Master Thesis

Exploring Segmentation Models for Chinese Ancient Landscape Paintings

Xin Lan 6690165

Supervisor: Dr. Almila Akdag

Second Supervisor: Prof. Dr. Arno Siebes

Graduate School of Natural Sciences
Utrecht University
Netherlands
29-07-2021

Abstract

Semantic segmentation is applied to various tasks such as road images, medical images and images that need to separate the main objects from the background. With high performance semantic segmentation models, the computer can actually see the images almost the same as how human can see it. The computer can know what are the objects in the image and where exactly are they. With this level of understanding, a lot of tasks can be turned automatic, therefore, saving the use of manpower. Also, with the context and location information extracted, a lot of hidden features which are hard for human to observe can be learnt by applying deep learning models. For semantic segmentation, the state-of-the-art models achieves high accuracy in tasks where there are large-scale training dataset as support. In this work, we explore a new field for semantic segmentation which is Chinese ancient landscape paintings. Developing a good segmentation model for ancient paintings can convenience and systematize the image retrieval for ancient paintings. Furthermore, it can speed up and upgrade the process of art digitalization. It can also promote new classification systems which offers new perspectives for art work interpretation. The main challenges lies in the following aspects: First, there is no annotated training data for Chinese landscape paintings. Second, the characteristics of Chinese landscape paintings make the task more challenging. Most of the paintings are drawn in black ink, therefore resulting in the lack of color information. Also, having a lot of blank space is a feature of Chinese landscape paintings. Such blank spaces tend to be the sky or water, therefore resulting in the lack of boundary information. Given the challenges above, we first tested the state-of-the-art models which are Unet, DeepLab-V2 and DeepLab-V3 on our manually annotated test set. Overall, DeepLab-V3 achieves the highest segmentation score among all the models. Furthermore, we proposed to improve the performance through text removal, style transfer and adding elastic augmentation to the training procedure. The combination usage of text removal and style transfer promoted the segmentation accuracy for sky and water classes while adding elastic data augmentation benefits the performance for mountain class.

Contents

1 Introduction 4

2 Research Questions 6

2.1 How do the state-of-the-art semantic segmentation algorithms perform on Chinese paintings? 6

2.2 How to improve the performance on Chinese paintings? 6

2.3 Can we find any patterns from the layout of Chinese landscapes? 6

2.4 Semantic image synthesis of paintings 7

2.4.1 How do existing image synthesis networks perform on generating Chinese paintings? 7

2.4.2 How to generate more realistic paintings? 7

3 Literature Study 8

3.1 Semantic Segmentation 8

3.2 Image-to-Image Translation 13

3.3 Computer Vision and Artworks 15

4 Dataset Preparation 17

4.1 Chinese Painting Dataset 17

4.2 Bob Ross Painting Dataset 18

4.3 COCO-stuff Dataset 18

4.4 Expanded Bob Ross Painting Dataset 19

4.5 100 Chinese Painting Test Set 19

4.6 Data Augmentation 19

5 Methodology 20

5.1 DeepLab model 23

5.1.1 DeepLab-V2 23

5.1.2 DeepLab-V3 25

5.2 Unet 26

5.3 Style Transfer: Cycle GAN 26

5.4 Text Detection 29

5.5 Image Inpainting 30

5.6 Evaluation 30

6 Results **33**

6.1 The state-of-the-art semantic segmentation algorithms performance 33

6.1.1 DeepLab 33

6.1.2 Using expanded Bob-Ross paintings as training data 34

6.2 How to improve the performance on Chinese paintings 36

6.2.1 Style Transfer 36

6.2.2 Text Removal 39

6.2.3 Elastic Data Augmentation 41

6.3 Statistical analysis for Chinese landscapes 42

6.3.1 Composition Proportion 42

6.3.2 Object location 42

7 Discussion **45**

8 Conclusion **48**

1 Introduction

With the development of computer vision technology, there are a lot of efficient models and algorithms developed for various tasks in image processing such as classification, object recognition, image segmentation, etc. With the deepening of these studies, various datasets have also been enriched. However, most of these researches are focused on real scene images, photographs, and videos. On the one hand, real images have more predictable patterns and they are easier to obtain, and doing research on real images can lead to more practical applications. For example, an efficient algorithm for object recognition or semantic segmentation for scene understanding on road images can benefit the development of self-driving vehicles. On the other hand, the lack of annotated training data in art work field makes it hard for researches to proceed.

When it comes to art objects like paintings, a lot of tasks are not well researched. Simply applying models that are designed for real images to paintings for common tasks like object recognition or segmentation suffers from poor performance [23]. Also, the lack of available well-labeled datasets prevents researchers from performing large-scale experiments.

The idea of this research project is to expand semantic segmentation algorithms for Chinese landscape paintings and to generate a dataset of Chinese Landscape objects such as mountains and rivers etc, which can be further used for an application like an AI-photoshop brush where you can change objects in a painting and the surroundings will adapt to your changes. By providing an algorithm for semantic segmentation problem on paintings, it contributes to the image retrieval field in terms of paintings. For example, you can upload a painting and through image segmentation, you can get search results with similar objects. It can also be used in art museums in order to help users locate and browse more artworks of the kinds they are interested in instead of having to follow the fixed flow. For example, if they are interested in mountains, by searching through the keyword 'mountain', with the help of semantic segmentation, they can browse different paintings with mountains in them and further compare different styles of mountain drawings from different artists and times. This research can also enrich the annotated data set of Chinese paintings and benefit humanities scholars.

The challenges for the project are numerous. To apply a pre-trained vision algorithm (like a deep network trained for object recognition) is not a straightforward task if applied on image-datasets generated from another era, as the historical dataset diverges from the training set considerably. Moreover, not only do the objects need to be recognized, but the contours need to be correctly found in order to generate a reliable dataset for the next stage. And lastly, even the AI-brush toolkit is already developed for painting from a modern image dataset, and works well on a semantic level (for example the tool does not let the user add a waterfall on top of a cloud), the semantic output of a Chinese landscape is very different from modern landscapes, and a new semantic network needs to be developed for this project.

Like the challenges, expected outcomes of the project are also numerous: The first contribution of the project is to generate a rich image database of objects found in the Chinese landscape paintings. The second contribution is to develop algorithms both for object recognition in Asian paintings as well as developing a semantic network for these objects adding to the literature in multimedia retrieval. We also expect to find certain patterns in Chinese landscape paintings that can be unearthed only by analyzing a big data set adding to the art historical research. We may also find strengths or drawbacks of certain algorithms and networks and ways to improve them. If time permits, we will extend the function to performing semantic manipulation on an input painting so that we can modify a certain part of the painting while still keeping the rationality.

This thesis is structured in the following way: First, we will introduce our research questions in Section 2, followed by the literature study in related field in Section 3. Then, we will present the datasets to be used in our research in Section 4. After that, we will explain the models and methods we used in detail in Section 5. Then, the results are shown and discussed in Section 6 and Section

7. In the end, we present some ideas and concepts for future research and the conclusion in Section 8.

2 Research Questions

In this section we specify our main research question, and the sub-questions which need to be answered in order to answer the main question and to reach our goal.

2.1 How do the state-of-the-art semantic segmentation algorithms perform on Chinese paintings?

Semantic segmentation is a task where we need to classify every pixel in the image to its corresponding class label. The main objects that would appear in Chinese landscape paintings are mountains, trees, water, etc. The main challenges for the semantic segmentation on Chinese paintings lie in two aspects: the absence of paired training data and the lack of color and texture information in the paintings. Because no similar research has been done before, there is no baseline performance for this task. Therefore, it is important to first test the state-of-the-art models so that we know where the models fail and where they perform well in order to propose further approaches for improvement accordingly.

We will first try some state-of-the-art semantic segmentation models on Chinese paintings and see how good the results are. The introduction of some existing models is shown in Section 3.1. Considering the computational resource, we consider trying the two models that are more efficient than others. One is ENet [22] and the other one is DeepLabV3 [7].

We will use the COCO-stuff dataset [4] as the training data in the experiments. It is a dataset that provides paired data of images and the corresponding semantic masks. The word 'stuff' refers to amorphous background regions like grass, sky, water, etc. It is built in order to include these objects that do not have a well-defined shape. This makes this dataset a good start for our research because in landscapes, we will need to recognize objects like water, sky, etc.

To be consistent with other works, we will use an often used metric called Intersection-Over-Union (IoU). The IoU for one object is the ratio of the intersection area of the ground truth and the predicted segmentation and the union of these two areas. It is shown in the formula below:

$$IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}} \tag{1}$$

For an image with multiple objects, the IoU of the image is the average of the IoU values for all the object classes.

2.2 How to improve the performance on Chinese paintings?

The existing models are all trained on realistic images, so the performance is probably not satisfying. We will use style transfer to convert the Chinese paintings to realistic images and assess how the models perform on the converted images. We will use the results obtained by directly applying algorithms on Chinese paintings as a baseline. The metric mentioned in the last research question can also be applied here.

2.3 Can we find any patterns from the layout of Chinese landscapes?

After getting a set of semantic masks over the paintings, we will further investigate whether there exists a certain pattern in terms of the layout such as which object tends to have the largest portion in the painting. If there exist certain patterns, it would also be helpful in other tasks like classifying real Chinese paintings and fake ones. The semantic masks we get can also help with information retrieval tasks if the algorithm can be applied in real-time.

2.4 Semantic image synthesis of paintings

If time permits, we will look into how a Chinese painting can be generated using a semantic layout. The goal here is to learn the mapping between the semantic layout to a Chinese painting. The generated paintings should be hard to distinguish from real paintings.

2.4.1 How do existing image synthesis networks perform on generating Chinese paintings?

Generating a painting based on the semantic mask is a suitable task for conditional GAN which is introduced in Section 3.2. We will first try the famous pix2pix system [12] which is a network used for image-to-image translation and evaluate the outcome. Using this model as a baseline, we will further explore other models and also make our adaptations. For training data, we will use the Chinese painting dataset collected from art museums and Google Image Search (introduced in Section 4) and the semantic masks obtained from the semantic segmentation step.

2.4.2 How to generate more realistic paintings?

To make the generated paintings more realistic, we would first want the generated objects to appear complete in their area and the boundaries of any two connected objects to be well rendered. Secondly, we would like the generated paintings to have high resolution to appear more clear. For the first task, we will train the model with a large amount of data using data augmentation methods. We will also adapt the network such as adjusting certain layers or adding certain loss functions to achieve better results. For the second task, we will explore high-resolution image synthesis networks and take advantage of useful parts from such networks to make the results more clear. The user study can be performed to evaluate this research question. We will also test the model's generating ability by adding one object to the semantic mask and observe if the surroundings also adapt according to the change in the newly generated painting.

3 Literature Study

This section is divided into three parts. One is focused on semantic segmentation models. This semantic segmentation subsection is mainly about the state-of-the-art models which are mostly based on deep neural networks. In the second subsection, we present image translation methods, among which we specifically explain the background knowledge of GAN and its application. In the last subsection we give insights on how computer vision technologies are applied in archival collections to study art historical artifacts.

3.1 Semantic Segmentation

Semantic segmentation is a computer vision task for predicting the class labels for every pixel corresponding to the region it belongs to or surrounding region area. It gives an understanding of an image at the pixel level. The prediction is not only about the class label but also the boundaries of each object class. It is the combination of two tasks, classification and localization. It needs to classify each pixel to its corresponding class label and also find out the exact boundary of each object. Before the appearance of deep learning, machine learning algorithms such as SVM (support vector machine), clustering algorithms are used to solve the problem. However, the main disadvantage of these algorithms is that the features that are used to perform the prediction need to be picked manually beforehand. Engineering the features that will be used in the algorithm becomes crucial during the experiments. Deep learning compensates for this disadvantage by learning the features by itself. This unlocks a lot of features that are neglected by humans because some features that are extracted from a very detailed level are not intuitive and hard to understand from a global point of view. Figure 1 shows an example of semantic segmentation.

