Team 067 Report

Xin Guo (xguo347), Yan Sheng (ysheng48), Jesse Watson (jwatson313),
Cody Westgard (cwestgard3), Tianhua Zhu (tzhu97)

1. Introduction

Art is one of the world's great reservoirs of cultural heritage, and the rise of digitized open source art collections has made the sharing of this resource easier. The Metropolitan Museum, and the Chicago Museum, and even Wikiart provide open source digitized versions of some of the world's greatest artistic heritage.

Digitized collections further allow cross-cultural interaction, for example Taipei's National Palace Museum museum of Art has also created an open source digitized collection.

Using these new resources, however, provides challenges and opportunities. The combination of different data sources, how to make data accessible, especially to non-experts, and how to leverage these datasets to reveal salient insights are all questions worth pursuing.

2. Problem Definition

Newly available datasets offer new resources and potential in art history, art appreciation, design, and more. While there are potential insights for experts, for example, cross-cultural, perhaps the greatest benefit would be for the uninitiated.

For non-experts, gaining insight into art can be a daunting task. If an artist or period is known, then it is relatively easy to find information and similar artwork. This is often not the case, however. Further, cross-cultural insights might also be difficult, adding a further dimension of uncertainty. How can the uninitiated explore the art available in these new resources? How can experts leverage disparate datasets for new insights? Having access to these datasets does not necessarily make them accessible. Our task then is two fold: to make parsing these large datasets manageable, and to further give salient insight. Essentially: the dataset and the model.

In terms of the dataset, we are left with several specific problems. How and where to store such large amounts of data? How to combine disparate datasets? How to make searching through them convenient? Clearly a good amount of processing is required, to make these large datasets manageable. Our project solves this using feature extraction. See the methods section

Once the dataset has been processed, the model has several tasks. How can we best leverage machine learning (ML) techniques to give users insight? Our group has chosen to use similarity as a point of entrance. With the ultimate goal of returning similar artwork to users.As a starting point, our group has decided on using a user provided image upon which to base our calculation of similarity. How can we use ML to calculate similarity?

Our task, plainly stated: using a publicly available database of artwork from multiple regions, we aim to build a web application that, given a user uploaded image, returns similar artwork across multiple aspects, ie. Western and non-Western, Fortunately, the application of ML to artwork is not uncommon.

3. Survey

Digitized art collections are a widely used ML resource [10]. Some studies apply traditional classification methods to artwork. Falomir et al (2018)[7] use SVM to reach 65% accuracy in classifying art by artistic period. Interestingly, they use a qualitative color palette to make complex quantitative data more digestible; of potential use for our project.

Shamir et al (2010) [14] have a similar goal[7], however, they first construct a "similarity matrix" which ranks works in relation to each other.

AlBadarneh et al (2017)[1] and AlyanNezhadi et al (2019)[2] likewise use SVM, but in this case to distinguish between forgeries and authentic work. AlBadarneh divides each artwork into smaller "patches" [1] for comparison, while Alyan Nezhadi focuses on feature extraction.

These studies [1][2][7][14] all involve classification by similarity, a problem our project also faces. However, they focus on subtle differences: detecting specific artists[1][2] or style[7][14]. Our project focuses on providing insight through comparable works, rather than detailed identification.

Further, all are scope limited, using small datasets[1][7], and limited artists[1][14] while our

project requires the integration of both disparate collections as well as expert supplied metadata.

Jiang et al (2006)[9] and Messina et al (2018) [11] provide potential guidance.

Jiang successfully identifies traditional Chinese art, which is further classified by period. While limited in scope, the study gives a clue to integrating artwork from disparate, especially non-Western, sources. Messina creates a recommendation system via integrating metadata with features extracted using Deep Neural Networks (DNN), achieving favorable results our project can seek to imitate. However, the study focuses on ecommerce and so is of limited applicability.

Central to these [11][9], and other[1][2][7], studies is feature extraction. Zujovic et al (2009) [16] study different classification models and feature extraction pairings, finding that a model's results can vary greatly with extraction method. Čuljaket et al (2011)[6] focus on texture and color features in artwork classification with a relative amount of success, but also find different combinations produce accuracy below 50%. While illustrative, these are earlier studies, limiting their direct use.

Liu et al (2021)[10], more recently, propose novel color and texture features used in concert with a Convolutional Neural Network (CNN) for art classification, achieving greater than 86% accuracy. Potentially informing our own extraction methods. However, the study is limited to only 927 portraits.

Condorovici et al (2015)[5] take a perception based approach, extracting features for shape, color, and boundary, which feed to individual classifiers. While potentially too time consuming for our purposes, these results[5] continue [10][6] to indicate the importance of color in classification. A commonality which can be used to simplify our own approach to feature extraction.

Finally, many studies focus on the application of Neural Networks to classifying artwork. Westlake et al. (2016) [15], Pinciroli Vago et al. (2021)[12], Gatys et al. (2015)[8], all apply CNN models. Westlake tests varying Convolutional layers in figure detection, finding lower layers desirable. While Pinciroli Vago uses Class Activation Maps (CAM) to improve the CNN model; identifying highly weighted prediction areas. Gayts, interestingly, finds that in applying CNN the content (subject) and style (filter) of an image is separable, a useful fact when sorting image by subject.

