Towards the Generation of Future Artwork: The Time Predictor

Jingyi Yang
New York University
jy4057@stern.nyu.edu

Chenyang Xu
New York University
cx2219@nyu.edu

Zhuohao Li
New York University
krisli@nyu.edu

Abstract

The characteristics of artworks have evolved over generations, reflecting changes in artistic styles and techniques. This paper explores the underlying relationship between artwork images and their creation dates. We employ Principal Component Analysis (PCA) and CLIP features and design both regression and classification models to predict the creation year of artworks. The models were evaluated against a ChatGPT baseline, achieving comparable performance. Our feature analysis reveals that CLIP, as a pre-trained multi-modal model, provides more informative features related to the artwork's creation date, enhancing prediction accuracy than PCA. The proposed time predictor opens up possibilities for developing a generative model conditioned on creation date, which could predict future artwork styles.

1 Introduction

This paper examined the relationship between time and artworks, hypothesizing that famous artworks from a similar period should encode information about cultural environments and a general understanding of the arts. Therefore, predicting the date of artworks in a rough manner should be a valid task based on the hypothesis above. Specifically, this paper investigated 1) What is the best type of Task for Artwork's chronology? Classification, Regression, or Generation? and 2) Which representation of artworks contains richer chronological information? CLIPS embeddings or PCA embeddings?

Our main contributions are summarized as the following:

1. We demonstrated that artworks in similar periods have common characteristics and a predictor of such could reach 85% accuracy with 20 years of tolerance.
2. We showed that predicting year information of artworks by combining simple traditional machine learning techniques and pre-trained image embeddings could achieve a similar performance with costly Large Language Models (LLM)

2 Related Work

There has been many fruitful research done within the domain of visual arts[1]. Tasks involves classification problems, such as predicting artists[2], genre[3] and style[4], and generation tasks, such as painting image augmentation[5], text-to-image tasks[6]. However, few researchers navigate to the underlying relationship between artworks and chronology, which is the direction of our research.

CLIP is a popular feature extractor that is trained on the image text pair to ensure the similarity of text and image embedding. We explore the possibility of utilizing the multi-modality model to help look into the latent features of images that help improve the prediction task. [7]

3 Method

3.1 Time Prediction Models

To build a predictor for the creation date of artworks, three kinds of models could be considered: 1) Regression Model 2) Classification Model 3) Generative Model. This section will discuss intuitions and the formulation methods for each task in detail. We train our models to complete the regression task and classification task and compare them to the baseline results from the generative model, ChatGPT.

Regression Model: Chronological information such as years is a continuous numeric value by default. We employed LightGBM[1] to predict the year of artwork.

Classification Model: To address the slow evolution in artistic characteristics across individual years, we grouped the artworks by generation. Specifically, paintings are categorized into 20-year intervals. For example, artworks created between 1800 and 1819 fall under the '1800s' class, while those from 1820 to 1839 are categorized as the '1820s' class. For the classification tasks, we utilized XGBoost[2], LightGBM, and MultiLayer Perceptron (MLP) Classifier models.

Generation Model: OpenAI recently developed a vision module for ChatGPT. We selected it as the large language model to evaluate the generation task, which saves us time for fine-tuning. Since it is widely used and generates promising results, we chose it as the baseline method. After inputting the image and prompt, an output year text will be generated. The year text can be regarded as an int to compare with the regression result or floored into a generation to compare with the classification result.

3.2 Feature Extraction, CLIP vs PCA

We choose two methods, 1) **Principal component analysis (PCA)**. and 2) **Contrastive Language-Image Pre-Training (CLIP)**, to extract features from images. And the embedding extracted will be fed forward to the models for regression and classification tasks. For the generation task, the inputs are raw images and prompts.

We explored the PCA embedding first, it is a way to reduce the dimensionality of the raw images of large size and keep only the most important image characters. The dataset WikiArt provides 512 dimensions of PCA embedding for each image. Since the PCA features only contain the raw image information, they might not be informative enough for our task and we also explore CLIP embedding.

CLIP uses a ViT-like transformer to get visual features and a causal language model to get text features. The dot product between the projected image and text features is then used as a similar score.[8] We propose that CLIP with its strong prior knowledge of multi-modality information learned from a large dataset can significantly boost our task.

The hugging face CLIP vision model generates a [CLS] token to serve as a representation of an entire image. It also provides the image features with projection in the space of text features. Due to the limitations of computability and time, we chose to try only the features with projection as we believe it will contain text information more directly e.g. year data.

Analysis was done on these two embedding forms. Correlation between each feature and year data was plotted and we found that CLIP embedding contains much more features strongly related to the year date. CLIP embedding also generally outperforms the PCA embedding. Please check the experiments for details.