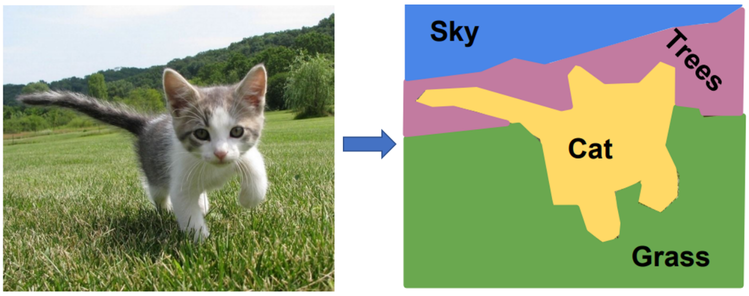

Figure 1: Semantic segmentation example. [9]

Semantic Segmentation on Realistic images [26] first proposed that training end-to-end, pixels-to-pixels on fully convolutional networks (FCNs) achieves better results than previous work based on convnets. A normal convolutional neural network (CNN) normally contains convolutional layers, pooling layers and fully connected layers. The convolutional layers are used to extract features from the input image. It is like a filter that will be applied repeatedly to cover the whole image and the weights in the filter will be learned during the training process. After passing the convolutional layer, we get a feature map of the input image. The pooling layers are used to reduce the number of parameters by using one value to represent one cluster of parameters. The most common one is called "max pooling" which only keeps the maximum value of a cluster of parameters. The dense layer is a fully connected layer where every neuron is connected to all the neurons in the last layer. This way, all the feature combinations are considered. Fully convolutional networks contain only layers like convolutional layers, pooling layers and upsampling layers (a simple layer without weights used for adding dimensions of the input) instead of dense layers. This study adapted previous successful networks like AlexNet [17] and VGG 16-layer net [27] which are designed for classification task into fully convolutional networks. By converting existing networks into fully convolutional networks, the networks can take different sizes of the input and get the corresponding sizes of output. Also, it reduces computational time required during the training process. As shown in Figure 2, after replacing dense layers with convolutional layers, when inputting a bigger image, the output is a heatmap of the object instead of just a class label. This is exactly what we need for the semantic segmentation task.

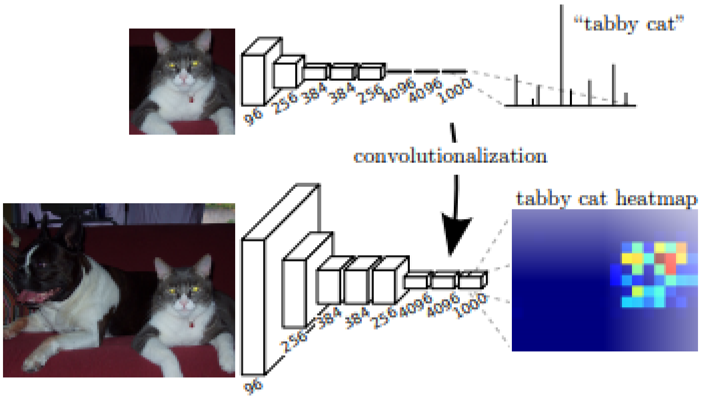

Figure 2: Transformation process. [26]

In most deep learning architectures, there exist two parts. One part is the encoder network and the other is the decoder network. The encoder is used for downsampling the image to reduce the size in order to subtract feature information and the decoder is used to upsample in order to get fine results with clear boundaries and original resolution. Instead of using interpolation algorithms, [26] proposes to use learned deconvolution layers to perform the upsampling. Since a lot of information is lost during the downsampling, they found the results not satisfying through direct upsampling from the output. So [26] further proposes new architectures FCN-16 and FCN-8 where information from previous pooling layers is also used. This allows the model to combine the global structure of the object while making local pixel prediction. The architectures are shown in Figure 3.

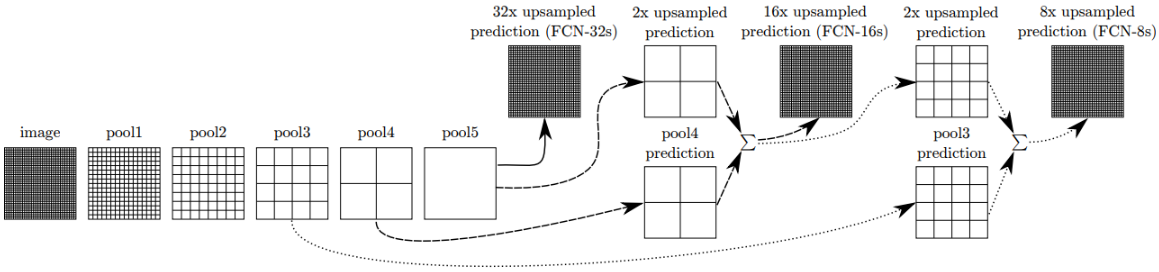

Figure 3: Skip layer architecture. [26]

One obstacle in semantic segmentation is how to deal with the relation between the global information which indicates what the object is (the semantic meaning) and the local information which indicates where the object is (the location information). In order to address this obstacle, the architecture is improved by adding skip layers. This skip architecture makes it possible to combine these two kinds of information from different level of layers. For example, [26] shows that adding links between lower layers and the final prediction layer to allow lower layers skipping ahead to higher layers leads to more accurate results.

On top of [26], [24] proposes a modified model designed for medical images. The new model needs fewer training images and therefore suitable for fields that do not have a lot of annotated data such as medical field. In [26], they combines information from previous pooling layers with the final feature layer to address the information loss problem. The modified architecture is called U-net. The network is also a fully convolutional network which consists of an encoder network and a decoder network. They expand the decoder network to concatenate more context information from previous layers, resulting in the U-shaped architecture shown in Figure 4. Different from [26], they propose to concatenate the feature map from the corresponding downsampling layer with each layer in the upsampling network. This way, more information is used during the upsampling process and this leads to better results.

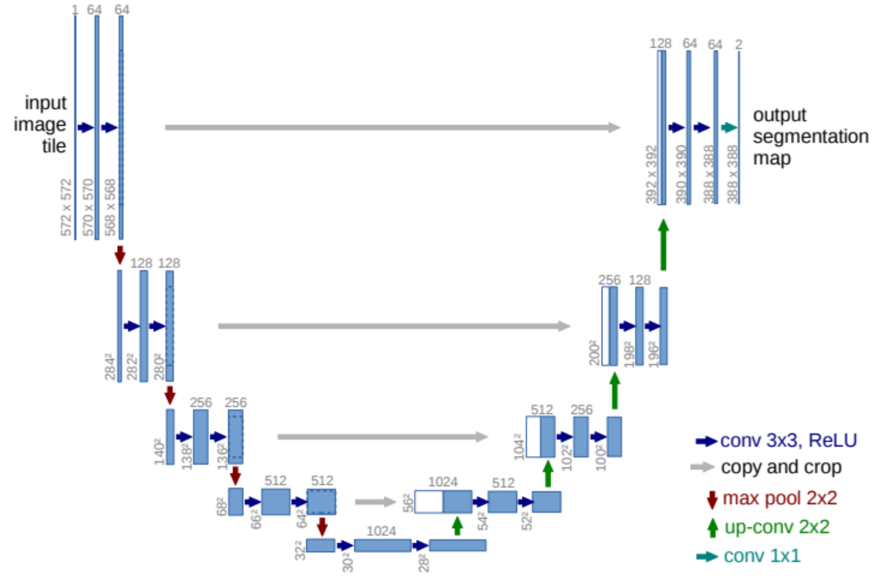

Figure 4: U-net architecture. [24]

In 2015, [2] proposes a new deep fully convolutional neural network architecture called SegNet. It is an efficient architecture designed for road and indoor scenes. They also use layers from the VGG16 network as the encoder network. The main architecture is similar to U-net. The novelty of their work lies in the decoder network. In order to make the network more efficient, instead of using the entire feature maps from previous layers, they propose a more efficient way to store important information. [24] stores every value in feature maps they need to use in the upsampling network. In this work, they only store the location of the maximum value in each pooling window in order to capture and store the boundary information in the feature maps. This way, they save a lot of memory and computational load and therefore make a network that is efficient and applicable in practical use.

Since VGG16 has a complicated architecture, the methods mentioned above have the disadvantage of requiring a lot of computation resources. In order to solve this problem so that the algorithm can operate in real-time on mobile devices, [22] proposes a new deep neural network architecture called ENet (efficient neural network). They attach more emphasis to the encoder network than the decoder network because they thought that the job of the decoder is simply fine-tuning the details while the main job of providing information lies in the encoder. Therefore, unlike U-net's symmetric encoder and decoder architecture, they choose to use a large encoder and a small decoder. They find that it is efficient to compress the information more at an early stage because the image contains a large amount of redundant information and it is computationally expensive to process them in the late stage. In order to avoid too much information loss, they use a pooling operation in parallel with a convolution and integrate the resulting feature maps. This expands the dimensionality in an efficient way. They further speed up the computational time and reduce the number of parameters by factorizing filters. It is mentioned in [14] that one $n \times n$ filter can be replaced with one $n \times 1$ filter and one $1 \times n$ filter. They adapted this idea in their network, resulting in large speedups and reduction of parameters. The result shows that ENet is significantly faster than SegNet (which is already one of the fastest segmentation models) while still achieving high accuracy.

The 'DeepLab' system proposed by [6] uses a different upsampling approach where a post-processing step is applied to improve the results. To deal with the information loss problem during the encoding process, instead of using a decoder network after the encoder network, they replace the downsampling operator from the last few layers with upsampled filter called atrous convolution. The upsampled filters are obtained by inserting zeroes between the filter parameters (shown in Figure 5). If there are n zeroes inserted between two parameters, then the dilation rate is $n - 1$.

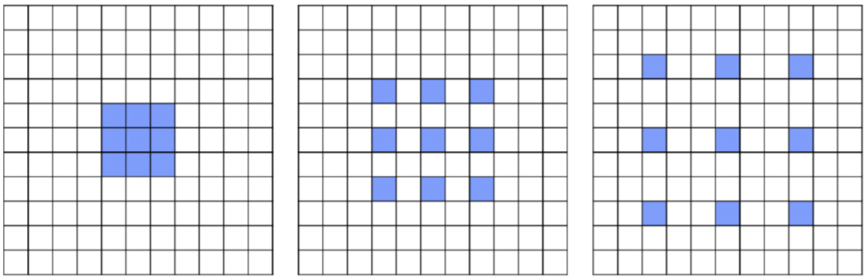

Figure 5: Atrous Convolutions with different dilation rates. [1]

In order to learn objects with different scales, they propose to use Atrous spatial pyramid pooling (ASPP) where multiple Astrous convolutional layers are used in parallel on a given feature map in order to capture different scales of the input context. They also propose to use Conditional Random Field (CRF) as a post-processing step to fine-tune the results to get more accurate boundaries. It combines the high-level information of the class label and the low-level information of the surrounding pixels and edges to make the final prediction. The overview of the whole network is shown in Figure 6.