Unfortunately, the above studies apply to only a small number of subjects [8] or themes [15][12], compared to our much wider project. Both Castellano1 et al (2021)[3] and Sabatelli et al (2019)[13] provide some help here. Castellano gives a broad overview of deep learning methods applied to large artwork dataset classification. While a valuable resource, they do not provide much guidance on method selection. Sabatelli not only applies CNN to multiple large datasets, but uses Transfer Learning to train their model on multiple datasets for improvement. Relevant, as our project will make use of multiple datasets.

Finally, Castellano et al (2021)[4] take the novel approach of applying CNN to feature extraction for clustering. While interesting, this is not a problem we must overcome as the databases provide labels for the paintings. These studies [3][4]][13] highlight the complexity in selecting and applying newer models to artwork classification. Our project, at its base, seeks to overcome this obstacle as a means of returning to the user salient works and information for comparison.

4. Proposed method

Although there is a wide breadth of literature on the application of ML to artwork, studies usually focus on using digitized artwork as a challenge. Most of the studies outlined above are primarily concerned with matching an artist or genre, essentially focusing solely on accuracy. While our project does focus on similarity as a metric, the primary goal is on giving users a wider range of connections.

As such, at present we have merged two collections from vastly different areas: Taipe's National Palace Museum collection, and the Metropolitan Museum of Art Collection

Intuition

Therefore, rather than simply using our model as a "black box" and returning a similar image or images, our goal is to grant the user insights based on similarity to their specified image. By combining ML with expert provided

labels to give information such as: the geographic origin of similar works, the time period similar works were produced, the cultural background of similar works in relation to each other, in addition to other basic information such as artist and title.

Our work is also focused on cultural comparisons. Rather than focusing on collections from a single museum or culture, our application seeks to integrate disparate collections to allow for cross-cultural insights. Essentially allowing users access to greater comparisons and contrasts.

Finally, our project is less focused on using artwork to improve ML techniques, and is instead focused on using ML techniques to grant insight into artwork. This is exemplified in the modeling techniques used; combining state-of-the-art techniques for feature extraction and data preparation, with relatively simple distance metrics. We are not focused on finding only works that match as accurately as possible, but indeed on finding a broad range of works that could be considered similar, but not identical.

Therefore, innovations include: **1)** the data set used, being the union of multiple disparate collections **2)** our choice of visualizations, focused on multiple insights **3)** as well as our combination of complex modeling for feature extraction with relatively simple heuristics for artwork selection.

Approach

Our proposed method, therefore, involves four parts: the processed dataset, a user submitted image, an algorithm which returns images based on similarity, and finally our visualizations. Each of these parts is explained in detail below.

The dataset itself is an important step in the process of providing user insight. As stated above, one of our group's primary focuses is on facilitating cross-cultural insight. Therefore, our group has chosen to combine open source museum provided collections from across the globe: as of now *The Metropolitan Museum of Art*, and *Taipei's National Palace Museum of Art.* We also chose to limit our dataset to paintings, instead of all art (such as pottery), in order to give users clearer direct comparison, which still leaves 2,359 images for the dataset.

Finally, we also store the meta-data for each image as a separate dataset. This metadata includes expert provided labels for artist, geographic information, and time-data; all of which will be used in the visualization step. Note that the original, unprocessed, images are stored separately on the Google Cloud Platform (GCP), while the reduced dataset and labels are stored on Github. Figure 1 illustrates the wide geographic and period distribution of the dataset as a stacked bar chart.

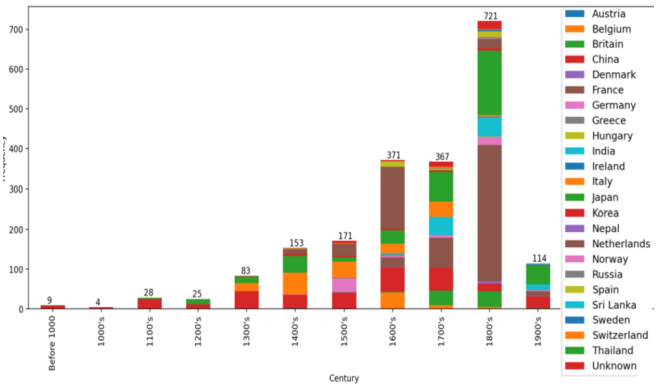

Figure 1. Frequency of Country by Century

Once combined, the size of the dataset becomes a problem. Feature extraction is used: simultaneously making the dataset smaller and more manageable for later tasks. Our group makes use of CNN models to accomplish this. CNN models can be used to essentially distill an image to its most salient features. Figure 2 provides a visual example of CNN as applied to an image from the dataset.

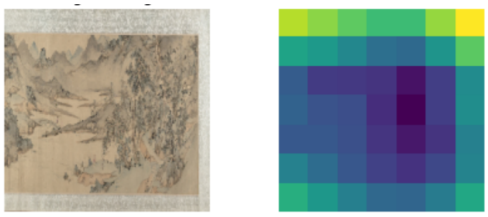

Figure 2. Feature Extraction Using CNN

The feature extraction process greatly reduces the memory needed to store an image, thus we have a more manageable dataset. More details on the algorithm and evaluation methods used to select the model, are included in the evaluations section.

