Self-supervised visual representation learning

(SimCLR & Deep Clustering)

African Master's in Machine Intelligence, AIMS-Senegal

June 2, 2024

Team Members:

1. Dzodzoenyenye Adjowa Senanou
2. Prince Mensah
3. Najlaa Mohamed
4. Leonard Sanya

Outline

Introduction

Understanding Learning Paradigms

- **Supervised Learning**: Relies on labeled data to learn the mapping between input and output.
- Aims to predict output labels accurately based on input features.

- **Unsupervised Learning**: Discovers the underlying structure or patterns in data without labeled examples.
- Aims to extract meaningful representations and relationships from data.

Self-Supervised Learning (SSL)

Why another paradigm?

Requirement: We want a model that learns from large amount of data and does various tasks.

- Supervised Learning: Requires abundant labeled data, which is often scarce and costly to acquire.
- Unsupervised Learning: Learns representations suitable for clustering and dimensionality reduction, but lacks the capability for tasks like classification, segmentation, and object detection.

Self-supervised learning

- Self-supervised learning harnesses abundant unlabeled data for training, yielding rich representations aligned with downstream tasks.
- Self-supervised models employ pretext tasks and contrastive learning to generate labels from data itself and foster representation learning by distinguishing between positive and negative samples.

Self-Supervised learning with Pretext tasks

Pretext tasks in computer vision are objectives used in self-supervised learning to train models without labeled data.

- Image in-painting: predicting missing parts of images, beneficial for tasks like image classification or object detection because it learns the spatial relationships
- Rotation prediction, predicting rotation angles of images, useful for tasks like pose estimation because it leans the shape's direction in-variance.

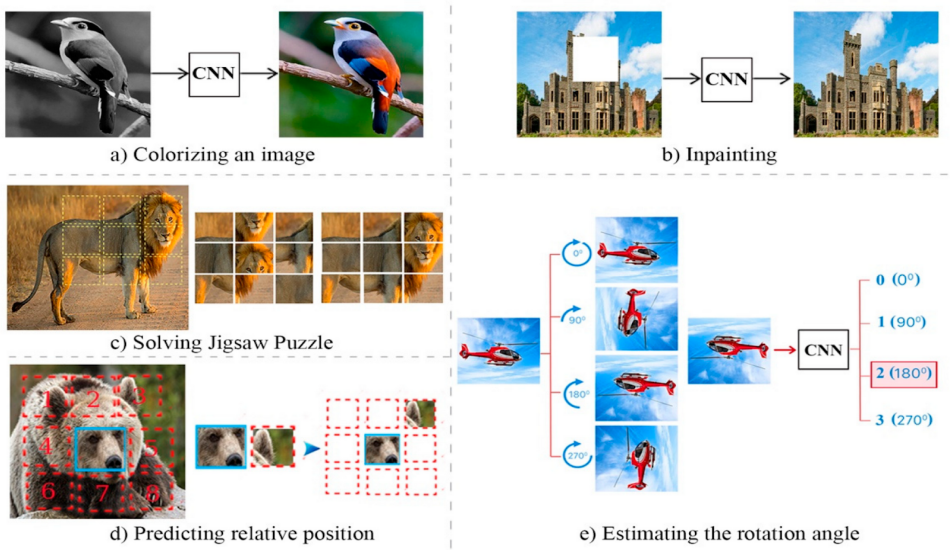


Figure: Examples of Pretext Tasks

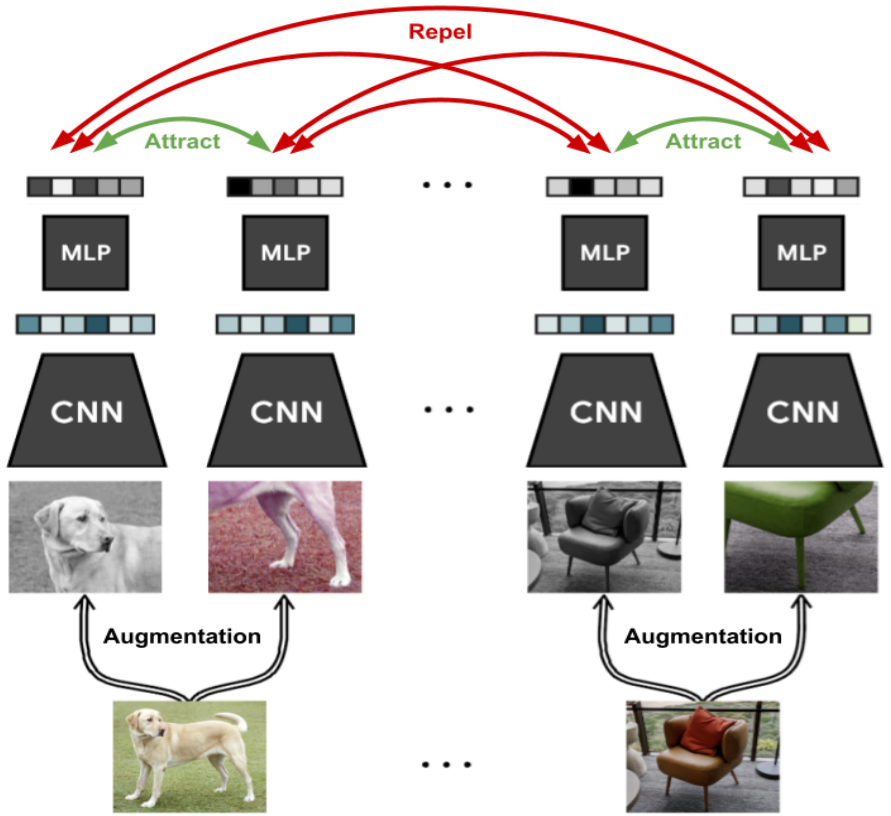
1

[1] source: `https://www.mdpi.com/1099-4300/24/4/551`

SimCLR: Simple Contrastive Learning Representations

It's a framework which aims to learn rich, high-quality representations of data.

- Generate Augmented Views
- Feature Extraction
- Projection Head
- Contrastive Loss
- Fine-Tuning



[a]

[a] source: https://research.google/blog/advancing-self-supervised-and-semi-supervised-learning-with-simclr/

How does SimCLR Solve the Problem?

1. Data Augmentation: SimCLR uses a combination of simple augmentations such as random cropping, color distortion, and Gaussian blur to create multiple views of each image. This encourages the model to learn consistent representations under different transformations.

