LowDINO - A Low Parameter Self Supervised Learning Model

Sai Krishna Prathapaneni,[1] **Shvejan Shashank** [2]

[1,2] Department of ECE, New York University
saiprathapaneni@nyu.edu, ssm10076@nyu.edu, https://github.com/saikrishna-prathapaneni/LowDINO

Abstract

This research aims to explore the possibility of designing a neural network architecture that allows for small networks to adopt the properties of huge networks, which have shown success in self-supervised learning (SSL), for all the downstream tasks like image classification, segmentation, etc. Previous studies have shown that using convolutional neural networks (ConvNets) can provide inherent inductive bias, which is crucial for learning representations in deep learning models. To reduce the number of parameters, attention mechanisms are utilized through the usage of MobileViT blocks, resulting in a model with less than 5 million parameters. The model is trained using self-distillation with momentum encoder (Grill et al. 2020) and a student-teacher architecture is also employed, where the teacher weights use vision transformers (ViTs) from (Oquab et al. 2023). The model is trained on the ImageNet1k dataset. This research provides an approach for designing smaller, more efficient neural network architectures that can perform SSL tasks comparable to heavy models.

Introduction

SSL has become popular in recent years, due to its ability to leverage large amounts of unlabeled data (Misra and van der Maaten 2019) and improve the performance of models mainly in downstream tasks with its ability to learn efficient representations of the data. SSL, in the field of Natural language processing(NLP), performs well in the paradigms of contrastive and generative setups due its limited dimensionality (Devlin et al. 2018) (Brown et al. 2020) and progressing into efficient models, on the other hand in computer vision (CV) a wide array of architectures have been proposed On extremely competitive benchmarks like as ImageNet, SSL approaches for computer vision have been able to match and in some cases even beat models trained on labeled dataset (**?**) and have been excellent KNN classifiers (Caron et al. 2021). SSL applied to multi-modalities such as audio and videos (Wickstrøm et al. 2022) with Frames dropped in between to predict. The objective of SSL is to mask a certain part of an image or a word in the text and try to predict the masked region. The objective of predicting the region leverages the model to capture complex relationships across various parts of an image or text without the necessity of providing any labels. In the research, we provide our results based on training from the data of CIFAR100 and ImageNet1k datasets. We use the methodologies of DINO where self-distillation of the model is done with the help of a momentum-encoded teacher network, In addition, we also perform the Distillation LowDINO models to calculate the KNN accuracies and Linear accuracies.

Literature Survey

One of the earliest SSL algorithms is developed by (Doersch, Gupta, and Efros 2015) essentially predicts the relative position of two randomly chosen patches in an image. This concept has been overtaken by "jigsaw" algorithms (Pathak et al. 2016b), which divide an image into an array of discontinuous patches and estimate their relative positioning and one includes to predict the rotation angle of an Image(Gidaris, Singh, and Komodakis 2018). Colorization-based SSL methods train to predict the original RGB values from grayscale (Zhang and Isola 2016). (Pathak et al. 2016a) trains a model to predict object motion in a single frame and then modify the resultant features to deal with single-frame detection challenges. (Agrawal, Carreira, and Malik 2015) predicts the ego-motion of a camera given multiple frames. (Owens et al. 2016) proposes to remove the audio track from a video, and then predict the missing sound. In Multiview variance, One of the most widely used methods for learning from unlabeled data involves labeling images with pseudo labels using a weakly trained network, then training using these labels in a conventional supervised manner (Lee 2013).

Though these contrastive-based techniques have gained interest and generated better representations, non-contrastive-based approaches where student-teacher-based networks in recent years have been showing comparative results to that of supervised and semi-supervised alternative paradigms.

The recent advancement with DINOv2 (Oquab et al. 2023), which is an extension from DINO (Caron et al. 2021) shows a promising path in the direction of SSL, with Imagenet1k top1 accuracy reaching $\approx 89\%$ showing a path to adopt. DINO extends from the extension of two other self-supervised learning algorithms, BYOL (Bootstrap Your Own Latent)(Mehta and Rastegari 2021a) and SimSiam (Simplified Self-Supervised Learning with Contrastive Pre-

