

# Self-supervised Image Classification using Convolutional Neural Network

*by NoShiyun Kong*

---

**Submission date:** 02-Aug-2022 08:51PM (UTC+0900)

**Submission ID:** 1878066539

**File name:** datacenter\_paper\_turnitin\_2022-08-02\_13082705.docx (2.73M)

**Word count:** 4196

**Character count:** 23827

# Self-supervised Image Classification using Convolutional Neural Network

Shiyun Kong\*

Shanghai United International School,  
Shanghai, China  
rjasinski@une.edu

**Abstract**—Image classification has been a trendy research topic in the field of pattern recognition and computer vision, which extracts different features of images and predicts the category of images. Thanks to the development of deep learning, powerful convolutional neural networks can be used in the field of image recognition. However, existing deep learning-based image recognition research mostly follows the framework of supervised learning, and model learning relies on a large number of accurate labels. Providing a large amount of label data will undoubtedly require laborious human effort and expensive costs. Therefore, image recognition based on unsupervised learning (without using any image category labels to achieve classification) has become a spotlight for research. In this paper, explorations on the image classification by self-supervised framework SimCLR on image classification successfully clusters a large number of images into an optimum amount categories. Qualitative results have shown SimCLR is particularly effective in recognizing the colors of images; both qualitative and quantitative results show SimCLR is great at identifying simple contours. However, when the colors are similar, and contour lines are complex, SimCLR does not obtain satisfying results. The accuracy on classifying Mnist dataset is 32%.

**Keywords:** Representation learning, ResNet, SimCLR, K-means, Clustering

## I. Introduction:

Image classification has been a trendy research topic in the field of pattern recognition and computer vision, which extracts different features from images and predicts the category of those images. With the popularity of the Internet and the rapid development of intelligent information processing technology, large-scale image resources continue to emerge.[1] How to accurately classify massive images, therefore, become particularly important. Thanks to the development of Convolutional Neural Networks (CNN), there is great success in the field of image classification and attracted a large number of researchers' interest.

Early image recognition algorithms rely on manual features, and its performance is difficult to meet the practical application needs. Convolutional

neural network can adaptively learn high-dimensional nonlinear transformation to effectively extract target features, and its powerful feature representation ability can significantly improve the accuracy of the classifier.[2] However, the existing image recognition research based on deep learning mostly follows the framework of supervised learning, and model learning relies on a large number of accurate labels.[3][4][5][6][7] Providing a large amount of labelled data will undoubtedly consume tremendous amount money, human effort and most importantly time. Therefore, image recognition based on unsupervised learning (without using any image category labels to achieve classification) has become a spotlight for researchers. Combining the supervised algorithm with the unsupervised method, a semi supervised learning model with decent amount of accuracy can be obtained. For example, the ladder network constructed by the nature of self-coding network, which is composed of an unsupervised network connected with a supervised network, can effectively screen the information related to the classification task from the data information.[8]

In the realm of self-supervised image classification, many discoveries are centered around the dataset of cifar-10, cifar-20, ImageNet, and etc. These are quite different from each other, and images' semantic environment are largely different. These discoveries were indeed significantly beneficial for the development of the Artificial Intelligence (AI) realm, but it is quite distant from the society. That means people would rarely not rely on computers to distinguish dissimilar objects because it is not laborious to do manually, but if images are all similar to each other, image classification can then become desperately needed in people's life. In this paper, I tried to conduct a practical approach by sending images that are very similar, and could possibly be taken as images for people's social media profiles into the CNN to generate embeddings and use K-means for clustering them into potential categories. Also, for providing the numerical measure of the accuracy, I send Mnist through the model, which is semantically similar, which matches my goal.

Because there is no place to obtain any ground-truth annotations when the people themselves are incognizant of the number of clusters of images, and the requirement to classify images, people require self-supervised algorithm to

classify images. Firstly, a Collate Function will be used to do image transformations in order to reduce the importance of minor details (assumed using empirical evidence) that lies on the fringes of all images on the clustering of images. Then, the pair of image<sup>5</sup> set after transformations will send through a simple framework for contrastive learning of visual representations—SimCLR with the Residual Neural Network (ResNet) backbone to improve quality neural network and reducing training time.[9][10] During which process, individual traits of images will be extracted and recorded with the output of 512 dimension embeddings. Inasmuch as the encodings are the incarnation of pictograms themselves, they were being singular-value decomposed into 2 dimensions using PCA for further Clustering process reserved by k-means and k-nearest neighbors. As long as the 2 dimensional data is obtained, they are being projected to a latent space, and in which imaginary space, Euclidian distance can be computed, thereby labelling images according to their closeness in that latent space. Through the Elbow Method, a favorable number of clusters can be chosen, and hence complete the whole clustering process by labelling all pictures with the reference of their closeness in sense of Euclidian distance. Finally, qualitative judgements will be given by comparing clustered images; and quantitative measures will be calculated by comparing the labels generated through the model with the truth labels of Mnist.

