Unsupervised Pre-Training using Proximal Sensitive Error (PSE)

Ali Golmakani

Introduction

- **Unsupervised Image Feature Learning:** A key research area in computer vision, aiming to extract meaningful representations from unlabeled data.
- **Labeling Challenges:** Overcoming the limitations and costs associated with acquiring large-scale labeled datasets.
- **Autoencoders:** Neural network architectures designed for unsupervised learning, enabling the extraction of informative image features without the need for explicit supervision.
- **Research Objective:** Investigating the effectiveness of Proximal Sensitive Error (PSE) loss in enhancing unsupervised image feature learning.

You can find the codes here:
https://github.com/Louagyd/TUDelft_Interview_Unsupervised_Pretraining_with_PSE

Research Background

- **Unsupervised Pre-Training:** Leveraging vast amounts of unlabeled data to overcome the challenge of acquiring expensive and time-consuming labeled datasets. By pretraining an autoencoder on the unlabeled data, the model can capture essential image features and extract high-level representations, which can be further fine-tuned for specific tasks.
- **Downstream Tasks:** Once the autoencoders are pretrained fine-tuning allows them to adapt the prelearned features to be able to generalize better and achieve superior results on specific tasks.
- **Content-Based Feature Learning:** Traditional Autoencoders use the Mean Squared Error (MSE) loss function for image reconstruction. They can be enhanced by focusing on content-based feature learning. This approach aims to capture more meaningful representations by prioritizing the extraction of high-level image features related to the content rather than pixel-level reconstruction.

Research Idea - PSE loss

- **Goal:** Develop MSE to be more sensitive on localized errors (semantic errors)
- **Approach:** Using a Kernel Density Estimation (KDE) on the Residuals.

$$\forall x \in P, \qquad \mathcal{R}_x = |\hat{Y}_x - Y_x|, \qquad MSE := \mathbb{E}_x[\mathcal{R}_x^2], \qquad PSE := \mathbb{E}_x[\hat{f}_x(\mathcal{R}_x)^2]$$

- For each pixel we can assume we have a weighted sample from the Residual
- Then we can expand the PSE formula by inserting the Weighted Kernel Density Estimation

$$\hat{f}_x(\mathcal{R}_x) = WKDE(x) = \sum_{x'} \mathcal{R}_{x'}.\, K_\sigma(x - x') = (\mathcal{R} * K_\sigma)_x$$

Validating Assumptions

$$MSE := \mathbb{E}_x[\mathcal{R}_x^2], \qquad PSE := \mathbb{E}_x[\hat{f}_x(\mathcal{R}_x)^2] = \mathbb{E}_x[(\mathcal{R} * K_\sigma)_x^2]$$

- **Kernel function:** The choice of the kernel function can vary depending on the task

- **Weighting effect:** Regions with higher density correspond to areas where errors are more concentrated, indicating potential semantic errors. Conversely, regions with lower density are often associated with syntactic errors or noise.

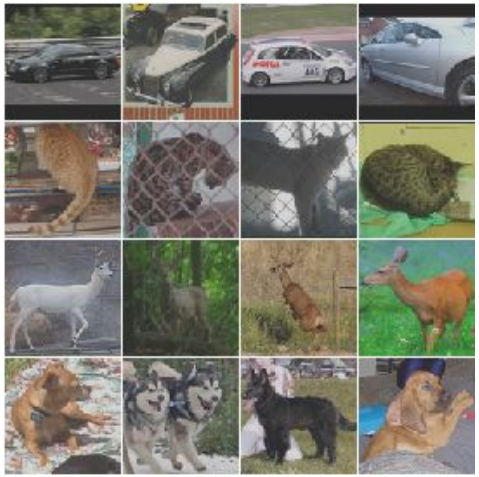
- **Robustness:** Outliers tend to have lower densities and therefore contribute less to the overall PSE loss. This robustness helps prevent outliers from excessively influencing the training process, allowing the model to focus on more representative and common patterns in the data.

