Unsupervised Feature Clustering for Animal Image Classification (Cats vs Dogs)

A PROJECT REPORT

in partial fulfillment for the award of the degree of

Bachelor of Technology
IN
Computer Science and Engineering

Lovely Professional University, Punjab

Submitted by

Name	Registration Number	Roll No.
Lexander Thakur	12323941	9
Ajay Gundu	12304294	28

December 2025

APPENDIX III

(A typical specimen of table of contents)
<Font Style Times New Roman>

TABLE OF CONTENTS

Unsupervised Feature Clustering for Animal Image Classification (Cats vs Dogs)

Lexander Thakur
School of CSE
Lovely Professional University
Jalandhar, Punjab , India
lexanderthakur3@gmail.com

Ajay Gundu
School of CSE
Lovely Professional University,
Jalandhar, Punjab, India
ajaygundu03@gmail.com

***Abstract*- This project an unsupervised approach to distinguish between cat and dog images using SimCLR-based contrastive learning implemented via a custom convolutional neural network encoder and finally clustering with PCA and K-Means. The dataset consists of approximately 24,000 images accquired through Kaggle, of which 20,000 used for training and 4,000 reserved for evaluation.This project relies on image augmentations, which were fed to the CNN enocder to produce 256 dimensional embeddings for each augmented image which was further passed to an projection head and trained using NT-Xent contrastive loss, which were then further processed by using PCA to get the important features to be then fed into K-Means clustering which then can be used for inference, the resulting clustering showed meaninful seperation between the embeddings of cats and dogs showing that the model properly learned the difference between the two without using their actual true labels, by understaing semantic differences.Overall, this work shows the effectiveness of contrastive learning as a label free approach to visual representation learning in an unsupervised pipeline with lightweight custom CNN encoder and relatively simple machine used for the actual training can provide meaningful results.**

Keywords- Convolutional Neural Network, Sim-CLR, Contrastive Learning, Principal Component Analysis, K-Means Clustering

I. INTRODUCTION

Image classification taks such as cats vs dogs usually relies on large labeled datasets where models learn to map already known categories. The cats vs dogs classification problem is a classic benchmark problem in computer vision to check weather a model can distinguish between two visually similar classes. While supervised approaches have high accuracy in such tasks they need need thousands of labelled images which can be time consuming if done manually which is impractical for real world scenario.

Unsupervised learning offers an alternative by learning the representations by how they are instead instead of learning what they are, however classical unsupervised methods directly on pixel data can not achieve this task, the challenge lies in feeding the model the data in such a way that it can separate the two classes based on the metrics that the unsupervised models can work on top of.

Recent advances in contrastive learning, which works on the concept that similar embeddings should be closer in data space and others should be farther apart, show that strong representation can be learned when two augmented images from the same source are used. Among these methods SimCLR (Simple Framework for Contrastive Learning of visual Representation) is used in this project. SimCLR does not need the original labels rather works by learning the semantic meaning of the actual image.

SimCLR's working is based on three cores, a strong augmentation pipeline that forces the model to learn the actual important and differentiating features, a deep CNN encoder that extracts the image representations, and finally a contrastive loss that forces postive pairs to be close together and pushes the negative pairs far apart.

In this project , SimCLR learning along with PCA and Kmeans is applied to Cats vs Dogs dataset to investigate if this model can learn to distinguish between cats and dogs. The contributions of this project are to provide a complete unsupervised learning pipeline using a custom lightweight CNN , a strong augmentation pipeline with SimCLR learning and finally feeding the important features decided by PCA into K-Means to finally do inference between the images of cats and dogs by not learning what they are but how they can be seperated in a data space according to their embeddings extracted by the custom CNN encoder.

II. LITERATURE SURVEY

The cats vs dogs dataset has been used a benchmark for computer vision models due to subtle diffenreces between the two. The usual solutions of this problem are supervised approaches such as VGG[1], AlexNet and ResNet[2] which can schieve accuracies exceeding 95%, these alothough relying on massive and well label datasets with deep and computationally expensive CNN layers, such approcahes being unsuitable for cases where labeling is expensive or unavailable.