After applying feature extraction to the original dataset and storing the result we move on to the main task: returning similar images to users. This is accomplished using distance metrics.

Essentially, we take an input image and apply the same extraction methods as on the

feature extracted dataset. We are then able to compare the input image to the dataset, and return images with the least distance. Figure 3, below provides a simplified illustration, with the dataset shown as a scatterplot.

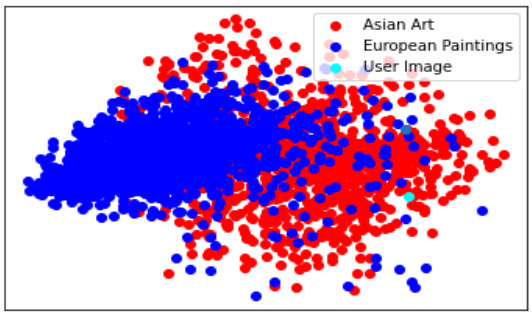

Figure 3. Art Visualized in 2 dimensions

Here, the dataset has been plotted using two features (for visualization purposes only) and colored by region of origin. We can see there appears to be some separability between regions. Likewise, a sample user image has been shown, which appears to have more in common with Asian art then European art.

This simplified example illustrates distance as a dissimilarity. Our method involves using Euclidean distance. An example is shown below in Figure 4. Given a new image of a horse and bamboo stalks, the model finds two images of a horse, and an image of bamboo stalks.

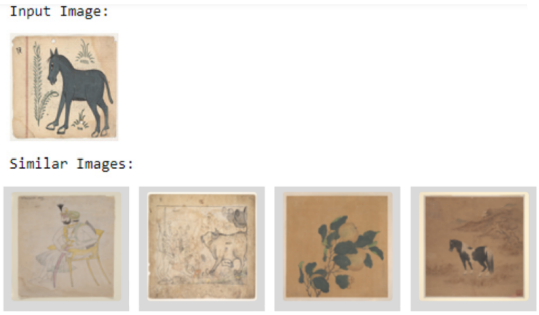

Figure 4. Sample Input and Similar Images

Further information on the selection of distance measurements used can be found in the Evaluation section.

The final model and its results provide the basis for our application, however, the visualizations are the main goal. Our team chose to use the popular Streamlit platform to host the application. One of the primary benefits of Streamlit is the ability to easily merge python, excellent for ML tasks, with visualizations.

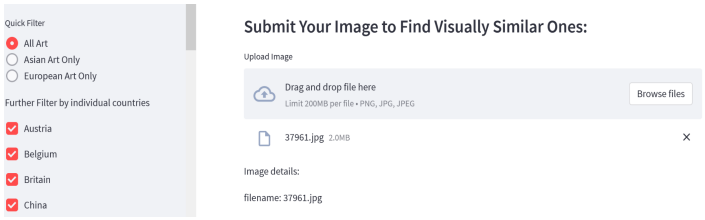

Figure 5. The application hosted on Streamlit

As stated above, our project is aimed at granting user insights they might not otherwise see. As such, the application offers many options for customization. For example, we allow users to filter for time as well as geography (Figure 6).

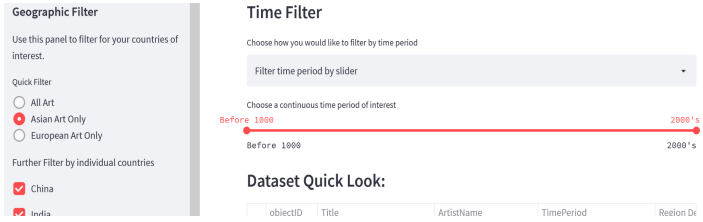

Figure 6. Filter options by Time and Geography

Besides filtering options, the application also provides several distributional visualizations, in addition to the returned similar images. Figure 7 shows a Choloropleth map, which colors regions based on the country of origin of the returned similar images. Additional visualizations included the stacked bar-chart, shown in Figure 1, which changes interactively based on the user's filter choices and image.

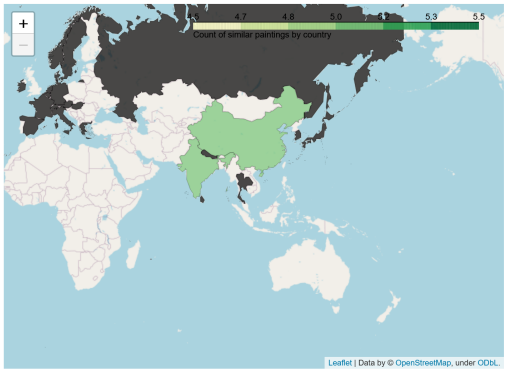

Figure 7. Similar Paintings distributed as a Choropleth

5. Experiments/ Evaluation

The primary aim of our project, in terms of ML, is to return visually similar images based on a user uploaded image. Therefore we are tasked with answering various questions 1) How to compare similarity between images? 2) How to evaluate the performance? 3) What method has the lowest testing error? 4) what method is most computationally efficient?

Our first attempt was to simply vectorize images and compare similarity using Euclidean distance. This caused two major problems: training time and visually dissimilar results. As mentioned above, to combat these issues, we turned to feature extraction. Through research we found that the current state of the art methods involve Convolutional Neural Network (CNN) algorithms which allow for both feature extraction and performance comparison. This allows for a streamlined approach, summarized in figure 8.

