Distilling Part-whole Hierarchical Knowledge from a Huge Pretrained Class Agnostic Segmentation Framework

Ahmed Radwan Mohamed S. Shehata
University of British Columbia
3333 University Way, Kelowna, BC
ahmedm04@student.ubc.ca

Abstract

We propose a novel approach for distilling visual knowledge from a large-scale pre-trained segmentation model, namely, the Segment Anything Model (SAM). Our goal is to pre-train the Agglomerator, a recently introduced column-style network architecture inspired by the organization of neurons in the Neocortex, to learn part-whole hierarchies in images. Despite its biological plausibility, we find that the original pre-training strategy of the Agglomerator, using supervised contrastive loss, fails to work effectively with natural images. To address this, we introduce a new pre-training strategy that aims to instill the model with prior knowledge of the compositional nature of our world. Our approach involves dividing the input image into patches and using the center point of each patch to generate segmentation masks through SAM. SAM produces three results per point to handle ambiguity at the whole, part, and subpart levels. We then train a simple encoder to utilize the intermediate feature maps of the Agglomerator and reconstruct the embeddings of the masks. This forces the network's intermediate features to learn objects and their constituent parts. By employing our pre-training strategy, we significantly enhance the classification performance on Imagenette, achieving an accuracy improvement from 58.6% to 91.2% without relying on any augmentation. Remarkably, we achieve this with a minimal parameter count of only 3.2 million, which is approximately 54 times smaller than the originally proposed Agglomerator. These results demonstrate both exceptional data and resource efficiency. Our code is available at: https://github.com/AhmedMostafaSoliman/distill-part-whole

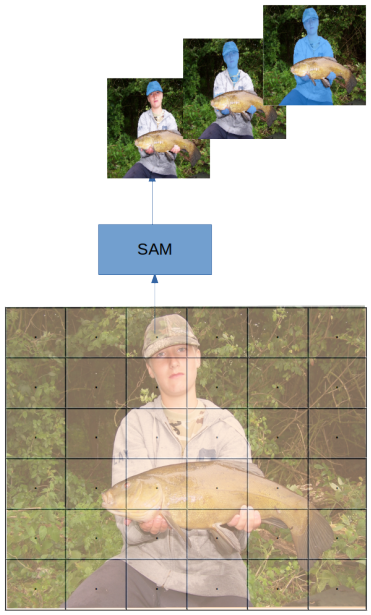

Figure 1. The image is divided into a grid of $p \times p$ patches. By using the central points of each patch as prompts, SAM generates multiple masks for each point, effectively resolving ambiguity. The figure illustrates this process by showcasing the three segmentation masks obtained when prompting SAM with a point within the boy's hat. These masks correspond to different levels of granularity: one representing the entire object (the boy), another depicting a part of the object (the complete face with the hat), and finally, a subpart (the boy's hat).

1. Introduction

In recent years, deep neural architectures have made significant advancements in various AI tasks, notably with convolutional neural networks (CNNs) [18] and more recently, architectures based on the vision transformer [5]. These architectures have achieved state-of-the-art results in many computer vision tasks. However, critics of deep learning argue that it struggles to handle compositional hierarchies effectively. CNNs, in particular, have been criticized for their limitations in capturing spatial relationships between identified patterns. For instance, if different parts of

a face image are swapped, a CNN may still identify it as a face since it focuses on static patterns rather than encoding explicit spatial relationships.

On the other hand, transformer-based architectures offer greater flexibility and fewer biases due to their self-attention mechanism. This mechanism allows any part of the input to attend to any other part, enabling the model to learn patterns more effectively from the data. However, this flexibility comes at the cost of increased data requirements and computational resources needed for training. Additionally, recent research [25] suggests that current state-of-the-art vision models struggle to generalize well to completely out-of-domain examples, highlighting the need for further investigation into the nature of human vision, which excels at domain generalization.

Furthermore, interpretability is a significant concern with these models. Using a model as a black box without understanding how it makes decisions can be problematic in various scenarios. The lack of interpretability raises questions about trust, accountability, and ethical implications. Therefore, it is crucial to explore approaches that address these challenges, aiming for models that better handle compositional hierarchies, generalize across domains, and provide interpretability.

Compelling evidence from cognitive science suggests that the human visual cortex processes scenes by hierarchically analyzing them into objects and their constituent parts. These hierarchical relationships involve transformations between intrinsic coordinate frames of reference, connecting the viewpoints of the parts and wholes [11].

