade-off for image classification (using the ImageNet dataset).

They combine a ViT with a CNN by reducing the number of layers in the ViT, stacking it on top of a pyramid consisting of convolution and pooling layers which extract local features and downsample the input.

This enables the ViT component to be computationally cheaper, since it operates on inputs of reduced resolution (yielding fewer tokens) and has fewer layers, leading to fewer attention operations.

During the initial training epochs, the training accuracy trend is similar to that of a CNN. However, the accuracy converges to a greater accuracy similar to that of ViTs in later epochs. The authors conclude that convolutional layers are more effective than attention to extract low level features in the early layers of the model, giving rise to faster convergence.

The authors also propose changes to the transformer portion of the architecture to make it computationally cheaper e.g. replacing the MLP with a 1x1 conv etc.

The insights offered by this paper showcase the complementary nature of convolution and attention and how combining the two leads to superior image classification accuracy while achieving greater throughput.

We hypothesize that this relationship will continue to hold when integrating a ViT with a CNN-GNN network to yield greater performance on the end task of image classification.

3 Methodology

3.1 Model Description

Our study will construct 1 model, with a combination of ViT, and ViG followed by a final MLP classifier head by inserting an Attention layer after every layer in ViG. We then evaluate it against the existing ImageNet baseline results.

The Vision Graph Neural Network (ViG) component models the relationships between the local features extracted in the first stage. By representing the image as a graph, where patches or local features become nodes, and edges represent their relationships, the model can capture complex, non-grid-like dependencies that might be missed by purely sequential or grid-based architectures. This graph-based representation could serve as a bridge between the underlying local features of the image and the sequence of patches required by the subsequent Vision Transformer (ViT). The ViG can potentially enrich the representation by explicitly learning which parts of the image are related and important, providing a more semantically rich input to the ViT.

The Vision Transformer (ViT) layers are included after each ViG layer to capture long-range dependencies and understand the global context of the image based on the graph-based features provided by the ViG. The self-attention mechanism within the ViT can then weigh the importance of different relationships learned by the ViG in a global context, allowing the model to integrate information from across the entire image to form a comprehensive understanding of its content and structure. After the ViG has modeled local relationships in a flexible way, the ViT can leverage its global receptive field to further process and refine this information.

Finally, the Multi-Layer Perceptron (MLP) at the end of the sequence serves as a standard classification head. It takes the high-level features learned by the model and maps them to the desired output classes. MLPs are well-suited for this task due to their ability to learn complex decision boundaries in the high-dimensional feature space produced by the preceding layers.

The role of the ViG in this sequence appears to be crucial as an intermediary, potentially transforming the features from the image into a graph structure that can then be effectively processed by the sequence-oriented Transformer architecture of the ViT. This transformation could enable the ViT to operate on a representation that is more semantically rich and aware of the relationships between different image regions compared to directly using patches from the original image. Furthermore, the graph structure has the potential to capture more intricate relationships between image regions compared to the fixed sequence of patches used in a standard ViT, as the edges in the graph can be dynamically determined based on the image, allowing for a more flexible and content-adaptive representation.

3.1.1 Model Diagram

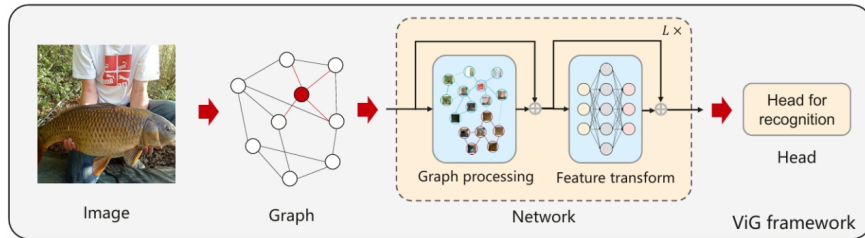


Figure 1: The existing ViG architecture.

Figure 1 shows the existing ViG architecture. Each layer of the ViG backbone consists of two core components: a Graph Processing module that performs message passing between connected nodes and a feedforward network. This structure allows ViG to model local and relational dependencies effectively, but it lacks an explicit mechanism to capture global context across distant patches.

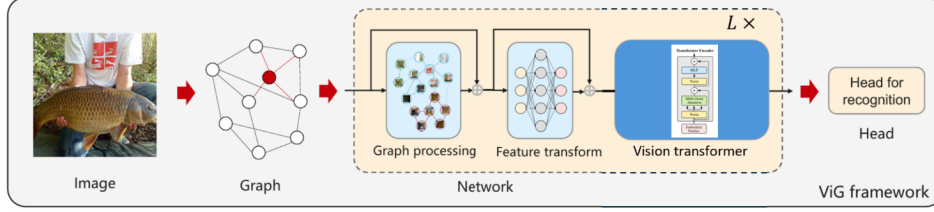


Figure 2: The proposed Graphormer architecture, with a Vision Transformer block inserted after each feedforward module.

