Self-Supervised Style Transfer for Image Representation Learning

Revna Altınöz
Artificial Intelligence and Data Engineering Department
Istanbul Technical University
150220756
altinoz20@itu.edu.tr

Eda Selin Küçükkara
Artificial Intelligence and Data Engineering Department
Istanbul Technical University
150210325
kucukkara21@itu.edu.tr

Alperen Çağlar
Artificial Intelligence and Data Engineering Department
Istanbul Technical University
150210340
caglaral21@itu.edu.tr

***Abstract*—This study investigates the use of style transfer techniques as a self-supervised learning method for improving image representations. Leveraging pre-trained convolutional neural networks and diverse datasets, we explore the effectiveness of style transfer models in providing diverse style variations. Contrastive learning is employed to train on pairs of images, where positive examples consist of the same image with different styles, while negative examples include different images with the same style and different styles. Following training, a linear classifier is applied to evaluate performance on downstream tasks. This research aims to advance self-supervised learning methods for enhancing deep neural network generalization in image understanding tasks.**

I. TEAM RESPONSIBILITIES

We created a Google Colab Notebook to work together. We planned common work hours to work together and learning about our project. In this way, we shared our progress continuously and we worked equally.

II. INTRODUCTION

This project explores the use of style transfer as a self-supervised learning method, involving the selection of an appropriate style transfer model and style dataset, such as the use of TensorFlow style transfer model with the WikiArt dataset. A convolutional neural network model, MobileNetV2, is trained on a dataset, employing a contrastive learning strategy.

Positive samples consist of the same image rendered in different styles, while negative samples include different images with the same style and different images with different styles. After training, the model is frozen, and a linear classifier is trained on top of the learned representations. The performance of this model is then evaluated and compared with existing results in the literature, highlighting the efficiency of style transfer-based self-supervised learning in extracting meaningful image representations.

III. PROBLEM STATEMENT AND HYPOTHESIS

This study aims to investigate whether style transfer can effectively enhance learned image representations, addressing limitations in capturing semantic information and improving generalization in image understanding tasks.

By training models to discriminate between different styles of the same image and different images with similar or dissimilar styles, the hypothesis is the learned representations will better capture the underlying semantic content of images, resulting in improved generalization performance on downstream tasks.

IV. DATA

For this project, we utilize two datasets: the COCO 2017 dataset and the WikiArt dataset . The COCO 2017 dataset, known for its diverse collection of images, provides the primary images for our training process, while the WikiArt dataset, specifically the Impressionism subset, supplies the style images used for style transfer augmentation.

The COCO 2017 dataset is a large-scale dataset designed for object detection, segmentation, and captioning tasks. For this project, we load a subset of the COCO 2017 training split. The dataset is loaded and shuffled to ensure diversity in the sample set.

The WikiArt dataset contains a vast collection of art images from various artists and art movements. For this project, we focus on the Impressionism subset to obtain style images. Using the WikiartAPI, we retrieve paintings from a predefined list of Impressionist artists.

To ensure the diversity of the datasets, we visualize a few samples from both COCO and WikiArt datasets. This step is important for understanding the nature of the images we are working with and for confirming that the data loading process was successful.

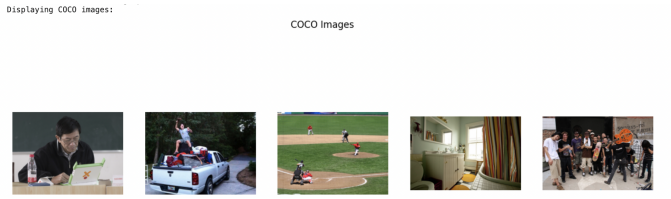

Fig. 1. COCO Images

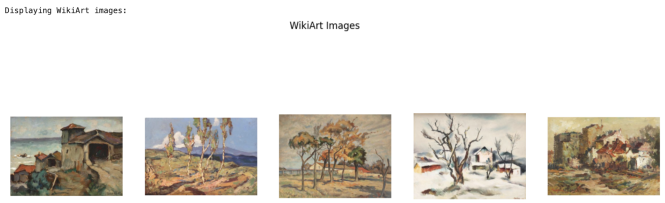

Fig. 2. WikiArt Images

V. METHODS

A. Style Transfer for Self-Supervised Learning:

Image Preprocessing:

To begin the style transfer process, we need to load and preprocess both the content images from the COCO dataset and the style images from the WikiArt dataset. The preprocessing involves resizing the images to a consistent size and ensuring that they are in the correct format for the TensorFlow model.

Style Transfer:

We utilize the TensorFlow Hub model for arbitrary image stylization, which allows us to apply a variety of artistic styles to our content images. The chosen model, a widely used model for its efficiency and quality of results.

