IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 25, Issue 2, Ser. II (Mar. – Apr. 2023), PP 32-38
www.iosrjournals.org

CLCGAN: Contrastive Learning and CGAN for Data Regeneration-based Continual Learning

Nassim Ammour
Department of Computer Engineering, College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia

Abstract
In the realm of scene classification, it is clear that deep learning models excel when a large amount of labeled data is available. However, continual learningsuffers from a lack of old tasks' data. Continual learning (CL) is a needed aspect of artificial intelligence (AI). Inspired from the human ongoing learning capacity, endowing a deep learning model with the ability to preserve previous knowledge is legitimate. Training a deep learning model for sequential learning of tasks leads to a continual decline in the performance for previous tasks due to the non-availability of their training data. This phenomenon is known, in the literature, as catastrophic forgetting. We proposed a two deep blocks model. A feature extraction module composed of an EfficientNet_B5 followed by a contrastive learning model to boost the reparability in the feature space. A conditional generative adversarial network (CGAN) to capture the latent structure of the previous tasks' data. Experiments are conducted on two scene datasets (Merced and Optimal31). The experimental results assert the outperformance and robustness of the proposed model.

Keywords: *Remote sensing, scene classification, data regeneration, contrastive learning, continual Learning.*

Date of Submission: 24-03-2023 Date of Acceptance: 06-04-2023

I. Introduction

Remote sensing images, collected from imagery sensors on satellites and airplanes, serve to detect and monitor the physical characteristics of an area, region, and Earth. Plentiful land-cover datasets are collected, and accurate classification models for remote sensing applications are developed thanks to the progress in remote sensing technology.

In the past, methods using handcrafted features followed by a classifier have been developed for scene classification, including bag-of-visual-words [1], sparse coding [2], midlevel visual elements-oriented features [3], a multi-scale local binary pattern. From multilayer perceptron (MLP) [4], and inspired by biological processes, convolutional neural networks (CNN) are introduced in 1980s and have proven as one of the excellent algorithms to extract visual information very efficiently [5], [6]. In fact, the big success of CNNs is due to their ability to learn the hierarchical features at intermediate layers automatically from the data. Several CNN based deep learning models such as Inception and ResNet are trained to extract good discriminative features from remote sensing images.

Inspired by the aptitude of humans to continually acquire knowledge and skills, a lot of efforts have been consecrated by researchers to overcome the catastrophic forgetting phenomena and develop continual deep learning models [7], [8]. Authors in [9] used a technique to penalize significant changes to the task-sensitive parameters when learning a new task. In [10], the authors expanded a trained model with additional layers associated with the new task. The authors in [11], [12] proposed a continual learning technique using a dynamically expanding model. Cumulative learning is realized in [13] by inserting new nodes to each layer in the model determined by a dedicated controller, freezing the previously learned parameters, and retraining the model on the new task. Authors in [14] used memory replay techniques to prevent losing old knowledge. To overcome catastrophic forgetting, authors in [15] used the average of the previously seen classes to generate a class exemplar and used them in training with the new task. In the remote sensing field, limited works were proposed. For instance, a continual learning technique for land-cover imagers classification is proposed in [16]. In this paper, a model composed of two trainable deep learning networks is proposed, the first module ensure the feature extraction and classification tasks, the second module learns maximize the separation between the tasks in order to identify each task.

Recently, a re-emergence of research in contrastive learning provided major advances in self-supervised representation learning [17], [18]. The key idea of contrastive learning is to pull together an anchor and a positive sample in embedding space, and push apart the anchor from many negative samples.

In this paper, we aim to endow the deep-learning architecture with the ability to learn sequentially to maintain its performance on the previous tasks. Each classification task contains a group of land-cover classes.

We propose a new deep learning model based on two trainable modules as illustrated in Figure 1. First, a feature extraction module extracts discriminative features from the remote sensing scene images. Then, and in order to boost the classifier performance, a contrastive learning module is added to increase the distinguishability between the classes in the embedded space. The weights of the first module are adjusted by discriminating between the land-cover classes within the new task using a contrastive loss. Normalized representations from the same class are pulled closer together, and representations from different classes are pushed away from each other. To preserve the old tasks' knowledge and avoid catastrophic forgetting, a second-deep learning architecture tries to discover the latent structure of previously seen tasks' data and to generate old tasks' data. Experimental results on two-scene datasets (Merced and Optimal31) demonstrate the advantage of the proposed contrastive learning and VAE based hybrid network in remote sensing images classification.