Figure 8. Algorithm Evaluation Approach

Feature Extraction

We test multiple CNN algorithms for feature extraction. specifically ResNet50[9], and VGG19[13], as well as a customized CNN model. PCA was also tested, however, the results were not as successful.

ResNet50 uses 50 layers, including 48 convolution, 1 MaxPool layer and 1 AveragePool layer. It attempts to fit residual mapping instead of using underlying mapping. VGG16, on the other hand, includes only 13 Convolutional layers, 3 dense layers, 5 MaxPool layers. We also create a customized CNN model, which takes the 4th block from the VGG16 model, 3 Convolutional layers and a max pooling layer. Figure 9. summarizes the architecture of ResNet50 and VGG16 algorithms.

We apply all of the CNN models to the original dataset, thus creating three different feature selected datasets. All images were first re-sized to 224 * 224 * 3, rescaled, and trained with the same batch size and epochs. However, the output size and time required varies for each model. Training time took from 1 to 20 minutes, with VGG16 being the fastest.

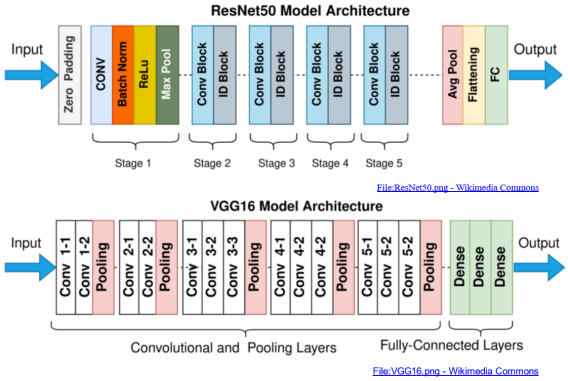

Figure 9. ResNet50 and VGG16 Architecture

Similarity Comparison

Distance metrics are the traditional method of measuring similarity. Once the feature extracted dataset is prepared, image features can be directly compared between images using distance to quantify dissimilarity. We experimented with Euclidean distance and cosine similarity metrics.

Euclidean distance measures the direct distance between two flattened image vectors. Cosine Similarity on the other hand calculates the similarity of two vectors by taking the dot product and dividing it by the magnitudes of each vector. That is: $\frac{|A||B|*Cos(\theta)}{|A||B|}$. Generally, we found that Euclidean distance outperformed Cosine similarity.

Performance Evaluation

Several test images were kept separate while training the feature extracted dataset. Similarity calculation was performed between these test images and the dataset. The process was repeated for each CNN generated dataset (ResNet50, VGG16, and our Custom model), and for each distance metric.

Many evaluation metrics were tried, including Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Multi-scale Structural Similarity Index (MS-SSIM)[18], Spatial Correlation Coefficient (SCC), Universal Quality Image Index (UQI). For the most part, the metric results are aligned. The results found RMSE to compare distance metrics and CNN models, across three test images, are shown in Figure 10. Complete results showing a comparison of RMSE and MS-SSIM on four images are shown in the appendix.

Based on the results, we can conclude that in general Euclidean distance outperforms Cosine.

Test Image	Extracted Features	Distance Metric	Top1 RMSE	Avg. RMSE	Test Time (sec)
37961	Original	Euclidean	48.14	48.82	**1554**
	ResNet50	Euclidean	132.16	101.66	**20.64**
		Cosine	130.47	103.70	**10.65**
	VGG16	Euclidean	49.10	57.88	**25.15**
		Cosine	122.11	111.52	**12.40**
	Custom	Euclidean	74.16	66.96	**62.90**
		Cosine	135.25	119.47	**12.75**
40011	Original	Euclidean	45.36	45.79	**1552**
	ResNet50	Euclidean	118.44	104.45	**18.57**
		Cosine	118.72	96.744	**10.76**
	VGG16	Euclidean	127.46	93.96	**24.44**
		Cosine	104.27	105.30	**10.67**
	Custom	Euclidean	81.67	69.01	**64.66**
		Cosine	104.02	100.32	**14.11**
435595	Original	Euclidean	47.83	48.88	**1838**
	ResNet50	Euclidean	60.04	68.91	**42.27**
		Cosine	99.58	118.01	**12.04**
	VGG16	Euclidean	54.33	59.07	**23.46**
		Cosine	115.73	101.76	**12.95**
	Custom	Euclidean	55.92	58.06	**51.96**
		Cosine	91.09	88.68	**14.08**

Figure 10. RMSE comparison by algorithm and distance

Results also seem to indicate that VGG16 performs relatively well when compared to the other techniques, while still being time efficient. Visual comparison further reinforces this. Figure 11 shows the top 5 most similar images for test image 40011, by approach. For full size comparisons for all test images, please see the appendix. For full size comparisons for all test images, please see the appendix.

