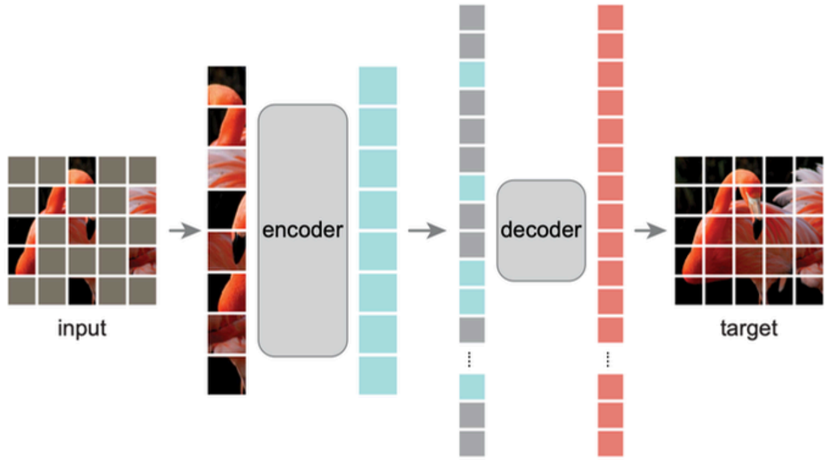
Project Presentation

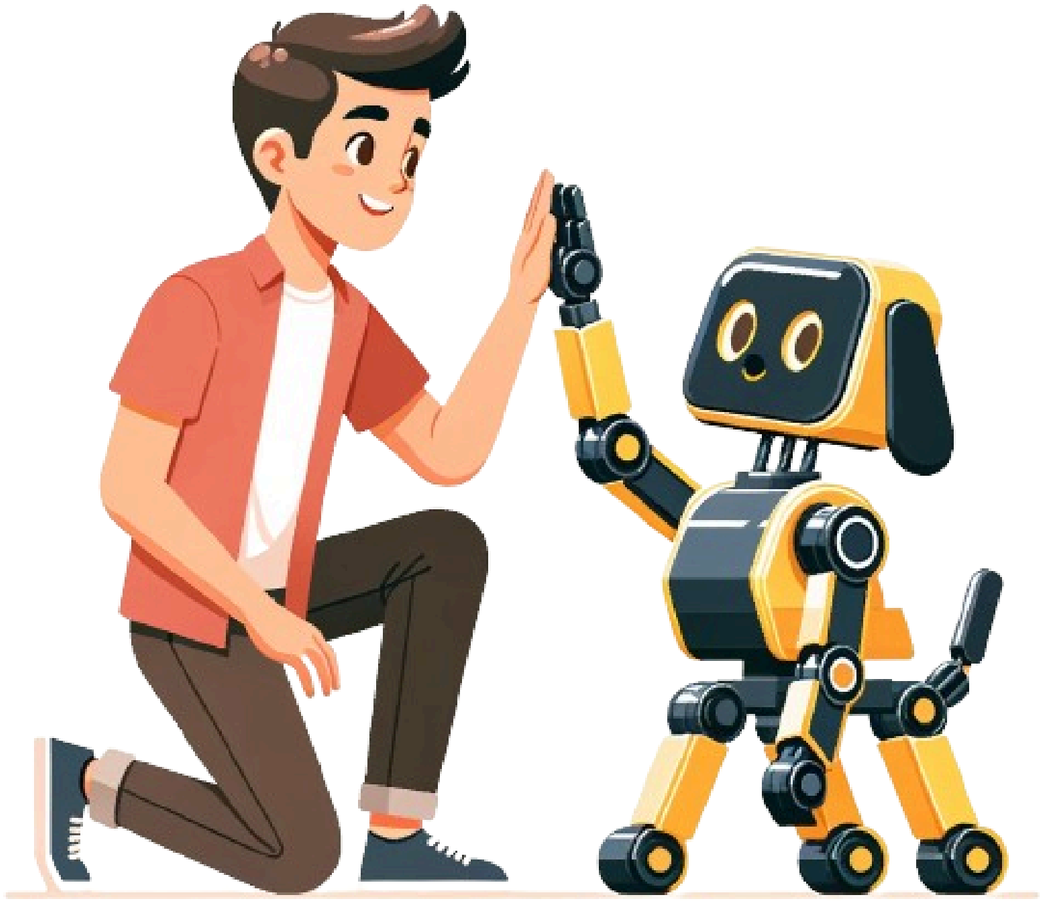


Enhancing Representation Learning in Masked Autoencoders by Focusing on Low-Variance Components

Turhan Can KARGIN

Outline

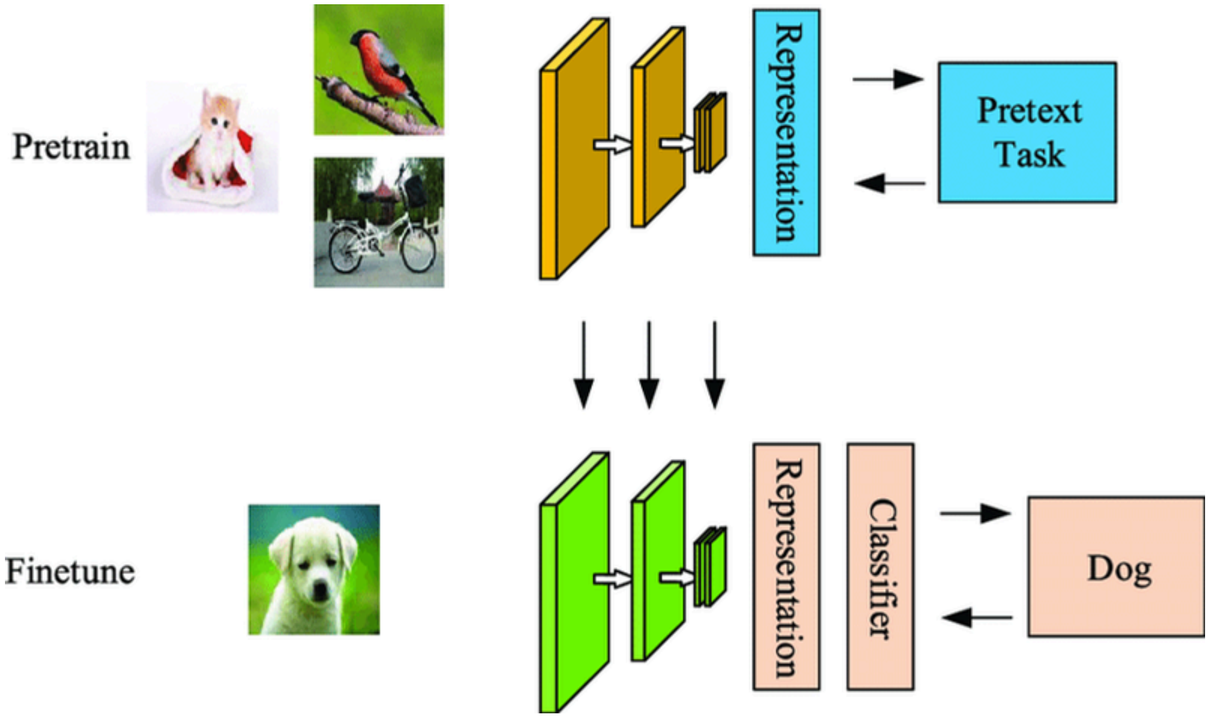
- Introduction
  - Background, Problem Statement, Objective
- Masked Autoencoders
- High and Low Variance Components of a Image
- Can we enhance the capability of the MAE to Learn Better Representations?
- Experiment Setup
- Results
- Conclusion



