Art Captioning : CS 7643

Maksym Kurashyn, Joseph Ferrin, Lewis Nikuze
Georgia Institute of Technology
{mkurashyn3,jferrin3,jnikuze3}@gatech.edu

Abstract

Art curation and style labelling are typically the domain of trained professionals. Moreover, artworks often contain subtle information that can be represented in different styles, making them difficult for non-experts to interpret. Art is a particularly interesting domain because the images are not real-world photographs, and their accompanying descriptions tend to be more abstract. For this reason, we worked on implementing a museum label generator capable of producing a textual description of an exhibit and determining its artistic style. Our system consists of two main components: a style classifier and an encoder–decoder model that translates CNN embeddings into textual descriptions. This problem brings together computer vision, natural language generation, and cultural heritage studies. It also aligns with current trends in multimodal learning, where vision and language models are trained jointly. Finally, we outline the different approaches we explored, the challenges we faced, and the results we obtained.

1. Introduction/Background/Motivation

Our goal for this project is to generate realistic captions for images of artwork and classify their style. This is unique from typical image captioning models since they are usually trained on real-life photos. We also wish for our captions to capture emotion and artistic meaning instead of only an description of the image. To achieve our goal, we train two separate models, one for classifying the style of the artwork, and one to generate a descriptive caption of the artwork. Our classifier is based on CNN architecture. Our caption generator uses encoder and decoder architecture, with a CNN for the encoder and either a transformer or LSTM model for the decoder. We also experiment with embedding predicted style and emotion labels with the encoded images.

Current methods for art classification are primarily based on convolutional neural networks due to their ability to learn visual patterns from pixel data that generalize across styles [8]. We used a fine-tuned ResNet-50 model. Previous research has also achieved strong results using architectures such as VGG [12], Inception-V3 [13], and others. More recent work explores ensemble approaches with two-stage classification pipelines [7], and transformer-based models [3].

There are notable limitations, including class imbalance [7] in available datasets and overlapping visual features across styles. Another challenge is decoupling content from style [5]. This becomes particularly problematic when certain subjects are over-represented in specific styles, for example, if many Impressionist paintings are landscapes, the model may incorrectly associate landscape content with the Impressionist style. Also, since pretrained models are based on real photos, we would need to fine tune our model to detect artistic representations of objects.

For classification, we used the WikiArt dataset [15], which contains 80,042 pieces of visual art. We focused on classifying artworks by style, with the goal of predicting the artistic style of each image. The dataset is split across 27 styles, with class sizes ranging from 93 to 13,028 samples.

Despite the challenges, automated art classification has useful applications, such as facilitating large-scale cataloging and improving accessibility for the general public by making it easier to search for and understand visual arts.

For the image captioning, we experiment with two datasets: SemArt [4] and Artemis [1] datasets. SemArt contains more than 20000 images, which are split into 3 samples: 19244 training images, 1069 test images and 1069 validation images. This dataset contains painting images associated with attributes and comments for semantic art understanding. For each of the images it contains one long description. The Artemis dataset uses the same images from WikiArt and can have multiple captions per image. For each caption, the annotators give free-form, natural-language explanations of the artwork and their emotional reaction. Each caption is then labeled with one of nine emotions based on the annotaters interpretation.

Our baseline captioning model will be based on the popular paper ”Show, Attend and Tell” [16] which uses a CNN encoder and and LSTM decoder with attention. We also try swapping the LSTM decoder with a transformer decoder.

An important aspect of captioning art is to capture the emotion and artistic meaning of the piece. To try to achieve this, we implement methods from a paper that uses style predictions to improve image captioning [11]. In the paper they embed style labels with the encoded images and train with contrastive learning.

2. Approach

We train the style classifier on the the WikiArt dataset. The dataset was reduced to 20 classes, each containing at least 1,000 samples. The dataset was then split into training, validation, and test sets in proportions of 0.70, 0.15, and 0.15, respectively.

We used a ResNet-50 model pretrained on IMAGENET1K-V2 weights. We first attempted feature extraction using only the final layer, but the results were poor; this indicated that artwork classification was too complex for the model to learn meaningful representations from the last layer alone. We fine-tuned the last four layers and trained the model for 200 epochs. The training and validation losses showed consistent learning over time, and the accuracy improved accordingly, reaching approximately 0.67 on 20 styles and around 0.80 when the dataset was reduced to 10 styles.