In their work [7], the authors propose an arrangement of neurons in the Neocortical regions, organized into columns and layers, that facilitates the learning of the structure of the world. The core concept is that each column specializes in processing a specific portion of the sensory input array, such as input from the retina. A column comprises multiple levels, enabling it to capture information about a particular location across multiple hierarchical levels. To illustrate this concept in the context of visual inputs, let's consider a column responsible for processing a small part of the sensory input from the retina, specifically representing a portion of a dog's eyes. At the lowest level of the column, it would encode features specific to that part of the eye. At the next level, it would represent features of the entire eye, and at the highest level, it would capture features of the entire dog.

In [12], Hinton presents a design document outlining a neural network architecture that captures the concept of the part-whole hierarchy. This proposal aligns closely with the ideas expressed by Hawkins in [7]. Hinton's motivation stems from various biological and mathematical analogies, as well as the success of recent ideas such as neural fields, the transformer, and contrastive learning. The resulting design document, referred to as GLOM, provides a comprehensive framework combining these influences. However, this paper didn't provide a fully implemented system but rather laid the groundwork for researchers to pursue more human-like vision systems based on these principles.

A realization of GLOM called Agglomerator [6] was developed and demonstrated comparable results to the state-of-the-art on various datasets, including MNIST, smallNORB, CIFAR10, and CIFAR100. However, the authors did not report results on more complex datasets. They acknowledged the need for further exploration of different pre-training strategies. While the system exhibits strong biological plausibility, interpretability, and alignment with neuroscience research on the columnar structure of the brain, achieving practical results on natural images has proven challenging. As anticipated by Hinton in [12], training a GLOM-based system is not as straightforward as training a convolutional network.

In this paper, we explore the capabilities of the Agglomerator on a subset of 10 classes from ImageNet, specifically Imagenette. We demonstrate that the current state of Agglomerator, employing the originally proposed pre-training strategy utilizing supervised contrastive loss, fails to effectively learn to classify natural images from Imagenette. To overcome this limitation, we propose a novel pre-training strategy that harnesses additional supervision information obtained from unlabeled images. This additional supervision is obtained by leveraging the Segment Anything Model (SAM) [16], a powerful pre-trained class-agnostic segmentation framework.

Our contributions can be summarized as follows:

- We provide evidence that the current state of Agglomerator, utilizing the originally proposed pre-training strategy based on supervised contrastive loss, struggles to effectively learn the classification of natural images from ImageNet.

- We demonstrate how to leverage the vast potential of unlabelled images to acquire extra supervision information for Agglomerator. This is achieved by utilizing the pre-trained Segment Anything Model (SAM), a class-agnostic segmentation framework.

- We devise a pre-training strategy that capitalizes on the extra supervision obtained from SAM. Through this approach, we achieve significant performance gains of over 30% on Imagenette [14]. Remarkably, these improvements are accomplished while using approximately 54 times fewer parameters compared to the original Agglomerator, all without relying on data augmentation techniques.

2. Background and Related Work

2.1. Knowledge Distillation

A significant factor contributing to the success of deep learning is the scaling up of both model parameters and training data. However, this approach often leads to deploying state-of-the-art models that are either unattainable or prohibitively expensive. In contrast, humans possess the remarkable ability to generalize from just a few examples, even when faced with completely unfamiliar scenarios. This suggests that scaling up data requirements may not be the optimal path toward achieving more human-like vision.

Knowledge distillation (KD) provides a solution for transferring knowledge from a teacher model to a smaller student model. Different types of knowledge can be conveyed from the teacher to the student. In the realm of image classification, one popular approach is response-based knowledge distillation. This method utilizes the teacher model's logits to guide the student model in generating similar final output responses. A well-known response-based KD algorithm, introduced in [13], distills the knowledge of soft targets from the teacher model, which are estimated using the softmax function. Intriguingly, aligning the student model's outputs with the teacher's soft targets has a similar effect to label smoothing [15], a technique known to improve model performance. However, directly supervising the final layer output of a model is applicable only to supervised classification tasks, as the softmax output represents class probabilities. Furthermore, this approach does not address the supervision of the student model's internal representations beyond the last layer.

