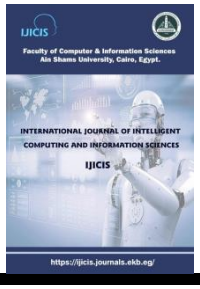
IJICIS, Vol.23, No.2, 62-73
DOI: 10.21608/ijicis.2023.193520.1255

International Journal of Intelligent Computing and Information Sciences

https://ijicis.journals.ekb.eg/



EXPLORING SELF-SUPERVISED PRETRAINING DATASETS FOR COMPLEX SCENE UNDERSTANDING

Yomna A. Kawashti*

Computer Science Department
Faculty of Computer and Information Sciences, Ain Shams University,
Cairo, Egypt
yomna.ahmed@cis.asu.edu.eg

Dina Khattab

Scientific Computing Department
Faculty of Computer and Information Sciences, Ain Shams University,
Cairo, Egypt
dina.khattab@cis.asu.edu.eg

Mostafa M. Aref

Computer Science Department,
Faculty of Computer and Information Sciences, Ain Shams University,
Cairo, Egypt
mostafa.aref@cis.asu.edu.eg

Received 2023-02-19; Revised 2023-05-02; Accepted 2023-05-05

Abstract: *With the rapid advancements of deep learning research, there have been many milestones achieved in the field of computer vision. However, most of these advances are only applicable in cases where hand-annotated datasets are available. This is considered the current bottleneck of deep learning that self-supervised learning aims to overcome. The self-supervised framework consists of proxy and target tasks. The proxy task is a self-supervised task pretrained on unlabeled data, the weights of which are transferred to the target task. The prevalent paradigm in self-supervised research is to pretrain using ImageNet which is a single-object centric dataset. In this work, we investigate whether this is the best choice when the target task is multi-object centric. We pretrain "SimSiam" which is a non-contrastive self-supervised algorithm using two different pretraining datasets: ImageNet100 (single-object centric) and COCO (multi-object centric). The transfer performance of each pretrained model is evaluated on the target task of multi-label classification using PascalVOC. Furtherly, we evaluate the two pretrained models using CityScapes; an autonomous driving dataset in order to study the implications of the chosen pretraining datasets in different domains. Our results showed that the SimSiam model pretrained using COCO consistently outperformed the ImageNet100 pretrained model by ~+1 percent (57.4 vs 58.3 mAP for CityScapes). This is significant since COCO is smaller in size. We conclude that using multi-object centric datasets for pretraining self-supervised learning algorithms is more efficient in cases where the target task is multi-object centric and in complex scene understanding tasks such as autonomous driving applications.*

Keywords: *self-supervised learning, transfer learning, scene understanding, SimSiam, autonomous driving*

*__Corresponding Author__: Yomna A. Kawashti

Computer Science Department, Faculty of Computer and Information Science, Ain Shams University, Cairo, Egypt

Email address: yomna.ahmed@cis.asu.edu.eg

1. Introduction

Great advances were made in the field of computer vision in the last decade. These achievements were mostly possible due to two main factors: the rapid evolution of Convolutional Neural Network (CNN) architectures and the publishing of several large-scale annotated datasets, most importantly, ImageNet [1]. Models pretrained on large scale annotated datasets are widely used in deep learning to achieve good results through transfer learning [2]–[5]. Though this paradigm proved to be extremely efficient, however, it suffers from a bottleneck, (i.e., a large enough annotated dataset needs to be available to achieve competitive results). Hand-annotated datasets are very expensive and time consuming to create, especially in specialized domains such as fashion, medicine, and autonomous driving. Even with the availability of an annotated dataset, there is still the disadvantage of not being able to utilize the massive amounts of unlabeled data being generated every second [6]. Self-supervised learning aims to solve these issues present in current supervised paradigms by leveraging and learning from unlabeled data. In self-supervised learning, a deep learning model is trained on a proxy/pretext task using unlabeled data. The term self-supervised means that, though the data is unlabeled, pseudo labels are automatically derived from the data, so the task is self-supervised. This pretrained model is then utilized to improve the results on a target/downstream task. Self-supervised learning has gained traction in recent years with several benchmarks achieved in the domain [7], [8]. Autonomous driving is one of the main domains that could benefit from self-supervised learning due to the massive amounts of street footage that are collected continuously using traffic cameras and car dashcams (dashboard cameras). Though this data would usually take years to annotate, the hope with self-supervised learning is that it could be used as soon as the data is generated. Several works were interested in applying self-supervised learning in this domain [9] [10].