Because we anticipated issues related to class imbalance, we balanced the styles by sampling an equal number of images (700 per class for training). Additionally, we expected challenges due to the intrinsic difficulty of artwork classification. One such challenge is the need to decouple content from style [5], which complicates the ability of CNNs to learn discriminative style features efficiently.

Our choice of pretrained model and fine-tuning strategy was inspired by previous work on CNN architectures for artwork classification. Although that work used ResNet-18, we found ResNet-50 to perform better. We also adopted their approaches for handling class imbalance and applying effective image transformations during training, such as resizing images to 256 pixels, performing random 224×224 crops, random horizontal flips, brightness and contrast adjustments, and normalizing pixel values with ImageNet mean and standard deviation [2].

In order to implement the image captioning, we decided to implement 2 different approaches and compare their performance. Both approaches are SeqToSeq models, but one is using a transformer-based decoder, and another one is LSTM-based. The encoder has the same idea in both approaches: it is CNN-based. Its main goal is to transform the image into patch embeddings. As a starting point, we chose ResNet-50 model and tried to train it from scratch. It would give the best results, but required a significant amount of time. Also we tried to use pretrained weights IMAGENET1K-V2, which increased the speed of training dozens of times, but the encoder wasn't able to generalize for our task, that's why we started fine-tuning layers from the last ones to find the proper balance.

For the model, first we feed the images through the resnet-50 up to the final convolution block. This gives us a spatial feature map that we can flatten into a sequence of vectors. This sequence is what we pass as a sequence of tokens to either decoder. For the LSTM decoder, the model uses cosine attention over the feature sequence. The transformer decoder, instead of recurrence, uses multi-head self-attention and cross-attention. After training, we can feed images to the model to generate captions.

To further add to our model, we embed emotion and style labels into the encoded image and add both cross-entropy and contrastive loss. First, the sequence of image tokens is averaged into a singe vector. Then that vector is projected with two linear heads, one for style and one for emotion. This gives logits to predict style and emotion. The cross entropy loss of these predictions is added to the loss function. During training, the actual labels are embedded, projected, and appended to the image encoding to then pass to the decoder. During inference, the predicted labels are used instead.

We also add in contrastive loss. The encoded image and the emotion and style labels are projected into the same space. Contrastive loss makes the learned embeddings of the same class closer together, and different classes farther apart. For contrastive loss we use InfoNCE [9]. The losses are then combined together using adjustable weights:

$$\begin{aligned}\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{caption}} + \lambda_1 \mathcal{L}_{\text{style}} + \lambda_2 \mathcal{L}_{\text{emotion}} \\ + \lambda_3 \left(\mathcal{L}_{\text{contrast-style}} + \mathcal{L}_{\text{contrast-emotion}}\right)\end{aligned} \tag{1}$$

3. Experiments and Results

For style classification, the primary metrics used to track learning were the training and validation losses. As shown in Figure 1, the results indicate clear evidence of learning, as both training and validation losses decreased during training. Moreover, Figure 2 shows that accuracy increased as the number of epochs grew.

The trained model was then evaluated on the test set, achieving an accuracy of 0.67. More detailed test metrics (recall, precision, and F1-score), are presented in Table 1. These results are significantly better than random chance, and given the limited dataset size and the complexity of art style classification, we were satisfied with the performance. Nonetheless, further improvements could likely be achieved by training on a larger and more diverse set of artworks.

These results also highlight the effectiveness of using a ResNet-50 backbone. The model was able to leverage transfer learning from the IMAGENET1K-V2 pretrained weights and successfully adapt them through fine-tuning for the task of art classification. The final performance further

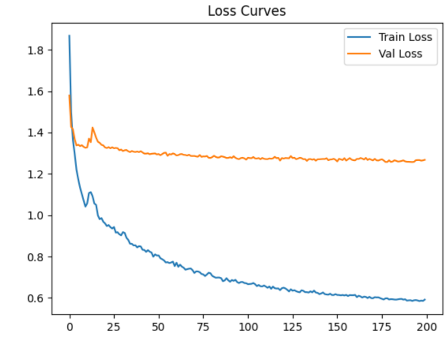


Figure 1. Loss curve of the image classifier on 20 styles

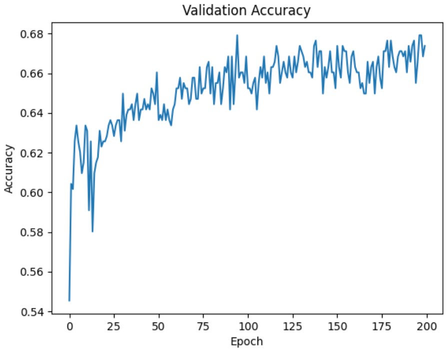


Figure 2. Accuracy of the image classifier on 20 styles

demonstrates the contribution of several architectural and training decisions, including the use of CrossEntropy Loss with label smoothing, the AdamW optimizer with weight decay, and the strategy of freezing and gradually unfreezing layers during training, approaches learned from previous work [10, 6].