Another approach in knowledge distillation is feature-based knowledge extraction, which involves capturing knowledge from the intermediate layers of the teacher model and training the student model to produce similar responses in its own intermediate layers. This method was initially introduced in [22], where a deeper and thinner student model received hints from the teacher's hidden representations to enhance generalization and runtime efficiency.

Our work closely aligns with this approach as we primarily focus on utilizing the response of the teacher model, SAM, to directly supervise the intermediate features of the Agglomerator. By leveraging the teacher's responses, the student model can learn from the acquired knowledge and incorporate it into its own intermediate features. This enables the student model to benefit from the expertise of the teacher model, enhancing its ability to capture essential characteristics of the data.

2.2. Related Foundation Models

Several efforts have been made to devise architectures capable of representing scene parse trees. One such architecture is the Capsule Network [23], which constructs layers of capsules, fundamental elements trained to respond to objects or their parts. Through dynamic routing by agreement, each capsule establishes connections with its corresponding parents, forming a parse tree of the scene. The performance of Capsule Networks has shown competitiveness on datasets like MNIST [4], CIFAR10, CIFAR100 [17], and smallNORB [19]. However, they struggle to match the performance of convolutional or transformer-based approaches on more complex datasets.

As noted by Hinton in [12], one major issue with capsules is that using a mixture of capsules to model the set of objects or parts forces a hard decision on whether two closely related parts are considered similar or different. If we use two different capsules to model each part separately, then we are not able to capture the similarity between them. On the other hand, if we use the same capsule to model them both, routing to a parent becomes ambiguous. This limitation in modeling relationships between parts can hinder the capsule network's ability to effectively represent complex hierarchies in more intricate datasets.

The Reversible Columns Network [2] serves as a foundational model inspired by the columnar structure observed in the brain. In this paper, the term "columns" refers to replicated networks, specifically ConvNext [20] blocks used as the individual columns. The macro design of the architecture consists of an array of columns with reversible connections between each consecutive pair of columns. This arrangement allows the columns to progressively disentangle information across different levels. The levels closest to the input capture lower-level information, while those closest to the output encode more semantic information. Intermediate supervision is also incorporated in this work by adding a decoder head to the last level of each column. The network is trained on an auxiliary task, which involves reconstructing the input image. By including this auxiliary task, the network receives additional supervision at intermediate stages, enhancing the learning process.

2.3. Agglomerator

The Agglomerator framework is a realization of the GLOM system introduced by Hinton and is the framework to which we apply our pre-training strategy. Within the Agglomerator framework, an initial step involves transforming images with dimensions of $H \times W$ into $N = h \times w$ patches. Where, $h = H/4$ and $w = W/4$. Each patch among the N patches corresponds to a specific column within the framework. These columns are responsible for encoding the representations of their respective patches across multiple layers, capturing different levels of abstraction. At each level of each column, the embedding

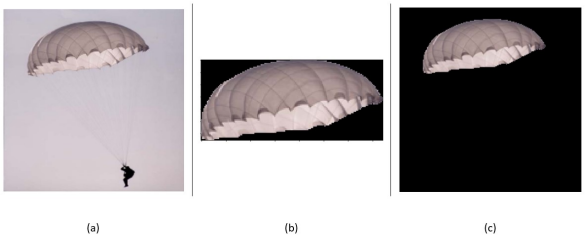

Figure 2. (a) the original image, (b) masking with cropping, and (c) masking without cropping retaining position information.

vector for each level is iteratively updated. The updated vector at each time step (t) is derived from four sources. Firstly, the vector from the previous time step ($t-1$) is considered. Additionally, the bottom-up prediction from the level below at the previous time step, the top-down prediction from the level above at the previous time step, and attention from all vectors at the same level from the previous time step are incorporated. These components are combined by employing four learnable parameters, resulting in the vector at the current time step (t).

The network is trained in two phases, firstly the pre-training phase where each image is duplicated, and has RandAugment applied to it then the network's contrastive head is pre-trained using the supervised contrastive loss. The process hopes that through the pretraining phase, the network is going to learn to successfully decompose the scene into objects and their constituent parts. While this could be true for easier datasets, and training on so many epochs, this proves to be quite challenging to do for natural images like in ImageNet. After the pre-training phase, a linear classifier on top of the contrastive head is trained using cross-entropy loss to predict the classification output.