II. Materials and Methods

An incremental multi-class classification problem consists of an long series of tasks $T_l = \left\{X_i^{(l)}, y_i^{(l)}\right\}_{i=1}^{n_l}, l = 1, \dots k, \dots, K$, where each task T_l includes a subset c_l of classes with n_l images $X^{(l)}$ and their corresponding categorical class labels $y^{(l)}$. We seek to train a unified classification deep model on a new task and make it able to maintain its performance on both the previously seen tasks and the new task. In the following, we illustrate the main steps for training this deep networks architecture for a continual learning process.

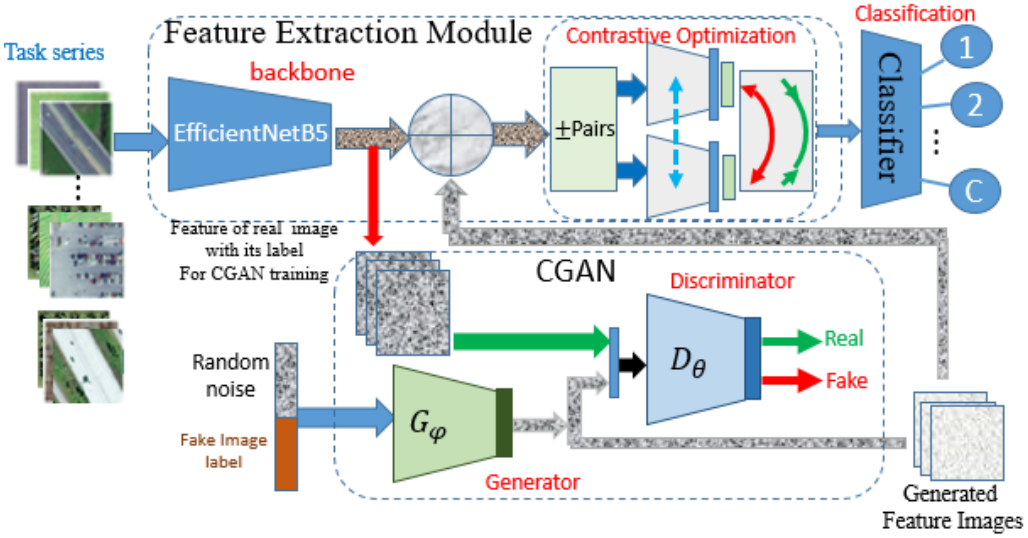

Figure 1. Contrastive learning and CGAN based hybrid model.

2.1 Learning the k^{th} task T_k

After $k-1$ tasks, we want to train the pre-trained model on a new task $T_k = \left\{X^{(k)}, y^{(k)}\right\}$. The model uses EfficientNet Network (EfficientNet B5) as a feature extractor backbone; however other well-known CNN models could be used as well. The feature extractor module is trained only for the first task and then frozen for the rest of the tasks. In [19], the authors observed that carefully balancing the depth, width, and resolution of a deep model can produce better accuracy and efficiency. Based on this observation, they proposed a new family of CNN architectures, called EfficientNet, that uniformly scales all dimensions of depth, width, and resolution of the network. We prune this backbone network by removing the softmax classification layer and add a 40 by 40 reshaping layer. Then, we placed a contrastive learning sub-module at the top of the architecture. To train the deep model on the task $T_{\mathrm{k}} = \left\{X^{(\mathrm{k})}, y^{(\mathrm{k})}\right\}$, we first generated the old tasks' data using a trained VAE generator $\left\{X^{(1:\mathrm{k}-1)}, y^{(1:\mathrm{k}-1)}\right\}$and we add them to the new task's dataset. Then, we add a softmax layer as output layer with the total number of output classes C, and we use the $c_{1:\mathrm{k}}$ outputs related to the tasks $T_{1:k}$ number of classes and put zeros at the $c_{k+1:K}$ other outputs.

2.2 Contrastive feature optimization

The contrastive learning uses losses based on metric distance learning between similar samples and different ones as shown in Figure 2. These losses are used to learn powerful discriminative representations.This sub-module encapsulates a data-augmentation module $Aug(\cdot)$, an encoder $Enc(\cdot)$ to obtain a 2048-dimensional

vector $r = Enc(x)$ normalized embedding from each input sample$x$, and a projection network $Proj(\cdot)$ which maps the vector r to a 128-dimensional normalized vector $z = Proj(r)$. The projection network is discarded at inference time. To apply contrastive learning, we first create two copies from one batch of data using data augmentation twice and pass them through the encoder and the projection network. The supervised contrastive loss is calculated on the outputs of the projection network. A classifier on top of the frozen representations is trained using a cross-entropy loss for a classification purpose.