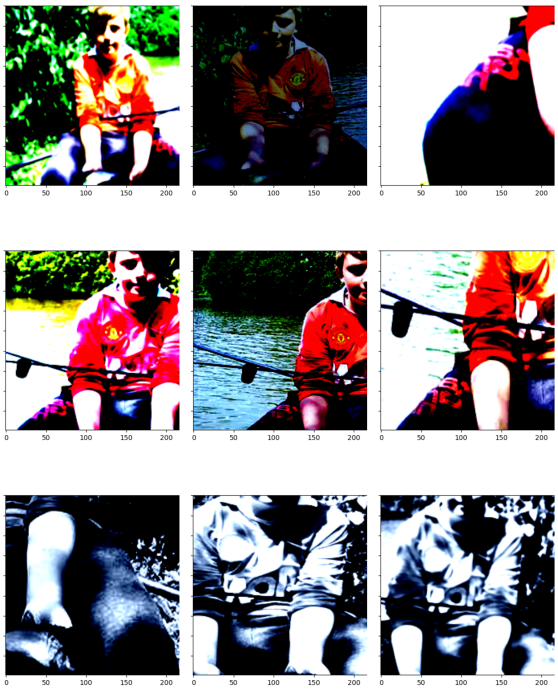

Figure 1: Crops generated per Image (resized to 224x224)

Model	Parameter	KNN
MobileVIT(LowDINO)	5.5M	53.05%
ResNet50(DINO)	23.5M	50.04%
ViT_S(DINOv2)	21.5M	55.64%
ResNet5M(LowDINO)	4.9M	34.3%
MobileNet(ImageNet)	3.5M	32.14%
ResNet50 (BYOL)	23.5M	50.14%

Table 1: KNN Accuracy on CIFAR10 with out finetuning

dictive Coding) (Chen and He 2020), is designed to work with discrete representations and relies on a momentum encoder and centering to avoid modal collapse. The relevance of the research paper lies in its exploration of the limitations of high-accuracy models based on supervised learning for image classification and segmentation applications, which require large amounts of labeled training data. The paper addresses the challenge of working with limited labeled data in certain applications, such as medical and astronomical imaging, by proposing the use of self-supervised learning (SSL). The paper highlights the benefits of SSL and provides evidence of the effectiveness of the DINOv2 model for image classification experiments on the ImageNet1k (Russakovsky et al. 2015) and extends itself from adopting properties from (Zhou et al. 2021). However, it also acknowledges the challenges posed by highly parameterized models like DINOv2, which have large computational and memory requirements and are unsuitable for low-powered devices. As such, the paper's relevance extends to the development of efficient models that can perform SSL tasks with lower computational requirements, making them more suitable for low-powered devices like embedded systems, mobile devices, and IoT devices. By addressing these challenges, the research paper contributes to the advancement of SSL in image classification, segmentation, and related applications, with the potential to impact various industries, including healthcare, robotics, and autonomous vehicles. One such attempt (Gao et al. 2022) Distilled Contrastive Learning (DisCo) by constraining student model to teacher embedding dimension. Our research work aims to explore ways to reduce the model's complexity without a significant decrease in performance as compared to recent SSL implementations. We trained a low parameter model with a similar training paradigm as DINO with MobileVIT blocks(Mehta and Rastegari 2021b) resulting in 4.9 million parameter backbone and distilling it into a series of smaller models such as MobileNets(Howard et al. 2017) and ResNets(Tomasev et al. 2022) (He et al. 2015).

Parameter	Value
batch_size	64
logging_freq	1
n_crops	4
n_epochs	100
out_dim	1024
optim	SGD
clip_grad	2.0
norm_last_layer	False
batch_size_eval	8
teacher_temp	0.04
student_temp	0.1
device_ids	[0]
pretrained	True
lr	0.0005
min_lr	1e-06
warmup_epochs	10
weight_decay	0.04
weight_decay_end	0.4
momentum_teacher	0.9995

Table 2: Hyper parameters for the Experiment for LowDINO. Showing experiment results with MobileVit small with 5.5M parameters

Methodology

The project aims to reduce the model size to less than 5 million parameters by considering the low-parameter architecture and reducing the size of DINO, and DINOv2 models considerably low. The architecture employed as shown in Figure 2 is with a replaceable backbone network, which was tested with MobileVit small networks and ResNet with 5M parameters approximately. We made use of properties from (Caron et al. 2021) and (Oquab et al. 2023) to construct our model. Table 2 shows the hyperparameters selected for the training of MobileViTs where the logging frequency is set to 1, indicating that the progress is logged after every training iteration. cilp_grad is the maximum value allowed for gradient elements during training to pre-