Created by DALL.E 3

Introduction

- **Background:** Self-supervised learning and Masked Autoencoders (MAE).
- **Problem Statement:** Limitation of MAEs focusing on high-variance components during reconstruction.
- **Objective:** Main goal of the project—improving MAE representation learning by focusing on low-variance image components.



Source: https://www.v7labs.com/blog/self-supervised-learning-guide

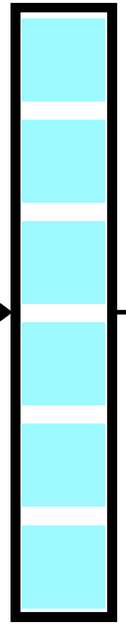
Masked Autoencoders

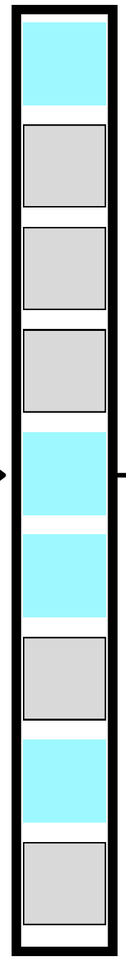
Loss function:
MSE between the reconstructed and original images in the pixel space

Input

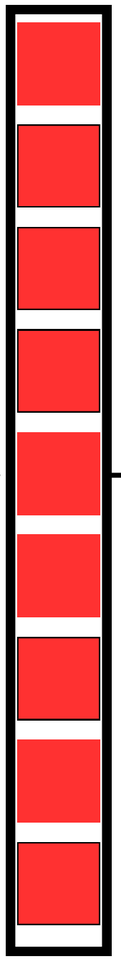
Assume %75 random subset of input image patches are masked out