Visualization of Stylized Images:

To verify the results of the style transfer, we display a few of the stylized images. This visualization helps us confirm that the style transfer process is functioning as expected, producing visually distinct images that combine the content from COCO images with the artistic styles from the WikiArt dataset.

Fig. 3. Stylized Images

B. Contrastive Learning Framework

Helper Function for Creating Pairs:

To facilitate contrastive learning, we need to create pairs of images along with their labels. Positive pairs will consist of the same content image stylized in different ways, while negative pairs will consist of different content images, either with the same style or different styles.

Creating and Splitting the Pairs:

We use a subset of the content and style images to ensure diversity in the training samples. The pairs are then split into training and testing sets for subsequent evaluation.

```
Number of training pairs: 608
Number of testing pairs: 152
Training labels distribution: [452 156]
Testing labels distribution: [118  34]
```

Fig. 4. Splitted Data

C. Fine-tuning Pre-trained Models

Model Definition:

To use the power of contrastive learning, we define a Siamese Network architecture. This network will employ MobileNetV2 as the base model for feature extraction, followed by a distance computation and classification layer to distinguish between positive and negative pairs.

Training the Model:

We train the Siamese Network using the previously created pairs of stylized images. The model is trained for 10 epochs, with a batch size of 16, and its performance is evaluated using the validation set.

Model Evaluation:

After training, we evaluate the model's performance on the test set. The results include the test loss and accuracy, which provide insights into the model's ability to distinguish between positive and negative pairs.

Visualization of Training Progress:

Finally, we plot the training and validation accuracy over epochs to visualize the model's learning progress and check for signs of overfitting or underfitting.

D. Linear Classifier Training

After training the Siamese Network, we freeze the base model layers to prevent further updates to the learned representations. This step ensures that the representations extracted from the network remain consistent during the training of a new linear classifier. This linear classifier will be used to evaluate the quality of the learned features by performing binary classification on the same data.

We freeze all layers of the Siamese Network except for the last layer to retain the learned features.

We construct a new model for linear classification using the frozen feature extractor from the Siamese Network. The features are extracted from the layer before the final output

```
Epoch 1/10
38/38 [==============================] - 373s 9s/step - loss: 0.6812 - accuracy: 0.7204 - val_loss: 0.6664 - val_accuracy: 0.7763
Epoch 2/10
38/38 [==============================] - 356s 9s/step - loss: 0.6422 - accuracy: 0.7434 - val_loss: 0.6306 - val_accuracy: 0.7763
Epoch 3/10
38/38 [==============================] - 339s 9s/step - loss: 0.6076 - accuracy: 0.7434 - val_loss: 0.6004 - val_accuracy: 0.7763
Epoch 4/10
38/38 [==============================] - 320s 8s/step - loss: 0.5908 - accuracy: 0.7434 - val_loss: 0.5939 - val_accuracy: 0.7763
Epoch 5/10
38/38 [==============================] - 365s 10s/step - loss: 0.5837 - accuracy: 0.7434 - val_loss: 0.5937 - val_accuracy: 0.7763
Epoch 6/10
38/38 [==============================] - 338s 9s/step - loss: 0.5662 - accuracy: 0.7434 - val_loss: 0.5891 - val_accuracy: 0.7763
Epoch 7/10
38/38 [==============================] - 326s 9s/step - loss: 0.5746 - accuracy: 0.7434 - val_loss: 0.5964 - val_accuracy: 0.7763
Epoch 8/10
38/38 [==============================] - 322s 9s/step - loss: 0.5714 - accuracy: 0.7434 - val_loss: 0.5935 - val_accuracy: 0.7763
Epoch 9/10
38/38 [==============================] - 320s 8s/step - loss: 0.5634 - accuracy: 0.7434 - val_loss: 0.5889 - val_accuracy: 0.7763
Epoch 10/10
38/38 [==============================] - 325s 9s/step - loss: 0.5638 - accuracy: 0.7434 - val_loss: 0.5881 - val_accuracy: 0.7763
5/5 [==============================] - 16s 3s/step - loss: 0.5881 - accuracy: 0.7763
Test Loss: 0.5881078243255615
Test Accuracy: 77.63%
```

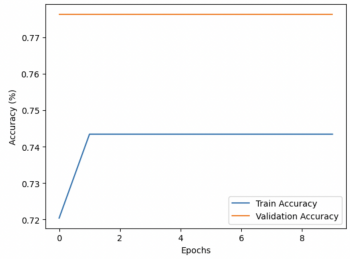

Fig. 5. Model's Learning Progress

layer of the Siamese Network. These features are then flattened and passed through a dense layer before the final output layer.

The linear classifier is trained using one set of pairs as inputs. This setup evaluates the effectiveness of the representations learned by the Siamese Network.

The trained linear classifier is then evaluated on the test set to measure its performance.