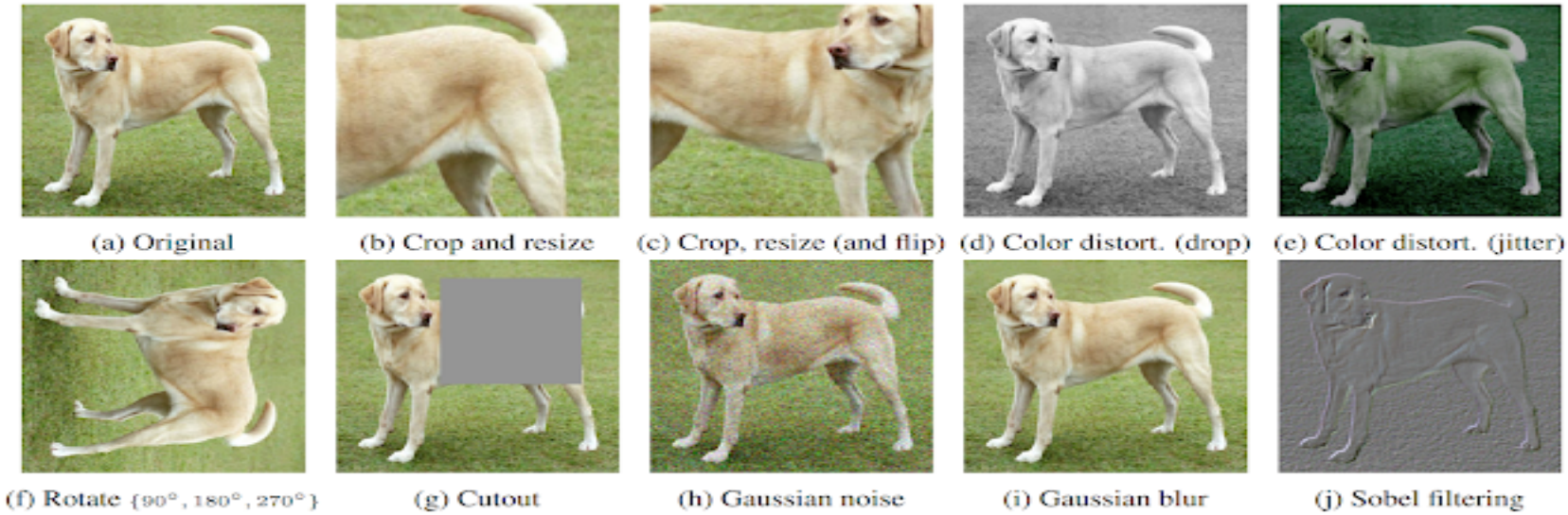


Figure: Data Augmentation examples

2

[2] source: https://research.google/blog/advancing-self-supervised-and-semi-supervised-learning-with-simclr/

2. Contrastive Loss :The framework uses a contrastive loss function to maximize agreement between different augmented views of the same image and minimize agreement between views of different images.

$$\mathcal{L}_{i,j} = -\log \frac{\exp(\text{sim}(z_i, z_j)/\tau)}{\sum_{k=1}^{2N} 1[k \neq i] \exp(\text{sim}(z_i, z_k)/\tau)},$$

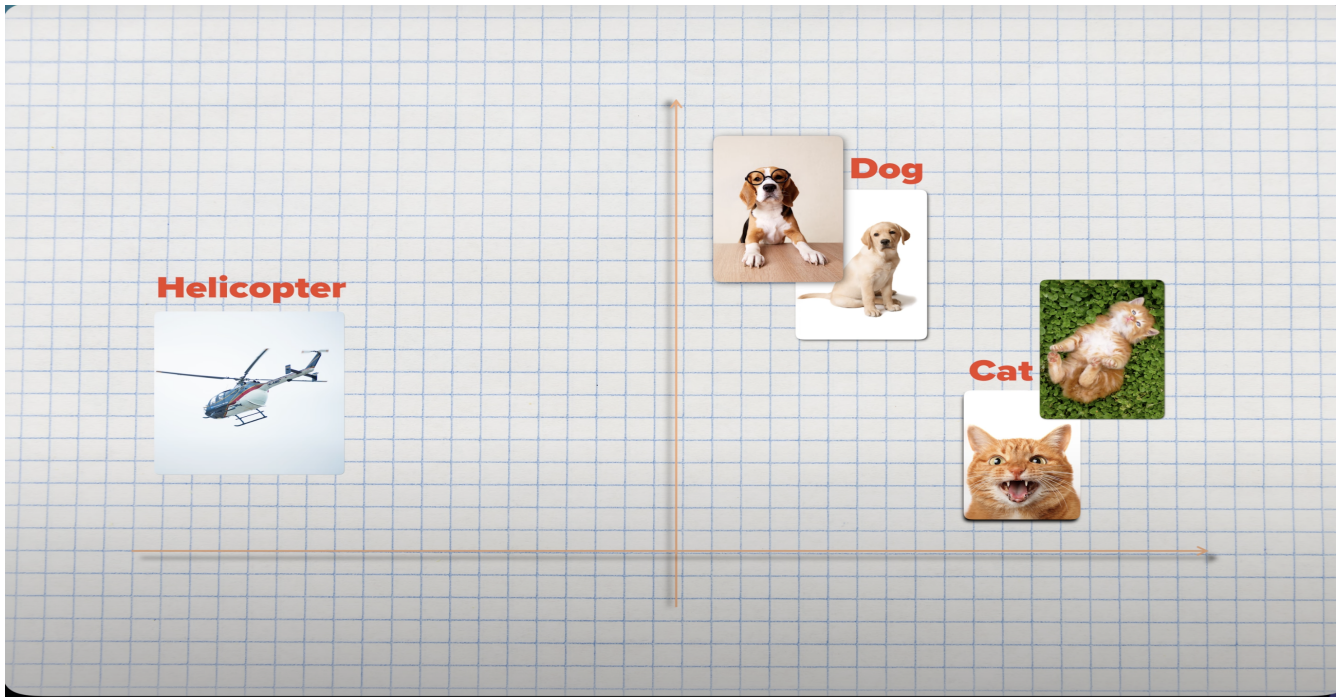


Figure: Embedding space of a contrastive loss

3

[3] source: https://medium.com/@boris.meinardus/what-is-self-supervised-learning-and-why-should-you-care-1c6dfc0e0f57

3. Projection Head: A non-linear projection head (MLP) is applied to the image representations before calculating the contrastive loss, which helps in amplifying invariant features and improving the quality of learned representations.
4. Large Batch Size: Training with large batch sizes is essential for SimCLR to achieve high performance, as it ensures a diverse set of negative samples for the contrastive loss.