Deeper ViT Encoder

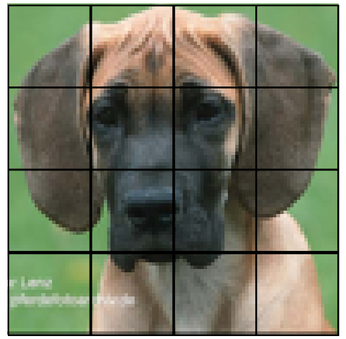




Shallower ViT Decoder



Output



Target image to reconstruct

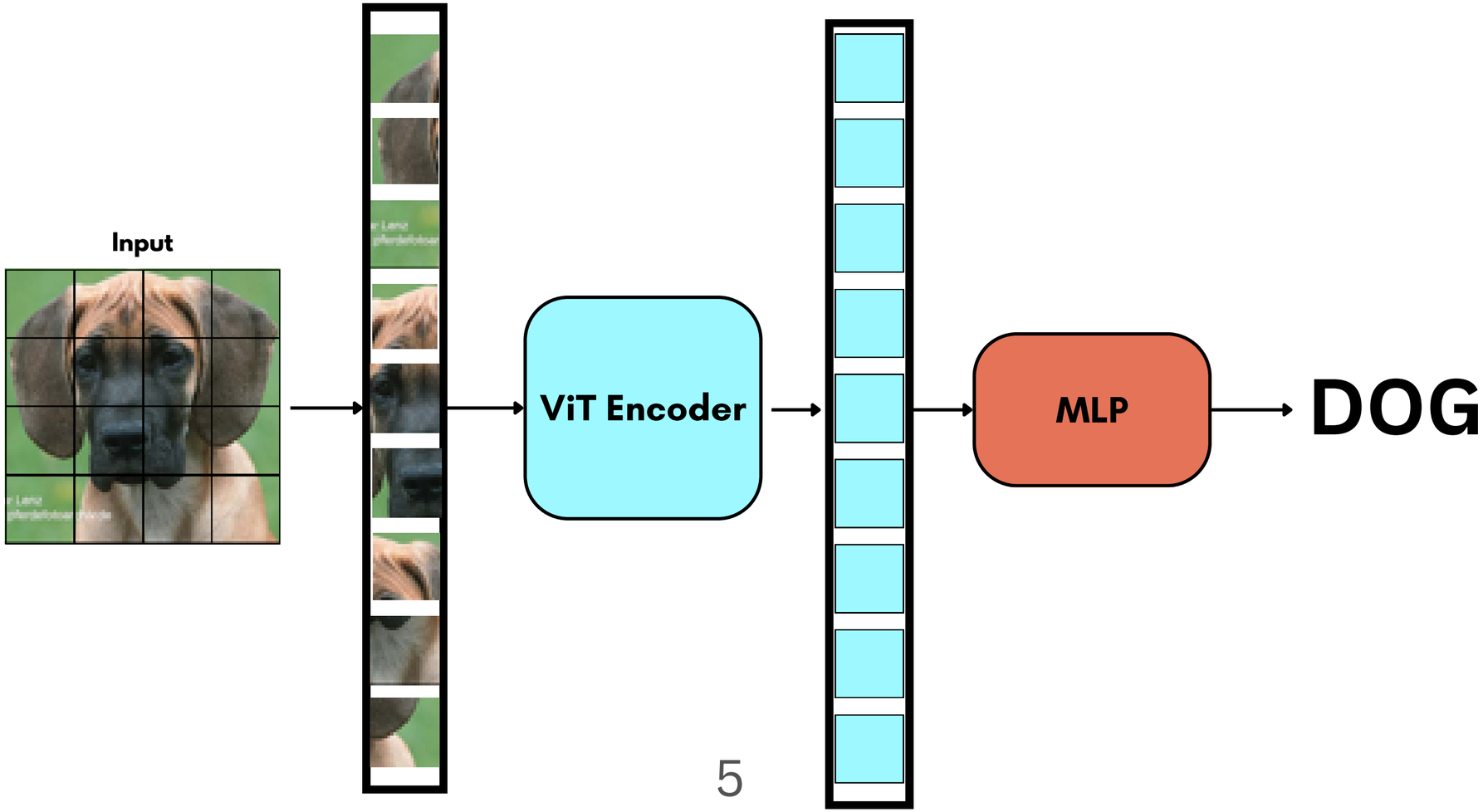
Learned Representations

Example sample from STL10 Dataset

Downstream Tasks

Classification via:

- fine-tuning
- linear probing (ViT Encoder parameters are frozen)



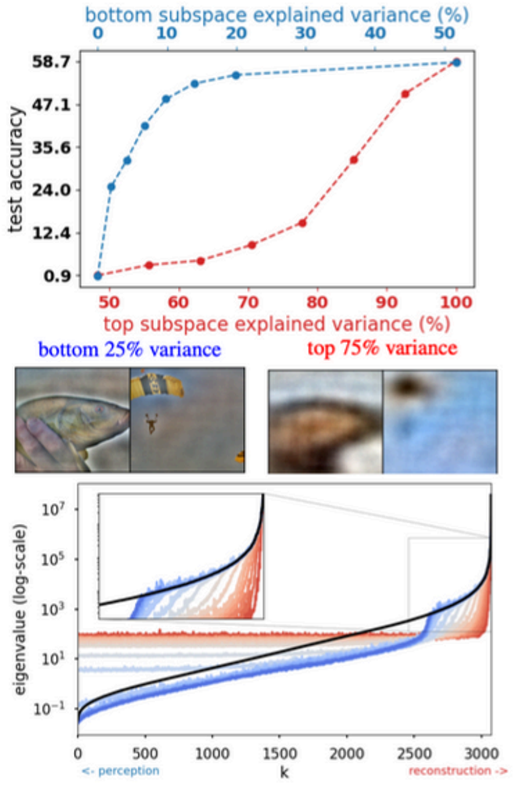
High and Low Variance Components of a Image

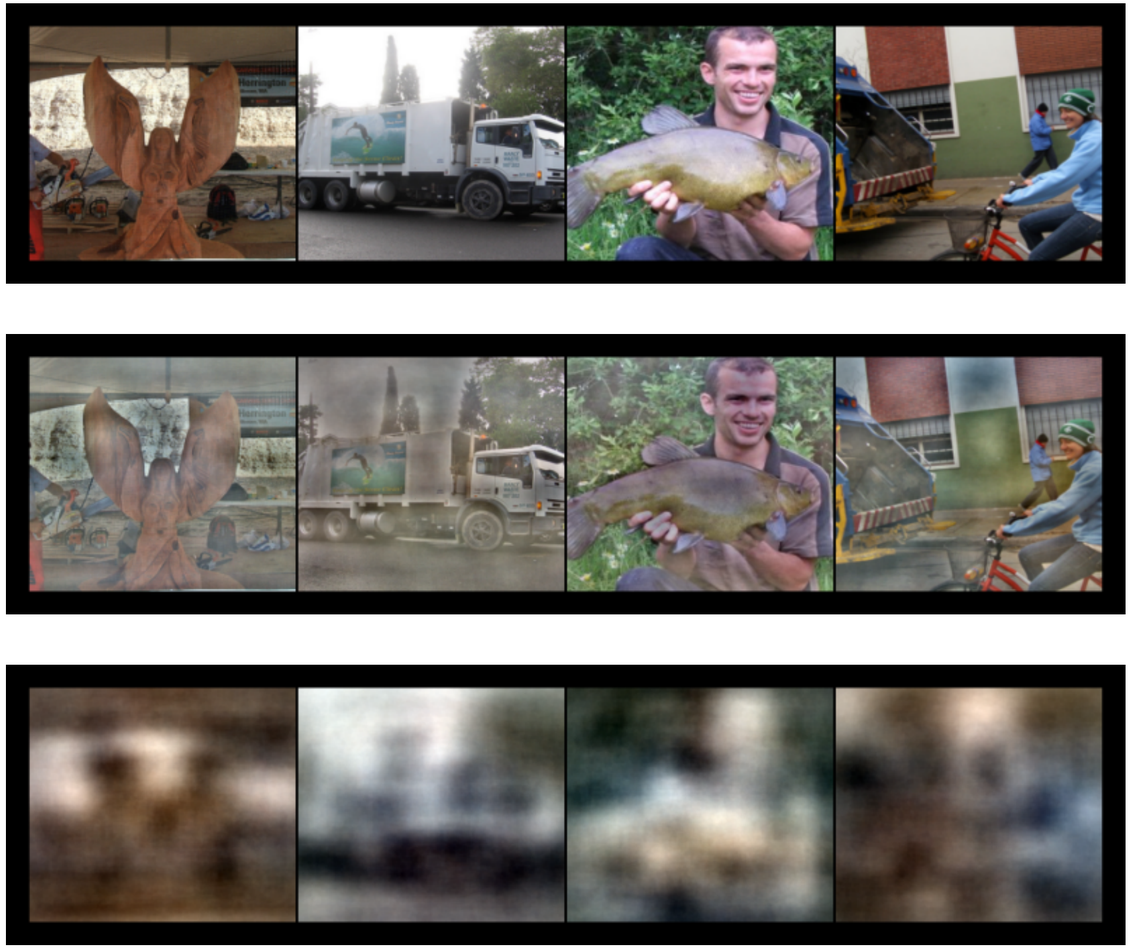
Learning by Reconstruction Produces Uninformative Features For Perception

Randall Balestriero [1] Yann LeCun [2]

Abstract

Input space reconstruction is an attractive representation learning paradigm. Despite interpretability of the reconstruction and generation, we identify a misalignment between learning by reconstruction, and learning for perception. We show that the former allocates a model's capacity towards a subspace of the data explaining the observed variance–a subspace with uninformative features for the latter. For example, the supervised TinyImagenet task with images projected onto the top subspace explaining 90% of the pixel variance can be solved with 45% test accuracy. Using the bottom subspace instead, accounting for only 20% of the pixel variance, reaches 55% test accuracy. The features for perception being learned last explains the need for long training time, e.g., with Masked Autoencoders. Learning by denoising is a popular strategy to alleviate that misalignment. We prove that while some noise strategies such as masking are indeed beneficial, others such as additive Gaussian noise are not. Yet, even in the case of masking, we find that the benefits vary as a function of the mask's shape, ratio, and the considered dataset. While tuning the noise strategy without knowledge of the perception task seems challenging, we provide first clues on how to detect if a noise strategy is never beneficial regardless of the perception task.