## II. Methodology:

### A. Image Transformation

Normally, from empirical evidences, when people create an image, or, look at an image, both processes would put their main focus upon the middle part, contours of individual graphics. Therefore, the fringes of images and the precise coloring are often regarded as a less important information, or, directly ignored by people. With the help of image transformations, it is crucial in reducing the importance of those less important information, and try to let the center and main contours be the most important information that a model is able to learn. With the collate function defined by Lightly, it will adjust the brightness, contrast, saturation, hue and size of images by specific probabilities. Furthermore, it will undertake cropping, Gaussian Blur, normalization and horizontal reflection under probabilities during the image preprocessing process. For each of the original image, a pair of transformed images will be produced for training the model to focus on the major details.[9]

### B. Backbone

The Backbone used in this experiment is the ResNet. ResNet was the winner of ImageNet

challenge in 2015 with accuracy of 3.57%. Before, people found it is hard to train deep neural networks because of the problem of vanishing or exploding gradient created by repeated multiplication though the process of backpropagation. However, Resnet addressed the issue by creating “Shortcut connections”. He, Kaiming et al. identified that the deepness of a neural network does not necessarily entail the degradation problem, rather it will at least obtain the same, or higher accuracy of its shallower parts if the deeper part only learns the identity function. The identity function, nonetheless, is hard to learn in CNNs directly because of different weights, therefore, it is better to focus on Residual Learning. Residual Learning is the process of learning the slightest differences (assumed) between previous outputs skipped from former convolution layer and current outputs from latest convolution layer (See Figure 1). The deeper the connection, the lesser Residual Learning is able to interpret, which then, learns identity function alternatively. Inside those Skip connections, the Residual Learning is undertaken, in conjunction with ReLU Activation, an accurate image classification CNN called ResNet is made.[11][15]

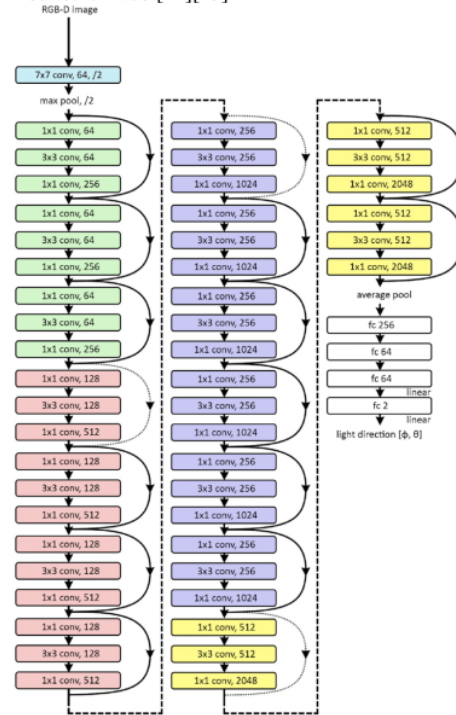


Figure 1. The basic structure of Resnet

### C. My Model

The model selected for this particular experiment is based on the Simple Framework for Contrastive Learning of Visual Representations

(SimCLR), which achieved top-1 accuracy on ImageNet. SimCLR will receive the transformed images explained in section 2.1, pass data through the ResNet backbone, send representation vectors from ResNet to a Multilayer Perceptron (MLP), and thereby find contrastive loss.

SimCLR improves the effect of prediction tasks by using data enhanced operations, and proves that unsupervised comparative learning can benefit more from data enhanced operations than supervised learning. SimCLR introduces a learnable nonlinear transformation between representation and contrast loss, which greatly improves the quality of learning representation. Compared with supervised learning, comparative learning can benefit from deeper and broader networks and larger batches and training. As shown in Figure 2, the main operations of SimCLR algorithm include unsupervised pre-training and supervised fine tuning, whose important innovations include: using data enhancement to improve the effect of unsupervised learning, using MLP network with hidden layer for nonlinear transformation, and introducing a counter loss function which can be seen in formula (1).

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

Here, SimCLR randomly samples  $N$  pictures as a training batch. After data enhancement,  $2N$  enhanced samples will be obtained. In a batch, for a certain picture (fixed as a reference), the loss is calculated by pairing between two, so there are a total of  $2(N-1)$  negative samples. Through this entire process, finds out the key features of images by contrasting the pair of input images with the loss function, and project relative vectors of each images into an embedding space, therefore, the embeddings can be recorded and ready for further processes.[10]

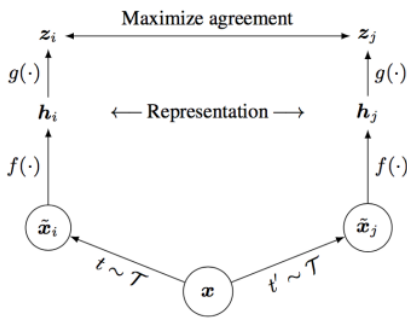


Figure 2. The basic structure of SimCLR

#### D. Feature dimensionality reduction

Because the representation vectors from ResNet are usually handled in 512 dimensions, it is

too much for K-means to determine the Euclidian Distance in between embeddings. Therefore, the need for Singular Value Decomposition (SVD) starting to realize. Principle Component Analysis (PCA), being the most widely used SVD methods, find the directions of maximum variance in high-dimensional data and projects into a new space with lower dimensions. With this mechanism, the PCA algorithm can reduce dimension whilst preserving the feature of original data.[12]