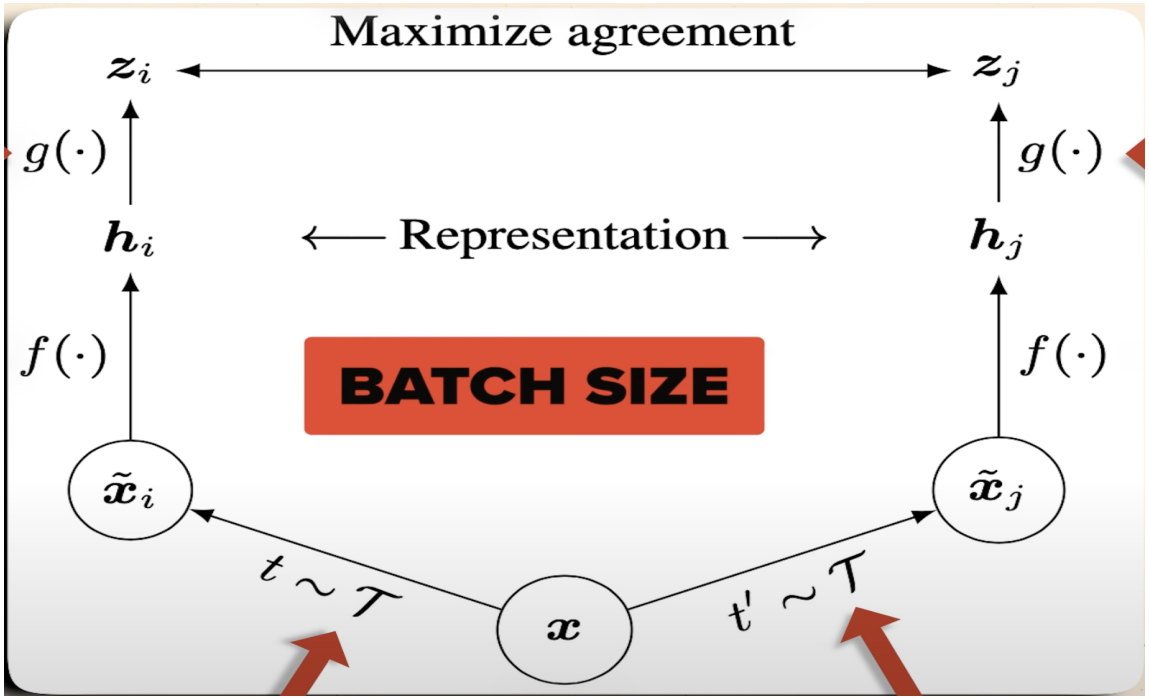


Figure: SimCLR process

4

[4] source: https://medium.com/@boris.meinardus/what-is-self-supervised-learning-and-why-should-you-care-1c6dfc0e0f57

SimCLR implementation

Model Components

- Feature Extractor: ResNet18
- Projection Head: 2 linear layers and a ReLU in between.
- Loss Function: Contrastive loss
- Optimizer: LARS optimizer

Challenges and Observations

- Batch Size: Limited to 128, leading to suboptimal model performance and less distinctive representations.
- Training Duration: Experiments with 50 epochs and 100 epochs yielded similar results, indicating possible saturation.
- Embedding Space Analysis: Used T-SNE dimensionality reduction to observe changes in the embedding space during training epochs.

SimCLR implementation

Observing the closest 10 embeddings to a specific Image through the training.