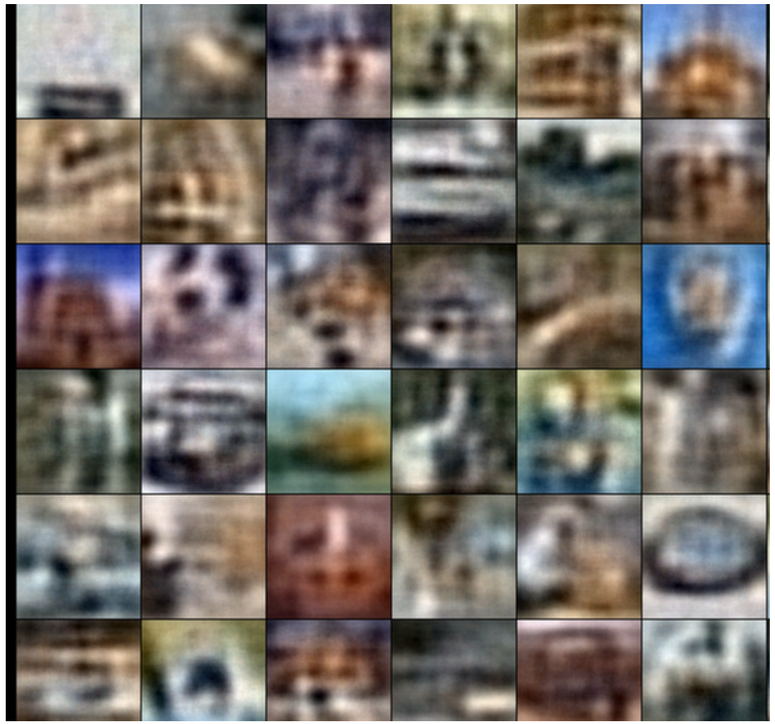


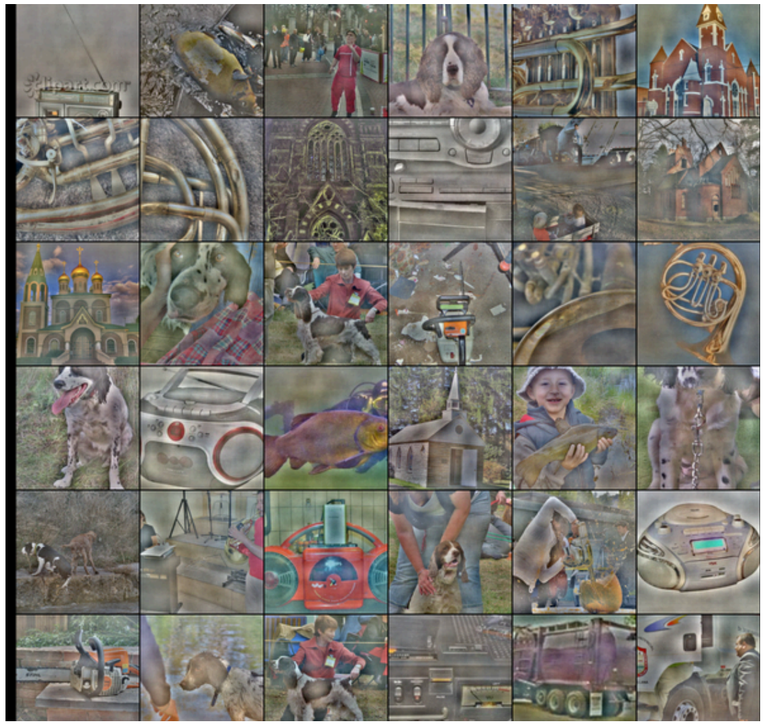


Original Set of Images (Top)
Images explains Bottom 25% PCA components (Middle)
Images explains Top 75% PCA components (Bottom)

Key Findings of the Paper:

- **Feature Alignment and Misalignment:**
  - Features learned by reconstructing the input data are not aligned with those required for downstream tasks. Specifically, they show that the principal components of the data (those that explain the most variance and are learned first in reconstruction tasks) are the least useful for classification tasks.
- **Order of Feature Learning**:
  - The study demonstrates that reconstruction-based models, such as autoencoders, learn features in an order that prioritizes reconstructing the input data accurately rather than focusing on features that are informative for perception tasks. This results in the most perceptually relevant features being learned last, which explains why these models often require long training times to perform well in tasks like classification.





Algorithm to Extract Top-Bottom PCA Components

Algorithm 1 Projecting Images onto Subspace Explaining a Percentage of Pixel Variance

Require: Image dataset X of shape (N, H, W, C), target variance percentage p

Flatten Images:
Flatten each image to a vector of size $D = H \times W \times C$

Center the Data:
Compute the mean vector $\mu = \frac{1}{N} \sum_{i=1}^{N} X_i$
Subtract the mean vector from each image vector: $X_{\text{centered}} = X - \mu$

Perform PCA:
Compute the covariance matrix $\Sigma = \frac{1}{N-1} X_{\text{centered}}^{\top} X_{\text{centered}}$
Perform eigen decomposition: $\Sigma = v_i \lambda_i v_i^{\top}$

Select the Desired Subspace:
Sort eigenvectors by eigenvalues in ascending order
Compute the cumulative variance explained by sorted eigenvectors
Select the number of eigenvectors k such that cumulative variance $\geq p$ (top subspace) or cumulative variance $\leq p$ (bottom subspace)
Form the projection matrix V_{top} or V_{bottom} using the selected eigenvectors