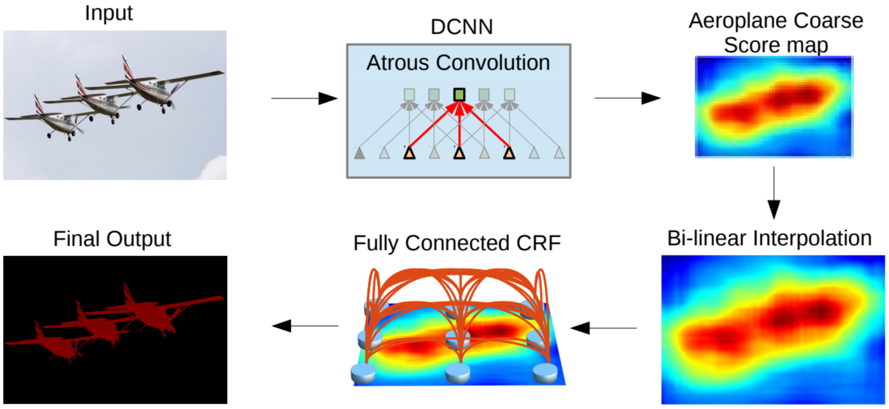

Figure 6: Overview of the Deeplab network. [6]

Figure 7 shows an overview of the networks mentioned above. They are divided into two main classes based on the upsampling methods. The Encoder-Decoder architectures show equal emphasis on both the encoder network and the decoder network. They present the idea that strong compression in the encoder network would require equally strong upsampling to restore the information. The architectures using Atrous convolution put more effort into the encoder network and use a more efficient way to perform the upsampling.

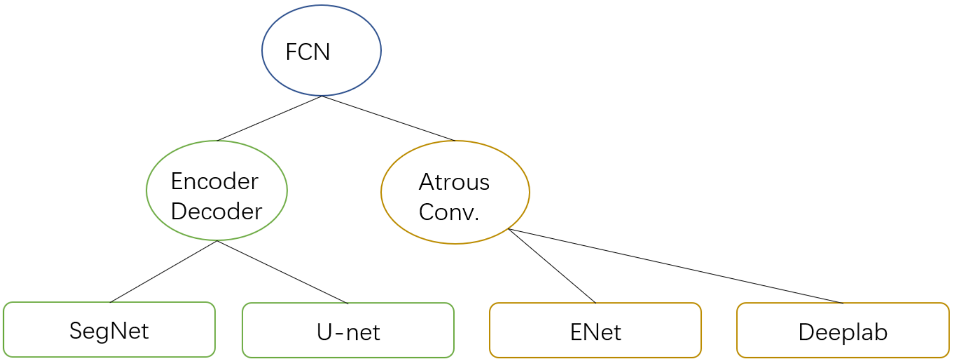

Figure 7: Overview of the previously mentioned networks.

Semantic Segmentation on Paintings Despite the promising results shown in the semantic segmentation task in realistic images, there are no effective models or fine-tuned models trained on paintings. One reason is that the number of paintings is much smaller than realistic images. The other reason is that the styles of paintings from different artists and eras are very distinguished, which makes the task even harder.

[23] proposes a solution for image processing on paintings. Since there are a lot of image processing algorithms well researched for realistic images, they propose to use style transfer to transform the paintings into realistic images. They adjust a GAN architecture [36] (the introduction of GANs is explained in Section 3.2) and a compound loss to achieve good results. Due to the lack of paired images of paintings and realistic images, they perform unsupervised training and realize style transfer in an unsupervised manner. They also compare the semantic segmentation results

(using DeepLab-V3 [7]) performed on original paintings and transferred images and prove that the accuracy improves significantly. In their work, they mainly focus on Chinese ancient paintings.

3.2 Image-to-Image Translation

One image can be shown in various formats such as edge maps, RGB images, semantic layout, etc. Image-to-Image translation is the process of transforming an image from one domain to an image from another domain. Depending on the input, there are different approaches and applications. Artistic painting generation is also an application that is getting more and more attention in the field of image-to-image translation. Although there are a lot of different image-to-image translation tasks (like style transfer, semantic segmentation), the essence of these tasks is predicting pixels from pixels. In this section, we will focus on researching different GAN architectures for generating realistic images from different inputs.

Since the appearance of Convolutional Neural Network (CNN), a lot of methods take advantage of the learning ability of CNN to learn the relation between the input image and output image by training on a large database with input-output examples. The principle behind the learning process of CNN is minimizing a certain loss function which is basically a score for how different the generated image is from the ideal one. Using different loss functions, you may get very different results since the objective value of the loss function tells the CNN which direction to go. This also means that different kinds of tasks require different loss functions.

Generative Adversarial Networks (GANs) bypass the process of constructing the right loss function because they can adapt a loss function according to the goal by itself. GANs are generative models. They learn the distribution of the training data. Take image synthesis as an example, there is a certain distribution in the training images that distinguishes them from other images. GANs try to learn this distribution and then generate a group of pixels that fit the distribution so that it would look like the training data. A typical GAN consists of two neural networks: a generator network G and a discriminator network D. The learning process is an adversarial process between the generator and the discriminator. The conceptual architecture is shown in figure Figure 8. The generator takes random noise as input and generates a candidate trying to fool the discriminator. The discriminator has the real image and learns to get better at classifying the real ones and fake ones. The generator is like a scammer who generates fake artworks and the discriminator is like an appraiser trying to discern real from fake.

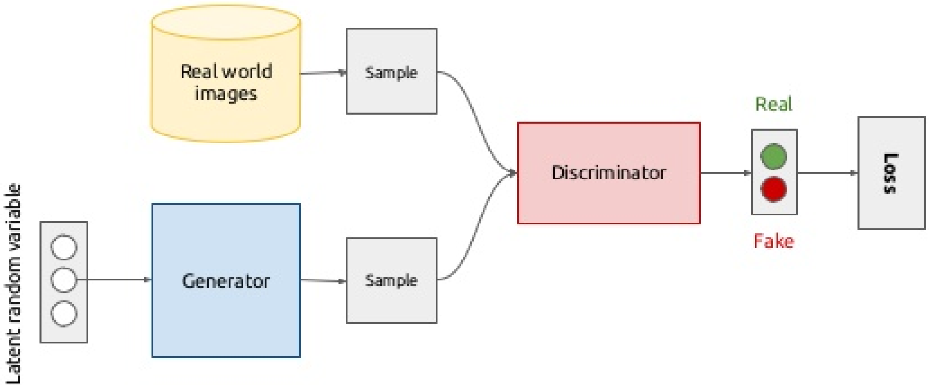

Figure 8: Conceptual architecture of GANs. [8]

The conceptual loss function format is as follows:

$$\min_{G}\max_{D} V(D,G) = \mathbf{E}_{x\ p_{data}(x)}[logD(x)] + \mathbf{E}_{z\ p_{z}(z)}[log(1-D(G(z)))].$$

p_g is the distribution that we want to learn of data x. The conceptual loss function consists of two parts. For some sample x, the value $D(x)$ is the probability that x is correctly classified by the discriminator. That is to say, if x is a real sample, then $D(x)$ is the probability that x is correctly identified as real by the discriminator. Alternatively if x is a fake, generated sample, then $D(x)$ is the probability that x is correctly identified as fake by the discriminator. $G(z)$ is the fake sample generated by G from a vector of random variables z, so $D(G(z))$ is the probability that a generated fake sample is classified as fake by the discriminator. The discriminator learns to maximize this function while the generator learns to do exactly the opposite. Under the objective function, the training process is an alternate process. Simply put, the process fixes G in one step, trains D, and then fixes D and trains G in the next step. When the generator G is fixed, the discriminator D learns from those samples it almost got fooled with in the last iteration. When the discriminator D is fixed, the generator G also learns from the classification results from D in the last iteration. This way, the two networks both learn to be better at their jobs. This alternate training process is used to make sure that the two networks are roughly at the same pace so that they can both be improved. For example, if the generator becomes way better than the discriminator, then the discriminator will always think that the generated image is real. Therefore, the generator will stop learning even though it did not reach an optimal performance yet.

Since GANs are widely used with image data, a lot of models use CNNs as the generator and discriminator. [12] proposes a general approach for image-to-image translation tasks. They apply GANs in the image-conditional setting so that it learns a conditional generative model to generate the output image based on the input image. Unlike normal GANs which learn the mapping from random noise to the desired output image, conditional GANs learn the mapping from certain input-output pairs. It adds a vector as an additional input layer to give the model more information. The match between the concept of conditional GANs and image-to-image translation enables this model to be adapted to various tasks including transform sketches, semantic segmentation maps, photos with missing parts, etc to normal realistic photos. They train different datasets on the model and it is shown that this is an efficient approach for many image-to-image translation tasks. The performance of this model is often used as a performance baseline when new models are proposed and the model itself is also used often in further research on conditional GANs.

[25] proposes a GAN architecture that takes user sketch as input and generates realistic images of cars, bedrooms or faces. Users can also indicate preferred sparse color strokes on the sketch to have more influence on the generated images. Based on [12], this approach is modified to be strictly feed-forward, which allows interactive user control of the generated images. By using augmented training data with different sketch styles, this model can generate high-quality images from imperfect human sketches.

For some tasks, paired training data is not available for training, however. [36] proposes an approach called CycleGAN that can translate images from a source domain to a target domain using unpaired datasets from each domain as training data. Since paired data is absent, the training process needs more constraints in order to get good results. Therefore, they propose to use an inverse mapping to translate the image from the targeted domain back to the source domain and use a cycle consistency loss to enforce the accuracy of this inverse mapping. This approach is also based on the framework of [12]. The method shows good results in terms of color and texture translations. When it comes to tasks that require geometric changes such as turning a dog into a cat, further improvement is still needed. This indicates that the model is more suitable for tasks like style transfer, season transfer etc.

[15] proposes a conditional GAN architecture called AL-CGAN which takes a semantic layout as input and generates a realistic outdoor scene. The users can not only control the semantic layout,

but they can also control the global appearance of the output image (sunny, foggy, etc). This is achieved by adding the scene attributes as another condition vector. This feature enables the model to generate more diverse images.

3.3 Computer Vision and Artworks

Most of the previous methods focused on generating photos with real-world scenes or simple objects. While more research is being done on realistic photos, the art field is gradually drawing more attention as well. [28] proposes a new architecture called ArtGAN that can generate paintings based on the input label of the style, genre or artist. This approach is similar to conditional GAN, but the difference is that in addition to feeding an additional input layer (the label) to the model, the generator also receives feedback regarding the label from the discriminator so that the generator can learn better within each label. The results show that this back-propagation on the label helps the model to generate art paintings with higher quality. Also, the comparison between the generated paintings and paintings used for training shows that this model does not generate paintings by memorizing the training set.

[18] proposes a new architecture based on [12] called Dual Scribble-to-Painting Network (DSP-Net). It takes a user sketch as input and generates a painting. Different from previous work, it imitates the human painting process from the basis of a scribble. A painter first needs to recognize the objects in the scribble and then apply colors. Therefore, recognizing objects can help the network to generate more accurate paintings from sketches. Based on this idea, multi-task learning is used, which consists of two networks of different tasks. The main network is used to learn to generate the artistic style based on the sketches. The secondary network is used to learn to recognize the semantic segmentation of the sketches. These two networks share the first few layers. So training the semantic network helps these shared layers to get a better semantic representation and further helps the main network to generate better artistic paintings.

[5] augments a CNN architecture (pre-trained VGG19) to concatenate it with semantic information that can be used during generation and shows that existing patch-based algorithms (neural patches algorithm) perform very well using this additional semantic information. The input of the model consists of three images: an original painting, its semantic annotations and your desired layout (the doodle). The output is a painting with your desired layout. This method also provides a way of content-aware style transfer by changing the input into a source style with annotations and a target content image with annotations.