#### E. Elbow Method

When the 512 dimension embeddings has become 2 dimensional, k-means can calculate the with-in-clusters-sums-of-squares between the embeddings. Each distance will be recorded in a list, and then being draw by matplotlib to show a descending curve with decreasing gradient. The curve will inevitably encounter vertexes, and the rightmost whilst obvious turning point hints at the optimum number of clusters.[13]

#### F. K-means for labelling

Once the optimum number of clusters are determined by the Elbow Method, K-means can calculate the average instances of every data points, and hence compute the centroids. With the Centroids and average Euclidian Distances, All data points can be grouped into clusters.[14]

### III. Experiments

#### A. Dataset

Firstly, before any machine learning can took place, a suitable dataset has to be chosen. Therefore, I went to Kaggle.com which contains many excellent image datasets, and picked a peculiar image dataset of my interest. Since this experiment aims to simulate a real life event in which no labels are present alongside the images, there was one particular dataset called "Re:Zero Rem Anime Faces For GAN Training" fits my requirement. Although the title of which stated that the intended use of this dataset is for training Generative Adversarial Networks, but the content is pure images, which is suitable for this experiment. Additionally, this dataset contains 725 illustrates of the character Rem.

Secondly, I used the Mnist dataset for validation purposes, which follows the tradition of machine learning. This experiment took 600 unique samples from the training set (60 images for each digit from 0 to 9) and send through the CNN without label for generating embeddings.

#### B. Evaluating indicator

The First experiment on the first dataset "Re:Zero Rem Anime Faces For GAN Training" will only be valued using qualitative descriptions ,



because the ground truth labels simply does not exist.

The second experiment using second dataset: The Mnist Dataset will compare the label from self labelling process with the truth label, and an ultimate accuracy will be determined.

### C. Performance analysis

#### Experiment 1:

We first compare the difference of various embedding, whose results can be seen in Table 1. The Table 1 above shows the 2 dimensional embeddings for the first 50 images with correspondence to their filename. The lowest number in embedding\_0 is -27.96, the highest number in embedding\_0 is 44.78; the lowest number in embedding\_1 is -10.59, the highest number in embedding\_1 is 21.57. The fact that there is a balance between positive and negative embeddings explains the ResNet has worked favorably as it tends to make the data close to 0. Also, from the numerical data itself, we could see a reasonable range of difference from those images, shows that there is some pattern recognition that is effective inside the entire SimCLR model.

filenames	embedding_0	embedding_1	filenames	embedding_0	embedding_1
100.jpg	-1.621	-2.345	1329.jpg	-3.869	0.752
1009.jpg	-8.940	0.531	1334.jpg	-13.930	-3.161
1019.jpg	35.658	2.587	1344.jpg	-6.432	0.259
1058.jpg	18.753	3.100	1351.jpg	13.776	-4.250
1086.jpg	20.827	4.103	1362.jpg	-13.626	1.236
1103.jpg	-27.965	16.951	1365.jpg	3.413	16.700
1108.jpg	-14.764	-6.185	1369.jpg	-6.275	-2.229
1130.jpg	-7.224	-5.051	1372.jpg	-10.156	-4.911
1147.jpg	-11.591	0.644	1394.jpg	-20.641	11.713
1149.jpg	-8.651	-2.915	1401.jpg	7.695	-3.832
1154.jpg	-4.801	-0.934	1414.jpg	10.130	-2.076
116.jpg	-11.506	-10.597	1429.jpg	24.583	-7.485
1160.jpg	-14.631	-1.393	1445.jpg	26.404	-4.887
1161.jpg	13.360	0.769	145.jpg	-13.806	-4.855
1169.jpg	-4.956	-6.823	1480.jpg	-14.860	0.192
1178.jpg	10.059	-6.182	1490.jpg	-3.958	-7.984
1179.jpg	-18.677	3.034	1493.jpg	16.741	-2.520
1181.jpg	3.376	-1.841	1495.jpg	7.181	-7.117
1213.jpg	-4.942	-1.728	1501.jpg	3.208	-1.141
1249.jpg	13.947	7.080	1538.jpg	-6.098	-6.400
1270.jpg	-7.540	-2.161	1541.jpg	-7.439	-0.556
1277.jpg	-17.129	-1.311	1554.jpg	-8.280	-3.257
1285.jpg	-13.422	-0.570	1556.jpg	3.296	-6.205
1296.jpg	-5.595	-4.647	156.jpg	-6.692	-2.218
1302.jpg	-10.879	-1.644	1561.jpg	8.660	-6.837

Table 1 Display of embeddings generated by CNN

Apart from initial embeddings, the plot using K-Nearest Neighbors method further validates that the Collate function actually worked. From the examples, it is easy to see that some images with white background are considered as close neighbor with colored backgrounds. This shows that the importance of colors are reduced, which is a desired effect. Also, as the Figure 3 shown, we could observe that two same picture is considered as close neighbor of each other, proves the representation learning process is effective.