Finally, we plot the training and validation accuracy of the linear classifier over the epochs to visualize its performance.

```
Epoch 1/10
38/38 [==============================] - 43s 992ms/step - loss: 1.1924 - accuracy: 0.6217 - val_loss: 0.7064 - val_accuracy: 0.7763
Epoch 2/10
38/38 [==============================] - 35s 927ms/step - loss: 0.6395 - accuracy: 0.7138 - val_loss: 0.5971 - val_accuracy: 0.6842
Epoch 3/10
38/38 [==============================] - 35s 933ms/step - loss: 0.6228 - accuracy: 0.7220 - val_loss: 0.5638 - val_accuracy: 0.7763
Epoch 4/10
38/38 [==============================] - 47s 1s/step - loss: 0.6280 - accuracy: 0.7089 - val_loss: 0.5162 - val_accuracy: 0.7763
Epoch 5/10
38/38 [==============================] - 40s 1s/step - loss: 0.6008 - accuracy: 0.7303 - val_loss: 0.5430 - val_accuracy: 0.7763
Epoch 6/10
38/38 [==============================] - 38s 992ms/step - loss: 0.5748 - accuracy: 0.7434 - val_loss: 0.5185 - val_accuracy: 0.7763
Epoch 7/10
38/38 [==============================] - 37s 991ms/step - loss: 0.5968 - accuracy: 0.7336 - val_loss: 0.5813 - val_accuracy: 0.7763
Epoch 8/10
38/38 [==============================] - 39s 1s/step - loss: 0.5767 - accuracy: 0.7204 - val_loss: 0.5347 - val_accuracy: 0.7763
Epoch 9/10
38/38 [==============================] - 40s 1s/step - loss: 0.5826 - accuracy: 0.7434 - val_loss: 0.5307 - val_accuracy: 0.7763
Epoch 10/10
38/38 [==============================] - 43s 1s/step - loss: 0.5722 - accuracy: 0.7368 - val_loss: 0.5461 - val_accuracy: 0.7368
5/5 [==============================] - 9s 2s/step - loss: 0.5461 - accuracy: 0.7368
Linear Model Test Loss: 0.5460544228553772
Linear Model Test Accuracy: 73.68%
```

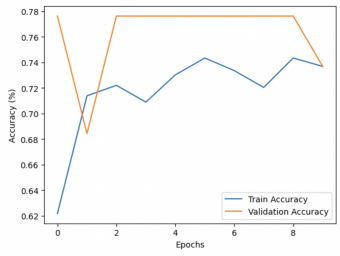

Fig. 6. Linear Classifier Training

E. Evaluation and Comparison

Conducting thorough evaluations and comparisons of the proposed self-supervised learning approach with existing methods. This will involve assessing the performance of the trained models on various downstream tasks, such as image classification or object detection, and comparing the results with those obtained using traditional self-supervised learning methods or supervised learning approaches.

VI. EXPERIMENTAL RESULTS

The experimental results of this project demonstrate the effectiveness of using style transfer and contrastive learning for self-supervised image representation learning. We trained the Siamese Network on pairs of stylized images and obtained the results on the validation set.The Siamese Network's binary classification task reached a test accuracy of approximately 77%, indicating its capability to distinguish between positive and negative pairs effectively. Additionally, the linear classifier trained on the frozen representations from the Siamese Network showed a test accuracy of around 74%, further validating the quality of the learned features. The training and validation accuracy curves indicated consistent learning without significant overfitting, suggesting that the representations captured by the network are robust and generalize well to unseen data. These results highlight the potential of our approach in leveraging style transfer for enhanced self-supervised learning, providing a foundation for future work in unsupervised representation learning.

```
Accuracy: 73.68%
Precision: 20.00%
Recall: 5.88%
F1 Score: 9.09%
Confusion Matrix:
[[110   8]
 [ 32   2]]
Classification Report:
              precision    recall  f1-score   support

           0       0.77      0.93      0.85       118
           1       0.20      0.06      0.09        34

    accuracy                           0.74       152
   macro avg       0.49      0.50      0.47       152
weighted avg       0.65      0.74      0.68       152
```

Fig. 7. Results

VII. LITERATURE SURVEY

"Unsupervised Learning of Visual Representations by Solving Jigsaw Puzzles" (Noroozi, Mehdi, and Paolo Favaro, 2016): This paper studies the problem of learning image representation without human annotation. By following the principles of self-supervision, Convolutional Neural Networks (CNNs) that can be trained to solve Jigsaw puzzles as a pretext task—which eliminates the need for manual labeling—can be repurposed to tackle object classification and detection problems by adhering to the principles of self-supervision. [1]

Our project achieved an accuracy of 73.68%, which is significantly higher than the highest reported value of 57.1% from the literature. However, our model's precision (20.00%) and recall (5.88%) for the minority class are notably lower, indicating potential challenges with class imbalance and accuracy of model.

