Explainable Representations in Self-Supervised Learning for Image Understanding

Darpan Gaur, Abhinav Vsk Kompella , Aditya Bacharwar, Naik Avaneesh Radhakrishnan
Indian Institute of Technology Hyderabad
{co21btech11004, es21btech11002, es21btech11003, es21btech11022}@iith.ac.in

Abstract

Self-supervised learning (SSL) has emerged as a powerful alternative to supervised learning, enabling models to learn meaningful representations from unlabeled data. However, these learned representations often lack interpretability, making it difficult to understand their decision-making process. In this project, we aim to investigate the explainability of SSL-based feature representations by integrating explainable artificial intelligence (XAI) techniques. Specifically, we will leverage contrastive learning methods such as SimCLR and employ Grad-CAM and similar concept-based techniques to interpret learned representations. Our objective is to evaluate the trade-off between explainability and performance, providing insights into how SSL embeddings capture meaningful features. By improving the interpretability of self-supervised representations, we aim to enhance their applicability in critical domains such as healthcare, autonomous systems, and scientific discovery.

1. Introduction

Supervised learning has traditionally been the dominant paradigm in computer vision, requiring large-scale labeled datasets for training deep neural networks. However, acquiring labeled data is costly and time-consuming. Self-supervised learning (SSL) addresses this issue by enabling models to learn feature representations from unlabeled data using pretext tasks. Contrastive learning, particularly SimCLR, has shown remarkable success in this domain by maximizing the similarity between augmented views of the same image while pushing apart different images.

Despite the success of SSL in learning powerful feature representations, a critical limitation remains: lack of interpretability. Unlike supervised learning, where class labels provide a semantic understanding of features, SSL representations are often opaque, making it challenging to understand what the model has learned. Explainability is crucial for:

- Trust and Transparency: Understanding SSL feature representations can improve trust in AI systems, particularly in safety-critical applications.
- Bias Detection: XAI techniques can help reveal biases in SSL models, preventing unintended discriminatory behavior.
- Downstream Task Adaptability: Interpretable SSL representations can improve transferability to various tasks by ensuring that learned features align with meaningful concepts.

2. Problem Statement

Our project aims to bridge the gap between self-supervised learning and explainable AI by investigating how to interpret SSL-learned representations. Specifically, we will:

- Train an SSL model using contrastive learning (e.g., SimCLR[3] or MoCo[4]).
- Apply Grad-CAM[6], RELAX[7] or other explainability techniques to visualize the important regions in SSL feature maps.
- Explore concept-based explainability techniques (e.g., TCAV, ProtoPNet) to align SSL representations with human-interpretable concepts.
- Evaluate the trade-off between interpretability and performance.

3. Literature Review

3.1. MoCo: Momentum Contrast for Unsupervised Visual Representation Learning

Momentum Contrast (MoCo) [4] is a self-supervised learning framework built on the idea of contrastive learning. MoCo maintains a dynamic dictionary of negative samples using a momentum-based encoder. This dictionary is implemented as a queue, where old representations are progressively replaced by new ones. The novelty that MoCo introduces in its momentum encoder, which helps maintain consistency in feature representations over time and reduces memory constraints by allowing the model to use a rela-

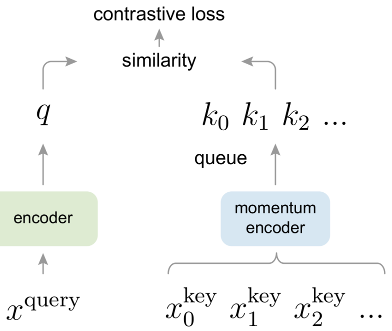

Figure 1. Momentum Contrast (MoCo) trains a visual representation encoder by matching an encoded query q to a dictionary of encoded keys using a contrastive loss. The dictionary keys $\{k_0, k_1, k_2, ...\}$ are defined on-the-fly by a set of data samples. The dictionary is built as a queue, with the current mini-batch enqueued and the oldest mini-batch dequeued, decoupling it from the mini-batch size. The keys are encoded by a slowly progressing encoder, driven by a momentum update with the query encoder. This method enables a large and consistent dictionary for learning visual representations.

tively smaller batch size while still accessing a large pool of negative samples.

The MoCo architecture consists of two encoders: a query encoder and a key encoder. The query encoder is updated via backpropagation, while the key encoder is updated using an exponential moving average (momentum update) to ensure stability as follows:

$$\theta_{\text{k}} \leftarrow m\theta_{\text{k}} + (1-m)\theta_{\text{q}}. \tag{1}$$

Given an image, two different augmentations are applied, producing a query image and a key image. The query image is encoded using the query encoder, and the key image is encoded using the momentum encoder. The contrastive loss, typically InfoNCE defined as follows:

$$\mathcal{L}_q = -\log \frac{\exp(q{\cdot}k_+/\tau)}{\sum_{i=0}^{K} \exp(q{\cdot}k_i/\tau)} \tag{2}$$

This loss is then applied to maximize similarity between the query and key embeddings while minimizing similarity with a queue of negative samples. This framework has been widely adopted in self-supervised learning due to its efficiency in handling large-scale datasets with limited computational resources.