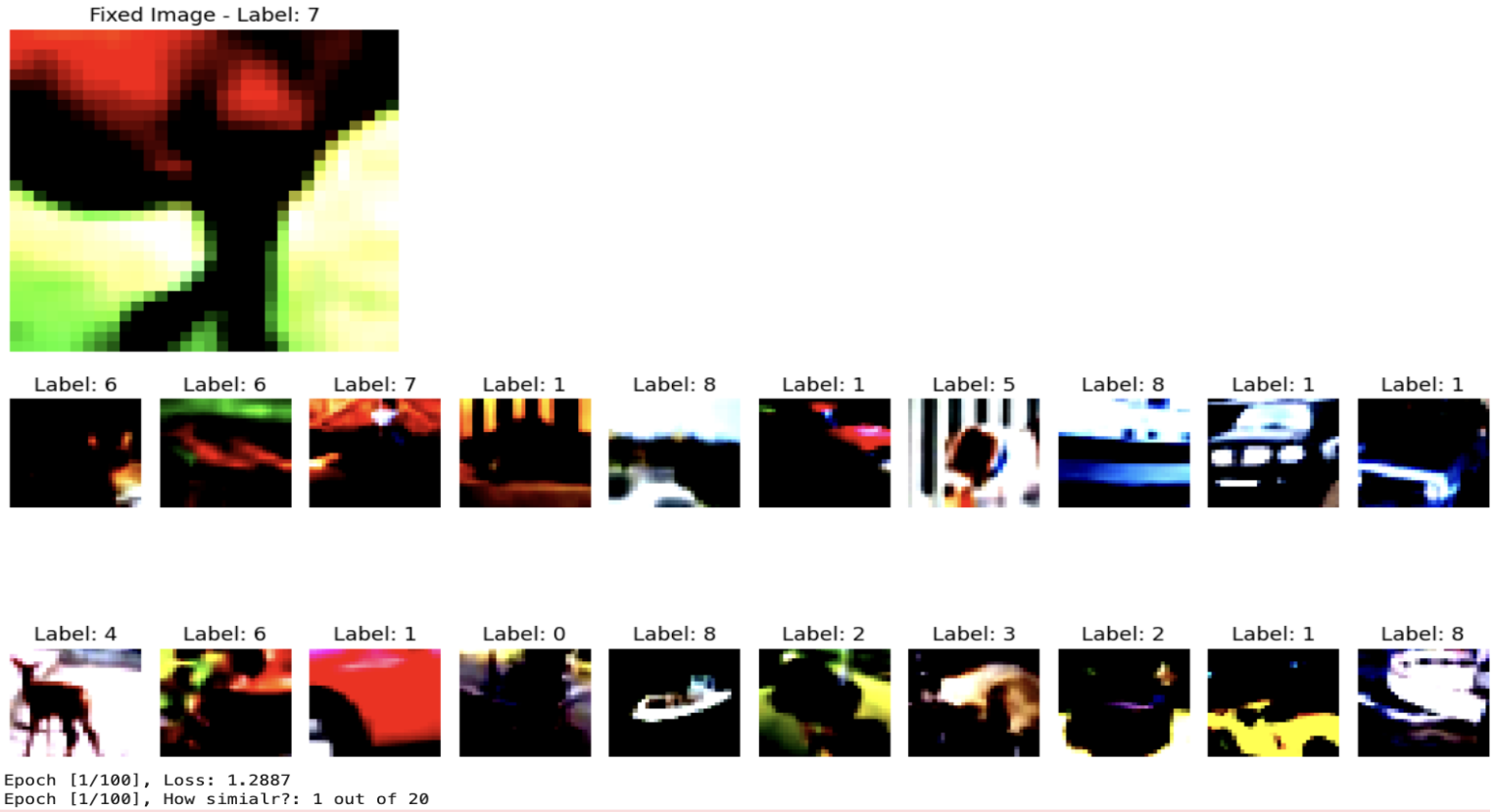


Figure: Epoch 1

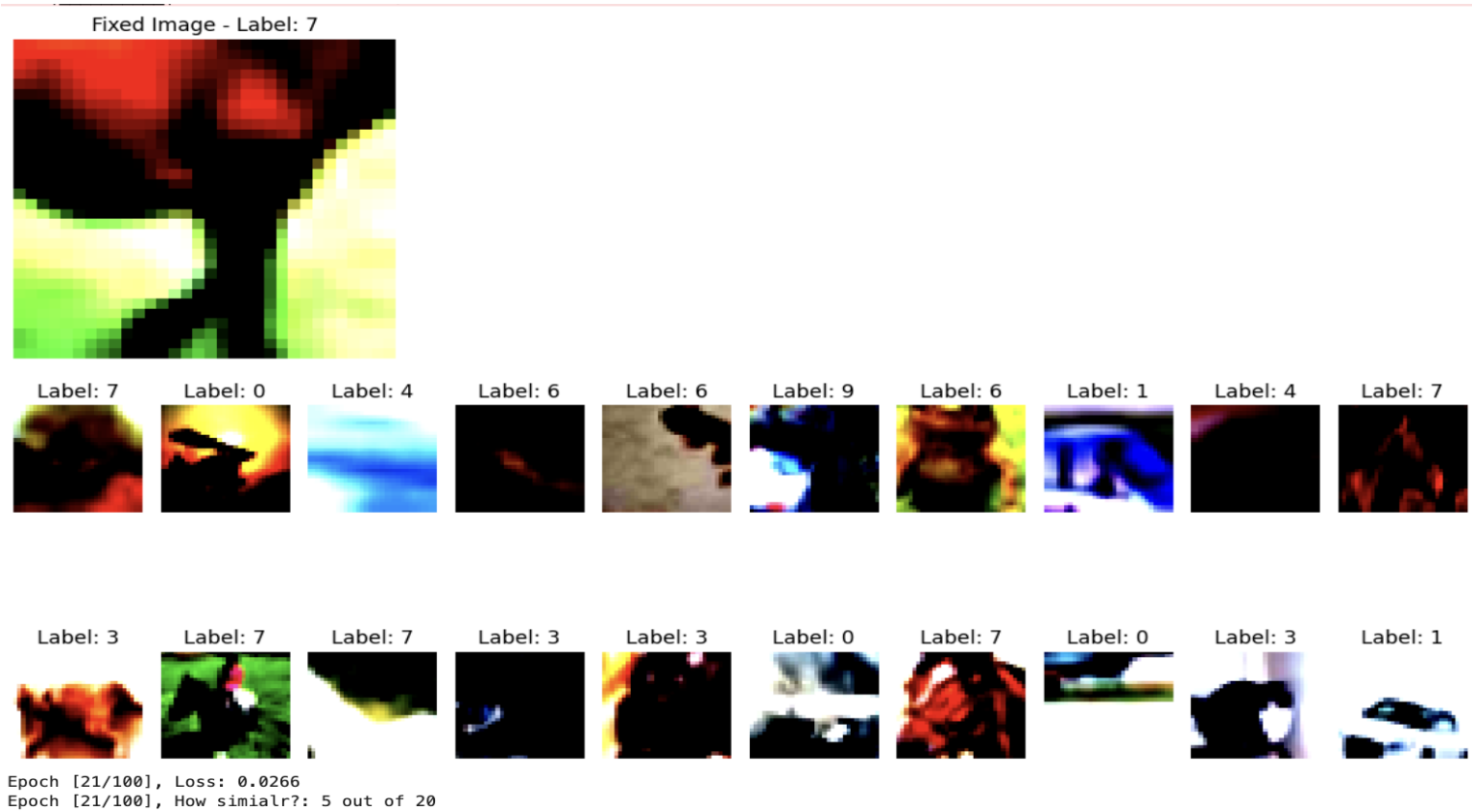


Figure: Epoch 21