[19] proposes a multi-scale deep neural network to transform a sketch into a Chinese painting. They train a deep GAN with multi-scale Chinese paintings with different levels of details so that the input sketch can be any size and the network can generate paintings with very fine detail in addition to a global framework of the painting. The results show that with this method, the model can generate a clear painting from a sketch regardless of the level of detail that is required. Furthermore, it also shows that by adding an edge detector, this model can also be used as a style transfer tool.

[31] proposes a mask-aware GAN called (MA-GAN) to generate traditional Chinese paintings. They trained their model to generate traditional Chinese portraits based on photos. For example, in [23], they also use unpaired training. They collect a dataset containing Chinese traditional figure images and a dataset with photos. In order to apply different strokes in different parts of the figures (face, hair, cloth etc.), they integrate the weights gained from the segmentation mask of the input photo into the loss function. The result is evaluated through Fréchet Inception Distance and user study. It is proved to be an efficient method to generate traditional Chinese paintings.

Different from other painting generation methods, [32] proposes a GAN-based method called Sketch-And-Paint GAN (SAPGAN) that can generate Chinese paintings without user input. The model mimics the process of painting in real life: first a sketch is made, which is then painted over. Therefore, the model consists of two GANs: the SketchGAN for generating edge maps which is trained on edge maps of Chinese landscape paintings, the PaintGAN for turning the sketch into the painting which is trained on paired edge maps and paintings. The result shows that this model can generate machine-original Chinese paintings. During user study, the generate paintings are mistaken as human-painted paintings with a frequency of 55%.

4 Dataset Preparation

In this section, we will introduce all the datasets we use in this study, including Chinese paintings, landscape photographs, COCO-stuff dataset and Bob Ross paintings, a manually annotated test set. We will also explain what data augmentation methods we apply to enrich our datasets.

4.1 Chinese Painting Dataset

For the Chinese paintings, we will use two datasets. One is provided by [32] and the other by [23]. The first dataset from [32] is collected from four museum galleries: 1301 from the Smithsonian Freer, 101 from Gallery, 428 from Metropolitan Museum of Art, 362 from Princeton University Art Museum and 101 from Harvard University Art Museum, resulting in 2192 paintings in total. The paintings are all landscapes and they are manually cropped and resized into 512×512 pixels. Figure 9 shows some samples from this dataset. This dataset is used as the test set in our study because of the high quality.

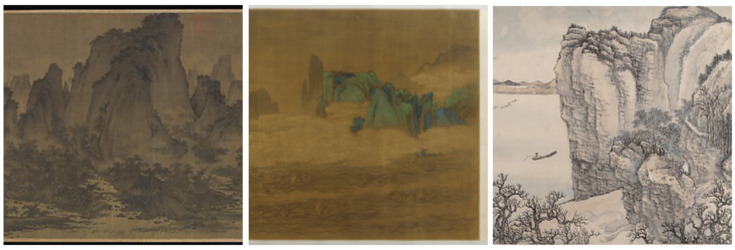

Figure 9: Examples of paintings from the first dataset.

The second dataset we found is from [23]. There are 2609 landscapes in the dataset collected from museums, picture albums and Google Image Search. According to the paper, they are all from Qing Dynasty. They are all cropped and resized into 256×256 pixels. Figure 14 shows some samples from the dataset. This dataset is used in the style transfer step in our study together with landscape photographs.

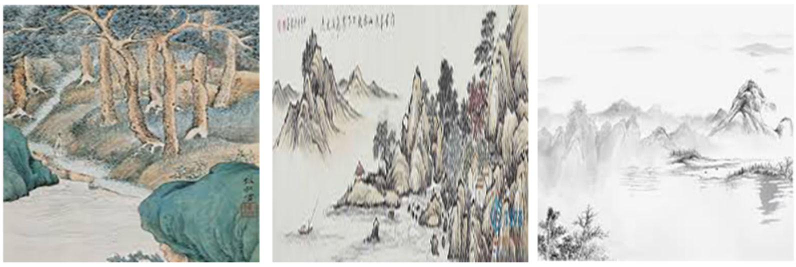

Figure 10: Examples of paintings from the second dataset.

In this research, they also provide two other datasets with 2719 bird paintings and 2935 flower paintings. In ancient Chinese paintings, landscapes tend to only have mountains and water. There are two other kinds of paintings which specialize in birds and flowers. We intend to include these two kinds of paintings to provide the users with more choices if time permits. An overview of the two datasets is shown in Table 1.

	Authors	Paper	Year	Painting type	Size	Number of samples
Dataset 1	Xue, Alice	[32]	2020	Landscape	512 x 512	2192
Dataset 2	T. Qiao etc	[23]	2019	Landscape	256 x 256	2609

Table 1: Comparison of the two datasets.

4.2 Bob Ross Painting Dataset

This dataset contains 250 Bob Ross paintings with semantic segmentation annotations. There are 9 classes in total in the annotations, which are sky, tree, grass, earth, mountain, plant, water, sea and river. This dataset is used as the training data for different semantic segmentation models because all the Bob Ross paintings are also landscape paintings. Although the style is different, the main objects in the paintings are similar. Therefore, the Bob Ross painting dataset is used as the main training data in this study. Figure 11 presents an Bob Ross painting from the dataset and the corresponding segmentation mask.

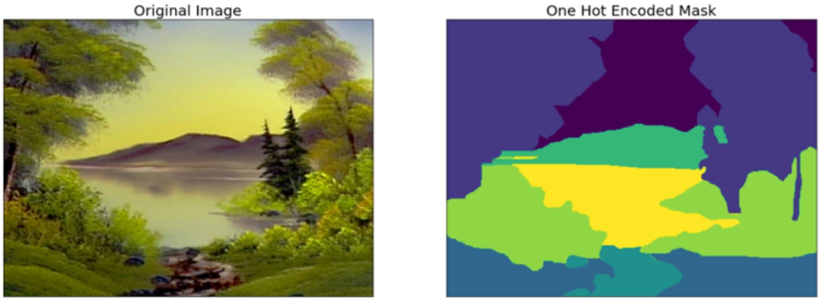

Figure 11: Example of the Bob Ross painting dataset.

4.3 COCO-stuff Dataset

The COCO-stuff dataset [4] contains 164.000 images with the semantic segmentation annotation. It has 182 classes in total including the classes we need such as trees, hills and rivers etc. It is used as a supplement to the Bob Ross painting dataset. We filtered out images with mountains and trees from the dataset and eliminated the remaining part of the image using an API provided by the COCO dataset. After filtering, we obtained 187 annotated images with mountains and 199 annotated image with trees from the COCO-stuff dataset. Figure 12 presents one example of an image with a mountain in it. The segmentation mask is shown in grayscale. This is done because the segmentation masks are required to be in greyscale most of the time when training a semantic segmentation model. Since the dataset was designed for the semantic segmentation task, the masks have already been transformed to a grayscale format.

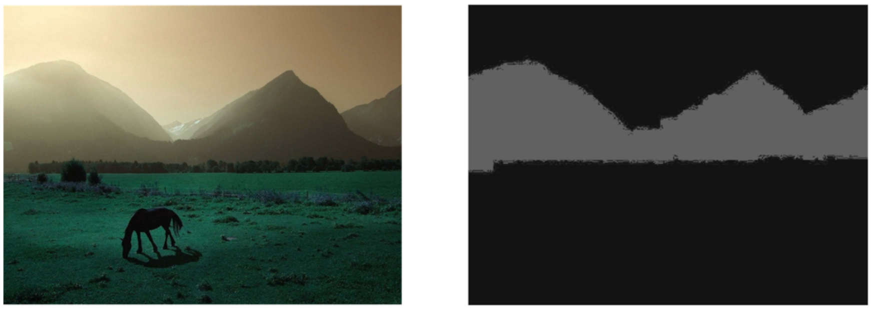

Figure 12: Example of an image from the COCO-stuff dataset.

4.4 Expanded Bob Ross Painting Dataset

An expanded Bob Ross painting dataset is used as the training data for training state-of-the-art models. It is the combination of Bob Ross paintings and images containing mountains and trees from the COCO-stuff dataset presented in the previous two subsections. The images filtered from the COCO-stuff dataset are an enrichment to the Bob Ross painting dataset. We will refer to this dataset as the "extended Bob Ross painting dataset" in the following sections. In total, the expanded Bob Ross painting dataset contains 636 images with semantic segmentation annotation.

4.5 100 Chinese Painting Test Set

In order to test the performance, we manually annotated 100 Chinese landscape paintings randomly selected from the 2192 paintings provided by [32]. This annotated dataset is used as the test set to evaluate the performance of all the models and proposed methods.

4.6 Data Augmentation

Since we have a relatively small dataset, we performed basic random rotation and cropping to enrich the training data.

5 Methodology

The inspiration of this research comes from [5] and [21]. [21] proposes a GAN-based architecture that can generate photorealistic images using a semantic layout as the input. [21] proposed a convolutional neural network that can generate a painting that has the user's desired layout based on an input painting and its semantic segmentation annotation. Figure 13 and Figure 14 show the final application results in these two papers. In Figure 13, different stylized results are obtained through style transfer and in Figure 14 the semantic maps are obtained by a manual approach. This indicates the lack of a semantic segmentation algorithm that can be applied to paintings in order to get paired data of real paintings and corresponding semantic masks. Our research intends to close this gap. Therefore, the main task in our research lies in semantic segmentation.

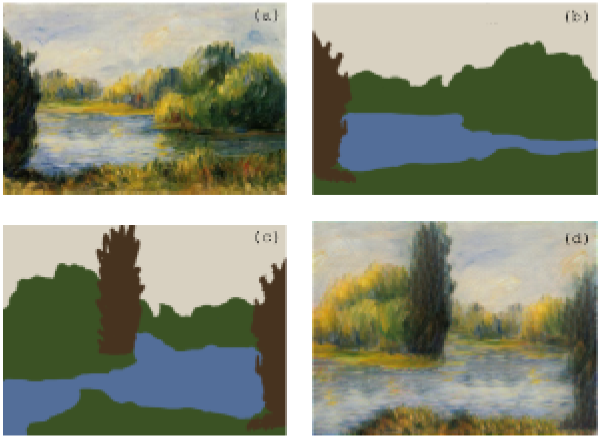

Figure 13: Application results (a) [5].

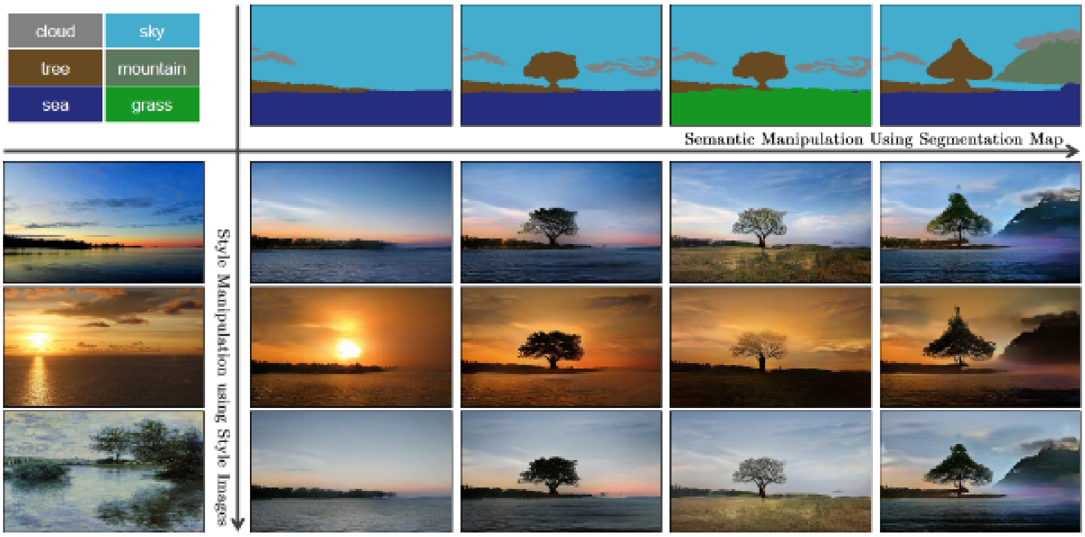
cloud
sky
tree
mountain
sea
grass
Semantic Manipulation Using Segmentation Map
Style Manipulation using Style Images

Figure 14: Application results (b) [21].