Empirical Experiments

- **Dataset:** We used **SLT10** dataset consisting 96x96 CIFAR10-like images in 10 classes
  - **Unlabeled Data:** 100,000 samples
  - **Labeled Data:** 5000 samples (500 for each class)
  - **Test Data:** 8000 samples (800 for each class)
- **Pre-Training:** Training Autoencoders using both PSE and MSE loss (unlabeled data)
  - **CNN Encoder**: Includes 5 Conv2D layers with latent size 256
  - **ResNet Encoder:** Includes 4 ResNet blocks with latent size 1024
  - **CNN Decoder:** Includes 5 ConvTranspose2D layers
- **Downstream:** Fine-tune pretrained encoders for image classification (labeled data)

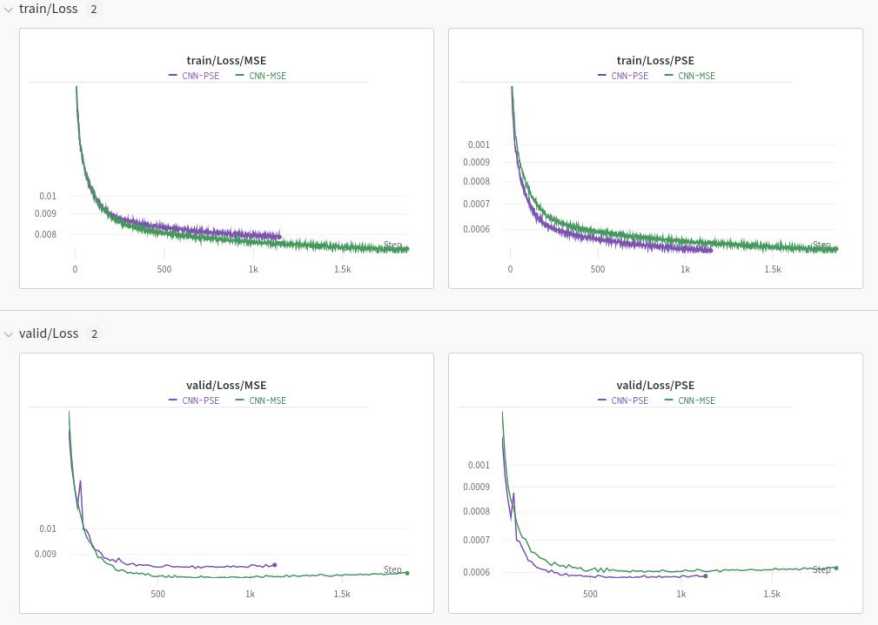
Unsupervised Pre-Training

- **Data Augmentation:** Small magnitude
- **Optimizer:** Adam, lr=1e-4
- **Batch size:** 128
- **Train-Validation split:** 85%
- **Device:** NVIDIA GeForce RTX 3060