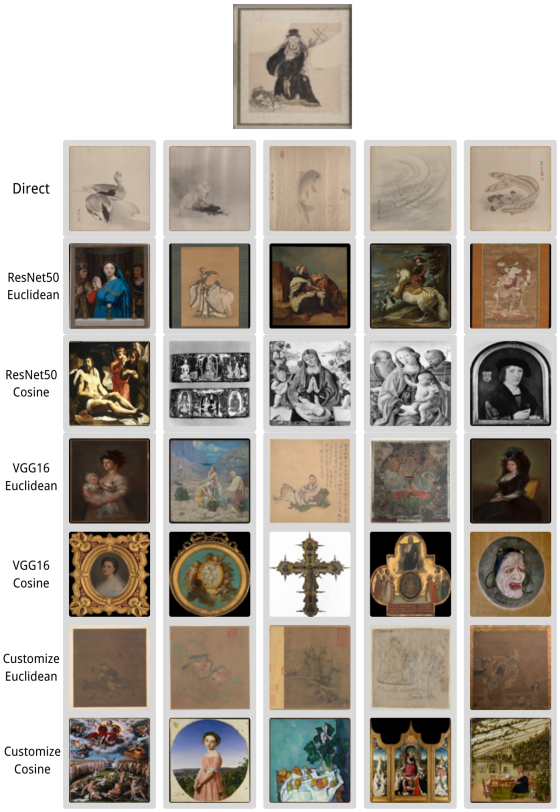

Figure 11. Top 5 images for each algorithm and metric

6. Conclusions and Discussion

The visual comparison reveals interesting insights not necessarily clear in the RMSE comparison. While direct comparison on the vectorized images seems to result in very similar images, in fact they are similar mostly in terms of color. Meanwhile, the feature extracted datasets (when compared using euclidean distance) made with both ResetNet50 and VGG16 seem to find similarity not only in color, but object as well. VGG16 seems to return especially similar figures, and images. Additionally it offers a good mix of European and Asian art, which aligns with our cross-cultural approach. Again, our goal is necessarily to match an image exactly, but to find similar images that grant users potential new insight. Therefore, due to relatively good evaluation performance, time, and visual analysis, we elect to use VGG16 with Euclidean distance for the final model. Our final approach is summarized in Figure 12.

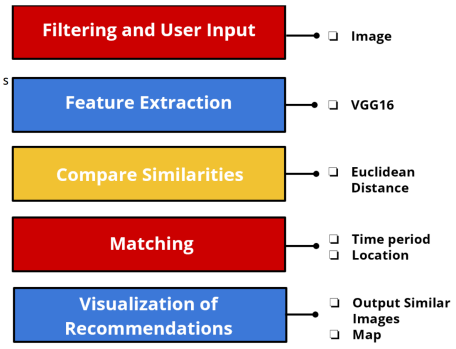

Figure 12. Final Model and Visualization

Future improvements could be made by expanding the datasets used. The current limitations relate largely to the museum collections available, rather than algorithm or modeling capability.

All team members have contributed a similar amount of effort

Works Cited

1. Albadarneh, I. A., & Ahmad, A. (2017). Machine learning based oil painting authentication and features extraction. *International Journal of Computer Science and Network Security* (IJCSNS), 17(1), 8.
2. AlyanNezhadi, M. M., Dabbaghan, H., Moghani, S., & Forghani, M. (2019, February). A painting artist recognition system based on image processing and hierarchical SVM. In 2019 *5th Conference on Knowledge Based Engineering and Innovation* (KBEI) (pp. 537-541). IEEE.
3. Castellano, G., & Vessio, G. (2021). Deep learning approaches to pattern extraction and recognition in paintings and drawings: An overview. *Neural Computing and Applications*, 33(19), 12263-12282.
4. Castellano, G., & Vessio, G. (2021, January). Deep convolutional embedding for digitized painting clustering. In 2020 *25th International Conference on Pattern Recognition* (ICPR) (pp. 2708-2715). IEEE.
5. 5. Condorovici, R. G., Florea, C., & Vertan, C. (2015). Automatically classifying paintings with perceptual inspired descriptors. *Journal of Visual Communication and Image Representation*, 26, 222-230.
6. Čuljak, M., Mikuš, B., Jež, K., & Hadjić, S. (2011, May). Classification of art paintings by genre. *In 2011 Proceedings of the 34th International Convention* MIPRO (pp. 1634-1639). IEEE.
7. Falomir, Z., Museros, L., Sanz, I., & Gonzalez-Abril, L. (2018). Categorizing paintings in art styles based on qualitative color descriptors, quantitative global features and machine learning (QArt-Learn). *Expert Systems with Applications*, 97, 83-94.
8. Gatys, L. A., Ecker, A. S., & Bethge, M.(2016) A Neural Algorithm of Artistic Style . *Journal of Vision* 16(12):326.
9. He, Kaiming et al. "Deep Residual Learning for Image Recognition." *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016): 770-778.
10. Jiang, S., Huang, Q., Ye, Q., & Gao, W. (2006). An effective method to detect and categorize digitized traditional Chinese paintings. *Pattern Recognition Letters*, 27(7), 734-746.
11. Liu, S., Yang, J., Agaian, S. S., & Yuan, C. (2021). Novel features for art movement classification of portrait paintings. *Image and Vision Computing*, 108, 104121.
12. Messina, P., Dominguez, V., Parra, D., Trattner, C., & Soto, A. (2019). Content-based artwork recommendation: integrating painting metadata with neural and manually-engineered visual features. *User Modeling and User-Adapted Interaction*, 29(2), 251-290.
13. Pinciroli Vago, N. O., Milani, F., Fraternali, P., & da Silva Torres, R. (2021). Comparing CAM Algorithms for the Identification of Salient Image Features in Iconography Artwork Analysis. *Journal of Imaging*, 7(7), 106.
14. Sabatelli, M., Kestemont, M., Daelemans, W., & Geurts, P. (2018). Deep transfer learning for art classification problems. In *European Conference on Computer Vision* (ECCV), 4th Workshop on Computer VISion for ART Analysis (VISART IV) (pp. 1-16).
15. Shamir, L., Macura, T., Orlov, N., Eckley, D. M., & Goldberg, I. G. (2010). Impressionism, expressionism, surrealism: Automated recognition of painters and schools of art. ACM *Transactions on Applied Perception* (TAP), 7(2), 1-17.
16. Simonyan, Karen and Andrew Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition." CoRR abs/1409.1556 (2015): n. pag.
17. Westlake, N., Cai, H., & Hall, P. (2016, October). Detecting people in artwork with cnns. In *European Conference on Computer Vision* (pp. 825-841). Springer, Cham.