Figure 3 The K-Nearest Neighbor Test on the embeddings

As the Figure 4 shown, the Elbow method worked perfectly, the within-clusters-sums-of-squares follows the declining pattern. And it is possible to spot a clear vertex with reasonable number of clusters. If the measure goes beyond the integer of 9, the within-clusters-sums-of-squares almost does not improve thereafter, so 9 is the last meaningful number of clusters to choose from.

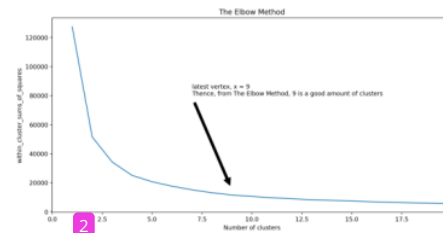


Figure 4 Elbow Method in determining optimum number of clusters

The Figure 5 displays the distribution of images according to their position in the latent space. The k-means algorithm created 9 distinct areas to cluster images according to their Euclidian distances relative to centroids. The 2 dimensional embeddings on both axes indicate that the conjunction of Representation learning and Clustering achieved a phenomenon effect in the pattern recognition of pictures. On the technical level, it is successful in rendering such descent result from pure pictures.

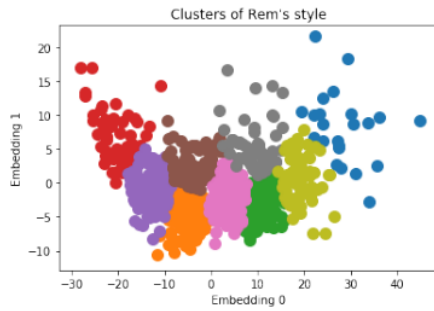


Figure 5 The distribution of images in the latent space with allocated clusters shown by coloring

Adding on to the prior 2 dimensional embeddings, a new row of actual label is appended to the pandas dataframe.(Table 2) With the data from k-means computation, each image has obtained its corresponding label from 0 to 8 (See Figure 6). Then, images can be easily printed into clusters according to the given labels. The array of images shows the clustered groups of images. Some of the clusters really does have semantically meaningful implications: for example, Group 0 has greater depth in the colors, Group 2 mostly involves smiley faces, Group 3 has the lightest color overall. Other than these mentioned groups, the clustering is not meaningful in the semantic sense.

file name	Embedding_0	embedding_1	Label	file name	embedding_0	embedding_1	Label
100.jpg	-1.621	-2.345	1	1329.jpg	-3.869	0.752	5
1009.jpg	-8.940	0.531	4	1334.jpg	-13.930	-3.161	4
1019.jpg	35.658	2.587	0	1344.jpg	-6.432	0.259	5
1058.jpg	18.753	3.100	8	1351.jpg	13.776	-4.250	2
1086.jpg	20.827	4.103	8	1362.jpg	-13.626	1.236	4
1103.jpg	-27.965	16.951	3	1365.jpg	3.413	16.700	7
1108.jpg	-14.764	-6.185	4	1369.jpg	-6.275	-2.229	1
1130.jpg	-7.224	-5.051	1	1372.jpg	-10.156	-4.911	4
1147.jpg	-11.591	0.644	4	1394.jpg	-20.641	11.713	3
1149.jpg	-8.651	-2.915	1	1401.jpg	7.695	-3.832	2
1154.jpg	-4.801	-0.934	1	1414.jpg	10.130	-2.076	2
116.jpg	-11.506	-10.597	1	1429.jpg	24.583	-7.485	8
1160.jpg	-14.631	-1.393	4	1445.jpg	26.404	-4.887	8
1161.jpg	13.360	0.769	2	145.jpg	-13.806	-4.855	4
1169.jpg	-4.956	-6.823	1	1480.jpg	-14.860	0.192	4
1178.jpg	10.059	-6.182	2	1490.jpg	-3.958	-7.984	1
1179.jpg	-18.677	3.034	3	1493.jpg	16.741	-2.520	8
1181.jpg	3.376	-1.841	6	1495.jpg	7.181	-7.117	2
1213.jpg	-4.942	-1.728	1	1501.jpg	3.208	-1.141	6

1249.jpg	13.947	7.080	7	1538.jpg	-6.098	-6.400	1
1270.jpg	-7.540	-2.161	1	1541.jpg	-7.439	-0.556	1
1277.jpg	-17.129	-1.311	4	1554.jpg	-8.280	-3.257	1
1285.jpg	-13.422	-0.570	4	1556.jpg	3.296	-6.205	6
1296.jpg	-5.595	-4.647	1	156.jpg	-6.692	-2.218	1
1302.jpg	-10.879	-1.644	4	1561.jpg	8.660	-6.837	2

Table 2 Picture and embeddings with respect to their labels



Figure 6 The display of images in labeled clusters

## Experiment 2:

Thereafter qualitative analysis has pointed out some advantages and disadvantages of the model, here is qualitative data for support.