Figure 2 illustrates the extended architecture. To address the limitations of the original ViG, the model is augmented by inserting a Vision Transformer (ViT) block after each feedforward module in the backbone. These transformer layers introduce self-attention mechanisms that enable global token interactions across all graph nodes, effectively complementing the local relational modeling of the GNN.

This research focuses on the ViG-Ti model for the purposes of simplicity and minimized computation time. The input shape is $224 \times 224 \times 3$ representing an RGB image. The output shape is 1000×1 vector containing class probabilities. The parameter count for the Graphormer model is 12M.

3.2 Dataset

This study will utilize the ImageNet 2012 dataset, a widely recognized benchmark for image classification tasks. The dataset consists of over 1.2 million annotated images spanning 1000 unique classes [5]. Each image in ImageNet is 224×224 RGB pixels.

This data set was selected specifically because it was utilized by Graham et al. and Han et al. for their model evaluations.

3.2.1 Data Preparation

The following steps were undertaken to prepare the ImageNet dataset for model training and evaluation:

1. The training and validation archives were downloaded from the official ImageNet repository.
2. A custom extraction script was employed to unpack all archives and organize the dataset into the required directory structure [4].

3.3 Evaluation Metric

To assess model performance, this study employs the standard Top-1 classification accuracy metric used in ImageNet evaluations [5]. Top-1 accuracy measures the proportion of samples for which the model’s most confident prediction matches the ground truth label. In classification tasks, models typically output a probability distribution over C classes via a softmax function. Let $\mathbf{p}_i \in \mathbb{R}^C$ denote the predicted class probabilities for the i -th input sample, and let $y_i \in \{1, 2, \dots, C\}$ denote its ground truth label. The predicted label is defined as:

$$\hat{y}_i = \arg \max_{c \in \{1, \dots, C\}} p_{i,c} \quad (1)$$

where $p_{i,c}$ is the predicted probability for class c .

Then, Top-1 accuracy is computed as:

$$\text{Top-1 Accuracy} = \frac{1}{N} \sum_{i=1}^N \mathbb{I}(\hat{y}_i = y_i) \quad (2)$$

where N is the total number of samples, and $\mathbb{I}(\cdot)$ is the indicator function that returns 1 if its argument is true and 0 otherwise.

This metric directly reflects the proportion of test examples for which the model’s highest-probability prediction is correct. It provides a simple yet effective benchmark for image classification tasks.

3.4 Loss Function

We utilize Cross Entropy loss to train the model since our use case is Image Classification. It is defined as follows:

$$\text{Cross Entropy}(Y, d) = - \sum_i d_i \times \log(y_i) \quad (3)$$

where d is an array of ground truth labels i.e. d_i is one-hot encoded and Y is an array of predicted classes.

4 Baseline & Extensions

4.1 Baseline Selection & Evaluation

LeViT implements a CNN-ViT hybrid, leveraging a CNN stem to perform downsampling and extract local features. Since ViTs rely on attention and are computationally intensive, the downsampling helps to accelerate training and inference by reducing the input size to the ViT. The ViT backbone aids in extracting global features. Hence, LeViT serves as a baseline for CNN-ViT hybrid which achieves superior accuracy and FLOPs to vanilla CNNs and ViTs.

ViG implements a CNN-GNN hybrid but arranges the layers in a structure similar to that of a ViT where there are a certain number of repeating blocks followed by an MLP classification head. Each block consists of a Graph Convolutional (GC) Layer and a Feed Forward Network. Thus, ViG replaces Attention with GC and achieves superior accuracy to vanilla ViTs.

We evaluate both baseline models to obtain the validation accuracy on ImageNet and report the results in Table 2.

4.2 Completed Experiments

- Naive integration of Attention layers with ViG by applying only 1 ViT layer at the end of the ViG model (just before the classifier head).
- Full integration of ViT layers with ViG by applying 1 ViT layer after each GC layer. This would enable both global feature extraction and utilizing GC layers to model relationships between K nearest neighbours of each patch [3].

5 Results & Analysis

5.1 Baseline Results

Table 2 shows the baselines obtained for various variants of LeViT and ViG. These baselines were obtained by evaluating available pretrained models using the Imagenet 2012 dataset. They are comparable to the expected baselines from each models’ respective papers.

Table 2: Comparison of calculated vs expected baseline Top-1 accuracy.