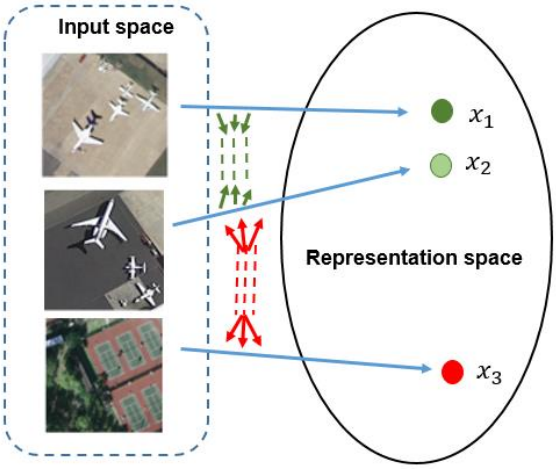

Figure 2. Contrastive learning representation.

From a batch of a set of N randomly chosen sample/label pairs $\{x_n, y_n\}_{n=1}^{N}$, a $2N$ multi-viewed batch $\{\tilde{x}_l, \tilde{y}_l\}_{l=1}^{2N}$ containing $\tilde{x}_{2k}$ and $\tilde{x}_{2k-1}$augmentations is built using data augmentation process (positives: sampled from the same class, and negatives: sampled from different classes). The contrastive learning sub-module is trained using supervised contrastive losses:

$$\mathcal{L}_{cont} = \sum_{i \in I} \frac{-1}{|P(i)|} \sum_{p \in P(i)} log \frac{\exp(z_i \cdot z_p/\tau)}{\sum_{a \in A(i)} \exp(z_i \cdot z_a/\tau)} \tag{1}$$

Here, $z_i = Proj\big(Enc(\tilde{x}_l)\big)$, the symbol $\cdot$ denotes the inner product, τ is the temperature parameter, $i \in I \equiv \{1 \ldots 2N\}$ is the index of an arbitrary augmented sample in the multi-viewed batch. $P(i) \equiv \{p \in A(i): \tilde{y}_p = \tilde{y}_i\}$ the set of indices of all positives distinct from i, $A(i) \equiv I \backslash \{i\}$, and $|\cdot|$ denotes the cardinality.

2.3 The Generative network

To control the forgetting phenomenon, we use a deep learning-based conditional generative network to memorize the latent structure of the previously seen data. Thus, we can generate samples of the old tasks needed to train the classification model on the new classification task without forgetting the previous tasks. The discriminator $D_\theta(\cdot)$ of a the CGAN estimates the probability that an input came from the true data rather than the generated data. The loss function of the discriminator gathers the error of predicting true sample coming from the dataset and fake sample coming from the generator given their labels

$$\mathcal{L}_{\theta,\phi}^{(D)} = -\mathbb{E}_{x \sim p_\phi} log D_\theta(x|y) - \mathbb{E}_{z \sim q_\theta} \log\Big(1 - D_\theta\left(G_\phi(z|y')\right)\Big) \tag{2}$$

The generator $G_\phi(\cdot)$ learns to map a noise input to the true dataset images space by minimizing its loss function, which is built using gathered prediction of the discriminator on generated samples conditioned on the specified labels.

$$\mathcal{L}_{\theta,\phi}^{(G)} = -\mathbb{E}_{z \sim q_\theta} \log\Big(D_\theta\left(G_\phi(z|y')\right)\Big) \tag{3}$$

The CGAN is trained by minimizing the alternatively the discriminator loss $\mathcal{L}_{\theta,\phi}^{(D)}$and the generator loss $\mathcal{L}_{\theta,\phi}^{(G)}$. The total loss function is:

$$\mathcal{L}_{\theta,\phi}^{(CGAN)} = \mathcal{L}_{\theta,\phi}^{(D)} + \mathcal{L}_{\theta,\phi}^{(G)} \tag{4}$$

III. Experimental results and discussion

3.1 Dataset Description

Two datasets of remote sensing images are used to evaluate the performance of the proposed continual learning architecture, Merced and Optimal-31 scene datasets. The Merced dataset was built from the United States Geographical Survey (USGS) National Map[20].Figure 3 (a)illustrates samples images from Merced dataset. It contains 21 classes of remote sensing images with 100 RGB images of dimension 256×256 pixels in each one. Besides, the Optimal31 dataset was collected from Google Earth[21]. Optimal31 dataset is built using 1860 RGB images separated into 31 categories of 60 images per class. This dataset is more challenging than Merced dataset and contains ten more classes. Figure 3 (b) shows image samples from the Optimal31 dataset.