In Table 3, the largest embedding in embedding\_0 is 22.981, the smallest is -10.441. The largest

embedding in embedding\_1 is 18.577, the smallest is -13.825. Indeed, the range of embeddings are much smaller than the previous experiment on anime pictures because the images of digits contains lesser depth of information.

filenames	embedding_0	embedding_1	filenames	embedding_0	embedding_1
img_0.jpg	-3.722	8.337	img_120.jpg	-6.449	4.256
img_1.jpg	6.605	1.984	img_121.jpg	3.291	-4.247
img_10.jpg	10.650	-5.056	img_122.jpg	-1.281	-2.825
img_100.jpg	-2.905	-7.567	img_123.jpg	1.860	3.284
img_101.jpg	0.174	3.295	img_124.jpg	1.672	14.258
img_102.jpg	-6.344	-7.786	img_125.jpg	-0.745	-1.540
img_103.jpg	-7.459	2.486	img_126.jpg	-0.122	7.027
img_104.jpg	4.936	-6.656	img_127.jpg	-1.885	-4.170
img_105.jpg	1.595	-5.666	img_128.jpg	-0.398	-3.336
img_106.jpg	0.927	-3.335	img_129.jpg	-6.528	-2.761
img_107.jpg	-6.855	-5.396	img_130.jpg	3.115	-3.190
img_108.jpg	9.486	0.236	img_131.jpg	1.023	4.859
img_109.jpg	5.191	0.991	img_132.jpg	6.577	-4.512
img_11.jpg	2.287	-3.386	img_133.jpg	9.255	0.907
img_110.jpg	14.904	2.755	img_134.jpg	5.150	-2.317
img_111.jpg	-4.351	-5.878	img_135.jpg	2.395	14.895
img_112.jpg	-1.666	-5.853	img_136.jpg	-2.651	4.497
img_113.jpg	-6.222	-2.943	img_137.jpg	-4.453	6.881
img_114.jpg	-6.662	-3.927	img_138.jpg	-8.245	-3.338
img_115.jpg	-1.782	-0.319	img_139.jpg	-0.488	2.408
img_116.jpg	-1.478	1.048	img_140.jpg	-9.903	-4.523
img_117.jpg	-2.405	0.554	img_141.jpg	-3.252	-9.120
img_118.jpg	0.433	16.508	img_142.jpg	0.663	15.684
img_119.jpg	-4.342	-2.090		15.014	4.281
img_12.jpg	-5.407	12.334		-3.356	5.164

Table 3 Display of embeddings generated by CNN

Because the images themselves contains digits, the performance of the experiment is clearly distinguishable. From figure 7, there is an inconsistency in the clustering of numbers: the digit "1" can be seen as clustered very well. The closest neighbor test has shown its 5 closest neighbor to be 1 itself, or something peculiarly similar to it, for example, the vertically extended "5" appear in Figure 7. Then, numbers with slightly more complex structure, in which case, is "2" and "7" distorted together. Others, with much more complex structure than "1" are entangled together, and unable to be distinguished clearly by the K-nearest neighbor test.

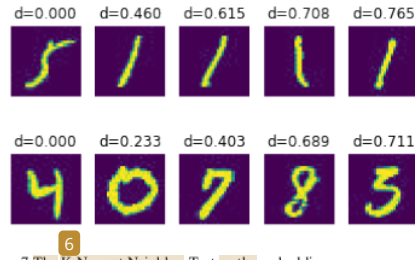
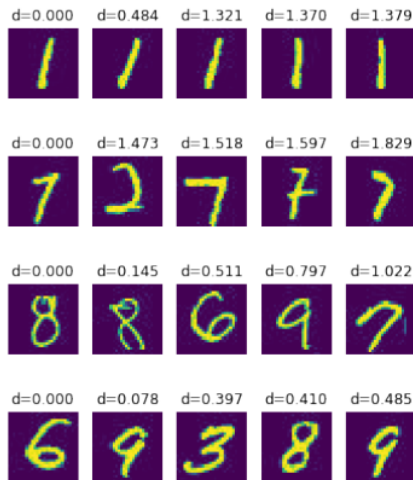


Figure 7 The K-Nearest Neighbor Test on the embeddings

Just for the purpose of showing the capability of the Elbow method, I have run the test even though the number of clusters are already understood. In Figure 8, The curve has shown the latest vertex when  $x=10$ , and hence the validity of this method is proven.

