DS-UA 301 AI Calligraphic Poet Project Proposal

Kevin Zhang, Millie Chen, Yuanheng Li

hz3223@nyu.edu, mc9362@nyu.edu, yl10337@nyu.edu

I. Motivation

Our project explores the integration of **image captioning, poetry generation, and calligraphy creation** using machine learning techniques. Specifically, we aim to address the question of **how deep learning can be applied to create artistic works** by extracting meaningful content from images, generating poetry based on these captions, and finally, producing calligraphy in Chinese. This project stands out because it goes beyond simply replicating existing works by **combining diverse forms of artistic expression—photography, poetry, and calligraphy—into a unified, innovative application**. Our motivation stems from our personal interests in these arts and the excitement of using machine learning in novel, creative ways to produce new forms of artistic expression. We provide a flowchart as figure 1 1.

II. Related Work

For image captioning, one popular model is the **Neural Image Caption (NIC)** created by engineers at Google. It uses a encoder-decoder approach where a convolutions neural network (CNN) encodes the given image into a vector and then uses a long-short-term-memory model (LSTM), which is a type of Recurrent Neural Network, to decode the vector and generate image captions([10]). Later works introduced an **attention based mechanism** where weights were given to different parts of the vector output from the CNN to improve the quality of the captions [16]. This attention based mechanism also enabled us to learn more about how the model was able to generate image captions as we can visualize what parts of the image was "given more attention". More recently, the **GIT (Generative Image-to-text Transformer) model** was developed with a simpler architecture and achieved better results (and can also be used for other purposes in addition to image captioning) [11]. It uses a Transformer architecture, which has a similar encoder-decoder framework but it does not use CNNs and RNNs in its encoder/decoder. Instead, it uses a **CLIP (Contrastive Language-Image Pre-Training) encoder** that is pre-trained to extract the features from the image and the decoder uses a self-attention method to help generate image captions [11]. **However, even though these models do well on generating descriptive and accurate image captions, the models were only limited to English image captions**. For our research, we aim to produce Chinese image captions so improvements and fine-tuning will have to be made to these existing models as we will be working with a different language.

For poem generation, the main challenges to be tackled are maintaining the special format and characteristics of the traditional Chinese poetry, while guaranteeing the integrity and truthfulness of the text. We found some previous works that address these problems, one of such frameworks is the **SongNet**. The backbone of this framework is a Transformer-based auto-regressive language model([7]). Another finding that would be helpful is **Generation with Planning based Neural Network**, it proposes a novel two-stage poetry generating method which first plans the subtopics of the poem according to the user's writing intent, and then generates each line of the poem sequentially, using a modified recurrent neural network encoder-decoder framework. The proposed planning based method can ensure that the generated poem is coherent and semantically consistent with the user's intent [13]. These models have implemented **constraint decoding** as a type decoding technique, and **Transformers** as the deep learning framework. These are the strengths of the models. Constraint decoding allows for precise control over certain elements of the generated text, such as rhyme schemes, meter, or thematic relevance. This is especially useful in structured text forms like poetry, where specific rules are critical. It also improves relevance, when generating text related to an input theme (e.g., turning a paragraph into poetry), constraints can ensure that the generated output stays focused on the input's central topic, which is essential when guaranteeing the integrity and truthfulness of the poem. The transformer architecture excel at capturing long-

range dependencies and context within text. This makes them well-suited for generating coherent and contextually aware poetry, where words and ideas need to be connected over multiple lines or stanzas. However, there do exist some weaknesses for these models. One major weakness is the **lack of proper evaluation metrics**. There is not a mature metric that measures how well the poem is written, and sometimes, human evaluation is needed, which could be subjective and leading to unsatisfactory results.

For calligraphy generation, most of the works we found mainly based on **GAN** or **Diffusion**. The first trial to generate Chinese characters is **zi2zi** [9], which is based on the first image to image translation model **pix2pix** [6]. It is basically a conditional adversarial network that generates styled calligraphy by encoding both character and category embedding, decoding them into output, and optimizing various loss functions through a discriminator to ensure accuracy and style consistency. But it only allows **single-style transferring** and it requires **too much paired training data**. To improve the quality of generation and to allow multi-style translation, researchers then come out with **CycleGAN** [2] approach and new **CalliGAN** [15], both of them are with some improvements. Using the cycleGAN approach allow **unpaired training data** which solve the problem that not all the Chinese characters have a picture of certain styles [2]. The CalliGAN use a different architecture that **concatenate Chinese characters' component features** and the style one-hot vector into the image feature maps which allows **multi-style transferring** [15]. And finally we have **Callifussion** model [8]. It's the first model that use **DDPM** to do Chinese calligraphy generation. This is a **SOTA** approach that gives us the **most high-quality result**. The models' architectures are shown in figure 2 2.