3. Methodology

Deep neural networks, with their large capacity, possess the risk of overfitting the training data. They can exploit inherent biases present in the dataset, allowing the network to take shortcuts and learn specific intermediate features as needed. Simply minimizing the contrastive loss during the pre-training phase and then the cross-entropy loss during the classification phase does not guarantee that the Agglomerator's internal representation will effectively capture the desired parse tree structure of the image.

To address this challenge, we propose an effective method for pre-training the model and subsequently leveraging the pre-trained model for the classification task. The core idea is to introduce additional supervision to guide the network in learning what it should represent at each level within each column. Having prior knowledge of the hierarchical composition of scenes, objects, and their constituent parts, allows the network to easily generalize while being data efficient.

By incorporating this additional supervision, we aim to ensure that the network learns to accurately capture the intended hierarchical structure. This approach helps the model acquire the necessary knowledge to represent the parse tree of the image, ultimately enhancing its performance in the classification task.

3.1. SAM masks generation

Since our image is divided into ($N = p \times p$) patches, we use the coordinates of the center point of each patch to prompt SAM to give us the segmentation output at each point. Since each point can belong to a sub-part, part, or object we do have an ambiguous situation and SAM will resolve this by generating a multi-level segmentation output that corresponds to the hierarchy at this specific point. Figure 1 demonstrates the mask generation process and the output we expect to get from SAM at any specific point. The multi-level segmentation output from SAM is given by the 5D tensor $M \in \mathbb{R}^{B \times N \times L \times H \times W}$. Where B is the batch size, N is the number of points used to prompt SAM to generate masks (which should match the number of patches in the Agglomerator), L is the number of outputs per point prompt (which should match the number of levels used in Agglomerator), H, and W are the output masks dimensions which match the input image dimensions.

3.2. Positional information

While the architecture of the Agglomerator implicitly includes position information, given that each column processes a specific patch of the input, we adopt a unique approach to incorporate the position signal into the masks. Rather than masking out the desired part of an image and cropping it separately to obtain its embeddings, we retain the desired part within a dark background and embed the entire image. In this approach, the background surrounding the desired part is masked out, while the object itself remains visible. This allows us to encourage the network features to be aware not only of the semantic information of the object but also of its precise location within the image. Recent neuroscience research [8] suggests that columns have a distinctive capability to represent location. Thus, by incorporating the location-awareness aspect, we aim to enhance the network's understanding of both the object's semantic attributes and its spatial placement within the image.