Dataset Samples

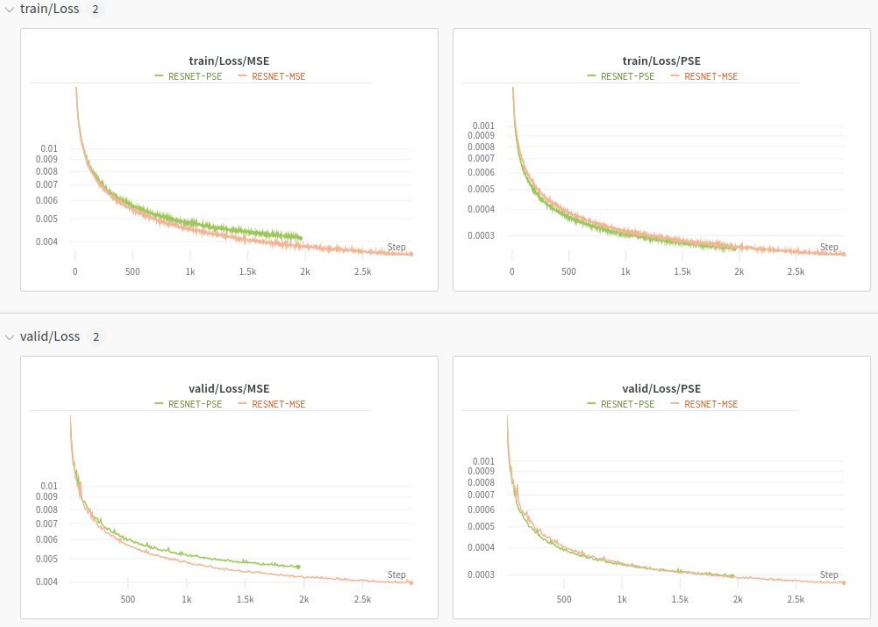
CNN Encoder



Train-Validation Loss

Image Reconstruction

ResNet Encoder



Train-Validation Loss

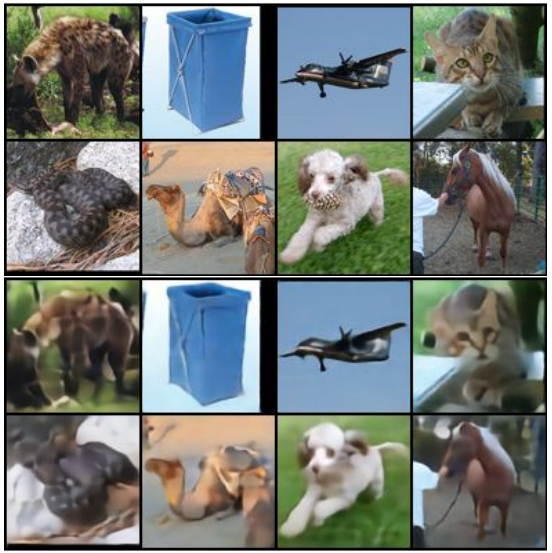
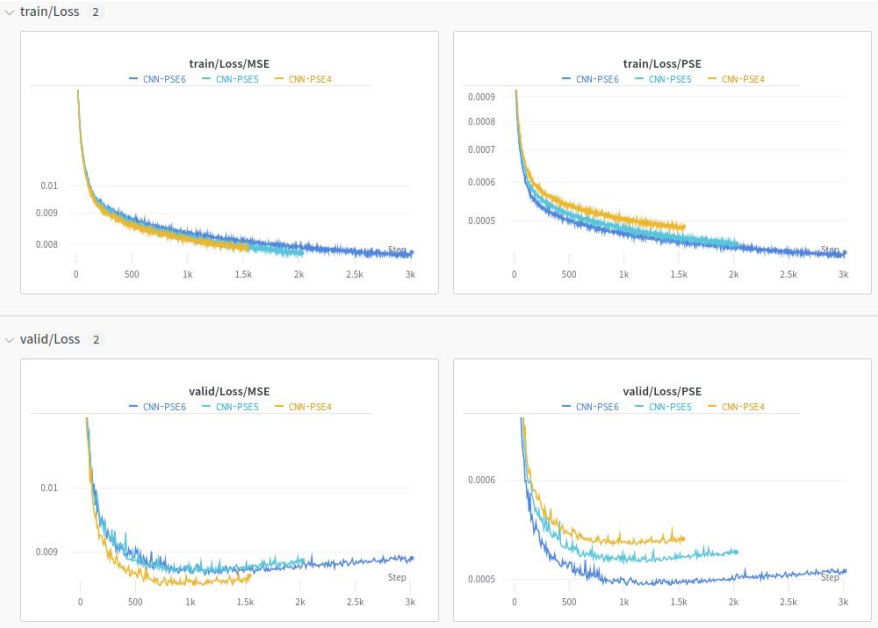