Regarding image captioning, it was decided to compare the captions generated with SemArt and Artemis. For SemArt we performed a cross-validation using K-fold method using different set of parameters. The parameters, tuned during the cross-validation, include encoder output layer size, decoder embedding size, decoder hidden layer size, dropout value and number of LSTM layers. We made the size of encoder output layer and decoder hidden layer the same to avoid the necessity to add an additional projection layer. Decoder hidden layer determines how far back the model remembers. Embedding size defines the representative capacity of words, it allows to find the difference or the similarity between words more precisely. The number of LSTM layers determines the representative capacity of sentences, allows to learn the sentence structure more efficiently. Dropout value becomes more important for longer captions, because it prevents overfitting. We trained each set 4 times during 5 epochs and found the one that allowed us to reach the best results. The CIDEr metric [14] was chosen because it is one of the best evaluation metrics for image captioning, which aligns with human perception and tolerates synonyms well. We struggled to get a CIDEr larger than 0 for SemArt. This may be explained by the fact that SemArt contains really long descriptions and most of the information in these descriptions is not factual. In SemArt each description is associated with an artistic comment, which contains historical context, symbolism, artist background, information about the time period etc. This information can't be easily captured by the decoder from the encoded images. In order to make it work for SemArt dataset, it is necessary to make captions shorter, normalize text structure, and remove as much non-descriptive information as possible. We tried to perform such manipulations on datasets using LLM. The results are presented in the figures 3 and 4.

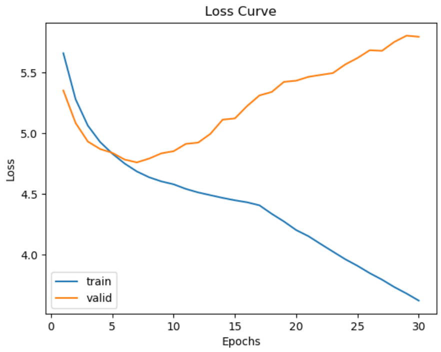


Figure 3. LSTM Loss on the training and validation datasets

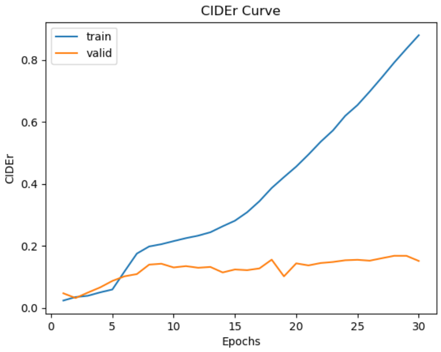


Figure 4. CIDEr for LSTM on the training and validation datasets

Huge validation loss might be explained by the fact that regular loss punishes any movement from the original text. Even if we use absolute synonyms. That's why we also use CIDEr to evaluate the model.

The highest CIDEr value we reached was 0.17. It was reached with parameters encoder output layer size = 1024, decoder hidden layer size = 1024, embedding size = 512, lstm layers = 2, dropout = 0.3 . It is a quite low value and

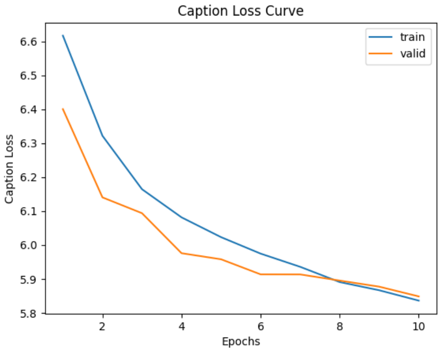


Figure 5. Loss for LSTM on ArtEmis

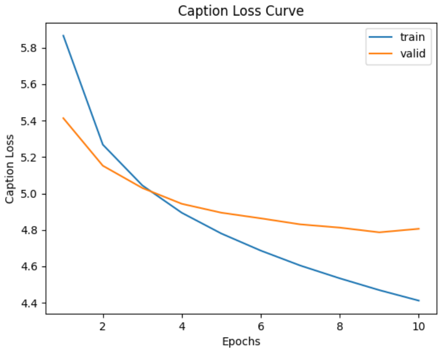


Figure 6. Loss for transformer on ArtEmis

might be explained by the huge variance in the format. Two similar images might have absolutely different captions and contain little information about what is actually depicted in the image. It is possible to increase the quality of captions further by tedious preprocessing, but it requires too much time and effort.