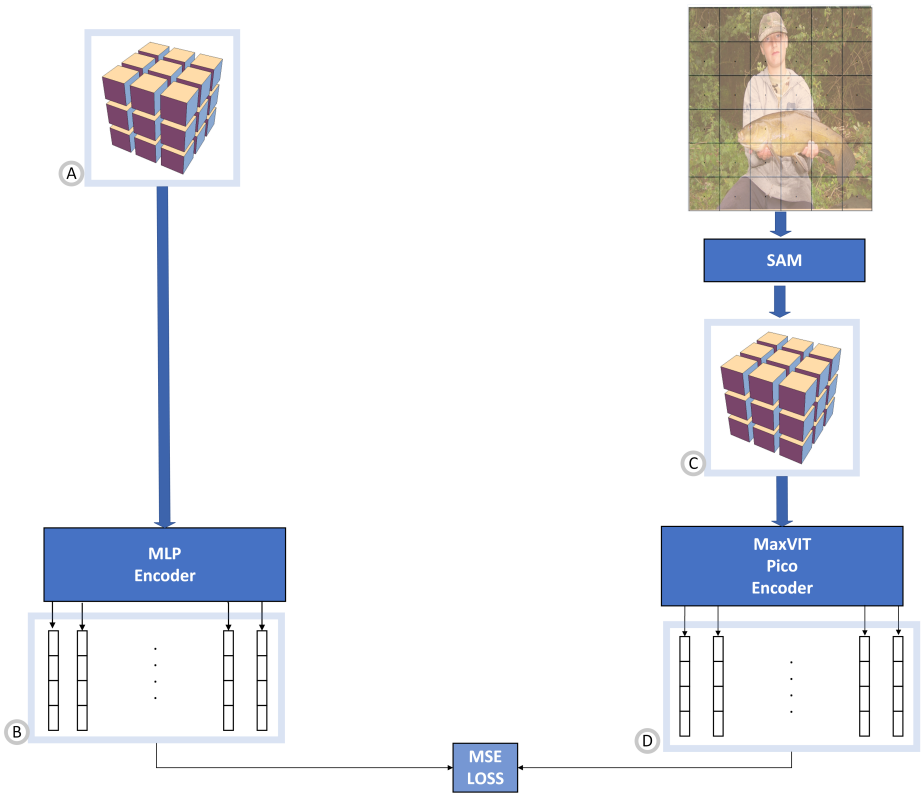

Figure 3. Our proposed pre-training procedure. (A) Agglomerator's internal representation Z, (B) Embedding of Agglomerator's internal representation Z_e, (C) Multi-level output of SAM masks M, and (D) Embedding of Multi-level output of SAM masks M_e. The procedure aims to minimize the mean squared error between (B), and (D) as described in 3.3

3.3. Intermediate Supervision

To supervise Agglomerator's hidden representation, firstly, we employ a trainable MLP encoder for Agglomerator's internal hidden representation which is given by $Z \in \mathbb{R}^{B\times N\times L\times D}$. Where D is the dimension of the vector representation of each patch. The output from the Agglomerator's encoder is another hidden representation which is given by the tensor $Z_e \in \mathbb{R}^{B\times N\times L\times D_e}$, where D_e is the size of the embedding of the hidden representation of each patch.

Similarly, we utilize a scaled-down version of the pre-trained MaxVIT [24] image encoder, known as pico MaxVIT, to create embeddings for the generated SAM masks as explained in 3.1. M is passed to the MaxVIT encoder to obtain the masks embeddings which output an embedding representation of M, $M_e \in \mathbb{R}^{B\times N\times L\times D_e}$, where D_e is the image embedding size we obtain from the MaxVIT encoder which matches the aforementioned hidden representation size in Z_e.

Since our objective is to align Agglomerator's encoded hidden representation with the embeddings of SAM masks, we would like to keep the mean squared error (MSE) between the Z_e and M_e as low as possible. Having this objective achieved means that Z is indeed representative of a valid part-whole hierarchy since it was able to predict the hierarchy we created using SAM.

Since we are mostly concerned with the classification task in this paper, we care the most about the semantic parsing of the image, while we do not care as much about the sub-part level detail in the hierarchy. Therefore we reflect this in the loss function by splitting up the loss into as many components as levels and then weighing each level's loss individually, with more weight being given to the last level's representation. The loss at each level k, $\mathcal{L}_k$, is the MSE between the k^{th} level in Z_e and M_e which is given by

$$\frac{1}{B\times N\times D_e}\sum_{b=1}^{B}\sum_{i=1}^{N}\sum_{j=1}^{D}(Z_e^k[b,i,j]-M_e^k[b,i,j])^2 \quad (1)$$

The total loss we use in our pre-training approach is given by:

$$\frac{\sum_{k=1}^{n}(\alpha_k\cdot\mathcal{L}_k)}{\sum_{i=1}^{n}\alpha_k} \quad (2)$$

Where $\mathcal{L}_k$ is the mean squared error between Agglomerator's k^{th} level embeddings, and the embeddings of the k^{th} level output of SAM masks, and α_k is the weight associated to $\mathcal{L}_k$ in the loss function.

3.4. Visualization

Following the footsteps of the original work in Agglomerator, we visualize the islands of agreement for multiple images and show that indeed the model has learned a good enough representation of the objects present in the scene by having similar embedding vectors all over locations representing the same object, or part of it. Figure 4, shows the islands of agreements at different levels. Patches with similar embedding vectors are assigned similar colors. As we go up in the hierarchy, i.e. the rightmost columns, we see that the embedding vectors converge to form two islands. One for the foreground, and the other for the background.

4. Experiments

4.1. Dataset

All the experiments were run on Imagentte[14], a subset of 10 classes from the widely used benchmark ImageNet. We chose this dataset since it offers the complexity of natural images to demonstrate the effectiveness of our approach while saving computational resources and computing time.