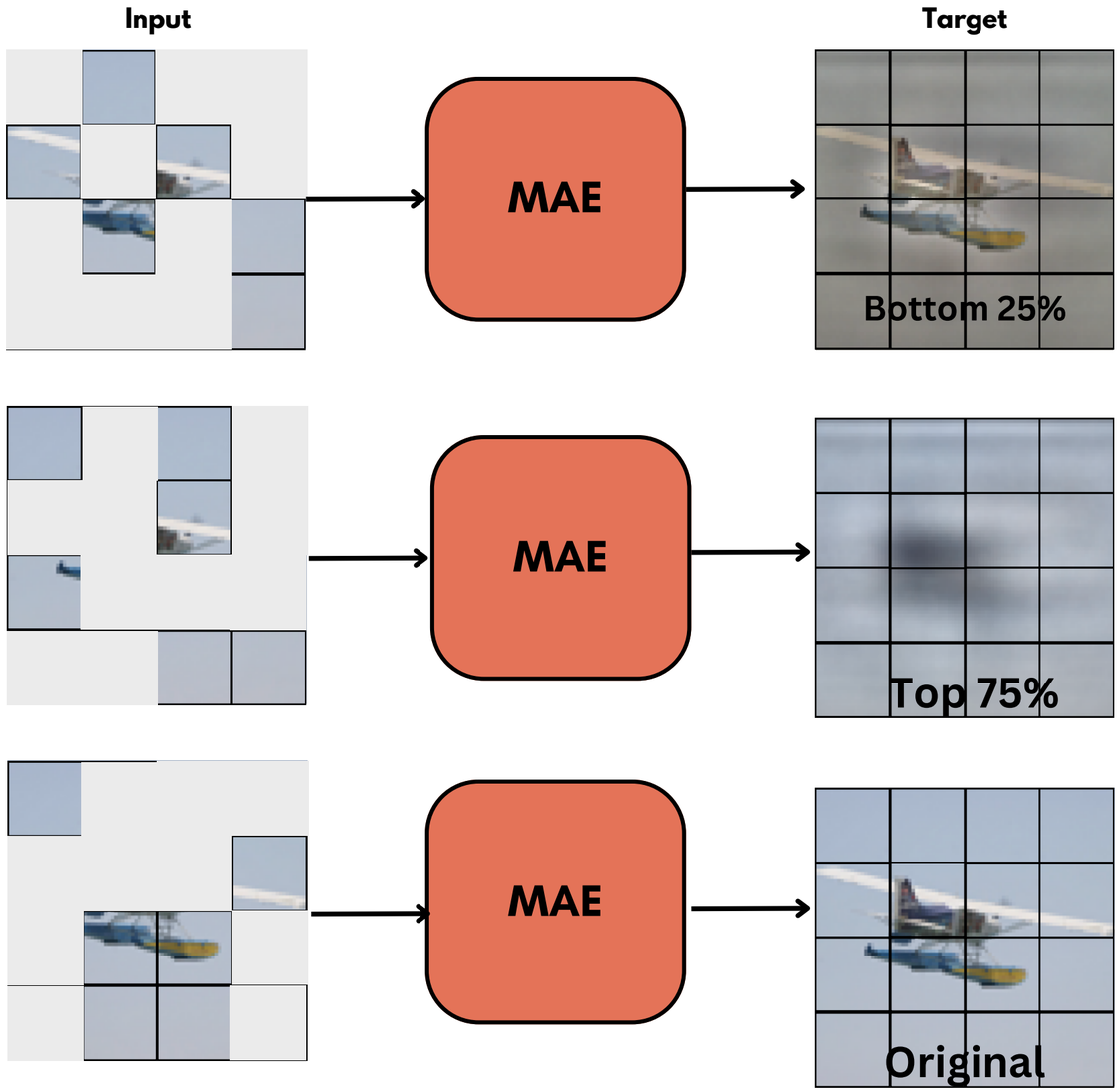
Project the Images:
Project each centered image vector onto the subspace: $X_{\text{top}} = X_{\text{centered}} V_{\text{top}}$
Or for the bottom subspace: $X_{\text{bottom}} = X_{\text{centered}} V_{\text{bottom}}$

Reconstruct:
Reconstruct the images: $\hat{X} = X_{\text{top/bottom}} V_{\text{top/bottom}}^{\top} + \mu$

Can MAE Learn Better Representations by Reconstructing Images With Filtered out High-Variance Components?

Methodology:

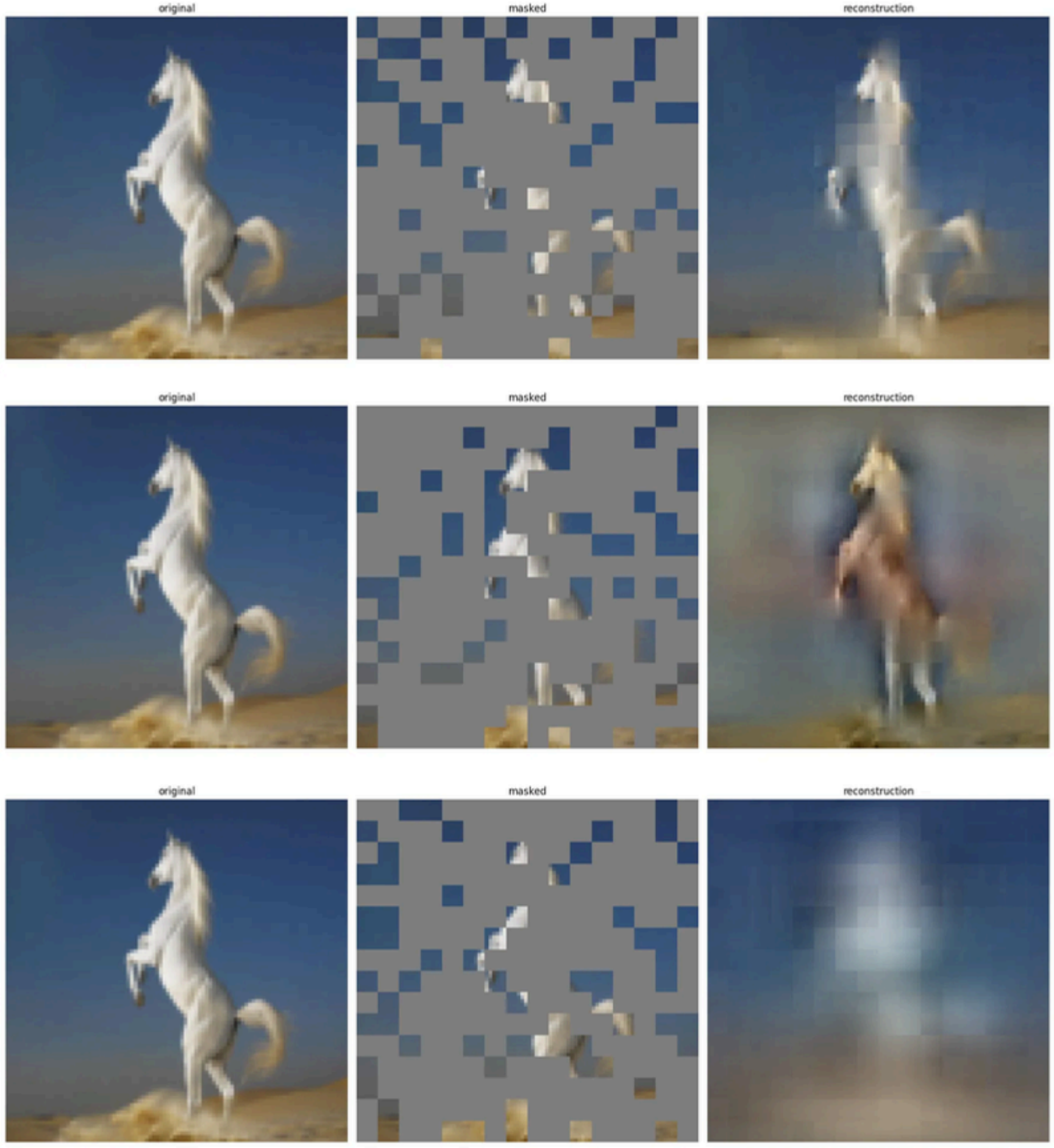
- **Step 1:** PCA Analysis
- **Step 2:** Data Preparation
- **Step 3:** MAE Training
  - Standard MAE reconstructing original images.
  - Modified MAE reconstructing low-variance and high-variance component images.
- **Step 4:** Evaluation
  - Validating the results using fine-tuning and linear probing.