III. Methodology

For the **image captioning** portion of our project, we have decided to use the **GIT model**. This is the most appropriate model to apply as it is the most recent model and it has a **simple architecture while achieving good results**. A transformer model is used and since we will learn more about transformer models later in the course, we plan to use this knowledge to help tune the hyperparameters of the model and make other adjustments so it is better fitted for generating accurate image captions in Chinese. We plan to use **BLEU Score (Bilingual Evaluation Understudy)**, **METEOR (Metric for Evaluation of Translation with Explicit ORdering)**, and **CIDEr (Consensus-based Image Description Evaluation)** since these are common metrics used for image captioning and text generating models.

For **poems generation**, we are going to use a **Recurrent Neural Network based model** as RNN is the most common and effective model for Natural Language Processing Tasks. A **transformers architecture** will also be implemented to capture long term dependencies so the best performance sequence could be captured. Decoding techniques such as **constraint decoding** will be used to ensure the format and constraints of the output text. Evaluation of the poems are a difficulty as it is hard to measure how well a poem. We have decided to use **a mix of machine evaluated metrics and human evaluated metrics**. For machine evaluated metrics, we are going use **Format, Rhyme, Integrity**. For human evaluated metrics, we are going to focus on 4 aspects that best evaluate how well a poem is written. They are: **Fluency, Coherence, Meaningfulness, Poeticness**. We think it is also interesting to investigate how **Large Language Models** such as Chatgpt evaluate poems, which are considered heavily human correlated. Thus, we would use Chatgpt to evaluate the poems, based on Fluency, Coherence, Meaningfulness, Poeticness and compare with human evaluation results.

For **calligraphy generation**, we will use the **Chinese Calligraphy Styles by Calligraphers dataset** [12] to train our **GAN-based** and **Diffusion-based model** with techniques learn in the course, for example **hyperparameter tuning** and **distributed training**. GAN and Diffusion are appropriate as they are mainstreamed generation model algorithm. For the evaluation metrics, we will consider both **content accuracy** and **style discrepancy** to evaluate the model. For content accuracy, we use a high accuracy pre-trained character classification model to evaluate our generated works accuracy. For style discrepancy, we used root-mean-square difference between the style representations of the target characters and the generated characters.

IV. Dataset

MS COCO subset: This is a subset of the MS COCO (Microsoft Common Objects in Context) dataset and it includes images of a variety of people, places, and objects. The Chinese captions for the images are created by Cai and his classmates who were enrolled in a Pattern Recognition course at Tsinghua University [1]. There are 503 labeled examples in this dataset that are available. This dataset is appropriate for our research as it is using images from a dataset that is commonly used for training image captioning models. It also includes captions that are in Chinese, so our model

can be trained to output Chinese captions that can be used in the poem generation step. A few examples of the images and corresponding captions are shown in Figure 3 3. Through exploratory data analysis, we found that this dataset only has images of color but the size of the images vary (Figure 4 4).

Poem Datasets: We have a list of THU datasets: Poetry Quality Evaluation dataset [17], Sentiment dataset [3], corpus of traditional Chinese poetry [4], and dataset of rhyme and rhythm[5]. We also have a corpus of traditional Chinese poetry with sufficient amount of data covering poems across all the dynasties [14] 5. These datasets all can guarantee the well-training of the model, also provide good reference of evaluating the model.

Chinese Calligraphy Styles by Calligraphers: This is a dataset of Chinese character writings in the style of 20 famous Chinese calligraphers. There are 1000 - 7000 jpg images in each subset (5251 images on average). Each image has size 64*64 and represents one Chinese character. Dataset is divided into training set (80%) and testing set (20%). The initials of calligraphers are used as labels [12] 6. This dataset is appropriate since it is quite large and contains multiple styles.

V. Work Plan

Currently, our group divided the project into 3 sections that corresponds to the 3 steps of our image to image task: image captioning (Millie), poem generation (Kevin), and calligraphy creation (Yuanheng). We each researched our topics and wrote the respective Related Work, Methodology, and Dataset sections for our topics. We collaborated on the Motivation and Work Plan sections.

Here is the weekly plan:

Week 1:

1. Image Captioning: Clean/format dataset so it is the correct format for the input of the model, do more research on the different aspects of the model (for example, self attention mechanism)
2. Poem Generation: Conduct more background research how the poem is generated, including reading more papers, testing out existing pre-trained models.
3. Calligraphy Generation: More research on Calligraphy Generation, data preparation and previous work reproduction.