4.2. Implementation Details

We resize all images to 224×224, which is a standard resolution to use with ImageNet. We divide the image into 14×14 patches. We prompt the huge version of SAM for a multi-level output, obtaining 3 masks that also match the number of levels we use in the Agglomerator. The pico version of MaxViT [24] trained on ImageNet-1k by [26], was used to obtain the embeddings of the generated masks with an embedding vector size $D = 256$. We chose $\alpha_3 = 0.7$, $\alpha_2 = 0.2$, and $\alpha_1 = 0.1$ (i.e. the weights for the loss function at levels 2, 1, and 0 are 0.7, 0.2, and 0.1 respectively). We use the recently introduced lion [3] to train all the networks. We test both strategies to create masks for the objects and their parts, as demonstrated in Figure 2. We used lion optimizer for both training and pre-training, with a batch size of 512, learning rate $5e^{-4}$, and weight decay 0.075.

4.3. Results

Firstly, we experiment with different design choices in our pre-training strategy. We try masking without cropping vs masking with cropping as demonstrated in Figure 2. We achieve better validation accuracy when we embed the masked image without cropping the object vs when we crop the masked object, validation accuracy for both approaches is shown in Table 1. We hypothesize that this is due to the fact that we retain information about the object's position in the embedding.

Mask Strategy	Accuracy
with cropping	80.56%
without cropping	**89.5%**

Table 1. Validation Accuracy for different masking strategies

Encoder	Parameters (M)	Accuracy
Included	14.2	89.5%
Removed	**3.2**	**91.2%**

Table 2. Validation accuracy when using an encoder for Agglomerator's intermediate features vs not using an encoder.

We also try using a simple Multilayer Perceptron (MLP) encoder to obtain a representation for the Agglomerator levels versus using directly these levels and pushing it to agree with SAM mask embeddings (i.e using an identity function as the MLP encoder and directly minimize the loss between Z and M_e). The MLP consists of 4 fully connected layers with GELU [10] activation and layer normalization [1] applied to the output of each layer. The results are shown in Table 2. Therefore, we find that not cropping the masks before embedding while removing the encoder yields the best-performing variant for our pre-training strategy.

We subsequently proceed to a comparison between our best variant and the supervised contrastive loss originally proposed by the authors of Agglomerator. Additionally, we include a baseline of Agglomerator without any pre-training for the purpose of comparison. Our approach outperforms the contrastive pre-training approach by a huge margin of more than 30% while using approximately 54 times fewer parameters and no data augmentation. Augmenting the final classification layer with RandAugment in our approach resulted in no change in the performance. Augmenting in the pre-training phase also is not needed since we can directly use any amount of unlabeled data in our pre-training approach. This huge performance gap shows that Agglomerator pre-trained with supervised contrastive loss fails to deal with the complexity present in natural images. While the supervised contrastive loss does encourage embeddings at the last level (i.e., scene-level embeddings) to be closer for images of the same class and farther apart for different classes, it falls short in compelling the network to learn a granular breakdown of scenes into distinct objects and their constituent parts. Additionally, Agglomerator's authors noted that their pre-training method demands substantial computational resources when

Pre-Training	Augmentation	Parameters (M)	Accuracy
None	RandAugment	172	49.48%
Supervised Contrastive Loss	RandAugment	172	58.6%
SAM pre-training (ours)	**None**	**3.2**	**91.2%**

Table 3. Validation accuracy of different pre-training strategies for Agglomerator vs our best pre-training variant

Size (px)	Epochs	Optimizer	BS	Accuracy
128	5	Ranger	32	87.43%
192	5	Ranger	64	86.76%
256	5	Ranger	64	86.85%
224	**5**	**Lion**	**512**	**90.17%**
224	**18**	**Lion**	**512**	**91.28%**

Table 4. Comparison of our approach's validation accuracy (highlighted in bold) with several leaderboard submissions using Fastai's implementation of "Bag of Tricks for Image Classification with Convolutional Neural Networks" on the Imagenette dataset.

applied to datasets like ImageNet, and that more efficient pre-training strategies are needed. The final comparison between the pre-training strategies is summarized in Table 3.

We also included the performance of a convolutional-based solution, XResnet-50 [9], which incorporates a mixture of state-of-the-art tricks to enhance its performance, as a reference for comparison. This solution primarily relies on the Fastai library, a popular tool that provides access to various cutting-edge techniques for rapid application. The results from the leaderboard for Imagenette, where several users utilized Fastai and reported their findings, are included in the comparison. Some of the techniques these solutions employed are Random Erasing Data Augmentation [28], Mish [21], and BlurPool [27]. A summary of the results in comparison with our best variant is presented in Table 4.