ImageNet is primarily used in State-Of-The-Art self-supervised methods in the proxy phase [8], [11], [12]. Though, there is no need for its annotations, it is still used as the benchmark due to its size, availability and spectrum. Most works are continuously focusing on generating competitive results using ImageNet as the pretraining dataset. Fewer research efforts were concerned with the implications of pre-dominantly using ImageNet in pretraining on complex downstream tasks [9], [13]. Since ImageNet is mostly an object centric dataset (i.e., the focus of the image is primarily a single object), it is worth investigating whether models pretrained on ImageNet transfer well to multi-object centric downstream tasks such as multi-label classification, object detection and segmentation.

In this work, we investigate this question by pretraining a self-supervised model (SimSiam [8]) using both a single-object centric dataset (ImageNet) and a multi-object centric dataset (MS COCO [14]). We choose the downstream task of multi-label classification to compare the performance of these pretrained models to investigate how the nature of the pretraining dataset affects the downstream performance. We evaluate the trained models in the downstream task using CityScapes dataset [15] for autonomous driving . Autonomous driving applications require complex scene understanding [16] and the images are mostly multi-object centric. The outline of this paper is as follows, in Section 2, we discuss the relevant works to our study. Section 3 illustrates the details of our experiment setup including the proxy task, the downstream tasks and the used datasets. The results are provided and analyzed in section 4 and compared with related work. Finally, in section 5 our conclusions are provided.

2. Background and Related Work

Self-supervised learning algorithms can be considered a subset of unsupervised learning. The standard self-supervised framework consists of two main components: the proxy/pretext task and the target/downstream task [6]. The term self-supervision refers to the proxy task which is supervised but the labels are automatically derived from the image data itself. Usually, a self-supervised algorithm is applied to a large unlabeled dataset to solve the proxy task. A CNN backbone is used within the algorithm to learn how to extract meaningful feature representations through the pretext task. This knowledge is then transferred to the target task, which is commonly applied on a small, labeled dataset. In this way, self-supervised learning can leverage the immense amount of unlabeled data that is constantly generated. This area of research has caught the interest of many researchers and witnessed a surge in State-Of-The-Art proposals in recent years[7]–[9]. Self-supervised learning can be divided into two categories: discriminative and generative [17]. In this work we are more concerned with the discriminative self-supervised algorithms.

Most self-supervised algorithms utilize a Siamese network to solve the self-supervised task. A Siamese network is an architecture that consists of identical subnetworks sharing the same weights (2 subnetworks or more). In earlier works, the pretext task involved generating explicit pseudo labels. The Jigsaw task [18] divided an image into 9 tiles and shuffled them around according to a random permutation. The CNN architecture was used to solve the puzzle by predicting the correct permutation. This was applied using a 9-way Siamese network. In this task, the "pseudo label" is the permutation. The label is explicitly generated and utilized. Several other works followed this pattern such as the rotation [19] and relative patch location [20]. Even though these works were efficient as a proof of concept for the self-supervised paradigm, they suffered from a specialization issue. It was observed that the features extracted from the convolutional layers closer to the classification layer would be more specific to the task at hand (e.g. solving the jigsaw puzzle) instead of having the generalization ability required to work well in transfer learning. This pushed the research in the direction of contrastive learning.