In this section, we will dive into all the methodologies we use in this research project. This section starts with an introduction to the semantic segmentation models we use in our research. Following the models, we introduce the methods we propose to improve the performance obtained by directly applying the models to Chinese landscape paintings.

As for the state-of-the-art models, we will choose to study Unet [24], DeepLab-V2 [6] and DeepLab-V3 [7] in our research. Unet is designed for biomedical image segmentation task where large-scale training data is not available. It therefore has an advantage for tasks where the training dataset is small. The DeepLab model uses special architecture features to extract higher resolution feature maps while still being computational efficient. The details of these three models will be explained in this section.

In order to answer the first research question, which involves testing the state-of-the-art performance on Chinese ancient landscape paintings, we will use Unet, DeepLab-V2 and DeepLab-V3, which will be applied directly to the paintings. We will use the pipeline shown in Figure 15. To answer the second research question, we will use the pipeline shown in Figure 16 to see how do these methods improve the performance. For the third research question of finding patterns in Chinese ancient landscape paintings, we will follow the pipeline from Figure 17. The pipeline allows us to find two types of features in the paintings. One is the composition proportion for each class and the other one is the pattern in the location of different classes. For the fourth research question, which involves generating Chinese paintings, we will follow the pipeline shown in Figure 18 if time permits.

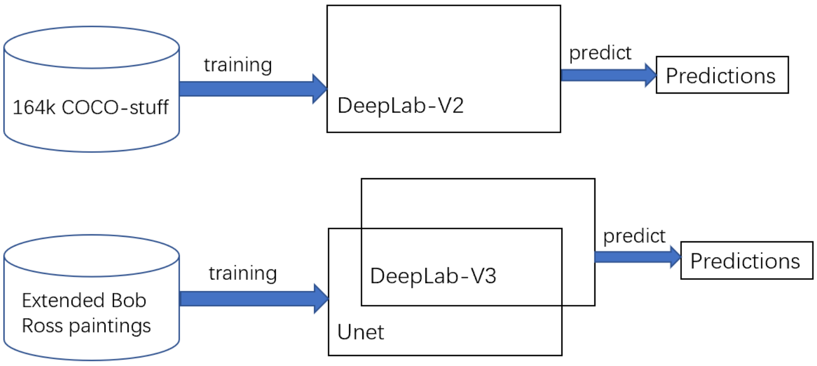

Figure 15: The pipeline for first research question.

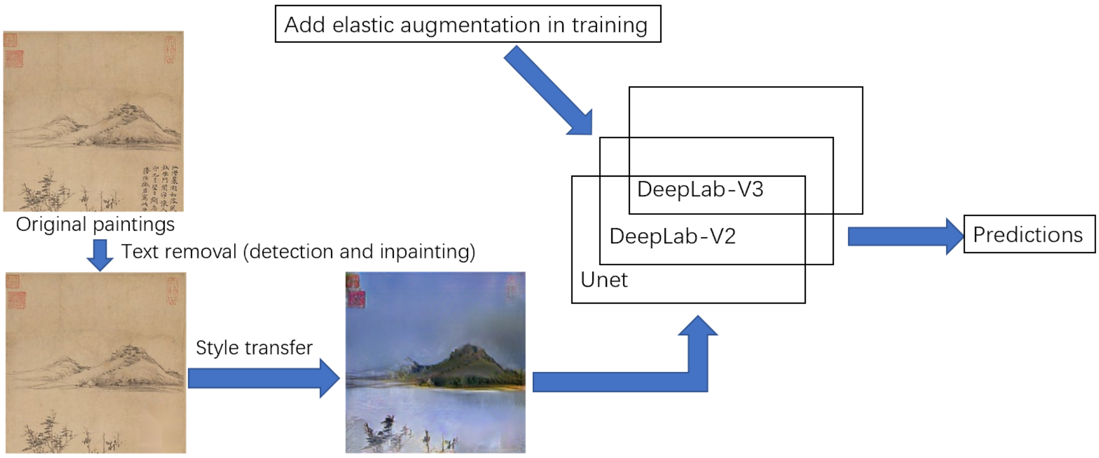
Add elastic augmentation in training
Original paintings
Text removal (detection and inpainting)
Style transfer
DeepLab-V3
DeepLab-V2
Unet
Predictions

Figure 16: The pipeline for the second research question.

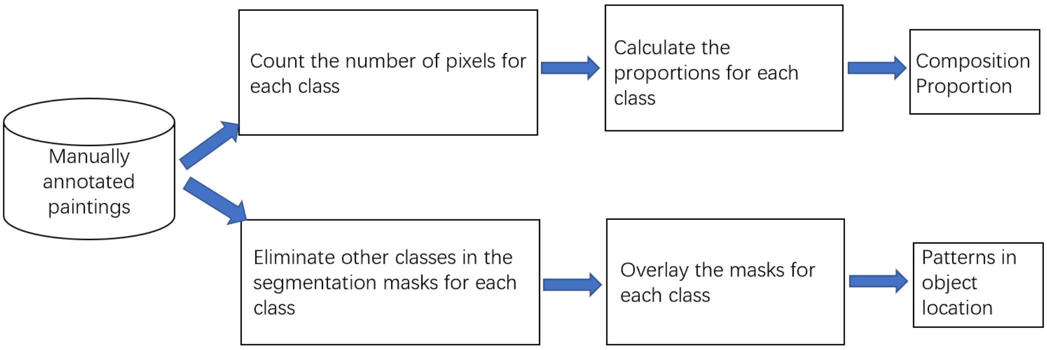
Manually annotated paintings
Count the number of pixels for each class
Calculate the proportions for each class
Composition Proportion
Eliminate other classes in the segmentation masks for each class
Overlay the masks for each class
Patterns in object location

Figure 17: The pipeline for the third research question.

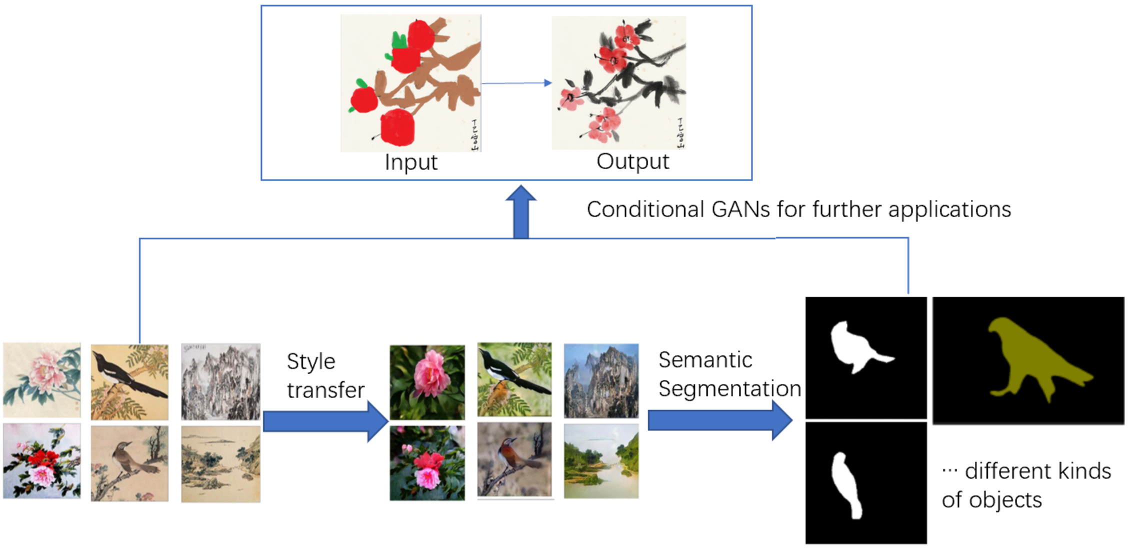
Input
Output
Conditional GANs for further applications
Style transfer
Semantic Segmentation
··· different kinds of objects

Figure 18: The pipeline for the fourth research question.

5.1 DeepLab model

5.1.1 DeepLab-V2

The use of the combination of max-pooling and downsampling in Deep Convolutional Neural Networks leads to the reduction in the resolution of the feature map. Since semantic segmentation is a low-level classification task which requires both high level and low level features, a coarse feature map can harm the prediction accuracy of the model. The DeepLab model [6] deals with this problem by using Atrous Convolution instead of the downsampling operator after the last few max-pooling layers. Afterwards, a simple bilinear interpolation is used to make a feature map the same size as the input images. This way, a denser feature map is computed efficiently. The DeepLab model computes a high resolution feature map efficiently by replacing downsampling filters with Atrous Convolution instead of deconvolutional layers. Figure 19 shows the difference between feature maps computed using the traditional convolutional way and those computed using the Atrous convolutional method.

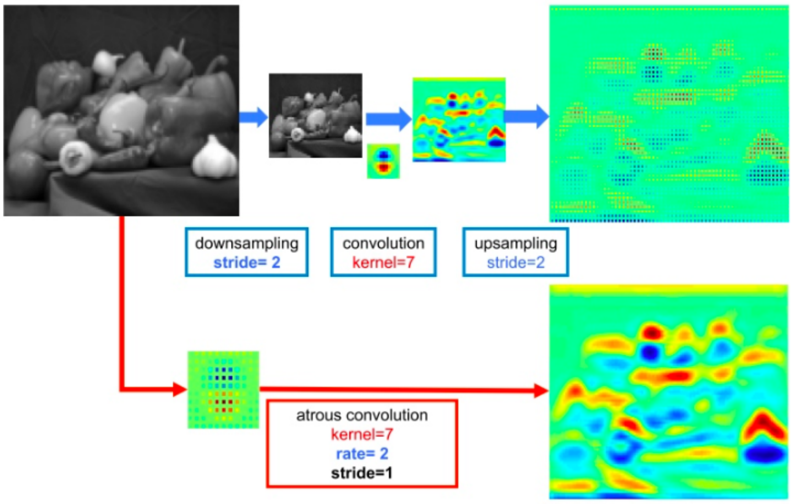

Figure 19: A feature map computed by different methods [6].

In Chinese paintings, it is a common situation that objects from one class appear in one painting at different scales such as mountains at the very distant or very close by. For object with different sizes, DeepLab-V2 model [6] deals with this problem by combining Atrous Convolution with Spatial Pyramid Pooling [11]. A traditional way to solve the scale problem is to extract different scales of feature maps from each image and interpolating them into the same size in order to train the model with images at different scales. However, this method is highly computationally expensive. By using Atrous Spatial Pyramid Pooling, instead of extracting feature maps at different scales, it only resamples the feature map using Atrous Convolution at different rates. This method helps with recognizing objects with various sizes at a low computational cost. Figure 20 shows the structure of Atrous Spatial Pyramid Pooling.