5. Conclusion

We have introduced a novel method to incorporate prior knowledge about the compositional structure of the world by distilling this knowledge from SAM. We exploit the fact that Agglomerator explicitly represents the part-whole hierarchy in the scene and find a connection between this representation and the multi-level output of SAM by guiding both representations to agree in the latent space. With this approach, we pre-train a significantly smaller version of the originally proposed Agglomerator, approximately 54 times smaller, while achieving a remarkable improvement in validation accuracy. Our approach is appealing because of its extreme data efficiency; we do not require any class labels, not even from the training set, during the pre-training phase. Additionally, our method demonstrates great promise in generalization by eliminating the need for data augmentation.

Through our research, we aspire to pave the way for further advancements in achieving a more human-like vision in neural network architectures. By leveraging prior knowledge and enhancing data efficiency, we hope that our work will inspire and guide researchers in their quest to push the boundaries of artificial intelligence and create models that exhibit even greater cognitive capabilities.

References

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization, 2016. 6

[2] Yuxuan Cai, Yizhuang Zhou, Qi Han, Jianjian Sun, Xiangwen Kong, Jun Li, and Xiangyu Zhang. Reversible column networks. In *International Conference on Learning Representations*, 2023. 3

[3] Xiangning Chen, Chen Liang, Da Huang, Esteban Real, Kaiyuan Wang, Yao Liu, Hieu Pham, Xuanyi Dong, Thang Luong, Cho-Jui Hsieh, Yifeng Lu, and Quoc V. Le. Symbolic discovery of optimization algorithms, 2023. 6

[4] Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012. 3

[5] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 1

[6] Nicola Garau, Niccolò Bisagno, Zeno Sambugaro, and Nicola Conci. Interpretable part-whole hierarchies and conceptual-semantic relationships in neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13689–13698, 2022. 2

[7] Jeff Hawkins, Subutai Ahmad, and Yuwei Cui. A theory of how columns in the neocortex enable learning the structure of the world. *Frontiers in Neural Circuits*, 11, 2017. 2

[8] Jeff Hawkins, Marcus Lewis, Mirko Klukas, Scott Purdy, and Subutai Ahmad. A framework for intelligence and cortical function based on grid cells in the neocortex, 2018. 4

[9] Tong He, Zhi Zhang, Hang Zhang, Zhongyue Zhang, Junyuan Xie, and Mu Li. Bag of tricks for image classification with convolutional neural networks. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 558–567, 2018. 7

[10] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus), 2023. 6

[11] Geoffrey Hinton. Some demonstrations of the effects of structural descriptions in mental imagery. *Cognitive Science*, 3(3):231–250, 1979. 2

[12] Geoffrey E. Hinton. How to represent part-whole hierarchies in a neural network. *CoRR*, abs/2102.12627, 2021. 2, 3

[13] Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. *ArXiv*, abs/1503.02531, 2015. 3