From here we tested the LSTM version on the ArtEmis dataset. We used the same cross-validated parameters from before to keep the models comparable. Training on this dataset led to a CIDEr score of 0.0613. This improvement could be due to the nature of the captions in each dataset. The SemArt dataset has captions with historical or cultural context, like ”this painting is said to have inspired Van Gogh.” It is difficult for a model to come up with knowledge like this just by viewing an image. ArtEmis, on the other hand, give captions based on reactions of average people. For example: ”I think the woman in the picture is beautiful and I love the soft features, the pinks and roses.” Captions like this make it easier for the model to learn features in an image such as people, colors, and shapes. The chose of using ArtEmis is also good because it uses WikiArt, which is what we train our style classifier.

Next we swap out the decoder for our transformer and train on ArtEmis. For parameters, we keep dropout at 0.03 and tune to get the following: number of heads = 8, layers = 6, embedding dimension = 256. The resulting best CIDEr score is 0.099. This is a noticeable improvement over LSTM. This follows current trends where transformers get better results due to being able to attend over the whole sequence of tokens at once.

At this point, our generated captions successfully identify things like humans, buildings, and animals. They also give a natural-language reaction to the artwork with emotion. For more abstract art, the captions just describe things like color, shape, and emotion.

From here we look to improve our model further by embedding the style and emotion labels from WikiArt and ArtEmis into the model. Our thought is that these labels can help the model to generate captions if it can first guess the style and emotion for an image. This is useful for the ArtEmis dataset since each image can have multiple captions, each with different emotions. For example, a caption with emotion “fear” could be “this painting is scary” and a caption with emotion “excitement” could be “this painting is cool.”

After the addition of style and emotion embeddings, the CIDEr score is 0.1013. This is only a marginal improvement. The captions seem to be about the same quality as

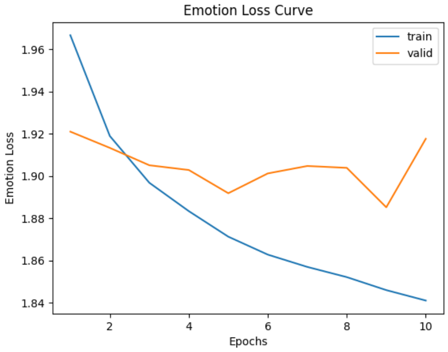


Figure 7. Emotion loss for transformer + style and emotions

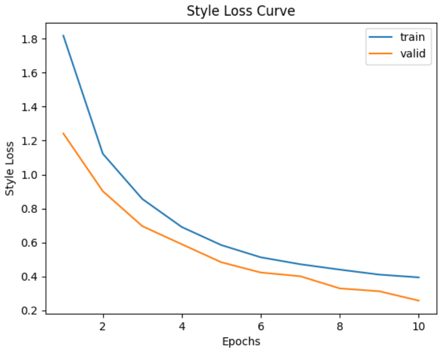


Figure 8. Style loss for transformer + style and emotions

Art Style	Precision	Recall	F1-score
Abstract_Expressionism	0.66	0.74	0.70
Art_Nouveau_Modern	0.67	0.58	0.62
Baroque	0.64	0.65	0.65
Color_Field_Painting	0.71	0.79	0.75
Cubism	0.77	0.83	0.80
Early_Renaissance	0.57	0.83	0.68
Expressionism	0.64	0.56	0.60
High_Renaissance	0.67	0.56	0.61
Impressionism	0.54	0.61	0.57
Mannerism_Late_Renaissance	0.61	0.66	0.63
Minimalism	0.79	0.75	0.77
Naive_Art_Primitivism	0.72	0.63	0.67
Northern_Renaissance	0.76	0.70	0.73
Pop_Art	0.72	0.61	0.66
Post_Impressionism	0.37	0.24	0.29
Realism	0.80	0.24	0.36
Rococo	0.71	0.67	0.69
Romanticism	0.33	0.14	0.19
Symbolism	0.43	0.18	0.26
Ukiyo_e	0.79	0.97	0.87

Table 1. Per-class performance metrics for WikiArt classification.