MAE Demo - a Hugging Face Space by turhancan97

Discover amazing ML apps made by the community

🤗 huggingface

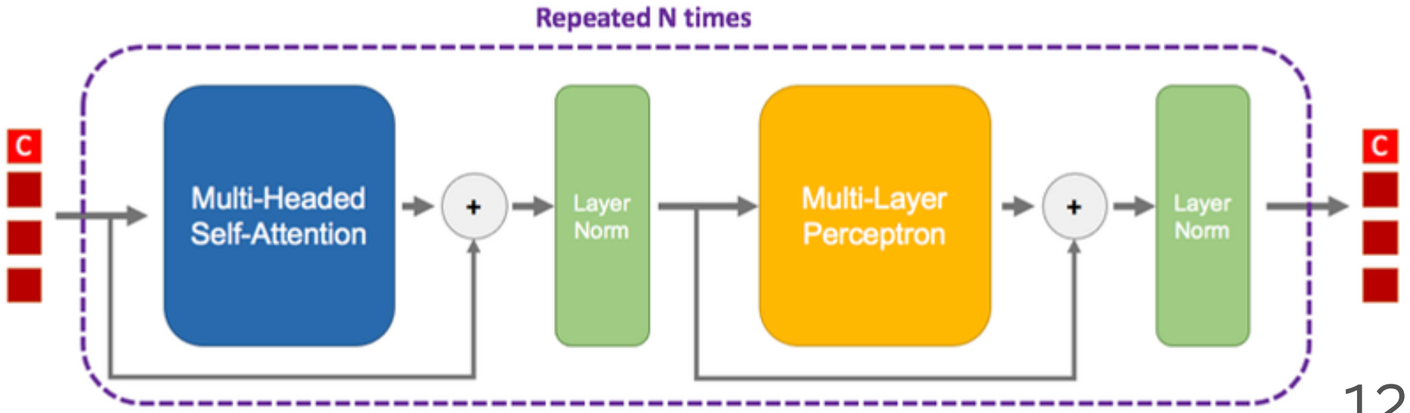


Experiment Setup

MAE Architecture and Training Details:

Standard ViT-Tiny Architecture

- **Image Size:**
  - 32 x 32 for CIFAR10
  - 96 x 96 for STL10
- **Patch Size:**
  - 2 x 2 for CIFAR10
  - 6 x 6 for STL10
- **Embedding Dimension:** 192
- **# encoder layer:** 12
- **# encoder head:** 3
- **# decoder layer:** 4
- **# decoder head:** 3
- **Mask Ratio:** 0.75



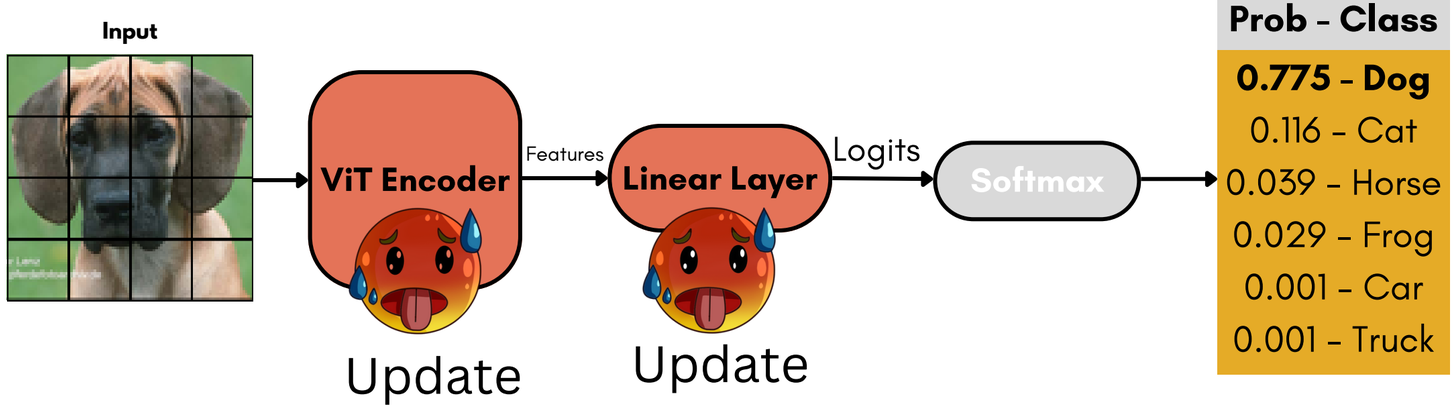
Single GPU: NVIDIA Tesla V100 32GB

config	value
optimizer	AdamW
base learning rate	1.5e-4
weight decay	0.05
optimizer momentum	β_1, β_2=0.9, 0.95
batch size	512
learning rate schedule	cosine decay
warmup epochs	40
epochs	400