Older papers that do rely on unsupervised approaches usually have to deal with manual feature extraction such as Gabor features[3] and then feeding them into PCA and K-Means, such approaches once again have the issue of cost of time spent in manually finding out the distinguishing edges and featrues in the images.

As deep learning rose into the picture CNN[4] and autoencoders[5] introduced nonlinear feature extraction saving on the time of manually finding features, but these structures leaned towards reconstruction qualtiy rather than semantic discrimination, thus these would not perform well when clustered into their classes.

Contrastive learning[6] has recently been on the most successful forms of learning that does not rely on the ground truth labels of the images, which is not available in an unsupervised setting, but rather works on augmentaions[7] of an image by comparing positive and neagtive pairs, main idea being positive pairs should be close together in latent space and negative pairs should be apart.

SimCLR[8] which is a Simple Framework for Constrastive Learning introduced a pipeline combining augmentation methods like random cropping, color jitter, Gaussian blur and gray scale conversion along with the use of a projection head for learning features in images that can distinguish them.

PCA[9] and t-SNE[10] are widely used for analyzing learned features and making sure to use the important ones making clustering more stable and meaningful, where t-SNE emphasizes local structure and is good for 2D visulization when number of features are high.

K-Means[11] being a simple yet powerful clustering method, can be used to find whether the learning embedding space can be used to distinguish between the classes naturally.

The literature shows a clear shift from manually deciding the distinguishing features to used approcahes such as contrastive learning[12], by extracting the features through CNN encoders. Thus this project aims to spply a custom CNN encoder with SimCLR learning method to then feed to PCA and K-Means to see if this model can successfully separate the cats and dogs classes with limited compute and relatively simple architecture.

III. METHODOLOGY

The overall workflow of the methodology is described in Fig. 1.

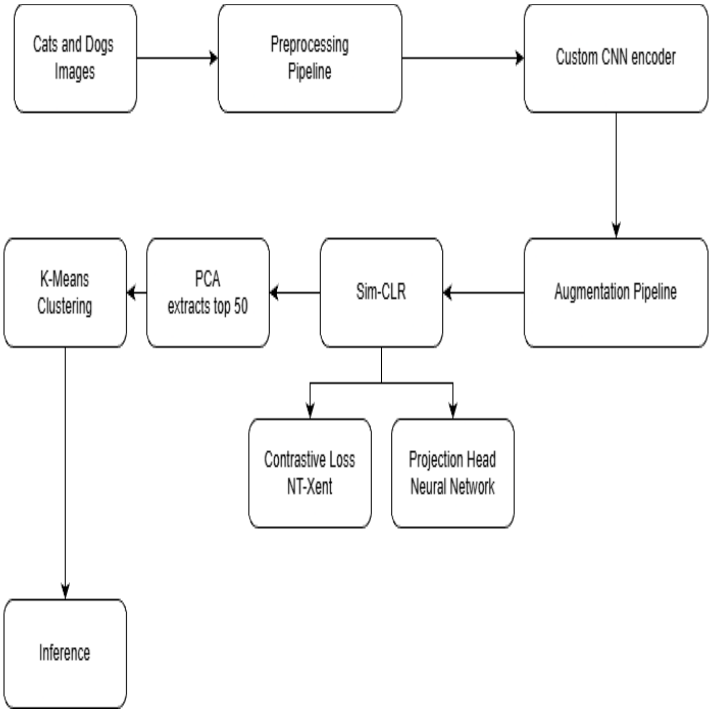


Fig.1. Workflow Model

A. Data Description

The dataset used in this project is the **Cats vs Dogs** image collection obtained from Kaggle. It contains images belonging to two categories: cats and dogs. In total, the dataset consists of **approximately 24,000 images**, with **12,000 cat images** and **12,000 dog images**.