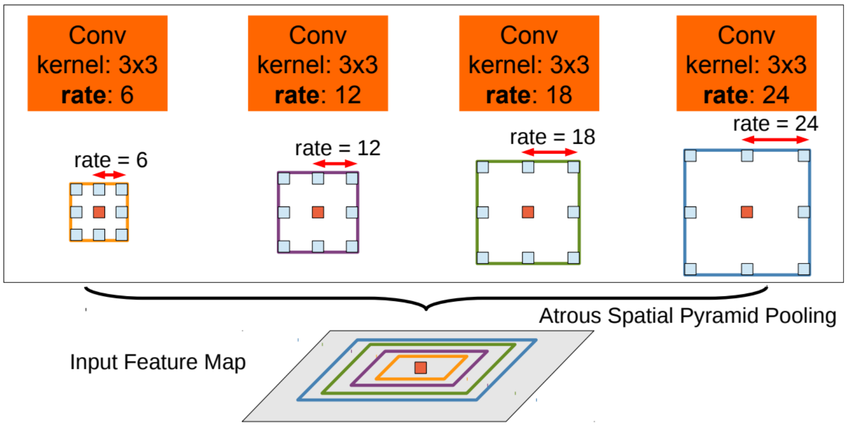

Figure 20: Atrous Spatial Pyramid Pooling [6].

In order to create smooth boundaries for segmented objects, the DeepLab-V2 model integrates Fully Connected Conditional Random Field with the model as a past-processing step. Conditional Random Field is used to smooth out the noisy parts in the segmentation masks. The model achieves this outcome by having two penalties. First, it penalises pixels for which the classifier has an uncertain prediction. Second, it penalises pixels that have different class labels compared to their adjacent pixels. The DeepLab-V2 model applies Fully Connected Conditional Random Field where every pixel is treated as if they were adjacent to each other. This is an efficient way to perform optimization on the whole segmentation mask. After this post-processing step, the edges and the pixels inside each object are refined. Figure 21 shows different outputs before and after CRF processing. We can see that those small noisy parts are getting filled in and the boundaries are becoming more and more accurate. In Chinese paintings, the edges of objects are normally not clear and the objects are often not fully covered with brush strokes. This extra step in the DeepLab-V2 model can reduce the difficulty of producing a clean and complete segmentation mask on a Chinese painting.

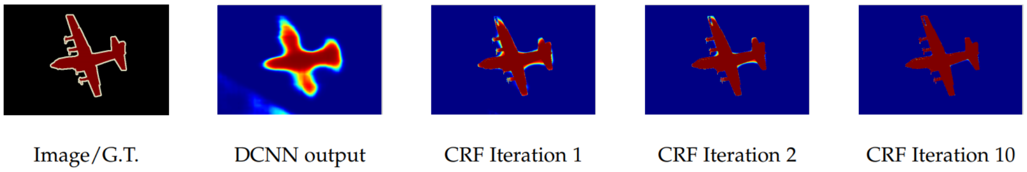

Figure 21: Outputs comparison before and after CRF [6].

Besides the architecture features above, Deeplab-V2 provides pre-trained weights on the 164K COCO-stuff dataset, which would have taken too much time and resources if we trained the model by ourselves from scratch. This pre-trained weight was used in another project [21] as the semantic segmentation model for landscape photographs. The similarity between their project and ours makes the pre-trained weights on the DeepLab-V2 model worth to try as a starting point and a baseline model in our research.

5.1.2 DeepLab-V3

We will also experiment with the DeepLab-V3 model [7]. We will train this model using the expanded Bob Ross painting dataset we built ourselves. Compared to DeepLab-V2, no significant architecture changes are made. The main difference between these two versions lie in the following aspects. First, in the DeepLab-V3 model, the Atrous Spatial Pyramid Pooling (ASPP) is augmented. Instead of resampling on an arbitrarily sized feature map, it uses an image-level feature map. The reason for this augmentation is that when resampling on an arbitrarily sized feature map, when the rate in Atrous Convolution becomes bigger, the valid weights in the filters (which are values computed on the valid feature map area) decreases. This causes some computational waste in addition to a loss of global feature information. By using an image-level feature map, the number of invalid weights is less than before as the rate increases. Figure 22 shows the relation between valid weight counts and atrous rate in a case where a 3x3 filter is applied on a 65x65 feature map. We can see that as the Atrous rate increases, the normalized count for when all 9 filter values are applied validly decreases and the count for when only one filter value is valid increases.

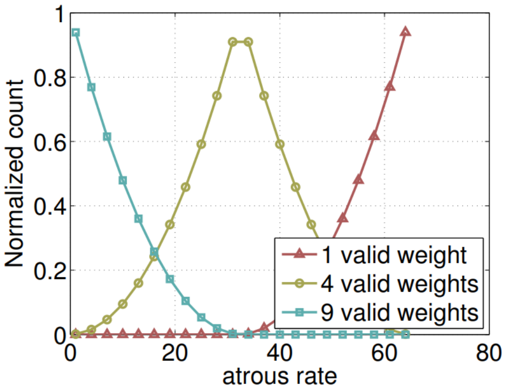

Figure 22: The relation between valid weight counts and atrous rate [7].

Besides the feature map, a batch normalization layer is also added in the Atrous Spatial Pyramid Pooling. [7] finds that adding and fine-tuning this batch normalization layer improves the performance of the model. Compared to DeepLab-V2, the performance of DeepLab-V3 imporved through the augmentation in the ASPP process. It is shown in [7] that even after removing the CRF post-processing step, DeepLab-V3 still outperforms DeepLab-V2. Therefore, we choose DeepLab-V3 as the model to train on our own customized training data (expanded Bob Ross painting dataset) and also consider the results as a baseline.

5.2 Unet

The Unet model [24] has a classic encoder-decoder architecture. This network can be trained with a small set of training data and the segmentation speed is also very quick. The Unet model is designed for biomedical images. Therefore, it is designed to be effective even on a limited dataset. In our case, training set is also small, which makes Unet a suitable model for our research. Unet has a symmetrical structure as shown in Figure 4. The left side of the structure consists of repetitive combinations of convolutional blocks and max pooling blocks for downsampling and the right side of the structure consists of repetitive combinations of upsampling operators and convolutional blocks. Through the connection between the downsampling part and the upsampling part, the upsampling part can take advantage on the feature maps extracted before which have higher resolution, allowing us to produce more accurate outputs. The Unet paper [24], applies extensive data augmentation such as flipping and elastic deformations due to the limited data they have, which achieved great performance with a small amount of annotated data. Therefore, we select Unet as one of our baseline models trained on the expanded Bob Ross painting dataset.

5.3 Style Transfer: Cycle GAN

In order to perform style transfer in the absence of paired training data, currently Cycle GAN is the only option. A style transfer method that is widely used in other researches [33] [13] and proven to be efficient is called Neural Style Transfer [10]. This method takes a style image and a content image as the input and output the combination of the content image and style image. It can preserve the content in the converted image very well by controlling the weight in the loss function. However, this is not suitable for our task because we need to do more than just transfer the texture of the style image to the entire painting. What we need to do is to transfer the water texture in photographs to the water area in the painting, transfer the mountain texture to the corresponding area in the painting etc. This requires the model to dig deeper and learn more about the correlation between the source domain (the Chinese paintings) and the target domain (the landscape photographs).

The working mechanism of GANs is explained in Section 3.2. Here we will focus on introducing the model we will use in our research which is obtained from [23]. It consists of two sets of generator and discriminator. Since the purpose of this Cycle GAN is to transfer the images from one domain (ancient Chinese paintings) to another domain (realistic photographs), this model is named as Domain Style Transfer Network (DSTN). The model learns two mapping functions which compose the cycle. Figure 23 shows the structure of DSTN. X and Y represent the two domains. G and F are the two mapping functions the model needs to learn. Each mapping function is connected with the corresponding discriminator (D_X and D_Y). The discriminator acts like an examiner. D_Y will supervise the images generated by mapping function G which is trying to learn to transfer images from domain X to domain Y and D_X will supervise the images generated by mapping function F which is trying to learn to transfer images from domain Y to domain X.

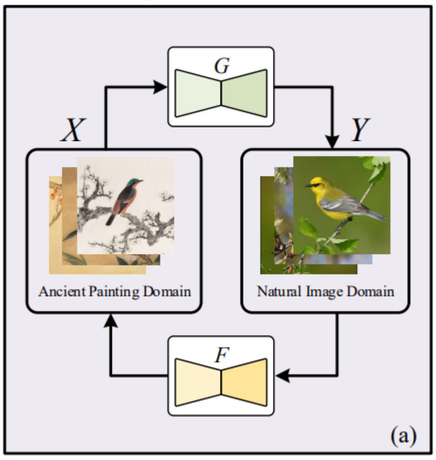

Figure 23: The structure of the Cycle GAN [23].

In DSTN, a compound loss is applied to ensure the color and content composition remains the same after the style transfer. The compound loss consists of four parts: an adversarial loss, a cycle consistency loss, a content loss and an identity loss.

Figure 24 shows the adversarial loss for one pair of the generator and discriminator. Adversarial loss is the core loss used in every GAN to train the generator to generate similar images as the target. x and y represent the image samples from domain X and domain Y respectively. $G(x)$ is the fake images generated by generator G from domain X to look like images from domain Y. The loss function for generator G is shown in Equation (2). In the training process, the generator G is trained to minimize the loss function while the discriminator tries to maximize the loss function at the same time since the generator wants to fool the discriminator with generated fake images and the discriminator tries to distinguish between the fake images and the real images. The complete adversarial loss function for the two pairs L_{GAN} is shown in Equation (3).

$$L_{GAN_G} = \mathbb{E}_{y \sim p_{data}(y)}[logD_y(y)] + \mathbb{E}_{x \sim p_{data}(x)}[log(1 - D_y(G(x))] \tag{2}$$

$$L_{GAN} = L_{GAN_G} + L_{GAN_F} \tag{3}$$

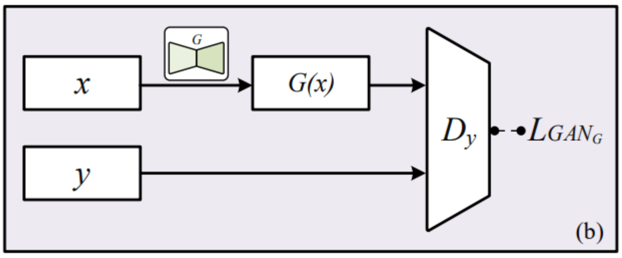

Figure 24: Adversarial loss [23].

The cycle consistency loss is used to solve the problem of multi-mapping in the cases where the training data from the two domains are not paired data. This loss function ensures the two generators to be consistent, which means that if one image is converted to another style through generator G, after putting it through generator F, the image should be converted back to the original image. Figure 25 visually presents how is cycle loss (in one direction) calculated. Generator G is used to fake an image from domain X and generator F is used to transform this fake image back to domain X. The same goes for the opposite direction (generating images from domain Y). The cycle consistency loss function L_{cyc} is defined as the difference between the real image from its domain and the fake image restored from another generated fake image. The formula of this loss function is shown in Equation (4) and Equation (5). α_G and α_F are the weights used to adjust two parts of the loss function.

$$L_{cyc_G} = \mathbb{E}_{x \sim p_{data}(x)}[|f(G(x)) - x|] \tag{4}$$

$$L_{cyc} = \alpha_G L_{cyc_G} + \alpha_F L_{cyc_F} \tag{5}$$

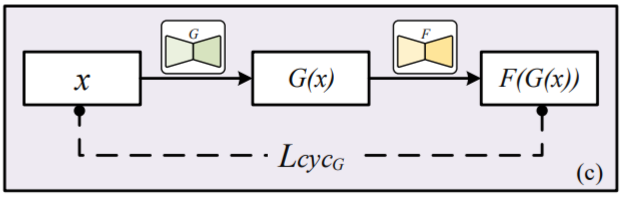

Figure 25: Cycle consistency loss [23].