3.2. Learning features by Swapping Assignments between multiple Views (SwAV) of an Image

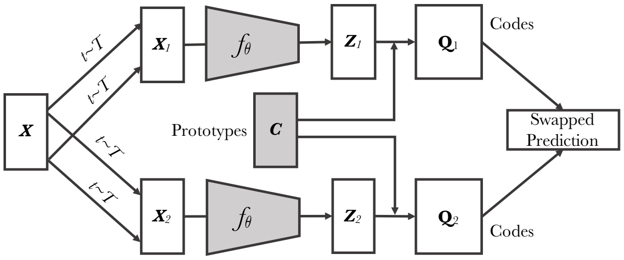

Figure 2. We first obtain "codes" by assigning features to prototype vectors. We then solve a "swapped" prediction problem wherein the codes obtained from one data augmented view are predicted using the other view. Prototype vectors are learned along with the ConvNet parameters by backpropragation.

SwAV (Swapping Assignments Between Views) [1] is a self-supervised learning method that introduces a clustering-based approach to learning visual representations without explicit contrastive loss. SwAV directly learns cluster assignments for different augmentations of the same image. This is achieved by using a swapped prediction mechanism, where the model predicts the cluster assignment of one view using the features of another as seen in the loss function:

$$L(\mathbf{z}_t, \mathbf{z}_s) \quad = \quad \ell(\mathbf{z}_t, \mathbf{q}_s) + \ell(\mathbf{z}_s, \mathbf{q}_t), \tag{3}$$

The terms in the loss function representing the swapped prediction terms are defined as the cross-entropy loss between the code and the probability obtained by taking a softmax of the dot products of z_i and all prototypes in C (set of learned prototypes). Some improvements that the SwAV paper proposes firstly is computing the codes in an online fashion and secondly is the choice of another image augmentation, namely Multi-crop. In the multi-crop strategy we use two standard resolution crops and sample additional low resolution crops that cover only small parts of the image because comparing random crops of an image plays a central role by capturing information in terms of relations between parts of a scene or an object.

3.3. Prototypical Part Network (ProtoPNet)

Prototype Part Networks (ProtoPNet) [2] introduce a novel interpretable deep learning framework by integrating prototype-based reasoning into convolutional neural networks (CNN). ProtoPNet learns a set of class-specific prototype representations, which correspond to meaningful image patches. During inference, an input image is compared with these prototypes, and classification is performed based on similarity scores. This approach provides inherent interpretability by allowing visualization of the prototypical features that influence model predictions. The network is trained using a combination of standard cross-entropy loss and a specialized prototype loss, ensuring that learned prototypes remain discriminative and aligned with human-interpretable visual concepts.

The issue in using the ProtoPNet net is that we assume

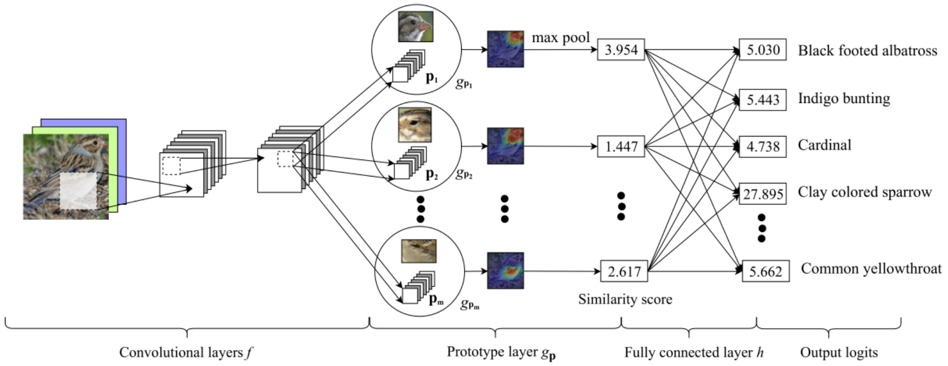

Figure 3. ProtoPNet Architecture

that our data to be unlabeled and hence will be unable to use the class specific prototype representations the model learns. Though the general idea of using prototype learning in self-supervised learning does prove to be useful and can be seen in other papers like Protocon [5].

4. Dataset

Since our objective is to obtain explainable representations using self-supervised learning, we are not restricted to a specific dataset. Instead, we aim to experiment with datasets that provide diverse visual concepts, enabling a better evaluation of both representation learning and explainability. For this purpose, currently we have considered the CIFAR-10 dataset for simplicity. For further works, we may shift to a dataset with more variety so that we can evaluate how well the model is able to explain various class representations.

5. Experiments

5.1. SimClr

We used SimClr as a feature extractor, used same architecture as in [3] with a ResNet-18 backbone, and a linear layer on top of the ResNet-18 features. We train the model on CIFAR-10 dataset.