- **10,000 images per class (~20,000 total)** were used during the training phase.
- **2,000 images per class (~4,000 total)** were held out and used later for evaluating the accuracy of the final classification.

B. Preprocessing Pipeline

All images from the dataset were passed through a structured preprocessing pipeline to ensure consistency in size, format, and further down the matrix calculations.

Each sample in the Cats vs Dogs dataset is stored as an **RGB image**, meaning it contains **three color channels**: Red (R), Green (G), and Blue (B). Every pixel in the image is represented as a triplet of intensity values across these three channels. This 3-channel representation is important because convolutional neural networks process color information by learning separate filters for each channel.

Each sample in the Cats vs Dogs dataset is stored as an **RGB image**, meaning it contains **three color channels**: Red (R), Green (G), and Blue (B). Every pixel in the image is represented as a triplet of intensity values across these three channels. This 3-channel representation is important because convolutional neural networks process color information by learning separate filters for each channel.

The data consists of images of various shapes, resolutions, and aspect ratios. To ensure uniformity, all images were resized to **224 × 224 pixels** so the CNN can operate properly.

Before the images can be used for training, they must be converted into **PyTorch tensors**. A tensor is a multi-dimensional numerical array similar to a matrix, but with PyTorch fast mathematical operations can be done using the GPU.

Since each Cats vs Dogs image has **3 color channels (RGB)**, every preprocessed image becomes a **3 × 224 × 224** tensor after resizing.

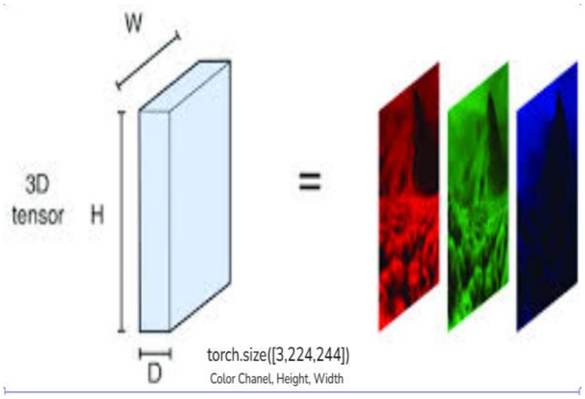
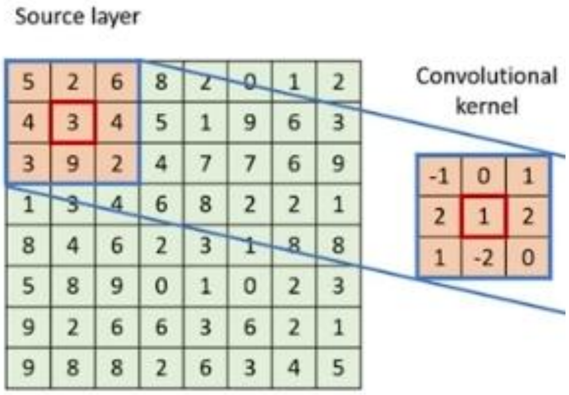


Fig.2. Image as a Tensor

C. *Convolutional Neural Network*

A Convolutional Neural Network is a deep learning architecture designed to process grid-like data such as images. Instead of manually engineering features like some earlier papers, a CNN learns them using stacked convolutional layers.

The core operation of a CNN is the **convolution**. A convolution layer applies small learnable filters (also called kernels) that slide across the image.



5	2	6	8	2	0	1	2
4	3	4	5	1	9	6	3
3	9	2	4	7	7	6	9
1	3	4	6	8	2	2	1
8	4	6	2	3	1	8	8
5	8	9	0	1	0	2	3
9	2	6	6	3	6	2	1
9	8	8	2	6	3	4	5

-1	0	1
2	1	2
1	-2	0

Fig.3. CNN kernel

Fig.2 shows how for an image represented as a matrix the kernel is just a smaller matrix that slides across the bigger matrix, extracting components of image, like edges, color, or in the case of cats vs dogs , the fur texture, ear shape, etc.