Model	KNN accuracy	Linear 10% data	Linear 30% data
MobileVit(LowDINO)	53.05%	61.93 %	66%
ResNet5M(LowDINO)	34.3%	39.12%	41.21 %
ResNet50(DINO)	50.04%	78.48%	79.72 %
Vit_S14(DINOv2)	55.04%	67.1%	69.9%
Mobilenet(INet1k)	32.14%	56.7%	59.9%

Table 3: Model fine tuned for 30 epochs on CIFAR10 with 10% and 30% of data

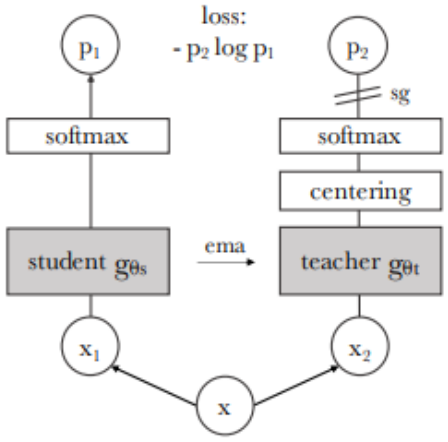

Figure 2: Architecture in DINO(Caron et al. 2021)

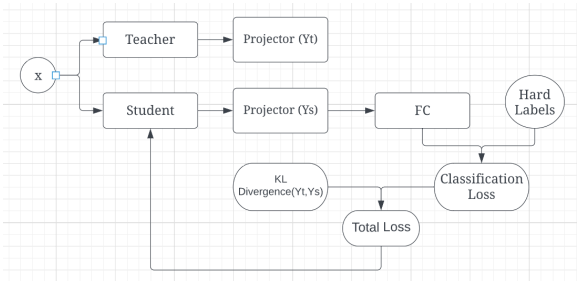

Figure 3: Architecture used to train via offline distillation of LowDINO models

vent gradient explosion. norm_last_layer is a flag indicating whether to apply normalization to the output layer of the head. In this case, the last layer normalization is set to False. batch_size_eval is the number of evaluation examples processed in one forward pass during each evaluation iteration. Here, the batch size for evaluation is set to 8. teacher_temp is used in the distillation process that controls the softness of the teacher's predictions. student_temp is similar to the teacher temperature but used for the student model. Here, a temperature of 0.1 is used. warmup_epoch is the number of initial epochs where the learning rate is gradually increased from the minimum learning rate to the initial learning rate. weight_decay_end is the weight decay value at the end of the training. Here, weight decay ends at 0.4. Teacher momentum is the momentum value used in the momentum encoder of the self-distillation process. Similar (Caron et al. 2021) we generated 4 and 10 crops for MobileVit and ResNet5m(saikrishna prathapaneni 2023) respectively of the same image i.e $x_1, x_2 \in crops(x)$ out of the n local crops generated 2 global crops consisting of scale between $(0.4, 1)$ and $n-2$ local crops with a scale between $(0.05, 0.4)$.where crops are generated based on the following augmentations.

- **Gaussian Blur** :50% of the time with a radius of 0.1 to 0.5 for local crops and 10% of the time in global crops.
- **Solarization**:20% of the times in global transforms.
- **RandomCrop**:$globalcrops \in (0.4, 1)$ and $localcrops \in (0.05, 0.4)$
- **Resize**:Bicubic interpolation to actual size of 224 x224 for local crops
- **Color jitter**: Applied for 80% of the time, changed parameters of color, contrast, brightness Grayscale(p=0.2)and saturation.
- **Flip**: Image flip horizontal and vertical flip of the images

After the generation of crops global crops are sent to the teacher model and local and global crops are sent to the student network.

$$\theta_{t+1} \leftarrow m\theta_t + (1-m)\theta_s, \tag{1}$$

$$\eta_t = \eta_{\min} + \frac{1}{2}(\eta_{\max} - \eta_{\min})(1 + \cos(\frac{T_c - t}{T_c}\pi)) \tag{2}$$

Weights are updated based on the teacher based on equation 1 where θ_t and θ_s are the parameters of the target (teacher) and online (student) networks respectively, and m is the momentum factor. This operation updates the target network parameters by taking a weighted average between the current target network parameters and the current online network parameters. The momentum parameter(m) is scheduled based on the cosine annealing schedule shown in equation 2 where $\eta_{\min}$ is the minimum learning rate, $\eta_{\max}$ is the maximum learning rate, t is the current training iteration, T_c is the total number of training iterations, and $\cos(\frac{T_c - t}{T_c}\pi)$ is a cosine function that varies smoothly between -1 and 1 over the course of the training schedule. To avoid the modal collapse between the teacher and student we introduce a centering and momentum encoder to update the teacher model from the student based on equation 1. Finally, the loss is a calculation between the probability distributions. unlike in DINO/DINOv2, we limited our dimensionality to 1024 to compute the softmax outputs.