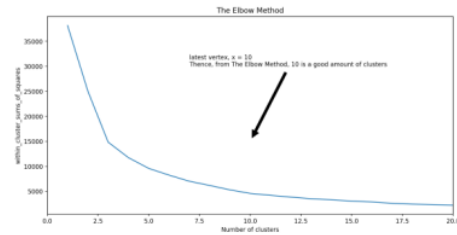


Figure 8 Elbow Method in validating optimum number of clusters

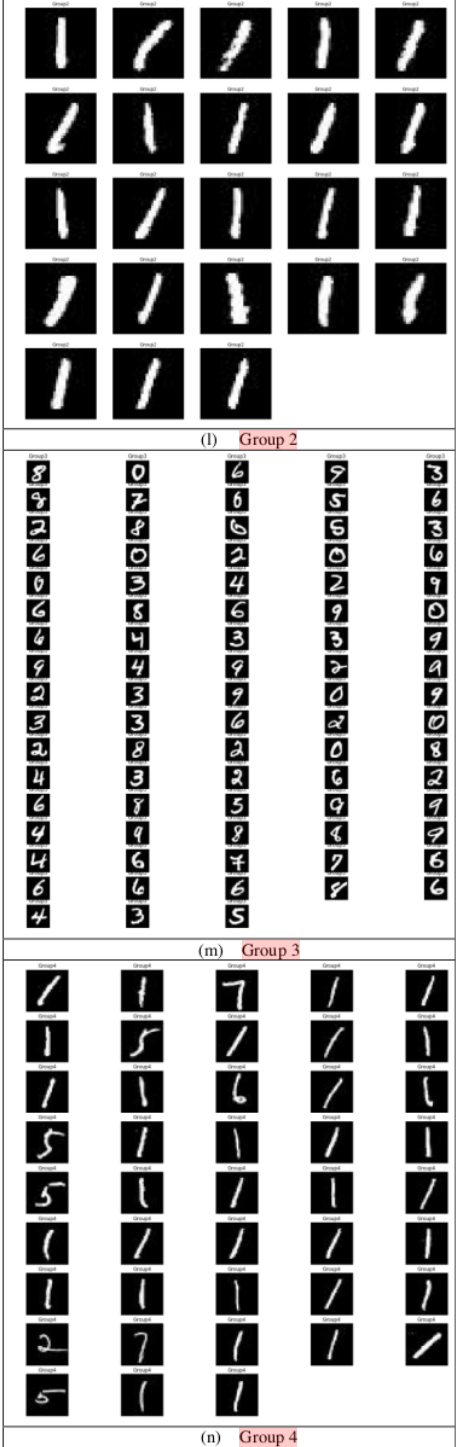
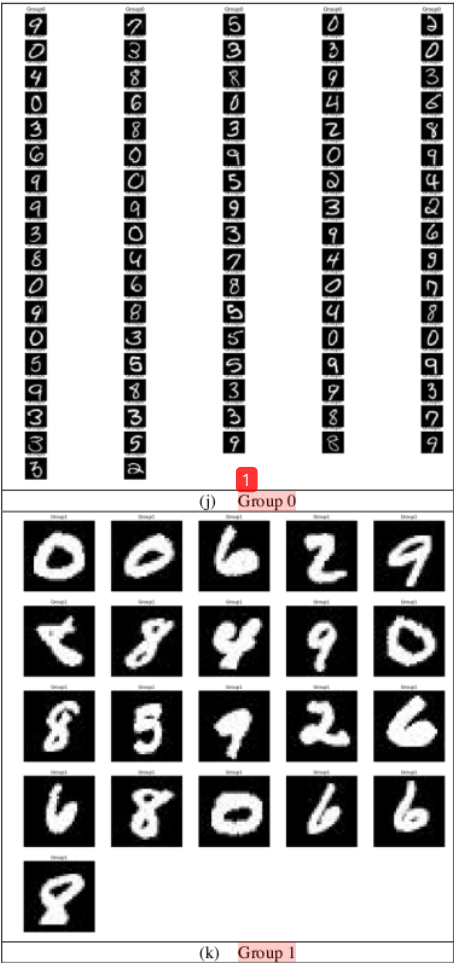
The locality of different images in the latent space provided by K-means Algorithm is shown in Figure 9. Although the area grouped is different from the first experiment, but conversion of embeddings into labels is again successful.



Figure 9 The distribution of images in the latent space with allocated clusters shown by coloring

In Figure 10, just like the results in K-nearest neighbor test, only the digit 1 are distinguished quite well, and even some other digits like 2, 5, 7 are mixed with 1. This presents the inability for SimCLR model to distinguish complex structure in the digits.

In this case, the labels generated by this self-supervised methods are not necessarily directly corresponds with the truth label (as anyone could tell by seeing the Figure 10), because computers themselves cannot determine the digit in the image and name the cluster with that digit. So, the clustered images are being relabeled manually by the most frequent digits that appeared in the clusters.