After each convolution, the encoder uses the Rectified Linear Unit (ReLU)[13] activation function. ReLU introduces non-linearity into the model giving it the ability to learn non linear patterns.

Between convolutional blocks, the encoder applies **MaxPool2d(2)**. Max Pooling reduces the spatial resolution by selecting the maximum value within a 2×2 window.

In this encoder, Max Pooling progressively shrinks:

- 224 → 112
- 112 → 56
- 56 → 28
- 28 → 14

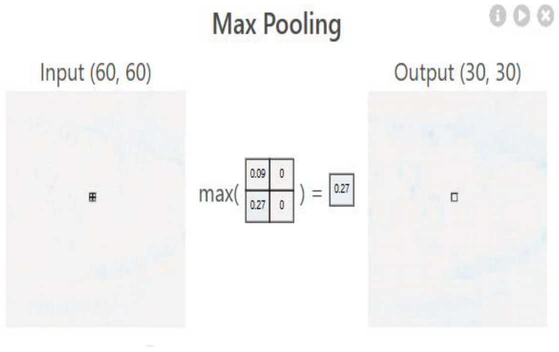


Fig.4. Max Pooling

Fig.4 shows how important information is compressed using Max Pooling for simpler representation and computational efficiency.

D. Image Augmentation Strategy

Since for an unsupervised model access to the labels for training the model is not possible, augmentations play a key role in this project for the later stage of SimCLR to work. Augmentation refers to a set of systematic transformations applied to an image—such as cropping, flipping, color distortion, or blurring—to create new, modified versions of the original sample.The augmentation pipleline in this project is as follows.

- RandomResizedCrop(224,scale=(0.2, 1.0))
- RandomHorizontalFlip(p = 0.5)
- ColorJitter(brightness,contrast, saturation, hue)
- RandomGrayscale(p = 0.2)
- GaussianBlur
- ToTensor()

These augmentations create a distorted copy of the image although keeping the important contents close to the original.

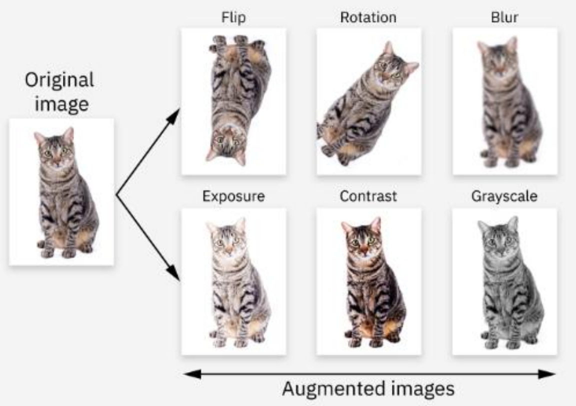


Fig.5. Image Augmentation

Fig.5 shows an example of what image augmentations look like.

E. Learning through SimCLR

SimCLR (Simple Framework for Contrastive Learning of Visual Representations) is a learning method that learns meaningful embeddings without using labels. The idea is to train an encoder so that two different augmented views of the same image end up with similar representations, while views from different images remain far apart.

SimCLR has two main components a CNN encoder that exracts the features from image, and Projection head and a contrastive loss that shape the representation space.

After the CNN produces the 256-dimensional feature vector as discussed previously in the paper, this vector is passed through a projection head which is basically a small MLP (Multi-Layer Perceptron).

An MLP[14] is a neural network with fully connected neurons with non-linear activations, in this project ReLU is used.