Contrastive learning was researched as a method to overcome the problem of degradation of feature representation quality that existed in classical self-supervised methods. This was achieved by designing the proxy task with the objective of feature representation learning. In most contrastive learning algorithms, a Siamese network is used to push the feature representations of similar images together and push the feature representations of dissimilar images away from each other [11], [21]. In this paradigm, data augmentation is applied to an image to generate a transformed view of the same image. The Siamese network is fed with two views of the image and their feature representations are extracted and contrasted. If both views are of the same image (i.e., the positive example), then, the contrastive loss will push the Siamese network to extract their feature representations to be more similar, otherwise, in the case of a negative example (i.e., two different images), the feature representations will be pushed apart. PIRL [21] is one of the earlier contrastive self-supervised methods that utilized this paradigm. Figure 1 shows a comparison from [21] between the standard pretext task and the one proposed in PIRL where "I" is the original image, and "I^t" is the transformed view of the image.

Several State-Of-The-Art contrastive methods were proposed after PIRL such as SimCLR [11] and MOCO [12]. These algorithms proved very efficient and generated competitive results. However, they had a significant drawback which is the need to generate a large-enough pool of negative samples to learn well. To overcome this, contrastive learning algorithms rely on the use of very large batch sizes or memory banks. Both solutions are computationally expensive. Non-contrastive methods [7], [8]emerged as a possible way to solve this issue. Where the authors in [7], [8] found that a Siamese network could

be taught to extract meaningful representations using positive samples only, thus reducing the computational overhead.

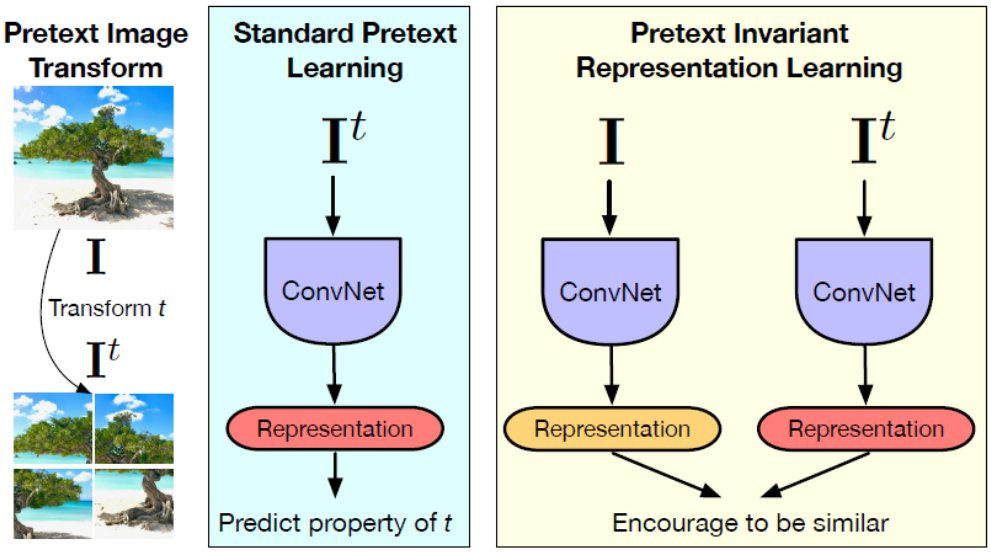


Figure 1. Comparison between standard self-supervised method and PIRL [21]

SimSiam [8] is one of these non-contrastive self-supervised learning techniques. In SimSiam, the task is essentially to predict the feature representation of an image from another view of the same image. Figure 2 shows the architecture of "SimSiam". Different data augmentation techniques are applied sequentially to image "x" to generate two augmented views of the image "x1" and "x2" each generated with random parameters for the augmentation operation. These two views are fed into a two-way Siamese network with a CNN backbone such as Res50. The CNN backbone is represented by encoder "f" in the figure. The encoded output is then fed to a prediction layer which is a multilayer perceptron (MLP). The negative cosine similarity between the predicted output from the first branch and the encoded output from the second branch is calculated and the error is backpropagated to update the weights accordingly. In practice, both encoded outputs are fed to predictor "h" and the average cosine similarity is acquired. Additionally, the authors found that applying a stop-gradient operation on the second branch was essential for the model to avoid collapsing. Model collapse occurs when the self-supervised algorithm generates the same feature representations for all images in order to minimize the objective function. A stop-gradient operation means that the output of the second branch is treated as a constant and does not contribute to the gradient calculation used in backpropagation.