Model	Experimental Top-1 (%)	Published Top-1 (%)
LeViT-128S	76.53	76.6 [1]
LeViT-128	78.57	78.6 [1]
LeViT-192	79.89	80.0 [1]
LeViT-256	81.58	81.6 [1]
LeViT-384	82.59	82.6 [1]
ViG-TI-224	74.29	73.9 [2]
ViG-S-224	80.62	80.4 [2]
ViG-B-224	82.544	82.3 [2]
DeiT-B	81.8	81.8 [6]
DeiT-S	79.82	79.9 [6]
DeiT-T	72.13	72.2 [6]

5.2 Proposed Architecture Results

- **Naive integration:** 1 ViT block after the ViG backbone just before the classification head.

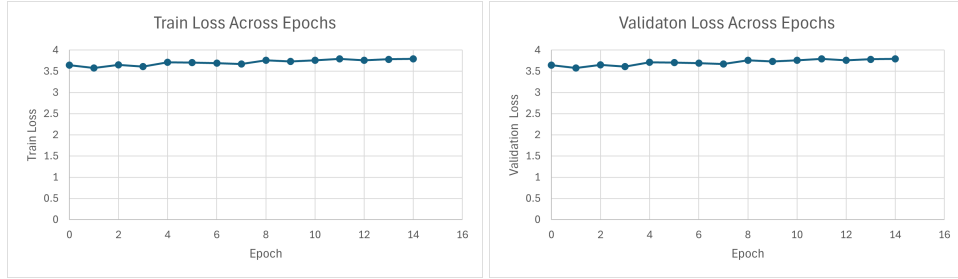


Figure 3: Naive Integration: Train Loss (Left); Validation Loss (Right)

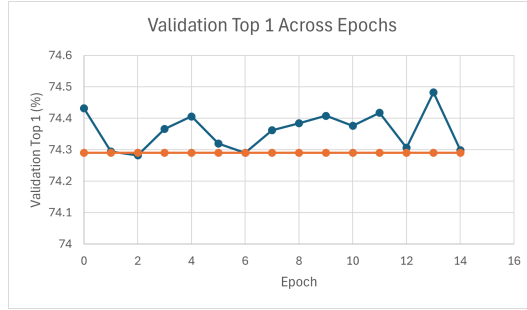


Figure 4: Naive Integration: Top 1 Classification Accuracy

- **Full hybrid model:** 1 ViT Ti (Tiny) layer inserted after each ViG Ti (Tiny) layer in the model.

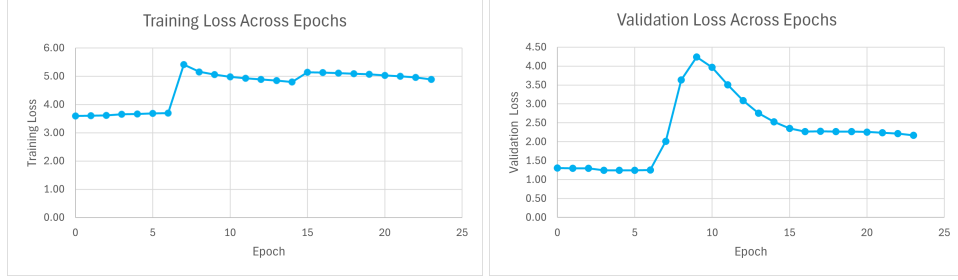


Figure 5: Full hybrid model: Train Loss (Left); Validation Loss (Right)

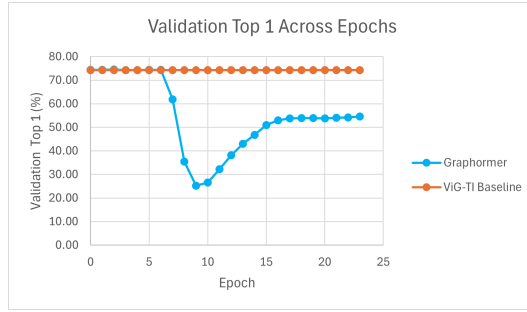


Figure 6: Full hybrid model: Top 1 Classification Accuracy

5.3 Discussion

- **Accuracy Progression:** The full hybrid model (ViT after every ViG layer) achieves a Top-1 accuracy of 74.53% at epoch 3 with all other layers frozen except for the attention layer(s). This is a 0.24 % improvement over the ViG-Ti baseline’s 74.29%. Upon unfreezing the weights, the accuracy drops sharply and recovers gradually; The model would require more epochs to converge.
- **Compute–Accuracy Trade-off:** Inserting a ViT layer after every ViG layer increases the parameter count by 69% (from 7.1 M to 12 M) while delivering an initial accuracy boost over the vanilla architecture (+0.24%) at epoch 3.
- **Resource Constraints:** Large dataset size and Compute limitations were the main bottlenecks for the project.