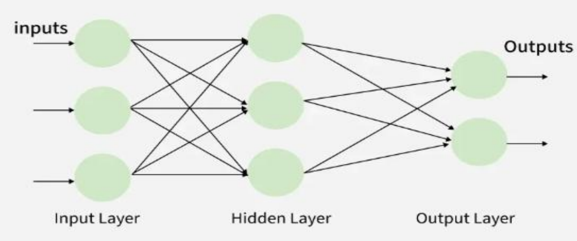


Fig.6. Example of an MLP

A projection head is needed because the CNN encoder learns what the image is and the projection head learn how to compare two images, the contrastive loss works on cosine similarity which struggles with the messy output of the encoder, so the projection head normalizes it and makes it compact so they eventually cluster well.

F. Contrastive Loss

The main idea of contrastive loss is that data points that are similar should be nearby in data space and dissimilar points should be far apart.The former is referred to as **positive pair** and latter is **negative pair.**

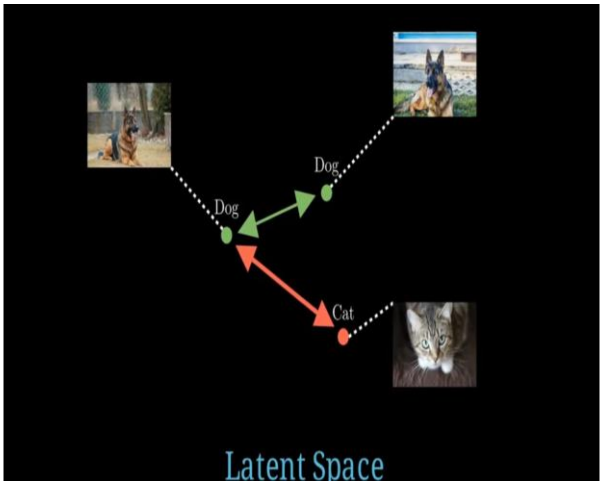


Fig.7. Positive and negative pair in latent space.

This is further utilized in a batch with creating one postive pair and multiple negative pairs, the number of negative pairs depending on the batch size, in this project batch size of 16 is used due to computational limits.

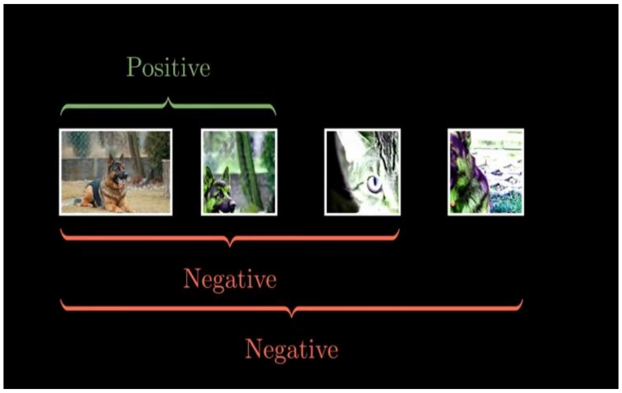


Fig.8. Positive Pair and Multiple negative pair

This project uses Normalized Temprature Scaled Cross Entropy Loss[15], the objective of this is simple, pull the embeddings of all postive pairs closer and push the negative pairs further apart.The loss function is given the two augmented images, if there and N images in the batch, we have 2N embeddings in total, then we concatinate these 2N images for both the augmentations into a 2N x 2N matrix in which every entry (i,j) is cosine similarity between the two, the elements of this matrix is then divided by the temperature value which is basically a scaling factor, when temprature is less than 1 it magnifies the differences and when temperature is large the differences are numbed down, temperature is used because often raw cosine similarities are close to each other.

Then the positive and negative pairs are extracted and a row of the logits are built in which the positive pair and kept at the 0th index, this 0th index then acts as an label upon which cross entropy loss can be calculated, so we get a pseudo label even though we did not have label for the original image.

Thus, NT-Xent trains the model to keep the two augentations of an image close while pushing the others far apart enabling unsupervised representation learning.

G. PCA and Kmeans

After the learning is compelete the encoder is used for inference, we convert the 256 dimensional vectors of the encoder into 50 taking the 50 most important ones by using Principal Componenet Analysis, then this in fed into kmeans with number of clusters as 2 as one for cat and one for dog, upon which then the inference is done.