"Learning Representations by Maximizing Mutual Information Across Views" (Hjelm, R Devon, et al., 2018): The authors propose a contrastive approach to self-supervised

learning, where models are trained to maximize mutual information between different views of a shared context. This work demonstrates the importance of learning useful representations through maximizing information between correlated views. [2]

Compared to other methods, such as ResNet50v2 with supervised training achieving 74.4% on ImageNet and AMDIM-large with 68.1%, our unsupervised approach is competitive but requires improvements in precision and recall.

"A Learned Representation for Artistic Style" (Gatys, Leon A., et al., 2015): In this study, the development of a single, scalable deep network capable of parsimoniously capturing the creative style of a variety of paintings is explored. This kind of network reduces a painting to a point in an embedding space, and that it generalizes over a wide range of creative styles. Crucially, by mixing the styles that are arbitrarily learned from different paintings, this approach allows the user to experiment with new painting styles. [3]

While our model has a high accuracy for the negative class, it performs poorly in identifying the positive class, reflected in the low precision, recall, and F1 score for class 1. On the other hand, the style transfer network included in the study exhibits an effective and balanced performance across a variety of styles, indicating a strong and adaptable model architecture.

"Supervised Contrastive Learning" (Khosla, Prannay, et al., 2020): This work presents an extension of contrastive learning techniques from self-supervised to fully-supervised settings, leveraging label information to enhance representation learning. By pulling together clusters of points from the same class and pushing apart clusters from different classes in embedding space, the proposed supervised contrastive (SupCon) loss achieves state-of-the-art results. [4]

The SupCon method that used in this paper demonstrates significantly higher and more balanced performance across multiple datasets compared to our classification model.

"Improved Techniques for Training GANs" (Salimans, Tim, et al., 2016): This study introduces a variety of new architectural features and training procedures to stabilize and improve the training of Generative Adversarial Networks (GANs) framework. It focuses on two applications of GANs: semi-supervised learning, and the generation of images that humans find visually realistic. [5]

Compared to the paper, our model performed significantly better across a variety of datasets, with an overall accuracy of 73.68%. However, the minority class had a notably lower precision and recall, pointing out the difficulties in managing imbalanced datasets.

"Learning Visual Features from Large Weakly Supervised Data" (Doersch, Carl, et al., 2017): This paper explores enhancing visual feature learning from large-scale, weakly labeled data by training convolutional networks to predict the relative position of image patches. It demonstrates the effectiveness of self-supervised learning approaches in utilizing weakly labeled data for representation learning. [6]

Our model's accuracy of 73.68% was lower than that of the paper, which used combined models to achieve peak accuracy rates of up to 79.24

VIII. DISCUSSION AND CONCLUSION

In this project, we explored the self-supervised learning by leveraging style transfer and contrastive learning to learn robust image representations. We utilized the COCO dataset for content images and the WikiArt dataset for style images, applying the TensorFlow Hub model for arbitrary image stylization to generate a diverse set of stylized images. A Siamese Network with MobileNetV2 as the base model was employed to distinguish between positive and negative pairs of stylized images. After training the Siamese Network, we froze its layers and trained a linear classifier to evaluate the quality of the learned representations. The results demonstrated that the self-supervised approach successfully learned meaningful features, as proved by the classifier's performance.

The integration of style transfer with contrastive learning in this project highlights several key insights and potential areas for further exploration. Firstly, the use of style transfer introduced a unique way to generate diverse training data, which proved effective in enhancing the robustness of the learned features. This approach can be extended to other types of data augmentation techniques to further improve self-supervised learning models.

However, there are some limitations to our approach. The choice of datasets and the specific style transfer model can significantly influence the results. While the COCO and WikiArt datasets provided a good mix of content and style diversity, experimenting with other datasets and more advanced style transfer models could yield even better results. Additionally, the current model architecture, while effective, might benefit from further tuning and optimization, including experimenting with different base models or feature extraction techniques.

Moreover, the evaluation metric focused primarily on binary classification accuracy, which, is also important, does not fully capture the richness of the learned representations. Future work could involve more comprehensive evaluation methods, such as downstream task performance or unsupervised clustering metrics, to provide a better understanding of the learned features.

In conclusion, this project successfully demonstrates the potential of using style transfer combined with contrastive learning for self-supervised image representation learning.

IX. PROJECT LINK

https://colab.research.google.com/drive/1squG15RDxChEe_0TzkH5P1tiFz6dWPq0?usp=sharing

REFERENCES

[1] https://doi.org/10.48550/arXiv.1603.09246
[2] https://doi.org/10.48550/arXiv.1906.00910
[3] https://doi.org/10.48550/arXiv.1610.07629
[4] https://doi.org/10.48550/arXiv.2004.11362
[5] https://doi.org/10.48550/arXiv.1606.03498
[6] https://doi.org/10.48550/arXiv.1511.02251