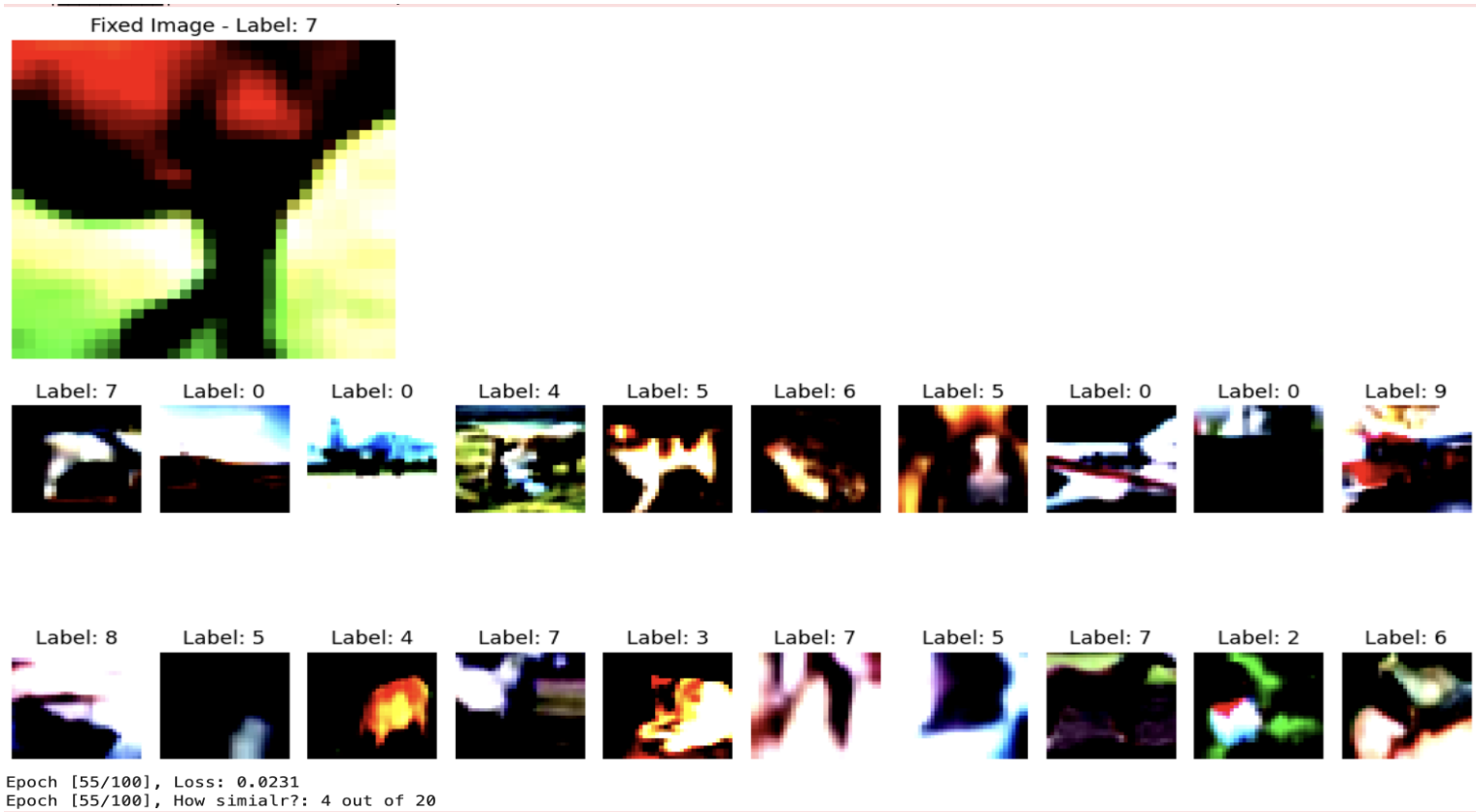


Figure: Epoch 55

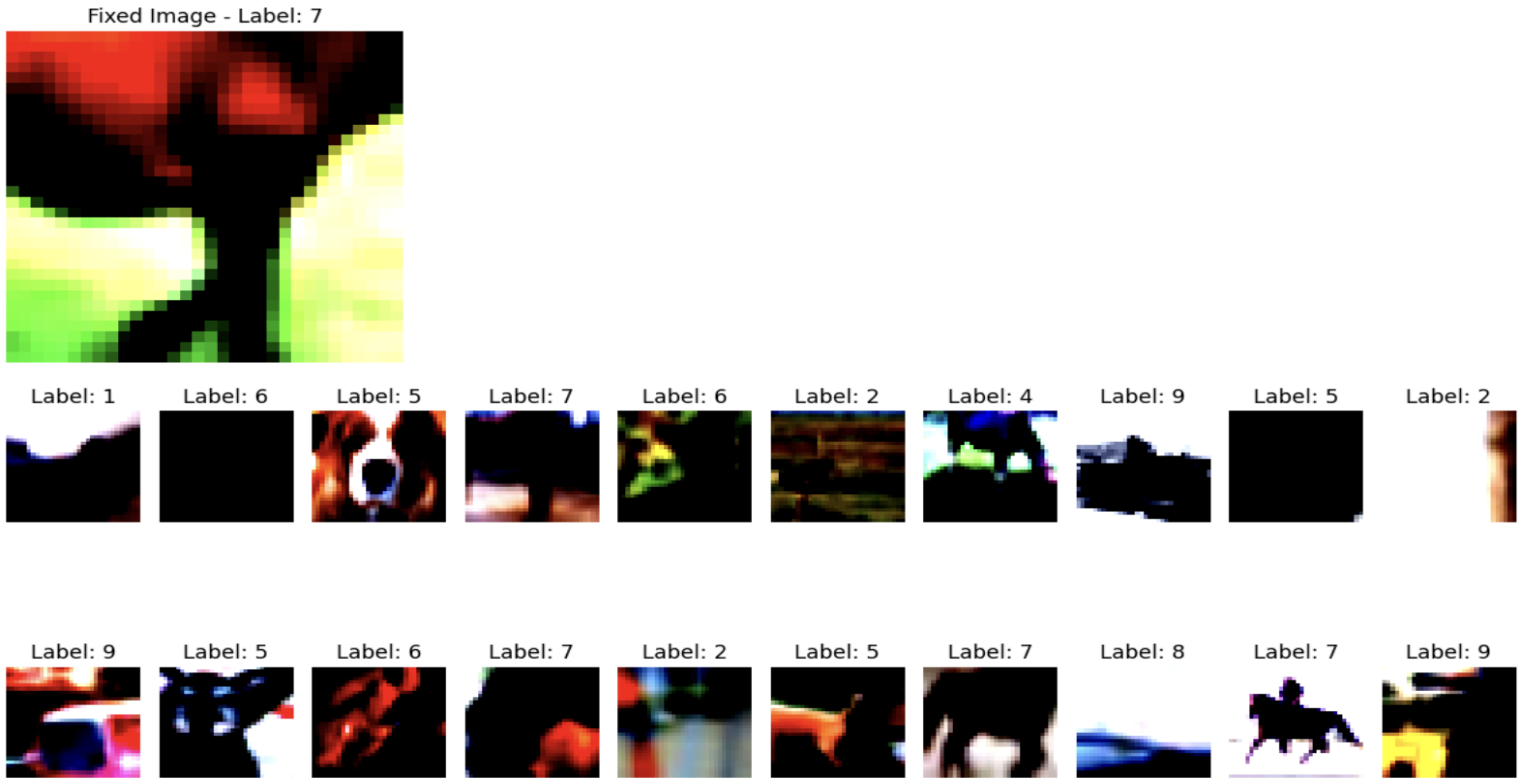


Figure: Epoch 98

Visualization of Embedding space