18. Z. Wang, E. P. Simoncelli and A. C. Bovik, "Multiscale structural similarity for image quality assessment," The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003, 2003, pp. 1398-1402 Vol.2, doi: 10.1109/ACSSC.2003.1292216.
19. Zujovic, J., Gandy, L., Friedman, S., Pardo, B., & Pappas, T. N. (2009, October). Classifying paintings by artistic genre: An analysis of features & classifiers. In 2009 *IEEE International Workshop on Multimedia Signal Processin*g (pp. 1-5). IEEE.

APPENDIX

Figures 13-17

Test Image	Extracted Features	Distance Metrics	Top1 msssim	Avg. msssim	Top1 RMSE	Avg. RMSE	Testing Time (sec)
37961	Original	Euclidean	0.166	0.156	48.14	48.82	1554
	ResNet50	Euclidean	0.070	0.044	132.16	101.66	20.64
		Cosine	-0.013	0.093	130.47	103.70	10.65
	VGG16	Euclidean	0.14	0.132	49.10	57.88	25.15
		Cosine	0.161	0.1421	122.11	111.52	12.40
	Customized	Euclidean	0.120	0.094	74.16	66.96	62.90
		Cosine	0.051	0.040	135.25	119.47	12.75
40011	Original	Euclidean	0.380	0.245	45.36	45.79	1552
	ResNet50	Euclidean	-0.014	0.059	118.44	104.45	18.57
		Cosine	-0.011	0.065	118.72	96.744	10.76
	VGG16	Euclidean	-0.021	0.077	127.46	93.96	24.44
		Cosine	0.167	0.134	104.27	105.30	10.67
	Customized	Euclidean	0.153	0.164	81.67	69.01	64.66
		Cosine	-0.006	0.061	104.02	100.32	14.11
435595	Original	Euclidean	0.275	0.204	47.83	48.88	1838
	ResNet50	Euclidean	0.181	0.153	60.04	68.91	42.27
		Cosine	-0.005	0.057	99.58	118.01	12.04
	VGG16	Euclidean	0.110	0.137	54.33	59.07	23.46
		Cosine	0.206	0.057	115.73	101.76	12.95
	Customized	Euclidean	0.015	0.044	55.92	58.06	51.96
		Cosine	0.059	0.037	91.09	88.68	14.08
435687	Original	Euclidean	0.389	0.364	29.26	30.89	1722
	ResNet50	Euclidean	0.139	0.213	39.37	55.67	78.25
		Cosine	0.091	0.019	101.03	99.14	11.08
	VGG16	Euclidean	0.602	0.468	40.41	38.26	32.74
		Cosine	-0.008	0.026	77.84	103.33	11.61
	Customized	Euclidean	0.462	0.267	32.69	56.48	68.71
		Cosine	0.035	0.095	69.01	72.36	13.41