Art Style	Precision	Recall	F1-score
Abstract_Expressionism	0.95	0.94	0.94
Art_Nouveau_Modern	0.97	0.93	0.95
Baroque	0.92	0.92	0.92
Color_Field_Painting	0.97	0.97	0.97
Cubism	0.91	0.95	0.93
Early_Renaissance	0.82	1.00	0.90
Expressionism	0.91	0.71	0.80
High_Renaissance	0.83	0.75	0.79
Impressionism	0.89	0.98	0.93
Mannerism_Late_Renaissance	0.85	0.87	0.86
Minimalism	0.98	0.96	0.97
Naive_Art_Primitivism	0.87	0.91	0.89
Northern_Renaissance	0.93	0.91	0.92
Pop_Art	0.88	0.98	0.93
Post_Impressionism	0.87	0.85	0.86
Realism	0.86	0.93	0.89
Rococo	0.92	0.98	0.95
Romanticism	0.90	0.84	0.87
Symbolism	0.90	0.80	0.84
Ukiyo_e	1.00	1.00	1.00

Table 2. Style performance metrics from the image captioner.

before. Now that we have multiple losses, we can plot them individually (Figures 7 and 8). The loss for the emotion label seems to have the hardest time improving. This makes sense as the emotion labels come from the captions, which means each image will multiple emotion labels and the model will not be able to always be right.

Since this model is now predicting style, the metrics can be compared to our standalone classifier (Table 1 and 2). Interestingly, the prediction accuracy is even better, with an accuracy of 0.91. This could be due to the fact that we are now training with additional information: the emotion and captions from ArtEmis. Since the model is learning to caption as well, the learned embeddings could be more meaningful than just the CNN trained on its own.

We can further evaluate our model by taking a look at the captions that our model generates on images outside our training set and judging their quality. Overall, the captions make sense. They show that the model can identify objects, colors, and overall feeling. While the captions aren’t always completely correct, we still consider our model successful in what we set out to do. Overall, to us the model using CNN and transformer without emotion and style make the best captions. The model with emotion and style added seems to misidentify objects in the art more and is more repetitive. For most abstract artworks it says ”I feel confused because i don’t know what this is supposed to be.” See Table 3 for a sample of images and their true and generated captions.

For future work on this topic, models could be trained on larger datasets to improve accuracy. Also, the model could be trained on different datasets to get a different style of captions. The Artemis data results in captions similar to the natural reaction of an average person, such as “The woman looks happy and the colors are pleasant.” A dataset with captions by art experts could result in more specific descriptions and historical context.

4. Project Code Repository

The Github repository of our project is at https://github.gatech.edu/mkurashyn3/DL_Project We have also submitted a zip file of our project code.

5. Work Division

Each member has contributed an equal amount of work for the project. Please see Table 4 at the end of the document for a description of each group members’ contributions.

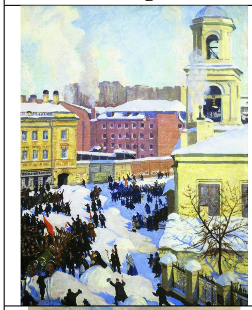
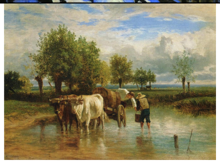
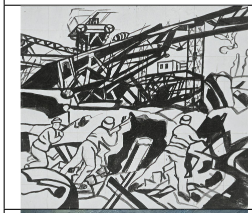
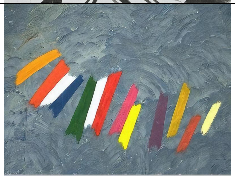
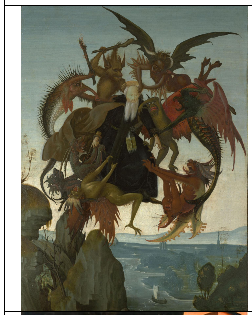
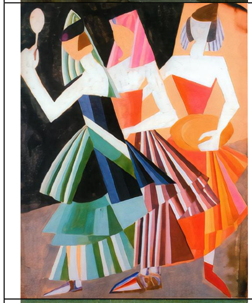
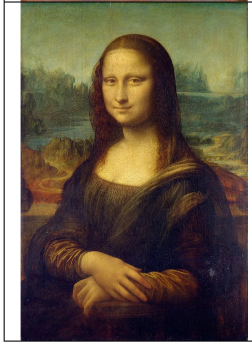
Image	Real Caption	Transformer Caption	Transformer + Style / Emotion Caption
	“The clear blue sky against the snowy ground plus all the people makes it look like everyone is coming out after a big storm finally passed.”	“the scene is very well done and the buildings are so bright and the buildings are very well done.”	“the buildings are so tall and the people are all gathered together.”
	“The beautiful landscape in this picture blends well with the naturalness of the cows working in the field and water.”	“the cows are very calm and serene.”	“the cows are enjoying a nice day in the field.”
	“It seems like a war scene which makes me anxious.”	“the black and white colors are very dark and the people are very dark and the people are in a way.”	“i feel confused because i don ’ t know what this is supposed to be.”
	“A strange conception of striped colors on a hard blue background.”	“the colors are bright and the painting is very colorful and the colors are bright and the subject matter is very calming.”	“the colors are bright and the subject matter is very interesting.”
	“something nefarious and devilish is happening here and its quite terrifying, like demons are yanking and pulling on somebody”	“the red and black colors are very striking and the animal is very detailed.”	“the angels are flying in the sky and the angels are very powerful.”
	“Bold colors and angular shapes, combined with the spoon waving in the air, tell me these women are filled with power and purpose!”	“the colors are bright and the shapes are bright and the colors are bright and the shapes are bright and cheery.”	“the colors are bright and the shapes are bright and fun.”
	“the look of vague amusement on her face and the subtlest smile is still captivating”	“the woman ’ s expression is very calm and she looks calm.”	“the woman ’ s expression is calm and content.”