4 Dataset

We used the publicly available dataset - WikiArt [3], which contains paintings ranging from 1059 to 2013. This dataset includes painting images and meta attributes, such as the name of the painting,

[1] `https://lightgbm.readthedocs.io/en/stable/`

[2] `https://xgboost.readthedocs.io/en/stable/`

[3] `https://huggingface.co/datasets/Artificio/WikiArt`

artist names, painting styles, and painting topics. Table 1 shows the attributes. There are 76706 samples in the dataset, including 57529 training samples and 19177 testing samples.

Table 1: Attributes of the WikiArt Dataset

Attribute	**Description**
Title	`string`: The title of the painting.
Artist	`string`: The name of the artist. Note that two artists have special characters: "Marevna (Marie Vorobieff)" and "Petro Kholodny (Elder)".
Date	`string`: The published date of the painting in the format of year (e.g., 1950).
Genre	`string (categorical)`: The type of content for the painting. There are 42 unique categories, such as Portrait (13027), Landscape (11453), Calligraphy (12), Shan Shui (10), etc.
Style	`string (categorical)`: The artistic style of the painting. There are 133 unique categories, including Impressionism (8271), Realism (8167), Kinetic Art (2), Renaissance (1), etc. Note that some large categories could be merged, but they were kept separate as style variations are not the primary focus.
Description	`string`: A concatenated string in the format "artist/title/style/genre/date".
Filename	`string`: The filename of the image, ending with `.jpg`.
Image	`pixel data`: The actual image data of the painting.
Embeddings_PCA512	`sequence of floats`: A list of 512 float values representing the PCA-reduced embeddings of the image.

4.1 Preprocessing

The WikiArt dataset, though rich, is not directly suitable for our project's purpose due to the missing attributes of several samples. Therefore, a cleaning and reconstructing of the dataset is required. Specifically, we reformatted the value in date and artists, removing samples with missing values. Sample values in 'date' attributes represent the year of artworks expressed in different formats such as 'c.1986', '1904.0', 'September 1908', and '1921'. Therefore, we wrote scripts to reformat all values into four-digit years, like '1921'. Since the WikiArt dataset originally is scripted from the WikiArt.org website, there are mismatches when capturing meta-data from HTML. For example, the artist's name may contain the artwork's name instead of the artist's name. The same artists may appear in two similar but not the same names. Therefore, we removed the artworks that contained the wrong information and converted two different names of the same artist into a unified name with Python Code.

Moreover, we sampled the artworks by removing artworks that were from earlier than 1800 or later than 2000 because the samples in these time intervals are fewer compared with artworks from 1800 to 2000. The left histogram of figure 1 shows the original distribution of artworks to years after cleaning. Based on the distribution, we decided to implement cutoffs at 1800 and 2000. The final distribution of years in the dataset is shown in the right histogram of figure 1. Although the data is still not uniformly distributed, there isn't any outlier and the the sample size in each category is enough for training.

Lastly, we split the dataset into 25% testing and 75% training sets to compare the performance across all three tasks. For each task, we trained and validated our model on the training dataset and applied the final model on the testing dataset.

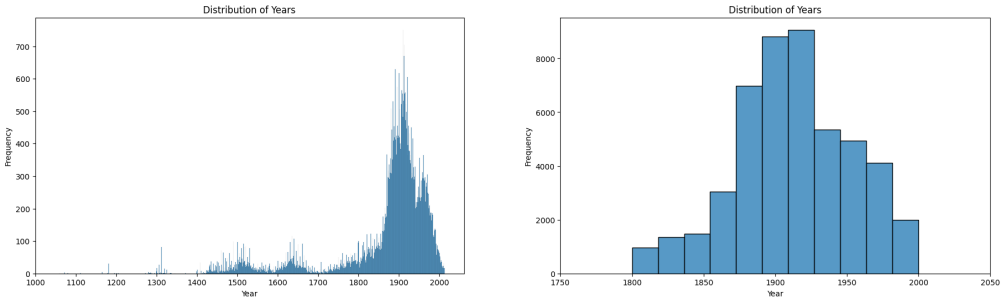

Figure 1: Sample Distribution

5 Experiments

5.1 Feature Analysis, CLIP vs PCA

We introduce CLIP embedding other than PCA embedding because we believe that CLIP embedding contains multi-modality information obtained from a big data set which can assist our task as prior knowledge. To evaluate this hypothesis and gain a further understanding, we checked the correlation between each feature and the year data for both CLIP and PCA and plot the correlation in descending order. As shown in figure 2, there are only 2 PCA features with an absolute correlation larger than 0.3 while there are 74 CLIP features with an absolute correlation larger than 0.3.