Figure 13. Full Results comparing MS-SSIM, RMSE for four Test Images

Figure 14. Evaluation results on test image 37961

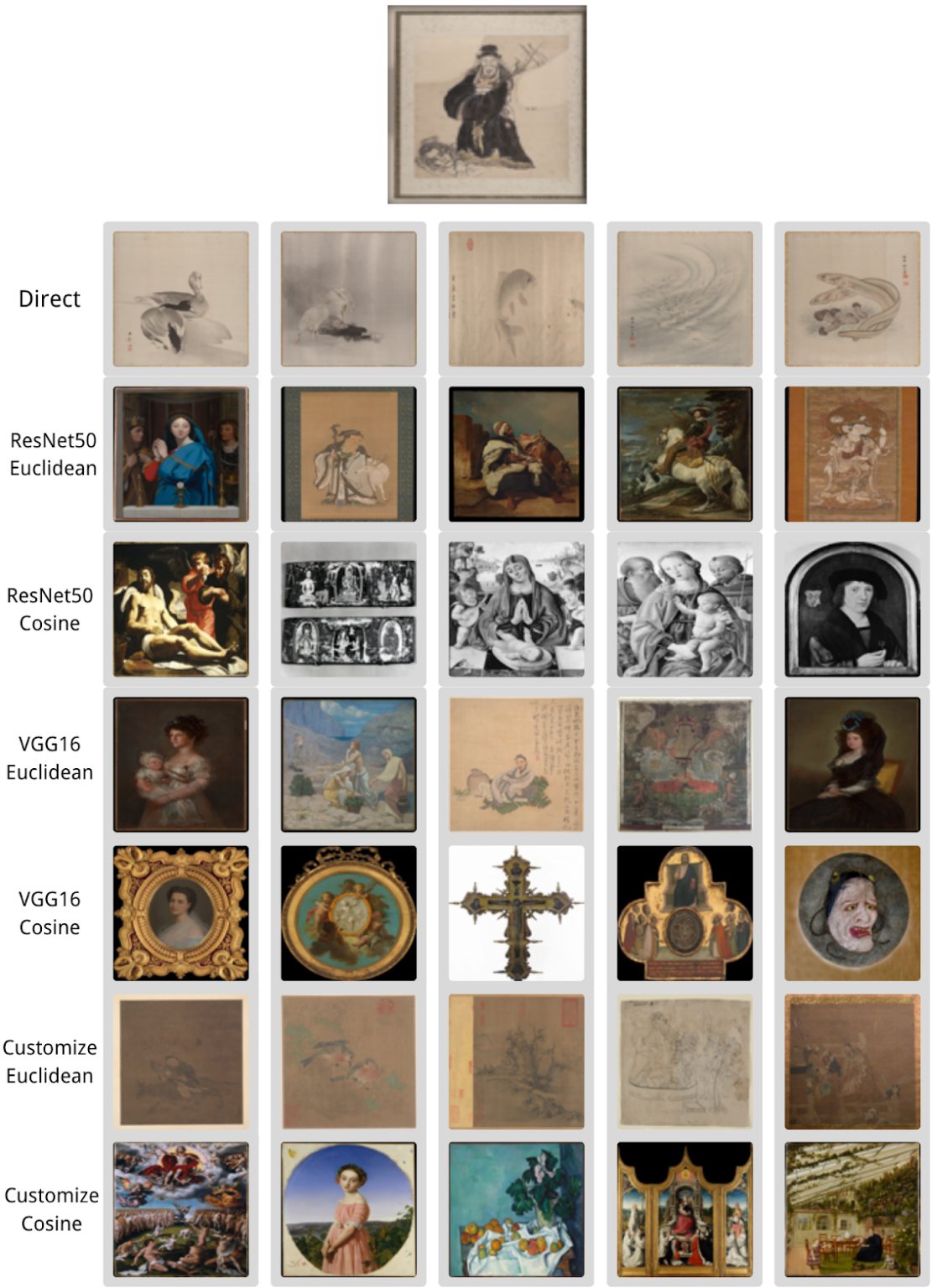

Figure 15. Evaluation results on test image 40011