In Table: 1 we show the KNN accuracy(Cunningham and Delany 2020) of two different backbone networks ResNet

with 4.9M parameters, and MobileVit with 5.5M parameters. We obtain comparable results for training the model from ImagNet1k constrained to 150K images of data, for 100 epochs, as shown in table 1 for our top performing model on the CIFAR10 dataset we obtain an accuracy of 53% outperforming DINO's ResNet50 model with 4times low parameters.

$$P_s(x)_i = \frac{\exp\left(\frac{g\theta_s(x)i}{\tau_s}\right)}{\sum_{k=1}^{K}\exp\left(\frac{g_{\theta_s}(x)_k}{\tau_s}\right)} \tag{3}$$

where $P_s(x)_i$ represents the probability of the i^{th} class for sample x under the softmax function s, $g_{\theta_s}(x)_i$ is the i^{th} element of the function $g_{\theta_s}(x)$, τ_s is the temperature parameter, and K is the output dimension. Loss is computed between the logits of the model of student and teacher models and updating just the student models at the end. At the end of the training we use backbone model for the downstream tasks.

Distillation

Model distillation is a popular technique for transferring knowledge from a large, complex model to a smaller, more efficient one. The process involves training a smaller model, called a student model, to replicate the behavior of a larger, more complex model, called a teacher model. The student model is trained on the same task and dataset as the teacher model, but with the added guidance of the teacher's predictions. The technique was first introduced in (Hinton, Vinyals, and Dean 2015). in the paper "Distilling the Knowledge in a Neural Network" and has since been widely adopted in the deep learning community. In a similar paradigm we introduce offline distillation (Gou et al. 2021) and SimKD (Chen et al. 2022) to distill into students where teacher weights are not updated but student weights are based on the KL divergence between logits of teacher and student networks are used. In addition, we introduce classification loss into the system with hard labels similar to equation 4 following show the networks used as a part of the distillation. Detailed architecture implemented for the distillation process is portrayed in the figure: 3. Figure 6 shows the accuracies generated of KNN after training with the architecture mentioned

$$L = (1-\alpha)CE(y,\hat{y}) + \alpha KL(T(y),T(\hat{y})) \tag{4}$$

where $CE(y,\hat{y})$ is the cross-entropy loss between the ground truth labels y and the predicted labels $\hat{y}$, $KL(T(y),T(\hat{y}))$ is the KL divergence between the softened probabilities $T(y)$ generated by the teacher model and the softened probabilities $T(\hat{y})$ predicted by the student model, and α is a hyper parameter that controls the balance between the two loss terms. The process implemented, Though the process involves the usage of cross entropy loss from the labels, we removed the hard loss from the procedure and just used soft labels to compute the loss and update the weights of students. following are the procedures used to train the model. we post a relative increase of $\approx 10\%$ for ResNet5m and approximately equal amount for MobileViT from the baseline training.

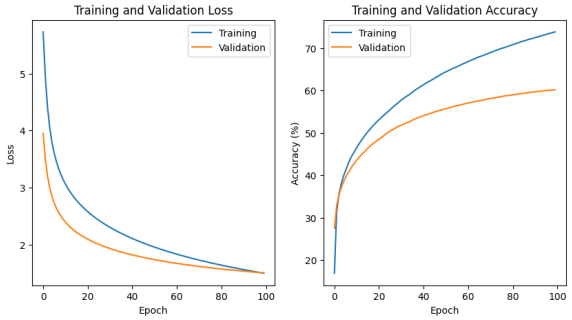

Figure 4: ResNet50 backbone(DINO) fine-tune on CIFAR 100, backbone frozen, FC updated