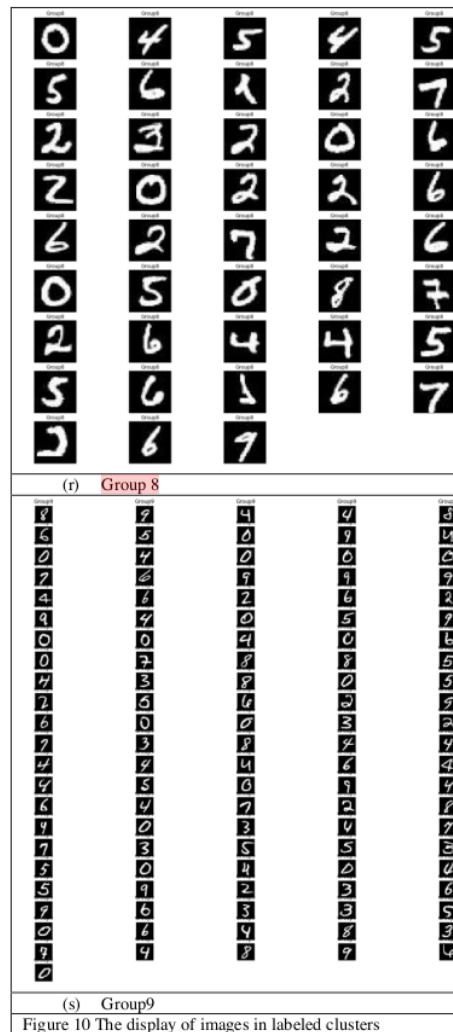
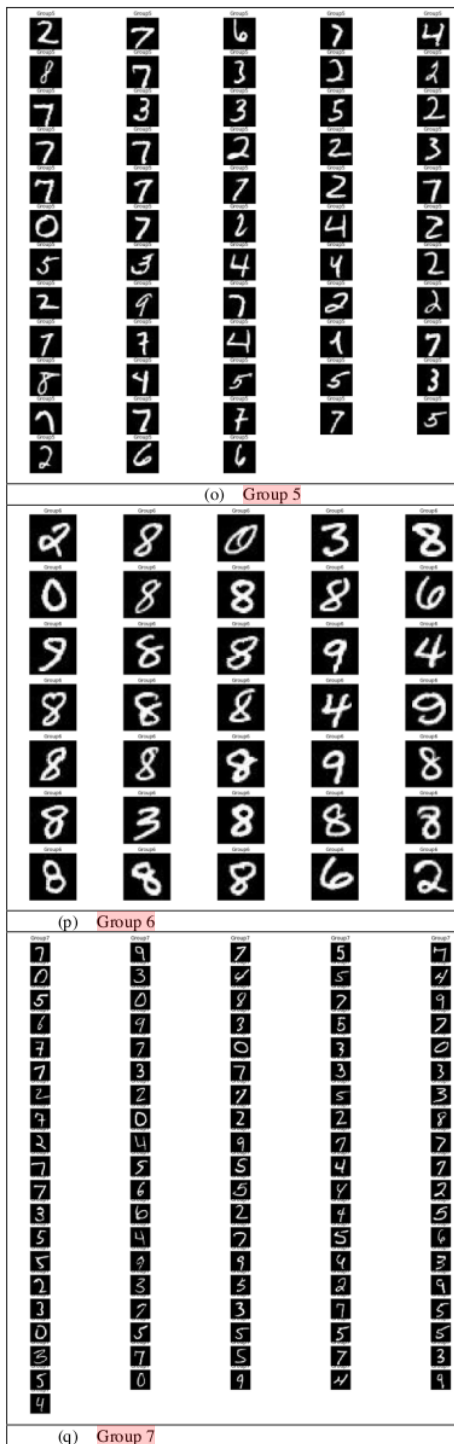


Figure 10 The display of images in labeled clusters

The Table 4 below is the display of comparison between labels generated by the model and the truth label comes with the Mnist dataset. By comparing two columns, there is 192 matching columns out of 600 columns, giving an accuracy of 32%.

Filenames	embedding_0	embedding_1	trained labels	truth label	T/F
img_78.jpg	-4.506	-2.823	4	4	1
img_79.jpg	-1.221	14.083	1	1	1
img_8.jpg	-2.724	-6.578	3	5	0
img_80.jpg	-3.601	4.483	7	5	0
img_81.jpg	-4.906	-1.161	6	4	0
img_82.jpg	0.182	-3.155	4	8	0
img_83.jpg	-2.624	-6.740	3	9	0
img_84.jpg	-5.043	4.358	7	2	0
img_85.jpg	-5.780	-0.094	7	9	0
img_86.jpg	0.299	-5.127	4	9	0
img_87.jpg	-8.686	-5.541	3	8	0
img_88.jpg	-6.377	-6.586	3	9	0
img_89.jpg	-2.749	-1.391	4	6	0
img_9.jpg	3.148	0.682	6	3	0
img_90.jpg	-5.304	-4.155	3	3	1
img_91.jpg	-0.064	2.522	7	6	0
img_92.jpg	-9.297	-1.576	7	4	0

img_93.jpg	-1.670	4.989	7	6	0
img_94.jpg	-2.373	-8.701	3	2	0
img_95.jpg	6.369	4.394	2	9	0
img_96.jpg	-6.858	10.611	1	1	1
img_97.jpg	1.069	-6.830	8	2	0
img_98.jpg	-0.998	-4.064	4	0	0
img_99.jpg	6.914	-2.226	6	5	0
					192
					32%

Table 4 the comparison between trained labels and truth label

#### IV. Discussion

At least partially, the experiment has resulted successful result, the clustering for obvious differences are persuasive. This is because the Representation Learning process is successful in when images are largely different. However, during the other half of the occasion where SimCLR handles with more complex images, the Representation learning process is not effective enough, with consideration of the subsequent clustering should be flawless. This is due to the SimCLR model is formed based on the assumption that there is a balanced distribution of different kinds of images across the entire dataset, unlike the everyday situation, where people cannot guarantee the distribution.[10]

#### V. Conclusion

The entire experiment is a simulation of a possible simple, quick self-labeling process in everyday life for separating largely distinct objects. Such requirement to organize people's picture storage in smartphones are to an extent achievable. And whilst helping people to select the favorable online persona, it is also usable. The future of this entire thought process can be integrated and implement into mobile programs that is much more user-friendly than these bare codes. But before any commercialized usage can be put forward, the accuracy of the image classification with semantically similar pictures should improve. Secondly, I am expecting the development of self-labelling can be more related to people's social life, so that more commercial opportunities can stimulate the development.