Downstream tasks such as image classification and object detection are typically used to evaluate the generalization ability of the weights learnt by the self-supervised model. According to [17], there are several paradigms for testing this generalization ability. The two main paradigms are finetuning and linear classification. Finetuning means that the weights of the CNN backbone trained on the self-supervised task are extracted and loaded to the downstream task model that contains the same backbone (e.g., Res50). The backbone resumes training such that the weights learnt during the proxy task are used as initialization. This finetuning paradigm generates competitive results, however, it is not considered an accurate understanding of how good the feature representations are on their own. In the linear classification evaluation paradigm, the weights of the CNN backbone are used without finetuning to

extract the feature representations of a classification dataset. A linear classification model such as SVM or logistic regression is trained on these extracted features and the testing accuracy is evaluated.

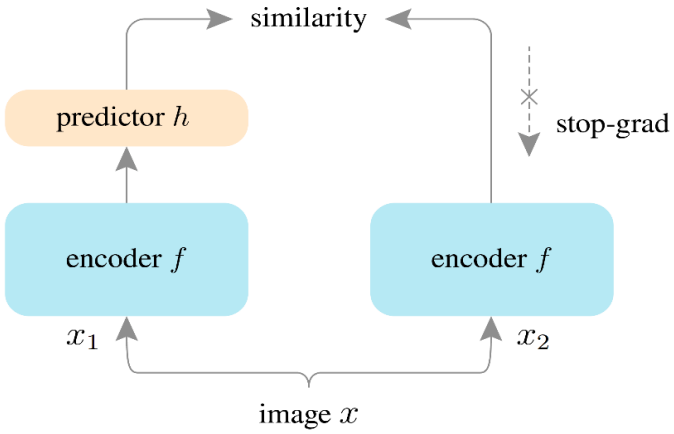


Figure 2. SimSiam Architecture [8]

Image classification can be divided into two main categories: Single Label Image Classification (SLIC) and Multi-Label Image Classification (MLIC) [22]. In SLIC, each image represents a single category/class and thus has a single label. In MLIC, each image contains several categories and thus has several labels. This means that, by definition, the task of multi-label classification requires the trained models to have more complex scene understanding than single label classification. The prevalent paradigm in self-supervised learning is to pretrain on ImageNet (an object centric dataset) and thus focus in the downstream task on ImageNet linear classification. Although other complex downstream tasks such as object detection and segmentation are also evaluated, however, they are rarely given the focus in the design of the pretraining framework.

Some researchers investigated the idea of changing the pretraining dataset or altering the self-supervised algorithm to better accommodate multi-object downstream tasks [9], [23]. In [13], the authors investigated the effect of pretraining transformer-based self-supervised models using different subsets of ImageNet and compared these results with using COCO (multi-object centric). They evaluated the transfer performance for downstream tasks mainly using image classification on the iNaturalist dataset as well as object detection and instance segmentation on the COCO dataset. The authors found that the COCO pretrained model achieved better results than the ImageNet pretrained model in the case of downstream evaluation on the same pretraining dataset (COCO). In [9] the authors proposed Multi-instance Siamese network (MultiSiam). MultiSiam is an alteration to the self-supervised task design that takes into account the existence of multi-objects in the same image by combining several techniques. These techniques include Intersection Over Union (IoU) filtering for generating image crops, feature map alignment and clustering. The authors also experimented with using large scale autonomous driving datasets in pretraining instead of ImageNet and found that it achieved better overall performance. A summarized comparison in both papers' setups is provided in Table 1 as well some of each papers' best acquired results for COCO object detection. It is important to note that dissimilar to our work in which we used multilabel classification for the downstream valuation on PascalVOC and CityScapes, the authors in [9] used the tasks of object detection for PascalVOC and semantic and instance segmentation for CityScapes.