Week 2:

1. Image Captioning: Train the model and tune hyperparameters for optimal performance, find more data if needed
2. Poem Generation: Prepare the datasets needed for training the model. Conduct data pre-processing and cleaning. Set up the foundation of the model, make sure the model is ready for training
3. Calligraphy Generation: Train GAN-based models and diffussion-based model, and do hyperparameter tuning.

Week 3:

1. Image Captioning: Continue training the model and tuning hyperparameters for optimal performance
2. Poem Generation: Training the model and tuning the hyperparameters
3. Calligraphy Generation: Do data augmentation and further tune the model to get the best result.

Week 4:

1. Image Captioning: Model evaluation and Milestone 2 write-up
2. Poem Generation: Evaluation of the model, and conduct further improvements
3. Calligraphy Generation: Evaluate models and do the write-up.

Week 5:

Gain insights into others' models to enhance our overall understanding of the project. Make potential improvements for the whole project.

Week 6:

Combine the models into one pipeline (ensure that the inputs and outputs for each step work). Make any necessary adjustments.

Week 7:

Finish the pipeline and connect all three parts of the model. Evaluate the project as a whole.

Week 8:

Final edits and finish Github documentation. Start writing Milestone 3 report.

Week 9:

Finish Milestone 3 report and start preparing for final presentation.

References

[1] Cai-Liwei. Github - cai-lw/image-captioning-chinese: Image captioning in chinese using lstm rnn with attention mechanism. Available at `https://github.com/cai-lw/image-captioning-chinese`, 2018.

[2] Bo Chang, Qiong Zhang, Shenyi Pan, and Lili Meng. Generating handwritten chinese characters using cyclegan. In *IEEE Winter Conference on Applications of Computer Vision*, 2018.

[3] Huimin Chen, Xiaoyuan Yi, Maosong Sun, Cheng Yang, Wenhao Li, and Zhipeng Guo. Sentiment-controllable chinese poetry generation. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, Macao, China, 2019.

[4] Zhipeng Guo, Xiaoyuan Yi, Maosong Sun, Wenhao Li, Cheng Yang, Jiannan Liang, Huimin Chen, Yuhui Zhang, and Ruoyu Li. Jiuge: A human-machine collaborative Chinese classical poetry generation system. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 25–30, Florence, Italy, 2019.

[5] Zhipeng Guo, Xiaoyuan Yi, Maosong Sun, Wenhao Li, Cheng Yang, Jiannan Liang, Huimin Chen, Yuhui Zhang, and Ruoyu Li. Jiuge: A human-machine collaborative Chinese classical poetry generation system. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 25–30, Florence, Italy, 2019.

[6] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[7] Piji Li, Haisong Zhang, Xiaojiang Liu, and Shuming Shi. Rigid formats controlled text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 742–751, Online, July 2020. Association for Computational Linguistics.

[8] Qisheng Liao, Gus Xia, and Zhinuo Wang. Calliffusion: Chinese calligraphy generation and style transfer with diffusion modeling. Available at Mohamed bin Zayed University of Artificial Intelligence and New York University Shanghai, 2024. Emails: qisheng.liao@mbzuai.ac.ae, gus.xia@mbzuai.ac.ae, zw2375@nyu.edu.

[9] Yuchen Tian. zi2zi: Master chinese calligraphy with conditional adversarial networks. `https://kaonashi-tyc.github.io/2017/04/06/zi2zi.html`, 2017.

[10] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: Lessons learned from the 2015 mscoco image captioning challenge. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(4):652–663, April 2017.

[11] Jianfeng Wang, Zhengyuan Yang, Xiaowei Hu, Linjie Li, Kevin Lin, Zhe Gan, Zicheng Liu, Ce Liu, and Lijuan Wang. Git: A generative image-to-text transformer for vision and language. *arXiv preprint arXiv:2205.14100*, 2022.

[12] Yuanhao Wang. Chinese calligraphy styles by calligraphers. `https://www.kaggle.com/datasets/yuanhaowang486/chinese-calligraphy-styles-by-calligraphers/data`, 2023. Accessed: 2023.

[13] Zhe Wang, Wei He, Hua Wu, Haiyang Wu, Wei Li, Haifeng Wang, and Enhong Chen. Chinese poetry generation with planning based neural network. *CoRR*, abs/1610.09889, 2016.

[14] Werneror. Poetry. `https://github.com/Werneror/Poetry`, 2023.

[15] Shan-Jean Wu, Chih-Yuan Yang, and Jane Yung jen Hsu. Calligan: Style and structure-aware chinese calligraphy character generator. Accepted to the AI for Content Creation Workshop at CVPR 2020, 2020. `https://arxiv.org/abs/2005.12500`.