IV. IMPLEMENTATION PLAN

The training was done on a local machine with a GPU of 3.6 GB, due to this limited GPU batch size could not be increased beyond 16.The training was done for 180 epochs , each epoch taking between 12 to 25 minutes, with a total training time of about 28 hours. For the training Adam optimizer is used and after every 10 epochs check points are made.

The training workflow is using the two augmented views and passing them to the encoder and projection head to obtain the embeddings which is then passed to NT-Xent loss , after which the gradients calculated by adam optimizer and backpropogated and loss is accumulated for each epoch which is then averaged.

V. EXPERIMENTAL RESULTS

The performance of this model is evaluated by checking the accuracy comparing to a test set in which true labels are compared to the cluster assignment, visualization of the embedding space via t-SNE to capture relationship between the high number of dimensions, and internal unsupervised clustering metrics.

The clustering accuracy we get from the PCA reduced 50 dimsenional features that is then fed into K-Means with 2 clusters shows a clear upward trend in accuracy as a the number of epochs increase, initially staring from 50% accuracy as it is binary classification and then slowly the accuracy increases.

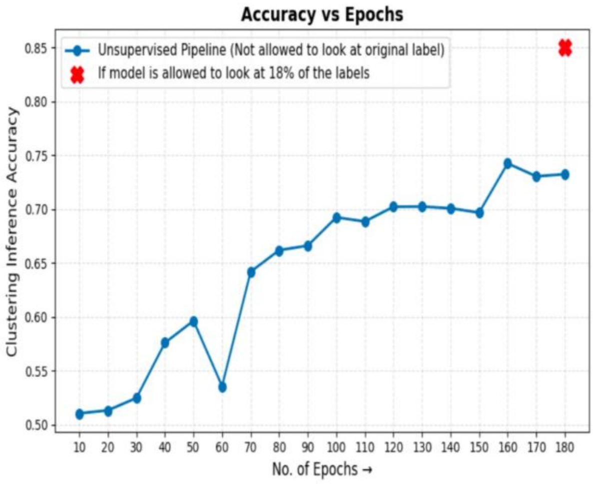


Fig.9. Accuracy vs Epochs Plot

The clustering accuracy early in training is performing slightly above random chance, as the training continues the accuracy stabilizes between .68 to .74 after a 150 plus epochs , peaking at 160 epochs and a slight decline after.For comparison if the same model pipeline is followed but with a slight change of letting the model see 18% of the true labels that is 2000 images, then the accuracy for this Sim-CLR bases system shoots upto 85% accuracy that is combining the best of both world of supervised and unsupervised training.

To analyze the visual representations in a 2D space where we are using such high number of features that is top 50 features decided by PCA, t-SNE was applied to get some meaningful 2D visualization.

z

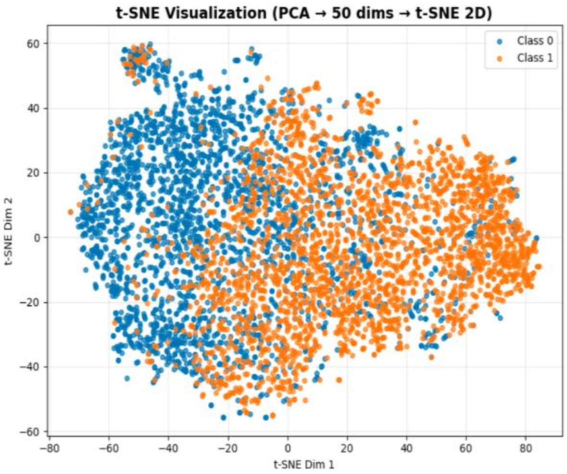


Fig.9. t-SNE plot of top 50 PCA features

The resulting 2D visualization shows two major clusters. There is also overlap between them which is expected in an unsupervised setting and also since cats and dogs are visually similar to some extent such as having the same eye shape or fur texture in some cases thus the encoder extracted features when plotted via t-SNE show features of one class overlapped into the other cluster.