For data augmentation, following techniques are used:

- `Color Jitter`: [0.8, 0.8, 0.8, 0.2] for brightness, contrast, saturation, and hue respectively, is applied to the input image with a probability of 0.8.
- `Random Resized Crop`: Randomly crop the input image and resize it to a given size of 32x32.
- `Random Horizontal Flip`: Randomly flip the input image horizontally with a probability of 0.5.
- `Gaussian Blur`: Apply Gaussian blur to the input image with a kernel size of 0.1*size of the input image, i.e., 3x3 kernel for 32x32 image.

Two views are created for each image using the above data augmentation techniques. They are passed through the feature extractor, and loss is minimized. Below given are the parameters used for training the model:

- Epochs: 1000
- Batch Size: 256
- Learning Rate: 3e-4
- Weight Decay: 1e-4
- Optimizer: Adam with $\beta_1 = 0.9$, $\beta_2 = 0.999$
- Scheduler: Cosine Annealing LR Scheduler

Figure 4 shows the training loss and accuracy of the model, in which we achieved an accuracy of 82.8125%.

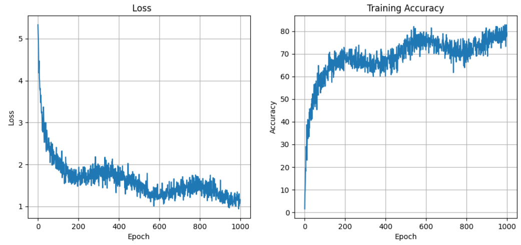

Figure 4. Loss and Accuracy plot for SimClr

5.2. Relax

Our SimCLR trained model is used as a feature extractor for the RELAX[7] model. Masks are generated for the input image and simlarity is found between the masked image and the original image. Importance and uncertainty are calculated for each pixel in the image using simlarity scores.

Figure 5, 6, and 7 show the RELAX explanation and uncertainty on CIFAR-10 dataset. In the figure 6, model is able to give importance to the aeroplane in the image and uncertainty is high in the background. In the figure 7, model is able to give importance to the bird in the image and uncertainty is high in the background. This shows that RELAX model is able to give importance to the object in the image and uncertainty in the background.

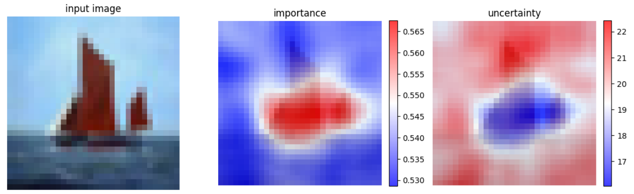

Figure 5. RELAX explanation and uncertainty on CIFAR-10 dataset

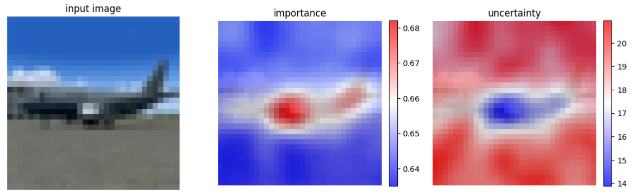

Figure 6. RELAX explanation and uncertainty on CIFAR-10 dataset

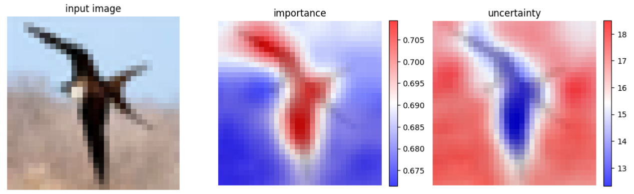

Figure 7. RELAX explanation and uncertainty on CIFAR-10 dataset

6. Observations

We have shown some few examples of the output from our SimCLR plus RELAX pipeline showing the most important parts of the image and also the certainty with which this importance is proposed. The high importance areas are highlighted as masking those key areas of the images changes the representation of the image by a large margin. We can see the model noting the main object from the images being the ship, airplane and the bird with high certainty. For future works we'd like to compare the explainability of this pipeline with other explainable self supervised models.

References

[1] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in neural information processing systems*, 33:9912–9924, 2020. 2

[2] Chaofan Chen, Oscar Li, Daniel Tao, Alina Barnett, Cynthia Rudin, and Jonathan K Su. This looks like that: deep learning for interpretable image recognition. *Advances in neural information processing systems*, 32, 2019. 2

[3] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PmLR, 2020. 1, 3

[4] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020. 1

[5] Islam Nassar, Munawar Hayat, Ehsan Abbasnejad, Hamid Rezatofighi, and Gholamreza Haffari. Protocon: Pseudo-label refinement via online clustering and prototypical consistency for efficient semi-supervised learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11641–11650, 2023. 3

[6] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017. 1

[7] Kristoffer K Wickstrøm, Daniel J Trosten, Sigurd Løkse, Ahcène Boubekki, Karl Øyvind Mikalsen, Michael C Kampffmeyer, and Robert Jenssen. Relax: Representation learning explainability. *International Journal of Computer Vision*, 131(6):1584–1610, 2023. 1, 3