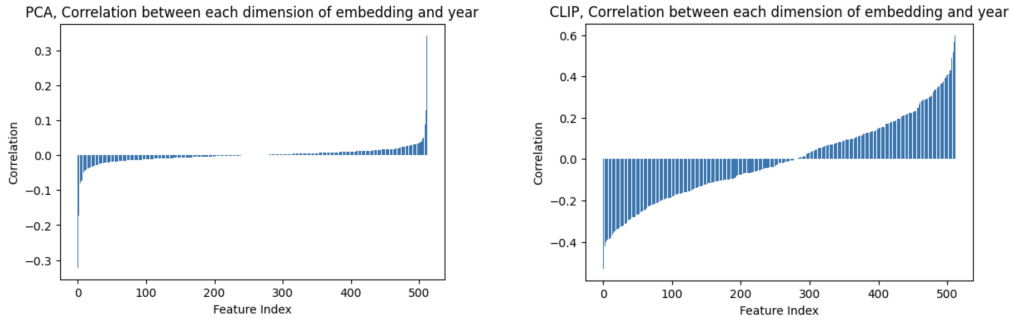

Figure 2: The correlation between each feature and the year data for both CLIP and PCA, the correlation sorted in descending order

5.2 Implementation Details

Regression Model: After splitting testing and training sets, the LightGBM datasets were created. The inputs of the model are image embeddings, generated by the method described in section 5.1. The outputs are year integer numbers. The hyperparameters for regression LightGBM were optimized using a grid search. A 5-fold evaluations of the training data set were implemented to train the optimized model and the testing data were used to evaluate it. The following parameters are the best results explored for CLIP embedding: number of leaves: 15; maximum depth: 6; number of leaves: 15; learning rate: 0.001. The model was trained for 10,000 epochs. All of the other hyperparameters are by default in scikit-learn 1.5.2. As CLIP embedding generally outperforms PCA embedding, we did not do a grid search for PCA embedding to save time and computability.

Classification Model: The dataset is split into testing and training sets and the image CLIP and PCA embedding as inputs are generated by the method described in section 5.1. We implemented only 2 fold evaluations on training data to train the optimized model as the tasks are computationally heavy. The testing data were used to evaluate it. The outputs are floored into generations with intervals of 20 years. The classes are in the form of an integer class index for the LightGBM classifier and XGBoost classifier or in the form of a one-hot code for the MLP classifier. The hyperparameters for LightGBM, XGBoost, and MLP classifier were all optimized using grid searches, which can be checked on the

table 2. The scoring standard is 'roc_auc_ovr', All of the other hyperparameters are by default in scikit-learn 1.5.2.

Table 2: Optimized hyperparameters, classification models

(a) Optimized hyperparameters for XGBoost

	max depth	min child weight	subsample
PCA	3	10	1
CLIP	10	10	1

(b) Optimized hyperparameters for LightGBM

	max depth	min child weight	num leaves
PCA	6	50	100
CLIP	no limit	50	100

(c) Optimized hyperparameters for MLClassifier

	N.Nodes on layer 1	N.Nodes on layer 2
PCA	4	8
CLIP	16	8

Generation Model: We implemented 'gpt-4o-mini' as the LLM on the Generation Task. The few-shot learning technique is used to maintain the output format of the model. We prompted the Openai model with 'Which year was the painting created? Please take the painting content, skill, style, cultural background, and other potential factors into account. Please only return the specific year.' and five examples from the training dataset with the years spread between 1800 and 2000 so that the generation of the open model will not be biased. The return format of the open model maintains only the exact year of the artwork, such as '1921'. We have also tried fine-tuning 'gpt-4o' because 'gpt-4o-mini' doesn't allow fine-tuning with images. However, we failed to get a valid evaluation of the fine-tuned data due to the server crash recently.

5.3 Evaluation Metrics

Regression Task: The model was trained to minimize the validation root mean squared error (RMSE), while also logging the mean absolute error (MAE). These metrics quantify the approximate deviation, in years, between the model's predictions and the actual values. To account for the slow evolution of art styles, we additionally assessed accuracy by measuring whether the absolute difference between the predictions and ground truths fell within 20-year and 30-year intervals. This approach resembles classification accuracy, making it more convenient for comparison with the results of classification tasks.

Classification Task:

- ACC (accuracy) is simply the percentage of correct predictions for a multi-class prediction task. It is a good metric for a uniformly distributed data set.