For the Naive integration in which we insert 1 ViT block after the entire ViG backbone, we have frozen the weights of the ViG backbone and trained only the ViT layer and the final classification head. Further, we initialize the ViT layer weights such that the weights are zero, resulting in the input features passed through to the output without any change. This is vital to ensure the accuracy of the model at the start of training begins at the pretrained ViG’s accuracy.

The loss plots show that the training and validation loss is stagnant across epochs, indicating that 1 ViT layer is not sufficient to capture global features as intended. Due to the minor fluctuations of the loss, we see that the classification accuracy also fluctuates across epochs.

The Full Hybrid model involves inserting 1 ViT layer after each ViG layer in the backbone. We apply the same weight initialization approach to ViT blocks as before, thus starting our training process with ViG’s pre-trained accuracy. The ViG layers’ weights are frozen for the first 7 epochs, allowing the ViT layer weights to train independently and bring them up to the required state. At epoch 3, we notice an increase in accuracy (74.53%) compared to the original baseline performance of 74.29 % i.e. +0.24 % accuracy.

We then unfreeze all the ViG weights and allow the entire model to train. At this point, we see a sharp increase in train and validation loss. This is expected since the optimizer attempts to move out of the local minimum found by the original ViG training. Since the model has 12M parameters to train, it would take a substantial number of epochs to converge. Due to compute limitations and the large dataset size of ImageNet, we were able to train a total of 24 epochs resulting in a Top-1 Classification Accuracy of 54.58 %.

From the initial proposal submission time to now, the primary challenge has been obtaining and training on the Imagenet dataset. The full set being over 150GB poses a challenge as most of the available computational resources (PSC, local machines) either cannot accommodate datasets that large or do not have the GPU capabilities needed for training/evaluation. Currently, the ECE cluster has been used as it has adequate storage resources. The caveat to this is that the storage directory has a file lifetime limit of 30 days, which means that the dataset will be deleted if it is not regularly copied/refreshed. Nevertheless, this has been the current working solution.

[5].

6 Future Directions

Future work will focus on conducting extended ablation studies to better understand architectural trade-offs. The number of attention heads in the injected ViT blocks will be varied to evaluate the balance between accuracy, computational cost, and inference latency. Transformer insertion frequency will also be tested, comparing placements after every feedforward block versus more sparse configurations such as every third block.

Beyond classification, the Graphormer architecture will be evaluated on object detection tasks using datasets such as COCO to assess its suitability for dense prediction problems. For deployment, trained models will be exported to ONNX and optimized using TensorRT to enable real-time inference.

7 Conclusion

In this work, we have introduced Graphormer, a novel hybrid architecture that seamlessly integrates Vision GNNs with Transformer blocks. Our key findings are:

- **Accuracy Trend:** The full hybrid model—injecting a ViT layer after every ViG block—achieves 74.53% Top-1 accuracy with only the attention layers unfrozen, improving over the ViG-Ti baseline’s 74.29% at epoch 3 of training. Further training is necessary upon unfreezing all the weights to attain convergence.
- **Compute–Accuracy Trade-off:** Adding ViT layers increases parameters by 69% (7.1M → 12M), resulting in the 0.24 % initial increase in accuracy. The tradeoff between computation complexity and performance boost will be quantified better upon further training of the model.

These results indicate that interleaving global self-attention within graph convolutional backbones to capture both local relationships and global context may yield performance gains. Future work will involve training Graphormer further until convergence and extending it to other vision tasks such as object detection.

8 Team Contributions

All team members contributed equally to this project.

Table 3: Team contributions

Team Member	Description
Om Anavekar	Prepare pipelines, train and evaluate models, handle compute resources
Pranav Venkatram	Develop hybrid models
Siddhartha Vanjari	Identification, Modeling, context and literature

9 GitHub

The repository can be found here: GitHub

References

- [1] Benjamin Graham, Alaaeldin El-Nouby, Hugo Touvron, Pierre Stock, Armand Joulin, Hervé Jégou, and Matthijs Douze. Levit: a vision transformer in convnet’s clothing for faster inference. *CoRR*, abs/2104.01136, 2021.
- [2] Kai Han, Yunhe Wang, Jianyuan Guo, Yehui Tang, and Enhua Wu. Vision gnn: An image is worth graph of nodes. *arXiv:2206.00272*, 2022.
- [3] Kai Han, An Xiao, Enhua Wu, Jianyuan Guo, Chunjing Xu, and Yunhe Wang. Vision gnn: An image is worth graph of nodes. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.
- [4] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. <https://github.com/pytorch/pytorch>, 2019.
- [5] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.
- [6] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. *CoRR*, abs/2012.12877, 2020.