Pre-training setting

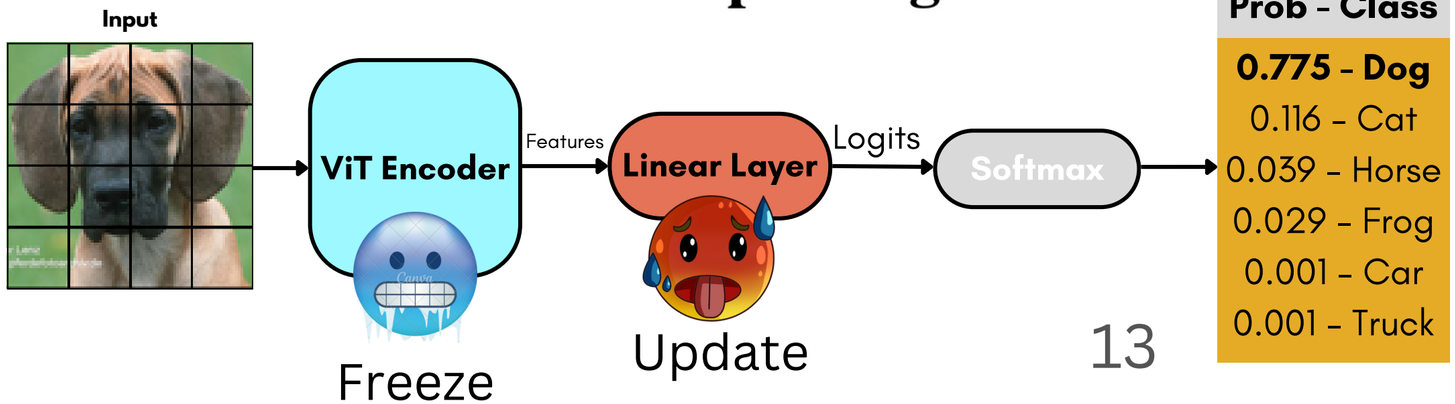
Experiment Setup

End-to-end fine-tuning



config	value
optimizer	AdamW
base learning rate	1e-3
weight decay	0.05
optimizer momentum	β_1, β_2=0.9, 0.999
batch size	1024
learning rate schedule	cosine decay
warmup epochs	5
training epochs	100

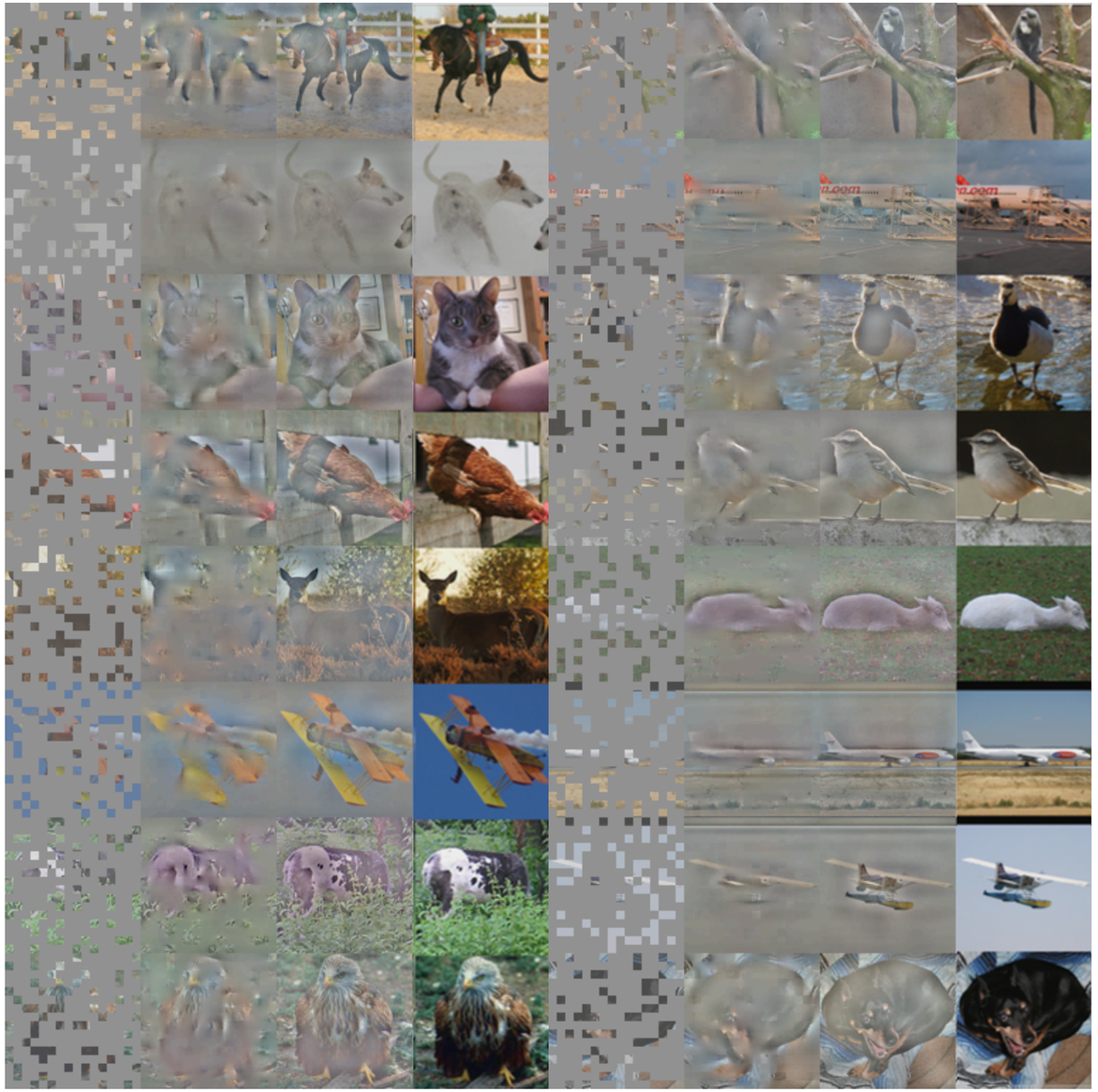
Linear probing



config	value
optimizer	SGD
base learning rate	0.1
weight decay	0
optimizer momentum	0.9
batch size	256
learning rate schedule	cosine decay
warmup epochs	10
training epochs	90