The identity loss is used to further restrict the generators. The idea of this loss function is simple. If an image is already in the style of the target domain, the generator G should output the exact same image. The same holds for the other generator F. The formula of this loss function is given in Equation (6) and Equation (7).

$$L_{id_G} = \mathbb{E}_{y \sim p_{data}(y)}[|G(y) - y|] \tag{6}$$

$$L_{id} = L_{id_G} + L_{id_F} \tag{7}$$

The last loss function, content loss, is used to preserve the content information through the style transfer. The content features are calculated by a pre-trained neutral network ϕ. With the high-level features extracted, the content loss function can be defined as in Equations (8) and (9).

$$L_{con_G} = \mathbb{E}_{x \sim p_{data}(x)}[|\phi_j(G(x)) - \phi_j(x)|] \tag{8}$$

$$L_{con} = L_{con_G} + L_{con_F} \tag{9}$$

After defining every loss function, the compound loss function can be defined in Equation (10). α_G, α_F, β and γ are the weights for each kind of loss function which can be adjusted.

$$L(G, F, D_x, D_y) = L_{GAN} + \alpha_G L_{cyc_G} + \alpha_F L_{cyc_F} + \beta L_{id} + \gamma L_{con} \tag{10}$$

5.4 Text Detection

In some paintings, there are text areas in them, which affects the performance of style transfer. So we will first detect the text areas, then remove them. For text detection, we will first test the traditional computer vision method because the text area is relatively isolated and often has a clean background. For text that has a clean and simple background, Maximally Stable Extremal Regions (MSER) is a quick and accurate method [16]. The processing process of MSER is as follows: First, the image is processed with gray-scale. Then different thresholds are applied to each image for binarization. The threshold will start from 0 to 255. As the threshold increases, the division of the image is changes. Some connected areas in the image have little or no change. This area is called the maximum stable extreme value area. Because the gray-scale value of the text area tends to be the same in an image, during the increasing of the threshold, the text area should show little change until the threshold increases to the same gray-scale value. Using this algorithm, we can roughly locate the text area in an image. We will use the pre-defined function from the `OpenCV` library to perform the algorithm.

We will also test the performance of neural network based methods. We will elect to use the method proposed in [3]. It is a text detection method based on character-region awareness. Their model uses an architecture that combines VGG-16 [34] with skip connections. The network is trained based on two scores in order to achieve better results when encountering a text area with an abnormal shape. As shown in Figure 26, the two scores are the region score and the affinity score. The region score indicates the location for each character. The higher the region score is, the more likely it is that this pixel belongs to a character. The affinity score indicates whether a certain area can form a text area. The higher the affinity score is, the more likely it is that the pixel is at the center of two adjacent characters. A wide range of datasets are used to train the model, therefore it can be applied to text detection tasks with different language requirements including Chinese. This method achieved impressive performance on different benchmark datasets and also shows great generalization ability.

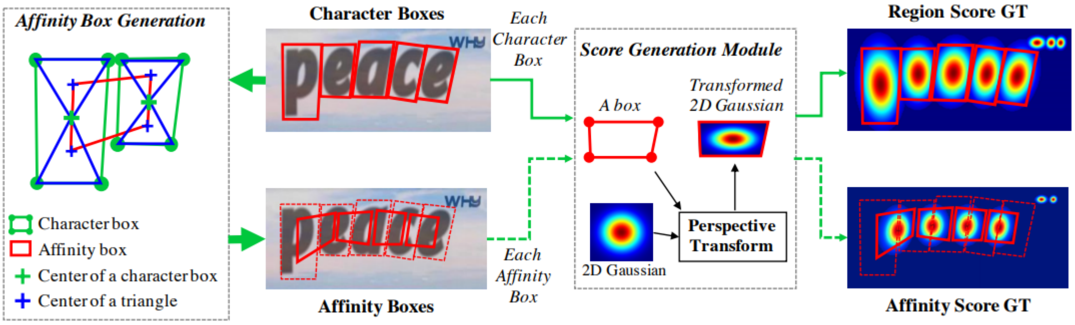

Figure 26: The process of generating the ground truth based on two scores [3].

5.5 Image Inpainting

Image inpainting is a task where certain algorithm is used to fill in the missing pixels in the image and the filled part needs to be realistic and also needs to fit the context of the remaining part of the image [35]. It can be used to remove unwanted objects from images. Figure 27 shows examples of results from image inpainting. In our case, our goal is to remove the text region. Therefore, after text area detection, we will fill the area with white color. Then, we need to use image inpainting to complete the image.

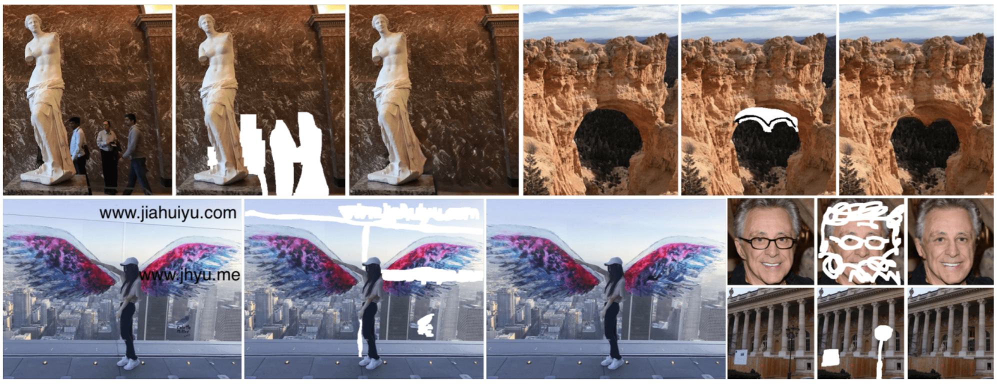

Figure 27: Examples of image inpainting applications [35].

Due to the simplicity of the surrounding area of the text region, we chose to use the `inpaint` function with `cv.INPAINT_TELEA` as the last parameter from the `OpenCV` library. This inpainting function is based on [29]. It starts with the boundary of the area that needs to be inpainted and uses Fast Marching Method to decide the next pixel to inpaint. The selected pixel is inpainted with the value of the weighted sum of certain amount of known pixels in the surrounding area. The range of the this area can be set by passing a parameter.

5.6 Evaluation

For the evaluation, we will use `chainercv.evaluations.eval_semantic_segmentation` function from `ChainerCV` library [20]. The output of the Deeplab-v2 model is a matrix of labels. Each element in the matrix corresponds to a pixel and the value of each element is the label index of the predicted class. The ground truth annotation is in json format and it does not contain the information for each pixel. Therefore, we transfer the annotated image into a matrix and create a dictionary that matches the RGB value for each class to its corresponding label index. This way, the output of the model and the ground truth can be compared. The IoU value is 0 when the class exists in the ground truth and there is no overlap in the prediction.

The output of the function contains following metric values: IoU of each class, mean IoU of all classes, pixel accuracy, class accuracy of each class and mean class accuracy of all classes. Pixel accuracy is the most intuitive one. It is the percentage of correctly predicted pixels in the whole image. Although this metric is easy to understand, it may cause a problem called class imbalance. Consider the image below as an example. There are two classes in the image, one is “ship” and the other one is “background”.

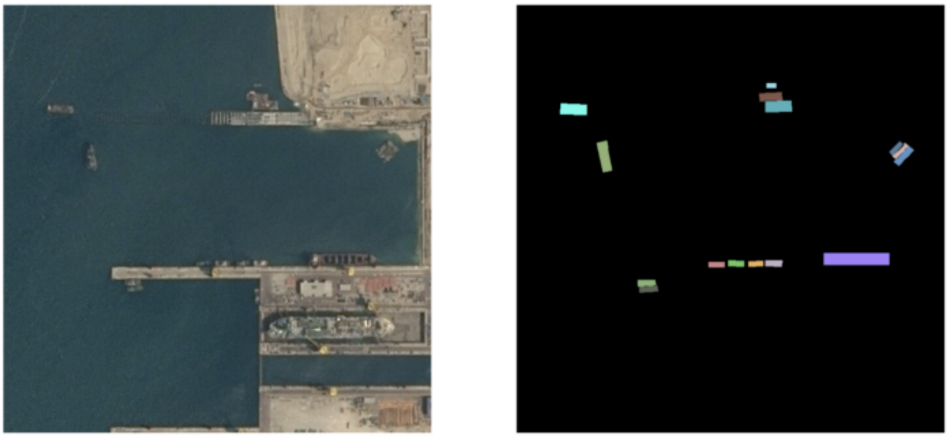

Figure 28: Pixel accuracy example [30].

The image on the left is the original image and the one on the right is the ground truth segmentation mask. For a predicted result that is shown in Figure 29, the pixel accuracy is actually 95% but we can see that the result is hardly useful. This is because the background class takes up a big part of the image. This class imbalance problem occurs due to the dominance of one class in an image and other classes only taking up a small portion, which results in the dominant class determining the final evaluation results for the whole image and the pixel accuracy value not expressing the actual accuracy of the results.

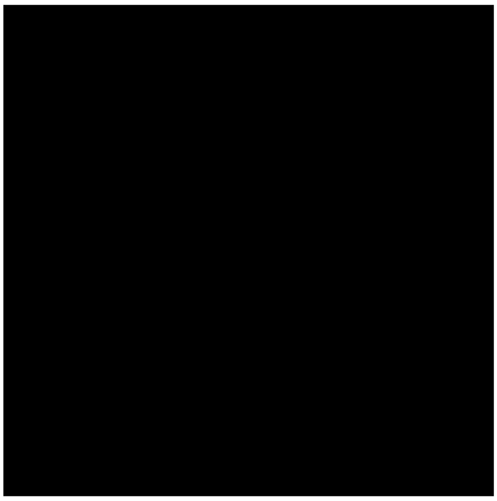

Figure 29: Pixel accuracy example [30].

Since the pixel accuracy is not a suitable metric for the evaluation, most of semantic segmentation researches use Intersection-Over-Union (IoU) as the evaluation metric. The IoU for one class is the ratio of the overlap area of the ground truth and the predicted segmentation and the union of these two areas. It is shown in the formula below:

$$IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}} \tag{11}$$

For an image with multiple classes, the IoU is the average of all the classes. The value of IoU ranges from 0 to 1 and the higher the value is, the better the segmentation results are. If we use IoU to evaluate the example above again, taking Figure 29 as the segmentation result, the IoU for the "ship" class would be 0 and the IoU for the "background" class would be 0.95, resulting in the mean IoU value of 0.48. This IoU metric gives the class that does not take up large portion of pixels more influence on the final result, making this metric a better evaluation method and widely used in research.

6 Results

In this section, we are going to present the results obtained from applying the methods mentioned in Section 5. We will report both quantitative results and qualitative results. We will start with reporting the performance of state-of-the-art segmentation models on ancient Chinese landscape paintings. Then, we will show the results of proposed methods and the performance effect they have on the segmentation models. The statistical analysis results are shown at the end.

For the qualitative result, there will be different color blocks. Each color represents a class. Figure 30 shows the correspondence between the color and class name.

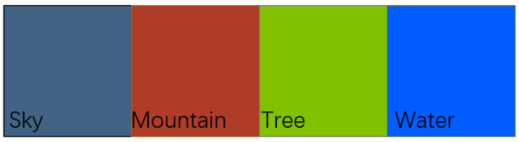

Figure 30: The correspondence between the color and class name.