Table 1. Comparison between multi-object centric related works

Work	Self-Supervised Algorithm	Pretraining Dataset	Downstream Dataset	COCO Downstream Performance (AP^b)
Chen et al. [9]	MultiSiam	Waymo, SODA10M	PascalVOC, CityScapes, BDD100K, **COCO**	42.1
El-Nouby et al. [13]	BeiT SplitMask	IN 1%, 10% and Full IN, COCO	iNaturalist, Food101, Stanford Cars, DomainNet subsets, **COCO**	46.8

3. Experiment Setup

In this work, we use SimSiam as our self-supervised algorithm. We use a complex scene understanding task as our downstream task which is multi-label classification. This task is inherently multi-object centric. We investigate the question of whether using a single-object centric dataset such as ImageNet [1] for pretraining would translate to better downstream performance than pretraining on multi-object centric dataset such as COCO [14]. Additionally, we further investigate the effect of such training on an autonomous driving dataset. This work differs from [13] as we focus on a non-contrastive self-supervised algorithm (SimSiam) instead of using transformer-based self-supervised algorithms. We also evaluate the transferability of the learnt models to a different domain dataset such as CityScapes [15]. In addition, our experiments differ from [9]since we use the original self-supervised algorithm (SimSiam) without any alterations. In the next sections we provide the details of our experiment setup including the proxy task setup (SimSiam), the downstream setup (MLIC) and the datasets used.

3.1 Proxy Task Setup

In the proxy task, we use SimSiam [8] as our self-supervised algorithm as provided in [24]. As previously mentioned, In SimSiam, several randomized data augmentations are applied to a given image to generate two views of it. The data augmentations applied are: random resized crop, horizontal flip, color jitter, random grayscale and random gaussian blur. We use Res18 as the CNN backbone since the datasets used in the pretraining phase are medium sized, so a shallower architecture performs better (Res50 is usually used with large datasets such as full ImageNet (ImageNet-1k) however Res18 is relatively used to train smaller datasets).

3.1.1 Proxy Task Datasets

We run the SimSiam training algorithm twice using two datasets: ImageNet100 (single-object centric dataset) and COCO (multi-object centric dataset). Table 2 shows the training configurations used in the pretext task to train both datasets.

Table 2. SimSiam pretraining hyperparameters

Hyperparameter	Value
Batch Size	512
Number of Epochs	100
Optimizer	SGD
CNN Backbone	Res18

ImageNet100 is a subset of ImageNet [1] that consists of 100 randomly chosen classes out of the original 1000 classes. Each category in ImageNet has approximately 1,300 images. There are several versions of ImageNet100. However, the most common version is the one used in [25]which we follow

in our experiments. This version contains 126,689 images in the training set. As we are working on self-supervised learning, this dataset is used without its labels. Figure 3a shows an ImageNet100 image sample.

Microsoft **Co**mmon **O**bjects in **Co**ntext **(MS COCO)** [14] is a benchmark dataset used for object recognition, detection and segmentation. It is, by definition, a multi-object centric dataset. There are several versions of COCO including "COCO2014" and "COCO2017". In our work, we use the 2017 version. It consists of 118, 287 images in the training subset and 5,000 images in the validation subset. In training SimSiam, we add both subsets to close up the size gap between ImageNet100 and COCO2017; so we train on COCO2017 "trainval" subset which consists of 123,287 images. Figure 3b shows an MS COCO image sample.

(a) (b)

Figure 3. Original samples from the pretraining datasets we used where (a) shows an ImageNet100 sample [1] and (b) shows an MS COCO sample [14].