(a) *(b)*

Figure 3. Sample images from (a) Merced and (b) optimal31datasets.

3.2Experiment Setup

To perform a continuous learning process, we split the dataset into a bench of subsets, where each subset is consecrated to one task during the sequential learning process. In the first experiment, seven tasks with three classes per task are performed using the Merced dataset. For the Optimal31 dataset, ten tasks with three classes per task are used (the last task for Optinal31 contains 4 classes). In the second experiment, we inspected the robustness of the proposed classification model to the number of tasks and the number of classes by task. As stated in the methodological section, the parameters of the proposed model are learned in a contrastive manner for each new task. The generator sub-module auto-generates the feature images of previous classification tasks and augments the training dataset by automatically adding the reconstructed feature images to the new task's feature images. The dataset of each task is divided into 80% for the training dataset and 20% for the testing dataset. The experiments were conducted on the Google Colaboratory cloud service using the available GPU to accelerate the deep learning process.

3.3 Results

Joint learning

In the first experiment, and to have an indication about the overall accuracy when we train the model jointly on all the dataset, an OA of 97% for Merced dataset (21 classes) and an OA 94 % for Optimal31 dataset (31 classes) are obtained when we trained the model on all the datasets in one task. These accuracies are used to be as baseline for evaluating the performances of the continual learning strategy.

Continual learning process

After the joint learning, and as a second experiment, we implemented the proposed continual learning model and reported the model performance results in Table1. For Merced dataset, the proposed model achieved an OA of 100% in the first task and then decayed slowly and reached the OA of 92% at the end of the last classification task, after executing sequentially seven consecutive classification tasks. The model performed with a biases of +3% with the first classification task and -5%with the last task compared to the joint training.

Similarly, for the Optimal31 dataset, the proposed architecture performs the first task with an OA of 99% and reaches an OA of 91% for the last classification task at the end of the continual learning process. We can notice from Table1, and compared to the joint learning, that the biases are +5% with the first task and -3%

with the last classification task. We can remark, also that the accuracy of the model decays slowly, with a rate of change of approximately 1% for Merced dataset, indicating that the model performs with a low forgetting effect during the continual learning process. For the Optimal31 dataset, we can remark that the forgetting effect is not important. The accuracy decreases slowly from 99% at the first classification task and reaches a classification accuracy of 91% at the end of the continual learning process after executing ten classification tasks. The proposed continual learning strategy combats this challenging forgetting effect and maintains high performance.In the third experiment, we investigated the sensitivity of the proposed architecture to the number of classes (the size of the data) during each classification task in the sequential process. The structure of the new classification task's dataset and the generated data for all the previous classification tasks are illustrated in Figure 4 after a dimensionality reduction.

TABLE 1. OVERALL ACCURACY (OA) IN [%] OBTAINED FOR MERCED AND OPTIMAL31 DATASETS.

Task	Accuracy (%)	
	Merced	**Optimal31**
Joint	97	94.22
1	100±0.00	99.07±0.03
2	98.75±0.03	98.38±0.02
3	97.33±0.02	96.05±0.03
4	95.90±0.03	95.13±0.02
5	94.53±0.03	93.31±0.02
6	92.90±0.01	93.21±0.03
7	92.00±0.02	93.19±0.01
8	-	92.47±0.02
9	-	92.61±0.13
10	-	91.13±0.03