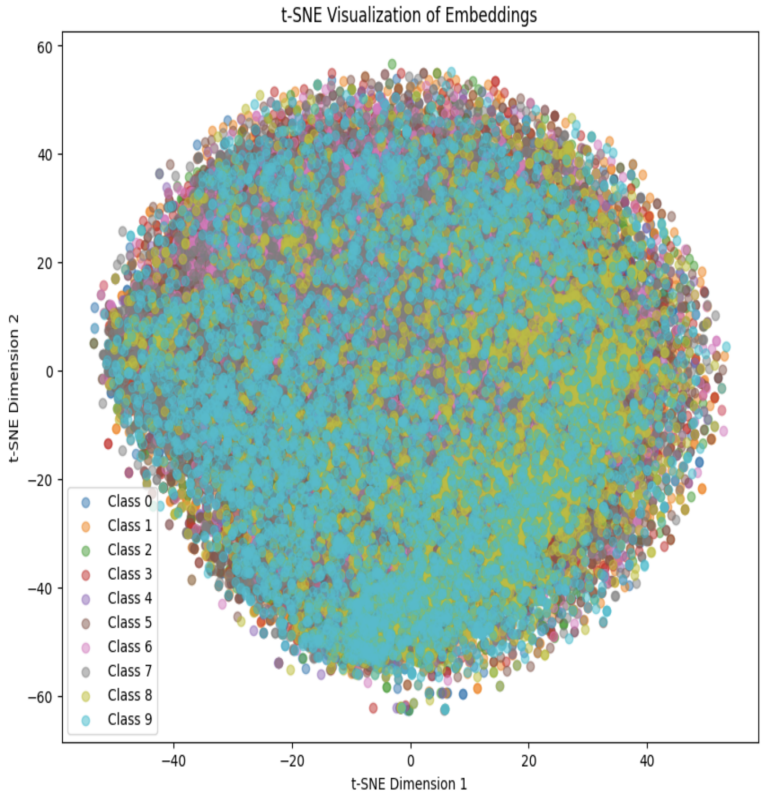


Figure: Epoch 1

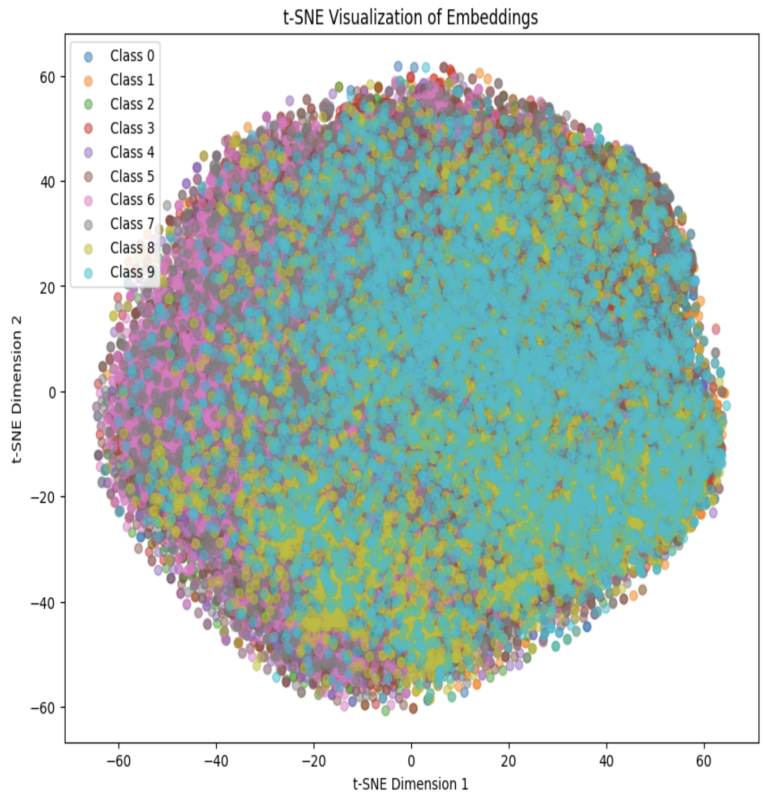
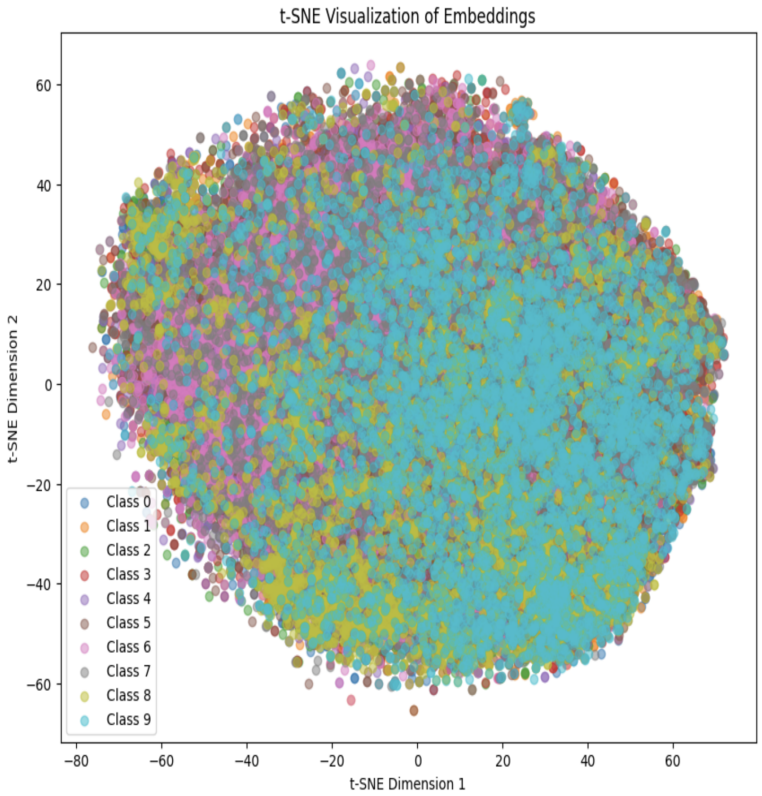