Results

Weights & Biases

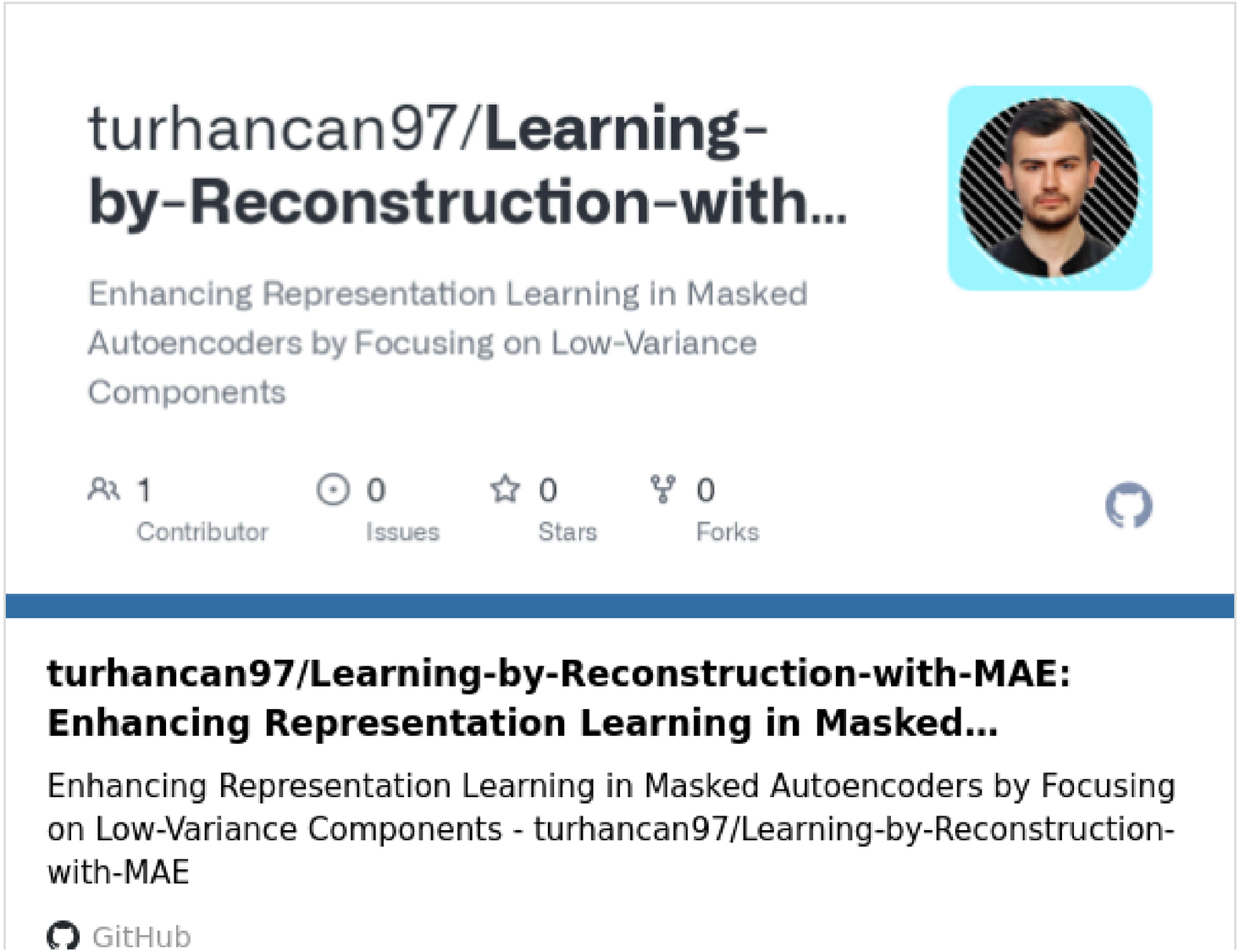


Results

Table 1: Validation Accuracy of Different Settings

Dataset	Target Image	Fine-Tuning	Linear Probing
STL10	Original Image	0.695	0.460
STL10	Without Pretraining	0.629	-
STL10	Components with Bottom 25 % Variance	**0.785**	**0.539**
STL10	Components with Bottom 10 % Variance	0.704	0.483
STL10	Components with Top 75 % Variance	0.649	0.398
STL10	Components with Top 60 % Variance	0.650	0.445
CIFAR10	Original Image	0.885	0.615
CIFAR10	Without Pretraining	0.813	-
CIFAR10	Components with Bottom 25 % Variance	0.902	**0.638**
CIFAR10	Components with Bottom 10 % Variance	**0.922**	0.636
CIFAR10	Components with Top 75 % Variance	0.894	0.607
CIFAR10	Components with Top 60 % Variance	0.880	0.560
CIFAR100	Original Image	0.671	0.370
CIFAR100	Without Pretraining	0.593	-
CIFAR100	Components with Bottom 25 % Variance	**0.682**	**0.373**
CIFAR100	Components with Top 75 % Variance	0.643	0.270

Codebase


turhancan97/Learning-by-Reconstruction-with...
Enhancing Representation Learning in Masked Autoencoders by Focusing on Low-Variance Components
1 Contributor
0 Issues
0 Stars
0 Forks
turhancan97/Learning-by-Reconstruction-with-MAE: Enhancing Representation Learning in Masked...
Enhancing Representation Learning in Masked Autoencoders by Focusing on Low-Variance Components - turhancan97/Learning-by-Reconstruction-with-MAE
GitHub

Conclusion

- **Interpretation of Results:** Discussion of how the results support the hypothesis.
- **Insights:** Why the modified MAE may have learned better representations.
- **Summary of Findings:** Recap the key findings from the experiments.

Source: Wikipedia Contributors. (2023, March 23). Cobot. Retrieved June 18, 2023, from Wikipedia website: https://en.wikipedia.org/wiki/Cobot

Project Presentation

Thank you for listening!

Turhan Can KARGIN