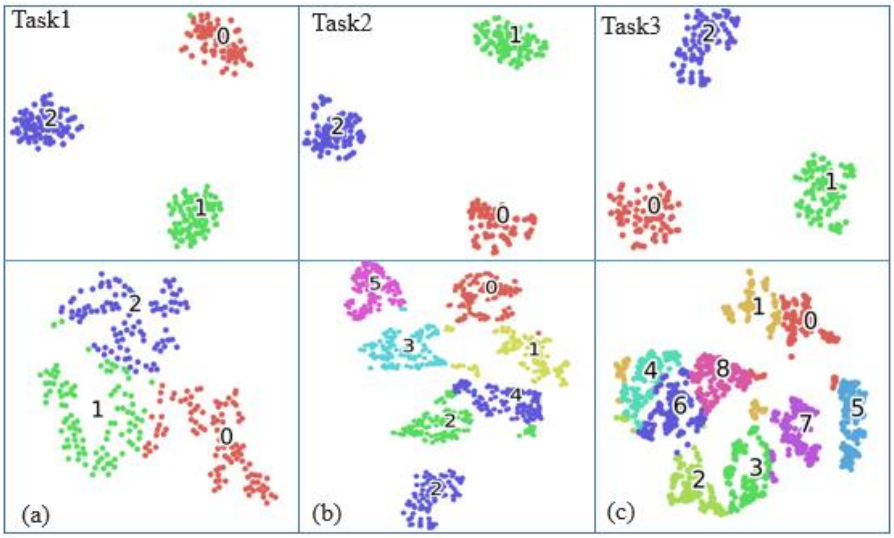

Figure 4. Original data on top and generated data on the bottom:(a) dataset for task 1, (b) Dataset for task 2, (c) Dataset for task 3

Sensitivity analysis regarding the task data size

In Figure 5, the sensitivity of the performance of the continual learning process to the number of classes per classification task (size of the data) is illustrated. To implement the sensitivity experiments, we increased the size of the data by augmenting the number of classes during each classification task. As shown in Figure 5, we can notice that the model gives better classification accuracies with a smaller set of classes by classification task during the continual learning process. We can interpret this attitude by the fact that data with a highernumber of classes has a bigger size, with a complex hidden structure, and thereby, is not easy to generate.

Figure 5. Sensitivity to the number of classes by task

Comparison to other methods

We compared our method with three recent methods proposed in the literature to show the eminence of the proposed model. The first method is developed in [22]; in this paper, the authors reduced the forgetting effect by extending the output of the model architecture for each new classification task and fine-tuning the parameters of shared layers. The second method isproposed in [16], like the method proposed in [22], the authors used shared layers and an extension of the model for the new classification task, and they improved the model performance by adding a deep learning selector to connect the input to the task-specific output layer. The third method is developed in [23] ; in this paper, the authors selectively reduce the impact of learning on the meaningful weights for the old classification tasks. The comparison experiments are conducted on Merced dataset and Optimal31 dataset, and the results are reported in Table II and III, respectively. The results reveal the promising effects of the proposed continual learning architecture.

TABLE II. COMAPRASION TO OTHER METHODS MERCED DATASET.

Task	J. Kirkpatrick*e t al.* [18]	Li. *et al.* [14]	N. Ammour et al. [21]	Proposed
1	97	99	99	**100**
2	95	77	99	**99**
3	91	49	98	**97**
4	86	29	95	**96**
5	76	25	92	**95**
6	68	18	93	**93**
7	58	17	89	**92**

TABLE III. COMAPRASION TO OTHER METHODS OPTIMAL31 DATASET

Task	J. Kirkpatrick*et al.*[23]	Li. *et al.* [22]	N. Ammour et al. [16]	Proposed
1	98	98	98	**99**
2	85	82	97	**98**
3	83	45	90	**96**
4	78	29	87	**95**
5	72	21	86	**93**
6	68	21	80	**93**
7	57	15	80	**93**
8	48	13	77	**92**
9	38	12	68	**93**
10	29	12	71	**91**

IV. Conclusions

In this work, we have developed a new continual-learning strategy for scene classification in remote sensing imagery. In this paper, we propose a continual learning technique using a contrastive learning process, used to boost the dissimilarity between the classes in a new embedding space. Furthermore, we train a deep learning generator model to learn the latent structure of old classification tasks' data. The generator is employed

to preserve acquired knowledge from old classification tasks and to fight the forgetting phenomena. The results of experiment conducted on Merced and Optimal31 datasets demonstrated the efficiency of the proposed continual learning method.

Conflict of interest

The author declare that they are no conflicts of interest.

References

[1] L. Zhao, P. Tang, and L. Huo, "Feature significance-based multibag-of-visual-words model for remote sensing image scene classification," *Journal of Applied Remote Sensing*, vol. 10, no. 3, p. 035004, 2016.

[2] J. Shi, Z. Jiang, H. Feng, and Y. Ma, "Sparse coding-based topic model for remote sensing image segmentation," presented at the 2013 IEEE International Geoscience and Remote Sensing Symposium-IGARSS, IEEE, 2013, pp. 4122–4125.