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}}$$

- As discussed in the section 4.1, the data set is not uniformly distributed, therefore, we introduce AUC to handle imbalanced datasets. For the multi-class classification, the AUC for each class can be obtained by calculating the True Positive Rate (TPR) and False Positive Rate (FPR):
  TPR (True Positive Rate) = $\frac{\text{True Positives}}{\text{True Positives}+\text{False Negatives}}$
  FPR (False Positive Rate) = $\frac{\text{False Positives}}{\text{False Positives}+\text{True Negatives}}$
  The ROC curve (Receiver Operating Characteristic curve) plots the true positive rate (TPR) against the false positive rate (FPR) at various thresholds. It is then plotted for each class, and the AUC for class i is the area under this curve. Following a one-vs-all manner, Multi-Class

AUC Formula (Weighted Averaged)

$$\text{AUC}_{\text{weighted}} = \frac{1}{N} \sum_{i=1}^{C} N_i \cdot \text{AUC}_i$$

Where: N is the total number of samples. N_i is the number of samples in class i. AUC_i is the AUC for class i.
For a non-random classifier, the AUC should be greater than 0.5 and less than 1.

Calibrated ACC: We propose that the shifting of artworks' styles, genres, and other image-related features is not as rapid as per year, it also does not change within a fixed interval, e.g. 20 years. Therefore, we introduce calibrated ACC, ACC with decades of years as tolerance.
We propose that a correct prediction calibrated should be when the difference between prediction and ground truth is less than the tolerance interval. And the calibrated ACC is defined as:

$$\text{Calibrated ACC} = \frac{\text{Number of Correct Predictions calibrated}}{\text{Total Number of Predictions}}$$

Please notice that this is a metric that can be fit into both regression and classification predictions, however, the same length of interval years does not mean the same tolerance for these two tasks. For example, a prediction of 1920 and a ground truth of 1945 with 20-year tolerance will be regarded as truth for the classification task as the ground truth will be floored to 1940, while a prediction of 1920 and a ground truth of 1945 will be regarded as false for regression. In conclusion, intervals of the same length always show a better tolerance for regression tasks. Calibrated ACC is only used to compare different models under the same task.

Correlation: Correlation is also a metric that can be implemented for both classification and regression. It can reflect on the similarity of trends between ground truth and prediction, however, ignores the local difference. It was used as an additional metric to help gain more insights.

5.4 Prediction Models Evaluation Results

The model results are shown in table 3. A visualization example of the LightGBM regression model and ChatGPT baseline can be checked in figure 3.

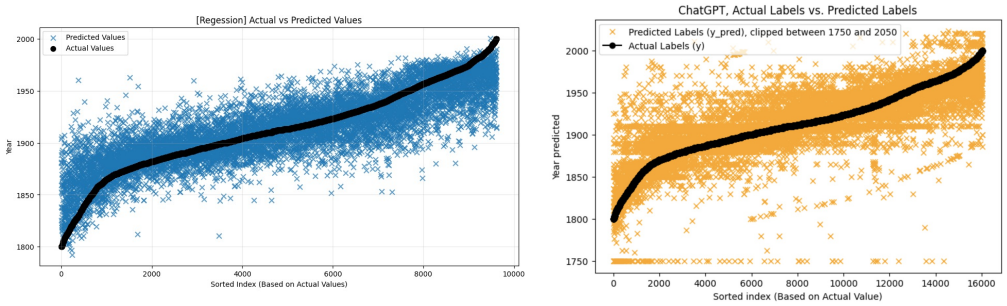

Figure 3: A visualization of LightGBM regression model(left) and ChatGPT baseline(right).

6 Discussion

6.1 What is the most suitable task type for determining the chronology of artworks?

These results suggest that generation tasks, leveraging large language models (LLMs), tend to outperform other approaches in determining the chronology of artworks. However, this performance advantage comes at a higher computational cost, making LLMs less practical in scenarios where resource efficiency is a priority. In contrast, both XGBoost classification and LightGBM regression demonstrate comparable performance to LLM-based generation tasks when a certain margin of error is acceptable. XGBoost is particularly effective for predicting discrete time categories, whereas LightGBM is better suited for continuous year estimations. These methods offer efficient and practical alternatives, balancing accuracy and computational demands, and are well-suited for applications with economic or computational constraints.