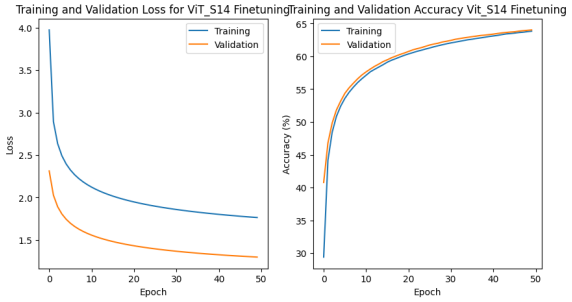

Figure 5: ViT_S14(DINOv2) fine-tune on CIFAR 100, backbone froze, FC updated

- DINO ResNet50 distills ResNet5M
- DINOv2 Vit_S distills MobileVit_S

Results and conclusion

The research aimed to show low parameter models can also learn better representations when trained with low dimensionality of only 1024 considered here and without a big drop in accuracy. The results compiled for ResNet5M and MobilVit_S model with self-distillation and offline distillation, from table 3, we can observe that although the best performing model is still DINOv2, with 21.5M parameters, the MobileVit(LowDINO) with 5.5M parameters is very close in terms of KNN evaluation, A similar trend can be observed after fine-tuning over 10 % and 30% data of CIFAR10, MobileVit(LowDINO) performs very close to DINOv2 while having almost 4 times less parameters and outperforming ResNet50 model of DINOv1 model. While these results seem promising in terms of parameter reduction, An extensive exploration needs to be done.

Future Scope

While the results posted here are promising with the amount of data used which is restricted to ImageNet1k with dataset sizes of 150k, 300k, and CIFAR100 datasets. It is worth exploring more further down the lane with similar paradigms of training by applying other loss functions and using larger dataset sizes including Imagenet22k/ LVM142M datasets

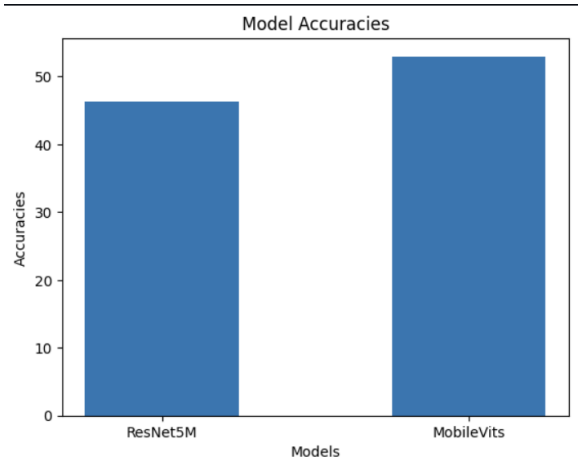

Figure 6: KNN accuracies generated for the models using distillation, Models use CIFAR100 Dataset to train

and aggresive abblation studies are required to conclude on the final models best performing models.

References

Agrawal, P.; Carreira, J.; and Malik, J. 2015. Learning to See by Moving. *CoRR*, abs/1505.01596.

Brown, T. B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; Agarwal, S.; Herbert-Voss, A.; Krueger, G.; Henighan, T.; Child, R.; Ramesh, A.; Ziegler, D. M.; Wu, J.; Winter, C.; Hesse, C.; Chen, M.; Sigler, E.; Litwin, M.; Gray, S.; Chess, B.; Clark, J.; Berner, C.; McCandlish, S.; Radford, A.; Sutskever, I.; and Amodei, D. 2020. Language Models are Few-Shot Learners. *CoRR*, abs/2005.14165.

Caron, M.; Touvron, H.; Misra, I.; Jégou, H.; Mairal, J.; Bojanowski, P.; and Joulin, A. 2021. Emerging Properties in Self-Supervised Vision Transformers. *CoRR*, abs/2104.14294.

Chen, D.; Mei, J.-P.; Zhang, H.; Wang, C.; Feng, Y.; and Chen, C. 2022. Knowledge distillation with the reused teacher classifier. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 11933–11942.

Chen, X.; and He, K. 2020. Exploring Simple Siamese Representation Learning. *CoRR*, abs/2011.10566.

Cunningham, P.; and Delany, S. J. 2020. k-Nearest Neighbour Classifiers: 2nd Edition (with Python examples). *CoRR*, abs/2004.04523.

Devlin, J.; Chang, M.; Lee, K.; and Toutanova, K. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *CoRR*, abs/1810.04805.

Doersch, C.; Gupta, A.; and Efros, A. A. 2015. Unsupervised Visual Representation Learning by Context Prediction. *CoRR*, abs/1505.05192.