The clusters also show that one region is densely populated by one class and the opposite region is dominated by the other class, the boundary areas show mixture and overlap due to ambiguous visual similarity.

The results of the clustering metrics are as follows:

Silhouette Score	**0.1052**	Weak–moderate structure; typical for image data
Calinski–Harabasz Score	**621.39**	Good overall cluster compactness
Davies–Bouldin Score	**2.60**	Moderate cluster overlap
Adjusted Rand Index (ARI)	**0.2198**	Moderate agreement with true labels

Fig.10. Clustering metrics

Collectively, these metrics show that learned embeddings capture structure relevant enough to distinguish between cats and dogs.

VI. CONCLUSION

This project shows that is possible to have meaningful representations for Cats vs Dogs without using their true labels using a SimCLR based contrastive learning. A custom CNN encoder trained solely through augmentations and NT-Xent loss captures features such as shapes, textures and structure enbaling effective unsupervised clustering and seperation of the dataset.

Through a unsupervised based pipline consisting of augmentations, CNN , projection head MLP, and clustering with PCA and K-Means, the project was able to separate cat and dog images without using their initial true labels.

REFERENCES

[1] Tang, Li. "Image classification based on improved VGG network." *2021 IEEE 6th International Conference on Signal and Image Processing (ICSIP)*. IEEE, 2021.

[2] Liang, Jiazhi. "Image classification based on RESNET." *Journal of Physics: Conference Series*. Vol. 1634. No. 1. IOP Publishing, 2020.

[3] Manohar, N., YH Sharath Kumar, and G. Hemantha Kumar. "Supervised and unsupervised learning in animal classification." *2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. IEEE, 2016.

[4] Purwono, Purwono, et al. "Understanding of convolutional neural network (cnn): A review." *International Journal of Robotics and Control Systems* 2.4 (2022): 739-748.

[5] Li, Pengzhi, Yan Pei, and Jianqiang Li. "A comprehensive survey on design and application of autoencoder in deep learning." *Applied Soft Computing* 138 (2023): 110176.

[6] Lim, Jit Yan, et al. "SCL: Self-supervised contrastive learning for few-shot image classification." *Neural Networks* 165 (2023): 19-30.

[7] Xu, Mingle, et al. "A comprehensive survey of image augmentation techniques for deep learning." *Pattern Recognition* 137 (2023): 109347.

[8] Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. (2020). A Simple Framework for Contrastive Learning of Visual Representations. *International Conference on Machine Learning (ICML)*..

[9] Maćkiewicz, Andrzej, and Waldemar Ratajczak. "Principal components analysis (PCA)." *Computers & Geosciences* 19.3 (1993): 303-342.

[10] Maaten, Laurens van der, and Geoffrey Hinton. "Visualizing data using t-SNE." *Journal of machine learning research* 9.Nov (2008): 2579-2605.

[11] Sinaga, Kristina P., and Miin-Shen Yang. "Unsupervised K-means clustering algorithm." *IEEE access* 8 (2020): 80716-80727.

[12] Schmarje, Lars, et al. "A survey on semi-, self-and unsupervised learning for image classification." *IEEE Access* 9 (2021): 82146-82168..

[13] Banerjee, Chaity, Tathagata Mukherjee, and Eduardo Pasiliao Jr. "An empirical study on generalizations of the ReLU activation function." *Proceedings of the 2019 ACM southeast conference*. 2019.

[14] Taud, Hind, and Jean-Franccois Mas. "Multilayer perceptron (MLP)." *Geomatic approaches for modeling land change scenarios*. Cham: Springer International Publishing, 2017. 451-455.

[15] Ågren, Wilhelm. "The nt-xent loss upper bound." *arXiv preprint arXiv:2205.03169* (2022).