Image Reconstruction

Multiple Kernel Bandwidths



Train-Validation Loss

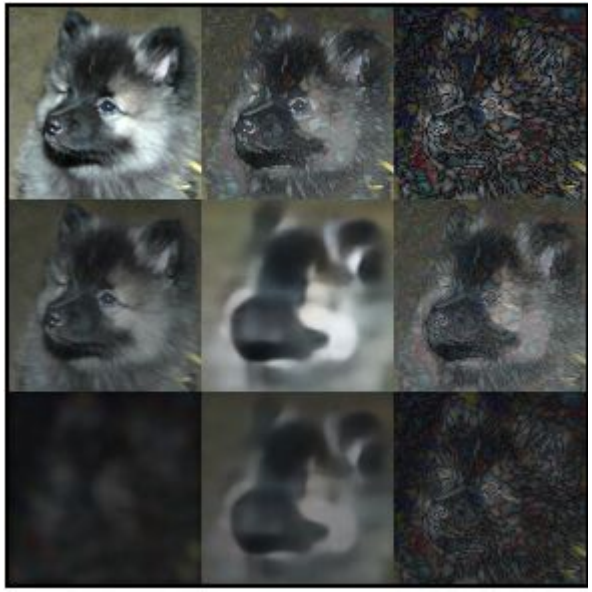


Image Reconstruction

Downstream

- **Data Augmentation:** High magnitude
- **Classifier:** 2 Fully-Connected Layers
- **Batch size:** 128
- **Train-Validation split:** 80% Cross-Validation
- **Device:** NVIDIA GeForce RTX 3060

Augmented Samples

Fine-Tuning Performances

Model Name	Ensemble Acc, Top2, Top4	Test 1 Acc, Loss	Test 2 Acc, Loss	Test 3 Acc, Loss	Test 4 Acc, Loss	Test 5 Acc, Loss
CNN-MSE	**48.96,** 68.90, 87.55	45.37%, 1.4847	47.26%, 1.4630	46.02%, 1.4819	45.21%, 1.4973	46.47%, 1.4792
CNN-PSE-3	**51.26,** 70.72, 88.71	47.74%, 1.4428	49.12%, 1.4299	48.95%, 1.4087	48.12%, 1.4223	49.54%, 1.4306
CNN-PSE-4	**51.43,** 70.88, 88.82	49.20%, 1.4144	48.96%, 1.4187	48.14%, 1.4232	47.88%, 1.4385	49.48%, 1.4367
CNN-PSE-5	**51.52,** 71.47, 89.13	48.96%, 1.4264	48.31%, 1.4594	48.62%, 1.4124	48.52%, 1.4442	49.57%, 1.4466
CNN-PSE-6	**50.73,** 71.15, 88.76	48.31%, 1.4364	48.95%, 1.4166	48.56%, 1.4183	47.89%, 1.4114	48.65%, 1.4313
ResNet-MSE	**54.40,** 72.66, 89,75	54.41%, 1.3184	53.13%, 1.3214	52.68%, 1.3193	53.63%, 1.3065	53.78%, 1.3001
ResNet-PSE-3	**56.45,** 74.53, 91.20	54.24%, 1.2749	54.90%, 1.2658	55.11%, 1.2500	56.03%, 1.2450	55.16%, 1.2634

Freeze Encoders Performances

Model Name	Ensemble Acc, Top2, Top4	Test 1 Acc	Test 2 Acc	Test 3 Acc	Test 4 Acc	Test 5 Acc
CNN-MSE	**28.15,** 44.90, 68.30	27.78%	27.63%	27.47%	27.16%	26.85%
CNN-PSE-3	**34.05,** 52.73, 76.67	33.32%	32.56%	32.42%	31.78%	33.22%
CNN-PSE-4	**34.52,** 52.67, 76.28	33.57%	33.54%	33.73%	32.61%	32.16%
CNN-PSE-5	**34.16,** 52.88, 76.68	33.05%	32.44%	33.35%	32.87%	33.05%
CNN-PSE-6	**33.30,** 51.62, 75.51	31.72%	32.56%	32.60%	31.80%	31.80%
ResNet-MSE	**37.07,** 54.70, 77.57	36.10%	33.67%	35.18%	36.24%	35.49%
ResNet-PSE-3	**39.38,** 57.47, 79.48	36.45%	36.07%	39.47%	36.58%	37.14%

Thank You