#### References:

- [1] L. Liu, Y. Wang and W. Chi, "Image Recognition Technology Based on Machine Learning," in IEEE Access, doi: 10.1109/ACCESS.2020.3021590.
- [2] O'Shea, Keiron and Ryan Nash, "An Introduction to Convolutional Neural Networks." ArXiv abs/1511.08458 (2015): n. pag.
- [3] Krizhevsky, A., Sutskever, I., Hinton, G.E., "Imagenet classification with deep convolutional neural networks." NIPS (2012)
- [4] Simonyan, K., Zisserman, A., "Very deep convolutional networks for large-scale image recognition." ICLR (2015)
- [5] He, K., Zhang, X., Ren, S., Sun, J., Deep residual learning for image recognition. CVPR (2016)
- [6] Xie, S., Girshick, R., Dollár, P., Tu, Z., He, K.: Aggregated residual transformations for deep neural networks. CVPR (2017)
- [7] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z., "Rethinking the inception architecture for computer vision." CVPR (2016)
- [8] Rasmus, Antti, Mathias Berglund, Mikko Honkala, Harri Valpola, and Tapani Raiko, "Semi-supervised learning with ladder networks." Advances in neural information processing systems 28 (2015)
- [9] Park, T., Efros, A.A., Zhang, R., Zhu, J.Y., "Contrastive Learning for Unpaired Image-to-Image Translation." Computer Science(), vol 12354. Springer, Cham. (2020)
- [10] Chen, T., Kornblith, S., Norouzi, M., and Hinton, G., "A Simple Framework for Contrastive Learning of Visual Representations", <i>arXiv e-prints</i>, 2020.
- [11] He, K., Zhang, X., Ren, S., and Sun, J., "Deep Residual Learning for Image Recognition", <i>arXiv e-prints</i>, 2015.
- [12] Abdi, Hervé, and Lynne J. Williams, "Principal component analysis." Wiley interdisciplinary reviews: computational statistics 2, no. 4 (2010): 433-459.
- [13] Bholowalia, Purnima, and Arvind Kumar, "EBK-means: A clustering technique based on elbow method and k-means in WSN." International Journal of Computer Applications 105, no. 9 (2014).
- [14] Teknomo, Kardi, "K-means clustering tutorial." Medicine 100, no. 4 (2006): 3.
- [15] Kán, Peter & Kafumann, Hannes. (2019). DeepLight: light source estimation for augmented reality using deep learning. The Visual Computer. 35. 10.1007/s00371-019-01666-x.

# Self-supervised Image Classification using Convolutional Neural Network

## ORIGINALITY REPORT

4%

SIMILARITY INDEX

2%

INTERNET SOURCES

3%

PUBLICATIONS

1%

STUDENT PAPERS

## PRIMARY SOURCES

1

[pdfs.semanticscholar.org](https://pdfs.semanticscholar.org)

Internet Source

1%

2

Stevan Milovanović, Zorica Bogdanović, Aleksandra Labus, Marijana Despotović-Zrakić, Svetlana Mitrović. "Social recruiting: an application of social network analysis for preselection of candidates", Data Technologies and Applications, 2022

Publication

<1%

3

[3dvar.com](https://3dvar.com)

Internet Source

<1%

4

Submitted to Georgia Institute of Technology Main Campus

Student Paper

<1%

5

[journalofbigdata.springeropen.com](https://journalofbigdata.springeropen.com)

Internet Source

<1%

6

W. Bruce Croft. "Combining classifiers in text categorization", Proceedings of the 19th annual international ACM SIGIR conference on

<1%

## Research and development in information retrieval - SIGIR 96 SIGIR 96, 1996

Publication

7

"Medical Image Computing and Computer Assisted Intervention – MICCAI 2020", Springer Science and Business Media LLC, 2020

Publication

<1 %

8

Submitted to Gwangju Institute Science and Technology

Student Paper

<1 %

9

"Medical Image Computing and Computer Assisted Intervention – MICCAI 2018", Springer Science and Business Media LLC, 2018

Publication

<1 %

10

Mingfu Xiong, Dan Chen, Jun Chen, Jingying Chen, Benyun Shi, Chao Liang, Ruimin Hu. "Person re-identification with multiple similarity probabilities using deep metric learning for efficient smart security applications", Journal of Parallel and Distributed Computing, 2019

Publication

<1 %

11

[www.springerprofessional.de](http://www.springerprofessional.de)

Internet Source

<1 %



---

Exclude quotes      Off

Exclude matches      Off

Exclude bibliography      On