Table 3: Model Results Across Tasks

(a) Regression Results for LightGBM

	LightGBM	
Embedding	RMSE	MAE
PCA	30.5271	23.3303
CLIP	22.2618	16.6982

(b) Classification Results for Different Models

	XGBoost		LightGBM		MLP Classifier	
Embedding	ACC	AUC	ACC	AUC	ACC	AUC
PCA	0.3361	0.7593	0.3506	0.7578	0.3361	0.7593
CLIP	0.4845	0.8788	0.4848	0.8642	0.4795	0.8655

(c) Classification vs. ChatGPT

	XGBoost	ChatGPT
ACC	0.4845	0.4718
Correlation	0.8005	0.8197
Calibrated ACC (20 years)	0.8362	0.8518

(d) Regression vs. ChatGPT

	LightGBM	ChatGPT
RMSE	22.2618	27.16
Correlation	0.8376	0.8326
Calibrated ACC (20 years)	0.6873	0.7459
Calibrated ACC (30 years)	0.8429	0.8650

6.2 How unbalanced dataset affect the prediction model?

Even we have filtered the dataset to the range between 1800 and 2000, the distribution is still unbalanced. There are lack of samples on tails, which can be seen in the figure 1. We can also observe that our regression model has a larger bias than the ChatGPT result in figure 3, which might be caused by the lack of training data on these two sides. These issues might be tackled by adding more samples at these ages or exploring the training method for the unbalanced data set. However, it might be also due to the high variance of image features in these generations. This is still an under-developed research field.

6.3 Which representation of artworks encodes the most comprehensive chronological information?

The results show that CLIP embeddings capture more temporal information than PCA due to their generation methods. While PCA reduces an image to its structural features, CLIP parses deeper semantic meanings, aligning visual and textual data. This process likely incorporates time-specific cultural and contextual nuances from the training data. CLIP's linking of visual elements to textual semantics enhances its ability to encode historical and stylistic features, resulting in richer temporal embeddings. This underscores the value of models integrating visual and semantic contexts for extracting temporal information from artworks.

6.4 Why does ChatGPT generate promising results?

ChatGPT provides promising results. Although it is a black box process, we can propose our hypothesis of its superiority in the aspects of multi-modality information in embeddings. Text embeddings encode semantic meanings into vector representations, inspiring the idea that embeddings of years as text could capture cultural or implicit semantic information embedded in training data. For example, an embedding of the year '2050' trained with science-fiction novels could indicate future elements, such as a spaceship and high technology. By considering year information as text,

our generation task aims at matching the embeddings of paintings to the embeddings of 'year', which could be possible through common culture information encoded in the embeddings.

7 Conclusion

This study investigated various methods for predicting the chronology of artworks, providing insights into the trade-offs between performance, computational efficiency, and resource requirements.

The findings demonstrated that generation tasks using large language models (LLMs) result in better accuracy in determining the chronology of artworks. However, their higher computational cost is not scalable for all tasks. Practical models such as XGBoost classification and LightGBM regression offer competitive performance while using a fraction of the resources needed for LLMs, making them suitable for applications where slight margins of error are acceptable.

Further analysis discovered the advantage of using CLIP embeddings over PCA, with CLIP embedding's semantic integration of visual and textual contexts enhancing its ability to encode temporal information. This result underscores the value of embedding models that capture deeper cultural and stylistic elements.

Future work could explore the development of generative models that utilize similar frameworks as our classifiers to generate artworks reflecting future trends. By integrating these classifiers as discriminators or guidance systems within generative adversarial networks (GANs) or other generative frameworks, it may be possible to produce art that not only reflects the cultural and stylistic elements of its time but also predicts and explores future artistic movements.

References

[1] Author(s). Recognizing the style, genre, and emotion of a work of art through visual and knowledge graph embeddings. In A. Editor and B. Editor, editors, *Proceedings of the International Conference on AI in Art and Design*. Springer, 2023.

[2] Roberto Leotta, Oliver Giudice, Luca Guarnera, and Sebastiano Battiato. Not with my name! inferring artists' names of input strings employed by diffusion models, 2023.

[3] Author(s). Multilabel genre prediction using deep-learning frameworks. *Journal Name*, 13(15):8665, 2024.

[4] Adrian Lecoutre, Benjamin Negrevergne, and Florian Yger. Recognizing art style automatically in painting with deep learning. In Min-Ling Zhang and Yung-Kyun Noh, editors, *Proceedings of the Ninth Asian Conference on Machine Learning*, volume 77 of *Proceedings of Machine Learning Research*, Yonsei University, Seoul, Republic of Korea, 15–17 Nov 2017. PMLR.

[5] Brandon Trabucco, Kyle Doherty, Max Gurinas, and Ruslan Salakhutdinov. Effective data augmentation with diffusion models, 2023.

[6] Qinghe Tian and Jean-Claude Franchitti. Text to artistic image generation, 2022.

[7] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021.

[8] Hugging Face. Clip model documentation, 2024. Accessed: 2024-12-13.