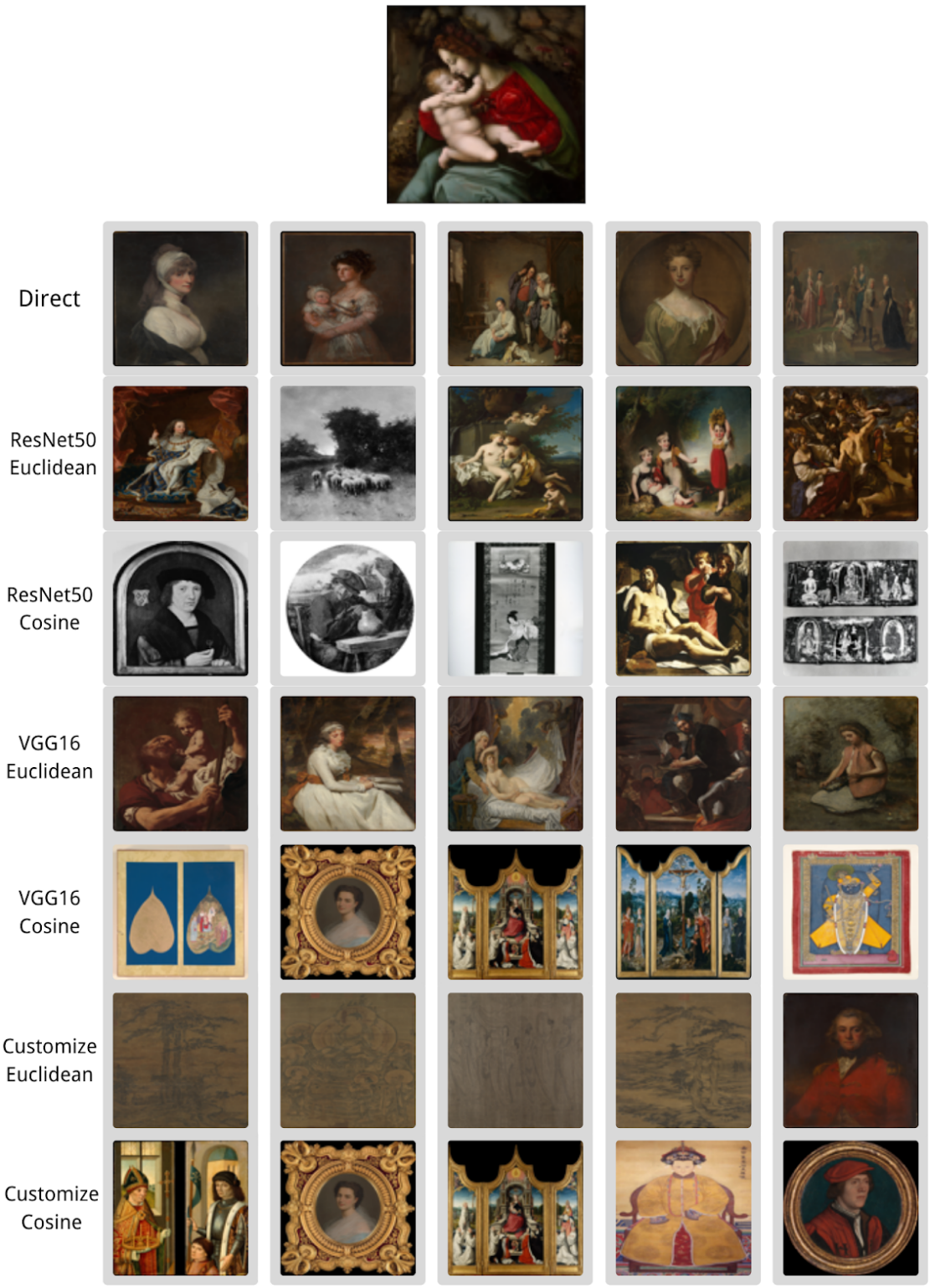

Figure 16. Evaluation results on test image 435595

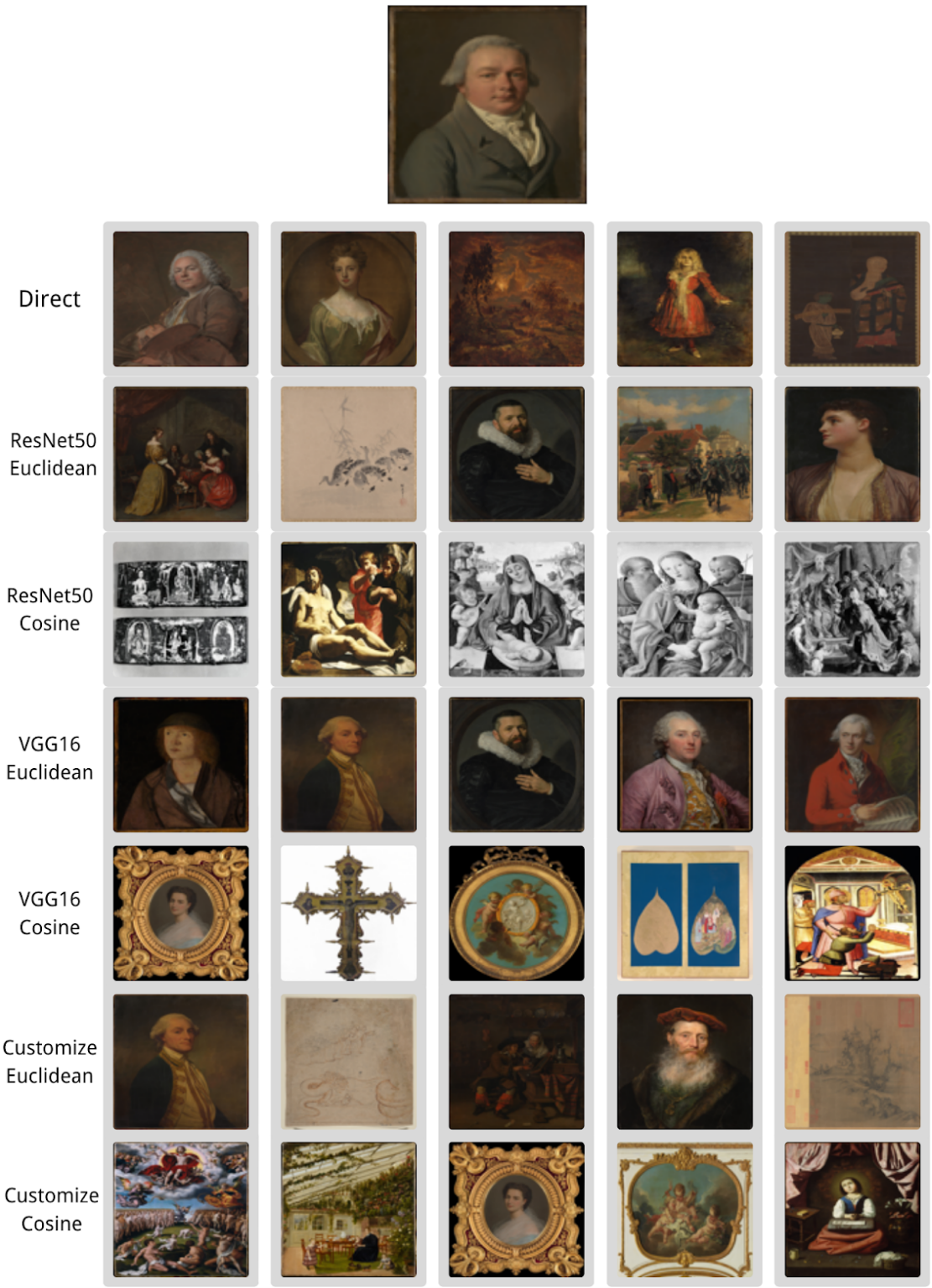

Figure 17. Evaluation results on test image 435687