Table 3. Examples of generated captions for a sample of images.

References

[1] Panos Achlioptas, Maks Ovsjanikov, Kilichbek Haydarov, Mohamed Elhoseiny, and Leonidas Guibas. Artemis: Affective language for visual art. *CoRR*, abs/2101.07396, 2021. 1

[2] Bengal1. The effect of dataset type on artwork classification. https://github.com/Bengal1/The-Effect-of-Dataset-Type-on-Artwork-Classification, 2023. GitHub repository. 2

[3] Alexandra Diem and Markus Mandl. Automatic classification of portraits: Application of transformer and cnn based models for an art historic dataset. In *CEUR Workshop Proceedings*, 2023. Compares ViT to ResNet for art classification. 1

[4] Noa Garcia and George Vogiatzis. How to read paintings: Semantic art understanding with multi-modal retrieval. In *Proceedings of the European Conference in Computer Vision Workshops*, 2018. 1

[5] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. A neural algorithm of artistic style. *Journal of Vision*, 16(12):1–15, 2016. This foundational work in Neural Style Transfer defined the separation of style and content features. 1, 2

[6] Tong He, Zhi Zhang, Hang Zhang, Zhongyue Zhang, Junyuan Xie, and Mu Li. Bag of tricks for image classification with convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019. 3

[7] Aditya Joshi et al. Art style classification with self-trained ensemble of autoencoding transformations, 2020. arXiv preprint. Discusses ensemble methods and achieving strong results on WikiArt. 1

[8] Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2012. The original AlexNet paper that popularized CNNs for ImageNet. 1

[9] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 2

[10] Tasfia Shermin, Shyh Wei Teng, Manzur Murshed, Guojun Lu, Ferdous Sohel, and Manoranjan Paul. Enhanced transfer learning with imagenet trained classification layer. *arXiv preprint arXiv:1903.10150*, 2019. 3

[11] Shivacharan S. Shivaji, Xiang Li, Kenneth T. Church, and A. M. Subramanyam. Style-aware contrastive learning for multi-style image captioning. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1472–1486, 2023. 2

[12] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations (ICLR)*, 2015. 1

[13] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1

[14] Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation, 2015. 3

[15] WikiArt.org. Visual art encyclopedia. https://www.wikiart.org/. Accessed: 20 Nov. 2025. 1

[16] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, pages 2048–2057, 2015. 1

Student Name	**Contributed Aspects**	**Details**
Maksym Kurashyn	Image captioning (SemArt, LSTM)	Trained Seq2Seq model using a CNN encoder and LSTM decoder on the SemArt dataset.
Joseph Ferrin	Image captioning (Transformers); Emotion and Style contrastive learning	Added a transformer decoder and contrastive learning to the pipeline, and trained multiple configurations on the ArtEmis dataset.
Lewis Nikuze	WikiArt Style classification	Data preprocessing, fine-tuning, training, and testing of the classification model.

Table 4. Contributions of team members.