[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, 2016.

[17] Xiaoyuan Yi, Maosong Sun, Ruoyu Li, and Wenhao Li. Automatic poetry generation with mutual reinforcement learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3143–3153, Brussels, Belgium, 2018.

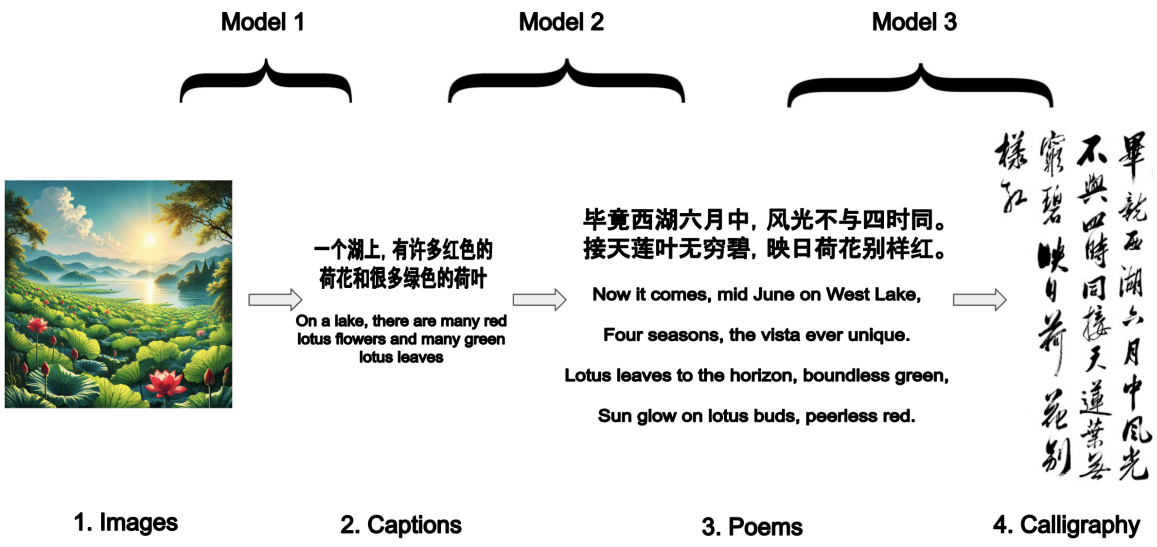
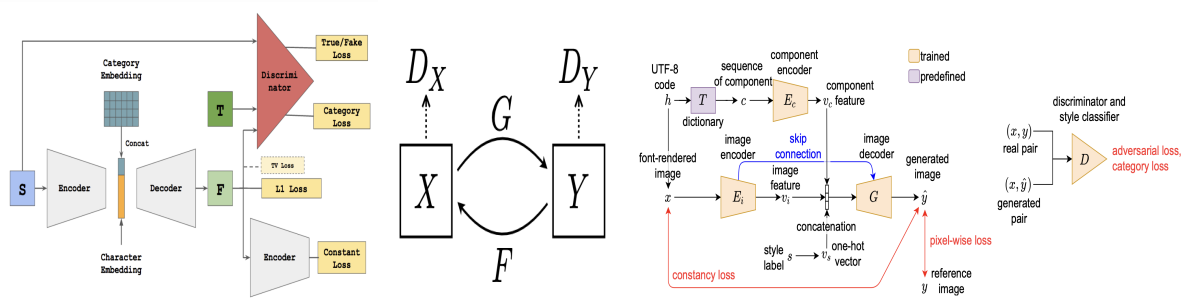


Figure 1: A brief flowchart of the whole project



(a) zi2zi (b) CycleGAN (c) CalliGAN

Figure 2: Calligraphy Generation Architectures

Figure 3: Examples from MS COCO dataset

Figure 4: EDA of MS COCO dataset: image sizes/dimensions

朝代	诗词数	作者数
宋	287114	9446
明	236957	4439
清	90089	8872
唐	49195	2736
元	37375	1209
近现代	28419	790
当代	28219	177
明末清初	17700	176
元末明初	15736	79
清末民国初	15367	99
清末近现代初	12464	48
宋末元初	12058	41
南北朝	4586	434
近现代末当代初	3426	23
魏晋	3020	251
金末元初	3019	17
金	2741	253
民国末当代初	1948	9
隋	1170	84
唐末宋初	1118	44

Figure 5: Visualization of characteristics of traditional Chinese poetry

Figure 6: Examples from Chinese Calligraphy Styles by Calligrapher Liu Gongquan