3.2 Downstream Task Setup: MLIC

The weights of the SimSiam CNN backbone at epoch 100 are utilized to extract the feature representations for the downstream task datasets. Following the benchmark paradigm in [17], first, the CNN weights are loaded into the same CNN architecture as that of the encoder of the pretext task (i.e, the same CNN architecture as the pretext task without the Siamese structure and the MLP layers). The images in the training subset of the downstream dataset are fed into the CNN architecture and the feature representations for each training image is extracted from the last convolutional layer without any weight updates applied to the CNN model. Average pooling is applied on the extracted feature map to generate ~9k features for each image. These features, combined with the corresponding labels are used to train a linear SVM model. The same feature extraction method is used on the testing subset of the downstream dataset. Lastly, the trained SVM model is used to predict the labels of the extracted features from the testing subset. Mean Average precision (mAP) is used to evaluate the accuracy since this is a multi-label classification task. It is calculated according to equation 1 where the average precision "AP" is acquired for each class and then the mean is calculated for "n" which denotes the total number of classes.

$$mAP = \frac{1}{n}\sum_{k=1}^{k=n} AP_k \qquad (1)$$

3.2.1 Downstream Task Datasets

We evaluate each pretrained model on two downstream datasets: PascalVOC [26] and CityScapes [15].

Pascal **V**isual **O**bjects **C**lasses (PascalVOC) [26] is a benchmark dataset used for object class recognition. It contains 20 categories. It is a multi-object centric dataset as multiple categories and objects exist in the same image. We use the 2007 version of PascalVOC. The "trainval" subset is used for training SVM. This subset consists of 5,011 images while the test subset consists of 4,952 images and is used for evaluation. Figure 4 shows a sample image from PascalVOC and the muli-labels associated with it.

labels
dining table
person
chair
bottle

Figure 4. PascalVOC [26] Image Sample with its labels

CityScapes [15] is an urban scenes benchmark dataset widely used in autonomous driving applications. The dataset describes 30 classes. There are several subsets associated with CityScapes. We use the fine annotated version of CityScapes where we train the SVM model on the training subset that consists of 2,975 images and evaluate the model on the validation subset that consists of 500 images. Figure 5 shows a sample image from CityScapes and the multi-labels associated with it.

labels
person
building
traffic sign
bicycle
rider
parking
dynamic
license plate
car
static
vegetation
sky
road
pole
terrain

Figure 5. CityScapes [15] Image Sample with its labels

4. Results and Discussion

In this section, we present the results acquired from each pretraining experiment on both downstream datasets. Table 3 shows the SVM downstream accuracies on the PascalVOC dataset. For base-reference, we added the SVM classification results provided in [27] when SimSiam was trained for 100 epochs using the full ImageNet dataset (ImageNet-1k). It is important to note that there are significant differences in the setup we used such as the CNN backbone due to the scale of the pretraining datasets.

Table 3. PascalVOC SVM results

Pretraining Dataset	SimSiam CNN Backbone	SVM Downstream Accuracy (mAP)
ImageNet-1k [8][27]	Res50	84.64
ImageNet100	Res18	56.2
COCO	Res18	**57.6**

From these results, we can see that SimSiam pretrained on COCO (multi-object dataset) achieved higher performance than ImageNet100 when transferred to a multi-object downstream dataset such as PascalVOC. We believe this improvement is also significant considering that the COCO dataset was smaller in size than the ImageNet100 (i.e., SimSiam was trained on ~123k versus ~126k respectively). This improvement in performance proves that the complete reliance on ImageNet for pretraining should be revisited and not taken as granted. Pretraining on full ImageNet achieved higher performance than COCO which was expected due to the large difference in size between the two datasets (ImageNet-1k consists of ~1.2 million images while COCO consists of ~123k images). These experiments point towards the effectiveness of substituting the ImageNet as a pretraining dataset in case of a multi-object downstream task, however, the scale of the pretraining dataset needs to be considerably large for a significant improvement.