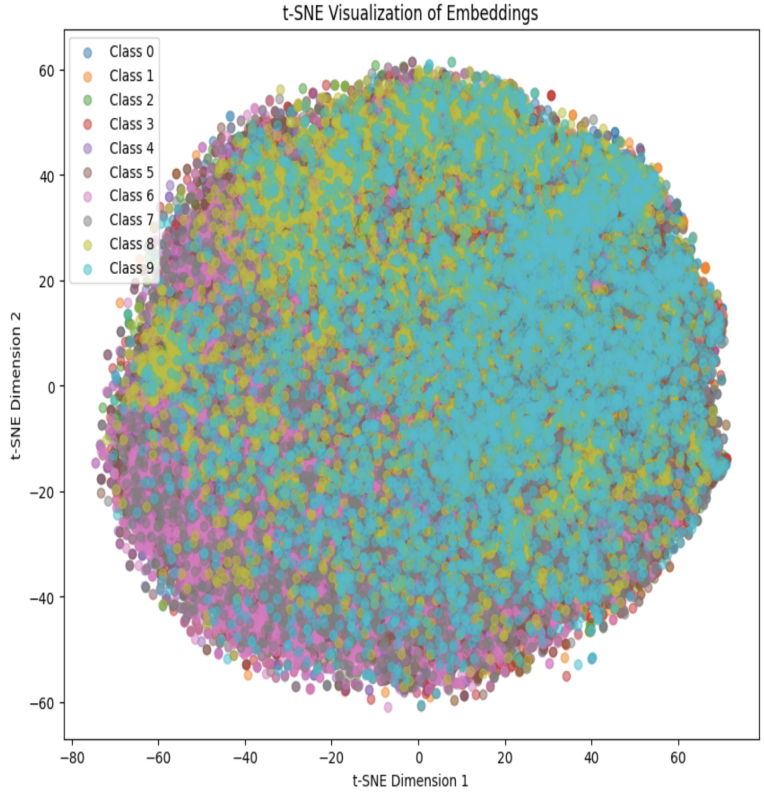
Figure: Epoch 21

Visualization of Embedding space



Epoch [51/100], Loss: 0.0177

Figure: Epoch 51



Epoch [91/100], Loss: 0.0171

Figure: Epoch 91

Deep Clustering

- Clustering methods have been adapted to end-to-end training of visual features [1].
- DeepCluster: a method used to learn deep representations of unlabeled datasets

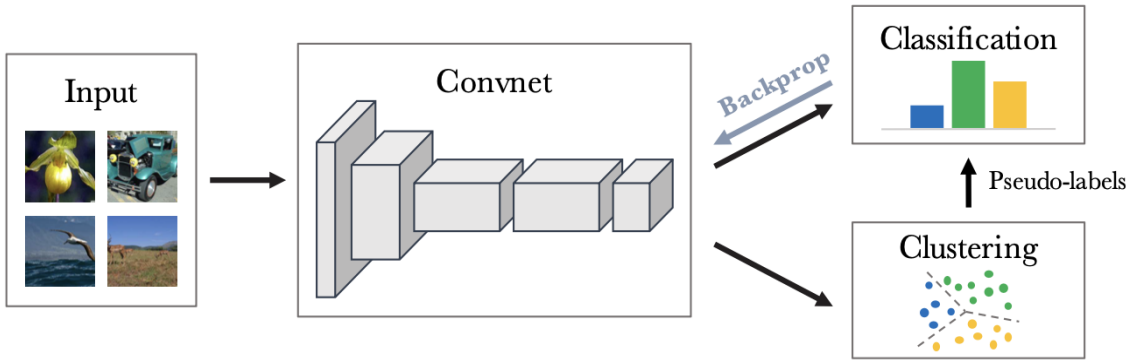


Figure: 1. Illustration of the DeepCluster method.

5

- Iteratively cluster deep features and use the cluster assignments as pseudo-labels.

[5] source: Mathilde Caron et al.

How it works: DeepCluster Pipeline

- Unlabelled images are taken and augmentations are applied to them.
- A convent architecture (AlexNet or VGG16) extracts the features.
- PCA is used to reduce the dimension of the feature vector along with whitening and L2 normalization.

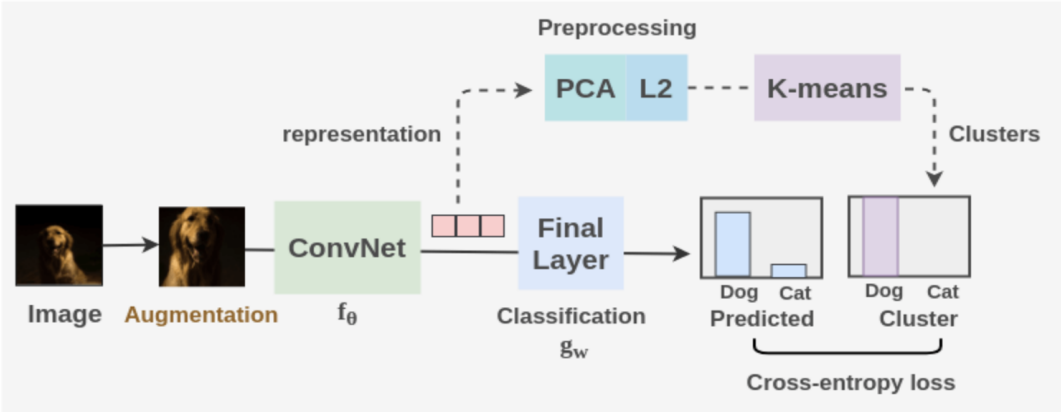


Figure: 1. Illustration of the DeepCluster Pipeline.

6

- The processed features are passed to K-means to get cluster assignments for each image.

[6] source: Amit Chaudhary

- The clustering is obtained by optimizing the objective function:

$$\min_{C \in \mathbb{R}^{d \times k}} \frac{1}{N} \sum_{n=1}^{N} \min_{y_n \in \{0,1\}^k} \left\| f_\theta \left(x_n \right) - C y_n \right\|_2^2 \quad \text{such that} \quad y_n^\top 1_k = 1$$

- The cluster assignments are then used as pseudo-labels for the CNN. We then train to predict the labels with a classification task using the cross-entropy loss:

$$\min_{\theta, W} \frac{1}{N} \sum_{n=1}^{N} \ell \left(g_W \left(f_\theta \left(x_n \right) \right), y_n \right)$$

DeepCluster: Key Results

- Deeper layers in the network can capture larger textural structures.