Gao, Y.; Zhuang, J.-X.; Lin, S.; Cheng, H.; Sun, X.; Li, K.; and Shen, C. 2022. DisCo: Remedying Self-supervised Learning on Lightweight Models with Distilled Contrastive Learning. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVI*, 237–253. Springer.

Gidaris, S.; Singh, P.; and Komodakis, N. 2018. Unsupervised Representation Learning by Predicting Image Rotations. *CoRR*, abs/1803.07728.

Gou, J.; Yu, B.; Maybank, S. J.; and Tao, D. 2021. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129: 1789–1819.

Grill, J.; Strub, F.; Altché, F.; Tallec, C.; Richemond, P. H.; Buchatskaya, E.; Doersch, C.; Pires, B. Á.; Guo, Z. D.; Azar, M. G.; Piot, B.; Kavukcuoglu, K.; Munos, R.; and Valko, M. 2020. Bootstrap Your Own Latent: A New Approach to Self-Supervised Learning. *CoRR*, abs/2006.07733.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2015. Deep Residual Learning for Image Recognition. *CoRR*, abs/1512.03385.

Hinton, G.; Vinyals, O.; and Dean, J. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Howard, A. G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; and Adam, H. 2017. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *CoRR*, abs/1704.04861.

Lee, D.-H. 2013. Pseudo-Label : The Simple and Efficient Semi-Supervised Learning Method for Deep Neural Networks.

Mehta, S.; and Rastegari, M. 2021a. MobileViT: Lightweight, General-purpose, and Mobile-friendly Vision Transformer. *CoRR*, abs/2110.02178.

Mehta, S.; and Rastegari, M. 2021b. MobileViT: Lightweight, General-purpose, and Mobile-friendly Vision Transformer. *CoRR*, abs/2110.02178.

Misra, I.; and van der Maaten, L. 2019. Self-Supervised Learning of Pretext-Invariant Representations. *CoRR*, abs/1912.01991.

Oquab, M.; Darcet, T.; Moutakanni, T.; Vo, H.; Szafraniec, M.; Khalidov, V.; Fernandez, P.; Haziza, D.; Massa, F.; El-Nouby, A.; Assran, M.; Ballas, N.; Galuba, W.; Howes, R.; Huang, P.-Y.; Li, S.-W.; Misra, I.; Rabbat, M.; Sharma, V.; Synnaeve, G.; Xu, H.; Jegou, H.; Mairal, J.; Labatut, P.; Joulin, A.; and Bojanowski, P. 2023. DINOv2: Learning Robust Visual Features without Supervision. arXiv:2304.07193.

Owens, A.; Wu, J.; McDermott, J. H.; Freeman, W. T.; and Torralba, A. 2016. Ambient Sound Provides Supervision for Visual Learning. *CoRR*, abs/1608.07017.

Pathak, D.; Girshick, R. B.; Dollár, P.; Darrell, T.; and Hariharan, B. 2016a. Learning Features by Watching Objects Move. *CoRR*, abs/1612.06370.

Pathak, D.; Krähenbühl, P.; Donahue, J.; Darrell, T.; and Efros, A. A. 2016b. Context Encoders: Feature Learning by Inpainting. *CoRR*, abs/1604.07379.

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.;

Berg, A. C.; and Fei-Fei, L. 2015. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3): 211–252.

saikrishna prathapaneni, s. s. 2023. ResNET5M: Residual Network with less than 5M parameters. https://github.com/saikrishna-prathapaneni/resnet5M.

Tomasev, N.; Bica, I.; McWilliams, B.; Buesing, L.; Pascanu, R.; Blundell, C.; and Mitrovic, J. 2022. Pushing the limits of self-supervised ResNets: Can we outperform supervised learning without labels on ImageNet? *CoRR*, abs/2201.05119.

Wickstrøm, K.; Kampffmeyer, M.; Mikalsen, K. Ø.; and Jenssen, R. 2022. Mixing up contrastive learning: Self-supervised representation learning for time series. *Pattern Recognition Letters*, 155: 54–61.

Zhang, R.; and Isola, P. 2016. Colorful Image Colorization. *CoRR*, abs/1603.08511.

Zhou, J.; Wei, C.; Wang, H.; Shen, W.; Xie, C.; Yuille, A. L.; and Kong, T. 2021. iBOT: Image BERT Pre-Training with Online Tokenizer. *CoRR*, abs/2111.07832.