In our next downstream evaluation, we examined the effects of pretraining on COCO and ImageNet100 for a multi-object dataset in a different domain such as the autonomous driving domain. Table 4 shows that SimSiam pretraining on COCO had higher transferability than pretraining on ImageNet100. This confirms the pattern established in the previous experiment on PascalVOC and further points toward the inaccurate use of ImageNet as a default pretraining dataset in autonomous driving applications.

Table 4. CityScapes SVM Results

Pretraining Dataset	SVM Downstream Accuracy (mAP)
ImageNet100	57.4
COCO	**58.3**

5. Conclusion

Self-supervised learning holds great promises for the future of deep learning in computer vision in general and for autonomous driving applications specifically. In this work, we focused on the effects of using a multi-object centric dataset in self-supervised learning pretraining instead of using ImageNet which is the current prevalent paradigm in self-supervised literature. We pretrained SimSiam using ImageNet100 (single-object centric) and COCO2017 (multi-object centric) using the same settings. We used multi-label classification as our downstream task to evaluate and compare the performances of both pretrained models. We applied linear classification using SVM on the PascalVOC dataset which is a benchmark setting for self-supervised learning multi-label classification. For further analysis, we also

used CityScapes; an autonomous driving dataset to evaluate the two models. We reported the downstream results and found that the COCO pretrained model consistently outperformed ImageNet100 pretrained models on both downstream datasets. The increase was approximately +1 percent in both cases (56.2 versus 57.6 mAP in the case of PascalVOC and 57.4 versus 58.3 in the case of CityScapes). This is significant because COCO is smaller in size than ImageNet100, nevertheless, it achieved higher results. We conclude that using multi-object centric datasets for self-supervised learning pretraining is essential when applied to a complex downstream dataset. These results also support the need for more research which focuses on the design of self-supervised algorithms that work best with multi-object centric datasets instead of the default focus on ImageNet.

References

1. J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A large-scale hierarchical image database,” 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255, 2009.
2. M. Soudy, Y. Afify, and N. Badr, “RepConv: A novel architecture for image scene classification on Intel scenes dataset,” International Journal of Intelligent Computing and Information Sciences, vol. 22, no. 2, pp. 1–11, Apr. 2022, doi: 10.21608/ijicis.2022.118834.1163.
3. M. Bassiouni, I. Hegazy, N. Rizk, E.-S. El-Dahshan, and A. Salem, “DEEP LEARNING APPROACH BASED ON TRANSFER LEARNING WITH DIFFERENT CLASSIFIERS FOR ECG DIAGNOSIS,” International Journal of Intelligent Computing and Information Sciences, vol. 22, no. 2, pp. 1–19, Apr. 2022, doi: 10.21608/ijicis.2022.105574.1137.
4. M. Alhumayani, M. Monir, and rasha ismail, “machine and deep learning approaches for human activity recognition,” International Journal of Intelligent Computing and Information Sciences, vol. 21, no. 3, pp. 1–9, Sep. 2021, doi: 10.21608/ijicis.2021.82008.1106.
5. E. Sadek, N. AbdElSabour Seada, and S. Ghoniemy, “Computer Vision Techniques for Autism Symptoms Detection and Recognition: A Survey.,” International Journal of Intelligent Computing and Information Sciences, vol. 20, no. 2, pp. 89–111, Dec. 2020, doi: 10.21608/ijicis.2020.46360.1034.
6. L. Jing and Y. Tian, “Self-Supervised Visual Feature Learning With Deep Neural Networks: A Survey,” IEEE Trans Pattern Anal Mach Intell, vol. 43, pp. 4037–4058, 2019.
7. J.-B. Grill et al., “Bootstrap Your Own Latent: A New Approach to Self-Supervised Learning,” ArXiv, vol. abs/2006.07733, 2020.
8. X. Chen and K. He, “Exploring Simple Siamese Representation Learning,” 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 15745–15753, 2020.
9. K. Chen, L. Hong, H. Xu, Z. Li, and D.-Y. Yeung, “MultiSiam: Self-supervised Multi-instance Siamese Representation Learning for Autonomous Driving,” 2021 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 7526–7534, 2021.
10. F. Chiaroni, M.-C. Rahal, N. Hueber, and F. Dufaux, “Self-Supervised Learning for Autonomous Vehicles Perception: A Conciliation Between Analytical and Learning Methods,” IEEE Signal Process Mag, vol. 38, pp. 31–41, 2019.
11. T. Chen, S. Kornblith, M. Norouzi, and G. E. Hinton, “A Simple Framework for Contrastive Learning of Visual Representations,” ArXiv, vol. abs/2002.05709, 2020.
12. K. He, H. Fan, Y. Wu, S. Xie, and R. B. Girshick, “Momentum Contrast for Unsupervised Visual Representation Learning,” 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 9726–9735, 2019.