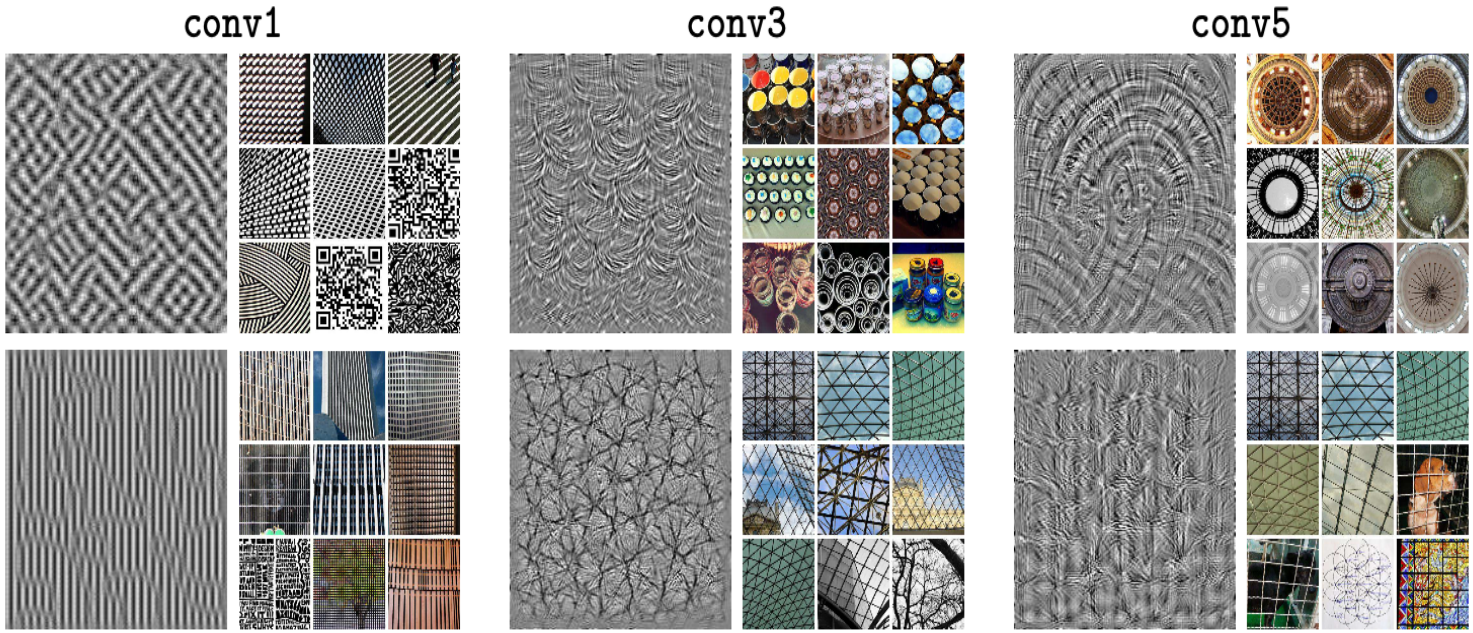


Figure: Filter visualization and top 9 activated images.

7

[7] source: Mathilde Caron et al.

- DeepCluster outperforms all previous unsupervised methods.

	Classification		Detection		Segmentation	
Method	FC6-8	ALL	FC6-8	ALL	FC6-8	ALL
ImageNet labels	78.9	79.9	–	56.8	–	48.0
Random-rgb	33.2	57.0	22.2	44.5	15.2	30.1
Random-sobel	29.0	61.9	18.9	47.9	13.0	32.0
Pathak *et al.* [46]	34.6	56.5	–	44.5	–	29.7
Donahue *et al.* [15]*	52.3	60.1	–	46.9	–	35.2
Pathak *et al.* [45]	–	61.0	–	52.2	–	–
Owens *et al.* [44]*	52.3	61.3	–	–	–	–
Wang and Gupta [63]*	55.6	63.1	32.8†	47.2	26.0†	35.4†
Doersch *et al.* [13]*	55.1	65.3	–	51.1	–	–
Bojanowski and Joulin [5]*	56.7	65.3	33.7†	49.4	26.7†	37.1†
Zhang *et al.* [71]*	61.5	65.9	43.4†	46.9	35.8†	35.6
Zhang *et al.* [72]*	63.0	67.1	–	46.7	–	36.0
Noroozi and Favaro [42]	–	67.6	–	53.2	–	37.6
Noroozi *et al.* [43]	–	67.7	–	51.4	–	36.6
DeepCluster	**72.0**	**73.7**	**51.4**	**55.4**	**43.2**	**45.1**

Figure: Comparison of the DeepCluster to state-of-the-art unsupervised feature learning.

8

[8] source: Mathilde Caron et al.

Results from our implementation

- The model performed well on less complex datasets.

Method	Accuracy	
	MNIST	CFAR10
Supervised	0.982	0.7409
Self-supervised	0.9193	0.3536

Table: Comparision of the proposed method and a supervised approach.

9

[9] Code Repo: `https://github.com/pmensah28/self-supervised-learning`

Challenges and Observations

- The proposed method (DeepCluster) was very expensive to train.
- The model performed poorly on the CFAR10 dataset due to its complexity.
- Complex architectures such as ALexNet or VGG16 will be robust in capturing the complex features in the dataset.

Conclusion

- The SimCLR paper introduces a solution to the challenge of limited labeled data, enabling improved performance on tasks like image classification, object detection, and image retrieval, through the use of contrastive learning.

- In our SimCLR implementation, the use of a batch size of 128 compromised performance, leading to poor representation learning. A T-SNE analysis over 100 epochs showed minimal changes in the embedding space with this batch size, indicating the need for longer training periods.

- DeepCluster achieves better performance than the previous state-of-the-art on every standard transfer learning task.

References I

`https://github.com/gkioxari/aims2020_visualrecognition/releases/download/v1.0/nuts.zip`.

Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European conference on computer vision (ECCV)*, pages 132–149, 2018.