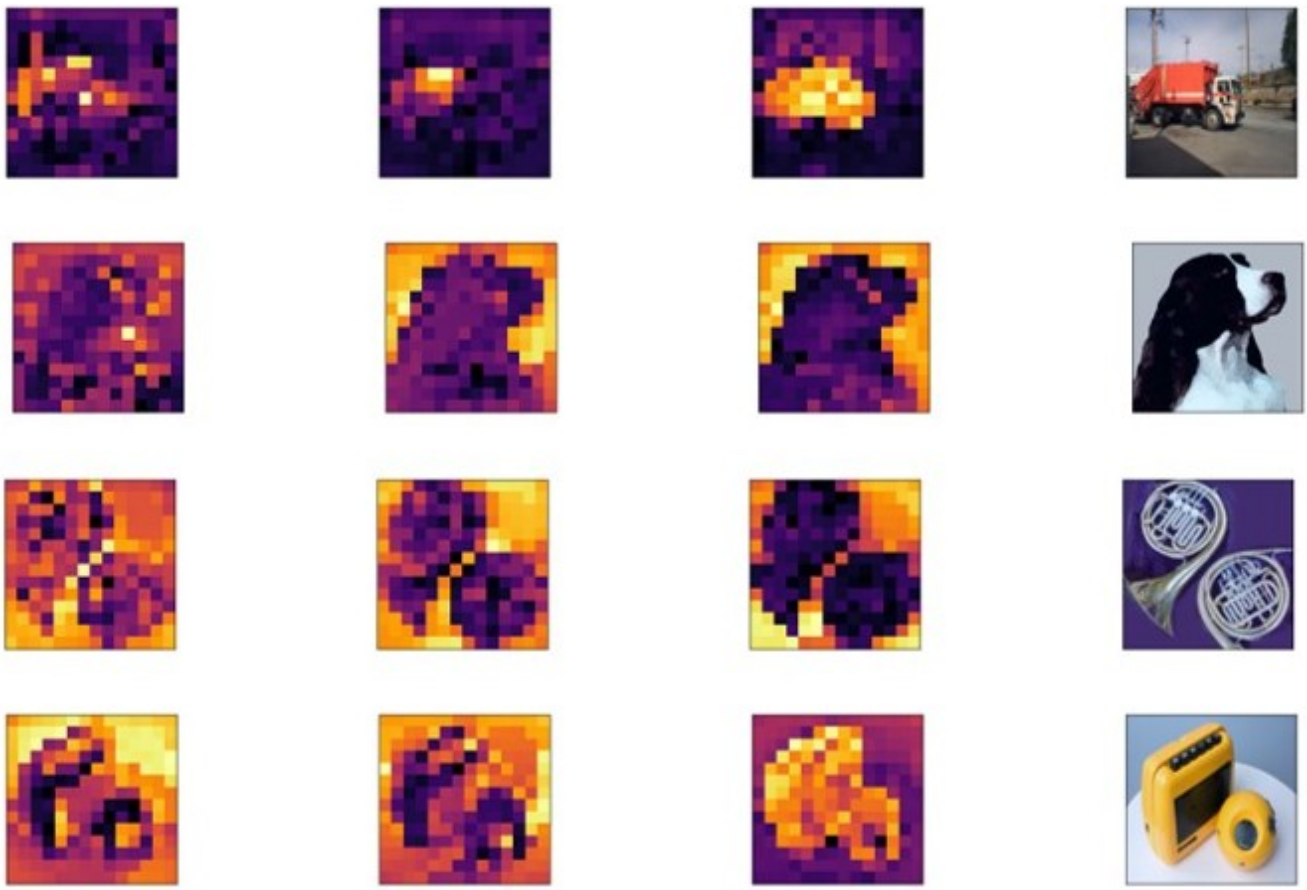

Figure 4. Visualization of the islands of agreement learned by Agglomerator using our pre-training strategy. The rightmost column shows the original image. Starting from the leftmost column we show the lowest level and then subsequently higher levels in the hierarchy.

[14] Jeremy Howard. Imagenette: A smaller subset of 10 easily classified classes from imagenet, March 2019. 2, 6

[15] Seungwook Kim and Hyo-Eun Kim. Transferring knowledge to smaller network with class-distance loss. In *International Conference on Learning Representations*, 2017. 3

[16] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. *arXiv:2304.02643*, 2023. 2

[17] Alex Krizhevsky. Learning multiple layers of features from tiny images. pages 32–33, 2009. 3

[18] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. 1

[19] Yann LeCun, Fu Jie Huang, and Léon Bottou. Learning methods for generic object recognition with invariance to pose and lighting. *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2:II–104 Vol.2, 2004. 3

[20] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 3

[21] Diganta Misra. Mish: A self regularized non-monotonic activation function. In *British Machine Vision Conference*, 2020. 7

[22] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. In *In Proceedings of ICLR*, 2015. 3

[23] Sara Sabour, Nicholas Frosst, and Geoffrey E. Hinton. Dynamic routing between capsules. NIPS'17, page 3859–3869, 2017. 3

[24] Zhengzhong Tu, Hossein Talebi, Han Zhang, Feng Yang, Peyman Milanfar, Alan Bovik, and Yinxiao Li. Maxvit: Multi-axis vision transformer. *ECCV*, 2022. 5, 6

[25] Reece Walsh, Mohamed H. Abdelpakey, Mohamed S. Shehata, and Mostafa M. Mohamed. Automated human cell classification in sparse datasets using few-shot learning. *Scientific Reports*, 12(1):2924, Feb 2022. 2

[26] Ross Wightman. Pytorch image models. https://github.com/huggingface/pytorch-image-models, 2019. 6

[27] Richard Zhang. Making convolutional networks shift-invariant again. In *ICML*, 2019. 7

[28] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 13001–13008. AAAI Press, 2020. 7