13. A. El-Nouby, G. Izacard, H. Touvron, I. Laptev, H. Jégou, and E. Grave, “Are Large-scale Datasets Necessary for Self-Supervised Pre-training?” ArXiv, vol. abs/2112.10740, 2021.
14. T.-Y. Lin et al., “Microsoft COCO: Common Objects in Context,” in European Conference on Computer Vision, 2014.
15. M. Cordts et al., “The Cityscapes Dataset for Semantic Urban Scene Understanding,” 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3213–3223, 2016.
16. M. Sadik, S. Moussa, A. El-Sayed, and Z. Fayed, “Vehicles Detection and Tracking in Advanced & Automated Driving Systems: Limitations and Challenges,” International Journal of Intelligent Computing and Information Sciences, vol. 22, no. 3, pp. 1–16, Jul. 2022, doi: 10.21608/ijicis.2022.117646.1158.
17. P. Goyal, D. K. Mahajan, A. K. Gupta, and I. Misra, “Scaling and Benchmarking Self-Supervised Visual Representation Learning,” 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 6390–6399, 2019.
18. M. Noroozi and P. Favaro, “Unsupervised Learning of Visual Representations by Solving Jigsaw Puzzles,” in European Conference on Computer Vision, 2016.
19. S. Gidaris, P. Singh, and N. Komodakis, “Unsupervised Representation Learning by Predicting Image Rotations,” ArXiv, vol. abs/1803.07728, 2018.
20. C. Doersch, A. K. Gupta, and A. A. Efros, “Unsupervised Visual Representation Learning by Context Prediction,” 2015 IEEE International Conference on Computer Vision (ICCV), pp. 1422–1430, 2015.
21. I. Misra and L. van der Maaten, “Self-Supervised Learning of Pretext-Invariant Representations,” 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6706–6716, 2019.
22. G. Li, Z. Ji, Y. Chang, S. Li, X. Qu, and D. Cao, “ML-ANet: A Transfer Learning Approach Using Adaptation Network for Multi-label Image Classification in Autonomous Driving,” Chinese Journal of Mechanical Engineering, vol. 34, pp. 1–11, 2021.
23. E. Xie et al., “DetCo: Unsupervised Contrastive Learning for Object Detection,” 2021 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 8372–8381, 2021.
24. P. Hua, “SimSiam”, https://github.com/PatrickHua/SimSiam (Accessed: 18 Oct. 2022).
25. Y. Tian, D. Krishnan, and P. Isola, “Contrastive Multiview Coding,” in European Conference on Computer Vision, 2019.
26. M. Everingham, L. van Gool, C. K. I. Williams, J. M. Winn, and A. Zisserman, “The Pascal Visual Object Classes (VOC) Challenge,” Int J Comput Vis, vol. 88, pp. 303–338, 2010.
27. Mms. Contributors, “MMSelfSup: OpenMMLab Self-Supervised Learning Toolbox and Benchmark.” 2021.