[3] G. Cheng, J. Han, L. Guo, Z. Liu, S. Bu, and J. Ren, "Effective and efficient midlevel visual elements-oriented land-use classification using VHR remote sensing images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 8, pp. 4238–4249, 2015.

[4] J. Määttä, A. Hadid, and M. Pietikäinen, "Face spoofing detection from single images using micro-texture analysis," presented at the 2011 international joint conference on Biometrics (IJCB), IEEE, 2011, pp. 1–7.

[5] Q. Wang, W. Huang, Z. Xiong, and X. Li, "Looking Closer at the Scene: Multiscale Representation Learning for Remote Sensing Image Scene Classification," *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[6] Y. Bazi, M. M. Al Rahhal, H. Alhichri, and N. Alajlan, "Simple Yet Effective Fine-Tuning of Deep CNNs Using an Auxiliary Classification Loss for Remote Sensing Scene Classification," *Remote Sensing*, vol. 11, no. 24, p. 2908, 2019.

[7] I. J. Goodfellow, M. Mirza, D. Xiao, A. Courville, and Y. Bengio, "An empirical investigation of catastrophic forgetting in gradient-based neural networks," *arXiv preprint arXiv:1312.6211*, 2013.

[8] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[9] L. Butyrev, G. Kontes, C. Löffler, and C. Mutschler, "Overcoming Catastrophic Forgetting via Hessian-free Curvature Estimates," 2019.

[10] S. S. Sarwar, A. Ankit, and K. Roy, "Incremental Learning in Deep Convolutional Neural Networks Using Partial Network Sharing," *IEEE Access*, vol. 8, pp. 4615–4628, 2020, doi: 10.1109/ACCESS.2019.2963056.

[11] S. Sadhu and H. Hermansky, "Continual Learning in Automatic Speech Recognition.," presented at the INTERSPEECH, 2020, pp. 1246–1250.

[12] J. Gideon, S. Khorram, Z. Aldeneh, D. Dimitriadis, and E. M. Provost, "Progressive neural networks for transfer learning in emotion recognition," *arXiv preprint arXiv:1706.03256*, 2017.

[13] J. Xu and Z. Zhu, "Reinforced continual learning," in *Advances in Neural Information Processing Systems*, 2018, pp. 899–908.

[14] D. Lopez-Paz and M. Ranzato, "Gradient episodic memory for continual learning," *Advances in neural information processing systems*, vol. 30, pp. 6467–6476, 2017.

[15] L. Guo, G. Xie, X. Xu, and J. Ren, "Exemplar-supported representation for effective class-incremental learning," *IEEE Access*, vol. 8, pp. 51276–51284, 2020.

[16] N. Ammour, Y. Bazi, H. Alhichri, and N. Alajlan, "Continual Learning Approach for Remote Sensing Scene Classification," *IEEE Geoscience and Remote Sensing Letters*, 2020.

[17] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin, "Unsupervised feature learning via non-parametric instance discrimination," presented at the Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 3733–3742.

[18] O. Henaff, "Data-efficient image recognition with contrastive predictive coding," presented at the International Conference on Machine Learning, PMLR, 2020, pp. 4182–4192.

[19] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," presented at the International conference on machine learning, PMLR, 2019, pp. 6105–6114.

[20] G.-S. Xia *et al.*, "AID: A Benchmark Data Set for Performance Evaluation of Aerial Scene Classification," *IEEE Trans. Geosci. Remote Sensing*, vol. 55, no. 7, pp. 3965–3981, Jul. 2017, doi: 10.1109/TGRS.2017.2685945.

[21] Q. Wang, S. Liu, J. Chanussot, and X. Li, "Scene Classification With Recurrent Attention of VHR Remote Sensing Images," *IEEE Trans. Geosci. Remote Sensing*, vol. 57, no. 2, pp. 1155–1167, Feb. 2019, doi: 10.1109/TGRS.2018.2864987.

[22] Z. Li and D. Hoiem, "Learning without forgetting," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 12, pp. 2935–2947, 2017.

[23] J. Kirkpatrick *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proceedings of the national academy of sciences*, vol. 114, no. 13, pp. 3521–3526, 2017.

Nassim Ammour. "CLCGAN: Contrastive Learning and CGAN for Data Regeneration-based Continual Learning." *IOSR Journal of Computer Engineering (IOSR-JCE)*, 25(2), 2023, pp. 32-38.