6.1 The state-of-the-art semantic segmentation algorithms performance

6.1.1 DeepLab

We tested the DeepLab-V2 model pre-trained on the 164k COCO-stuff dataset on 100 Chinese landscape paintings. The results are used as the baseline. The IoU scores for each class are shown in Table 2.

Sky	Tree	Water	Mountain
0.31	0.28	0.16	0.24

Table 2: Results from Deeplab-V2 model trained on COCO-stuff

Figure 31 shows some qualitative results. From the examples, we can observe the following points: first, for mountains, the model can sometimes recognize the mountain area, which means that although the model is trained on photographs and the mountains in Chinese paintings have a different appearance and texture compared to photographs, some features such as the shape can still be picked up by the model and used for prediction. The same goes for the sky area, although there are no strokes or color in the sky area, the model can still make some right predictions. This is possibly due to the spatial information. However, in terms of the classes “tree” and “water”, the model barely recognizes any, especially the water area. The water area in the middle is not a common structure in the training data. Moreover, there is no other information to help the model make the prediction, which results in the complete failure to recognize the water area.

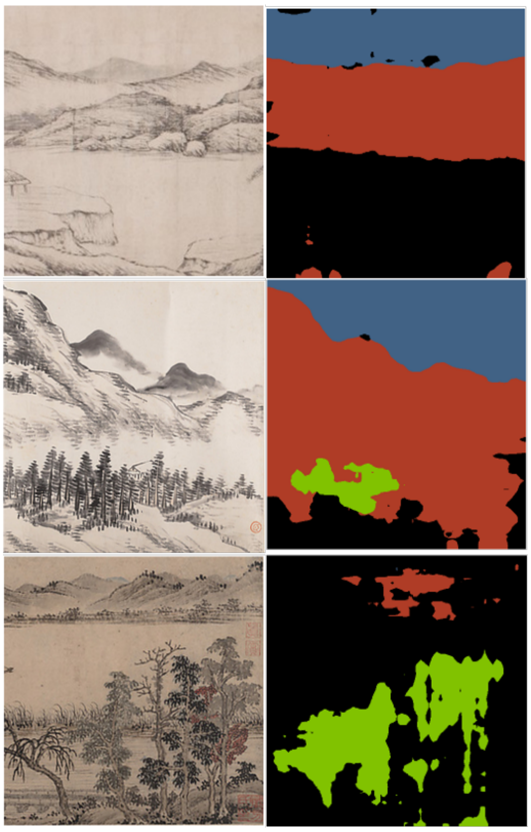

Figure 31: Qualitative results from DeepLab-V2 model trained on COCO-stuff.

To sum-up, the state-of-the-art image segmentation algorithm DeepLab-V2 fails to recognize certain classes such as water and trees even though it is trained on a large scale dataset.

6.1.2 Using expanded Bob-Ross paintings as training data

The performance of a neural network depends largely on the training data. Given the absence of annotated data of Chinese paintings, we found a dataset of Bob Ross's paintings together with semantic segmentation annotations. We trained two models, Deeplab-V3 and Unet, on Bob Ross dataset together with 187 images with mountains and 199 images with trees from the COCO-stuff dataset in order to enrich the diversity of mountains and trees in terms of shape and texture. The results obtained from the models Unet and Deeplab-V3 are shown in Table 3. We can see that the performance improved significantly on both models. Although the style is very different and the amount of data is much smaller than that of the COCO-stuff dataset, using the expanded Bob Ross dataset for training is a good step since this dataset is more focused on landscape objects such as mountains and trees whereas COCO-stuff contains a wide range of object classes.

Model	Sky	Tree	Water	Mountain
Unet	0.29	0.31	0.07	0.35
DeepLab-V3	0.44	0.30	0.26	0.40

Table 3: Results from Unet and DeepLab-V3 trained on the expanded Bob Ross dataset.

Figure 32 and Figure 33 show some qualitative results from the two models. From these qualitative results, we can see that a big difference between the results gained from these two models is that Unet's prediction has a lot of small blocks whereas DeepLab-V3's prediction tends to be more complete, containing less small color blocks. This is better for further research such as building a data set for Chinese painting objects.

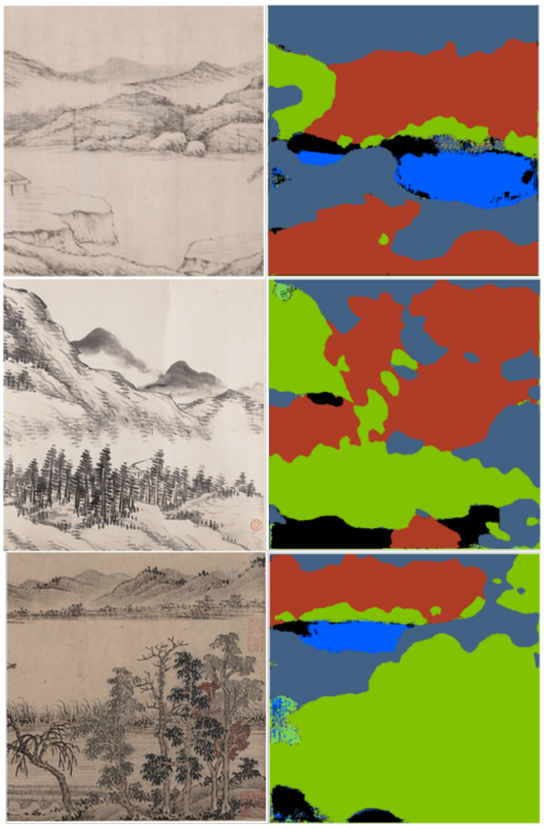

Figure 32: Qualitative results from the Unet model trained on the expanded Bob Ross dataset.

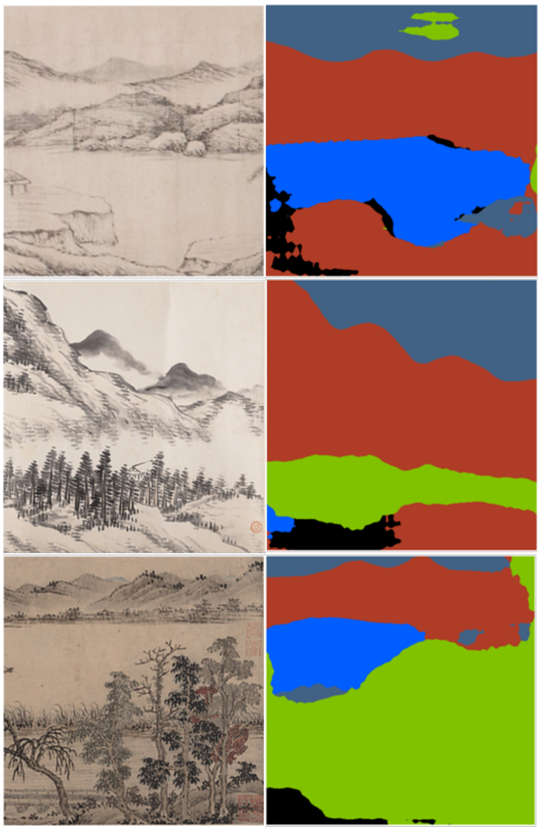

Figure 33: Qualitative results from the DeepLab-V3 model trained on the expanded Bob Ross dataset.

To answer the first research question, the state-of-the-art model DeepLab-V2 was tested directly on Chinese landscape paintings. We also trained two models on our expanded Bob Ross dataset and tested their performance on Chinese paintings. DeepLab-V2 model still fails at the task despite being trained with large-scale dataset. By constructing a training set that is more focused on landscapes, the performance is improved but still not satisfying. Among the selected models, DeepLab-V3 shows the best performance.

In the following subsection, we will show the results obtained from these three models above under different circumstances where we try to improve the performance in order to answer the second research question. As explained above, the three models are DeepLab-V2 trained on 164k COCO-stuff dataset, Unet and DeepLab-V3 trained on the expended Bob Ross paintings dataset. We will eliminate mentioning the training set from here for convenience.

6.2 How to improve the performance on Chinese paintings

6.2.1 Style Transfer

In order to improve the results, we performed style transfer using Cycle GAN to convert the ancient Chinese paintings to landscape photographs. Style transfer gives the paintings more information in terms of color and texture. As we can see from the paintings, ancient Chinese landscapes tend to lack color. Moreover, the boundaries between objects are blurry, especially between water and sky. Both are often shown as a blank space without any texture or color. Therefore, by performing style transfer, we expect an improvement of the model's performance given more information will be available in the images.

The results of style transfer are shown in Figure 34. One big advantage we can gain from style transfer is that the water area and sky area become more distinguished. Also, instead of only ink paint, the images now have different colors for different objects. However, there are also some problems. For example, in the painting on the third row, the mountains in the distance are almost eliminated in the converted image, which leads to failure of recognizing distant mountains for the following segmentation task.

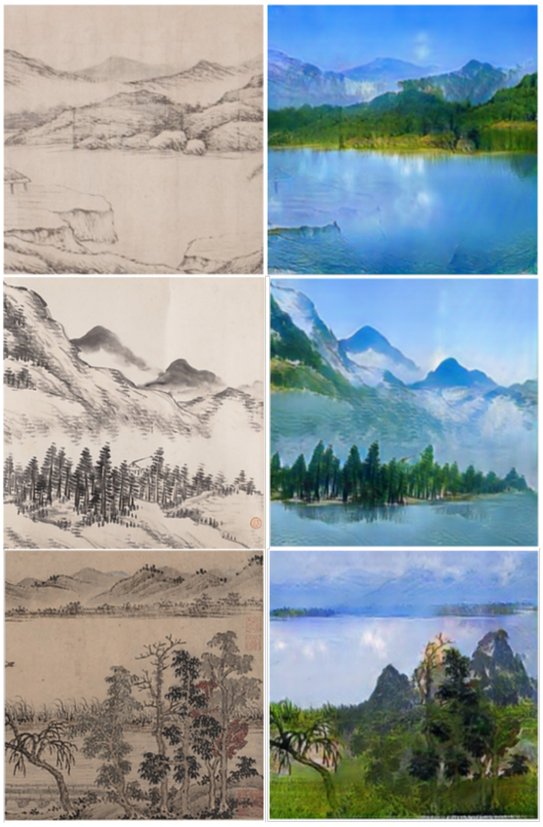

Figure 34: Results from style transfer.

We tested the three models mentioned above on the converted images. The results obtained from Unet model and DeepLab models are shown in Table 4. The results we obtained from applying the models directly on paintings are also shown in the table for comparison (with 'baseline' written in the brackets). For DeepLab-V2 model, water and mountain classes improved by 1% and 3% respectively while the other two classes' accuracy deteriorated. One possible reason for the deterioration is that some paintings contain hand written poems in the sky area. After style transfer, some of the handwriting area was kept as it was in the paintings, resulting in inconsistent texture and color (some examples are shown in Figure 35). For the Unet model, sky and water classes improved by 14% and 10% respectively and the other two classes' performance decreased. For the DeepLab-V3 model, no class shows improvement after the style transfer. Overall, we did not observe significant improvement after performing style transfer.

Model	Sky	Tree	Water	Mountain
Unet	0.43	0.25	0.17	0.21
DeepLab-V2	0.28	0.22	0.17	0.27
DeepLab-V3	**0.49**	0.22	**0.26**	0.18
Unet (baseline)	0.29	**0.31**	0.07	0.35
DeepLab-V2 (baseline)	0.31	0.28	0.16	0.24
DeepLab-V3 (baseline)	0.44	0.30	**0.26**	**0.40**

Table 4: Results from Unet and DeepLab on style-transferred landscapes

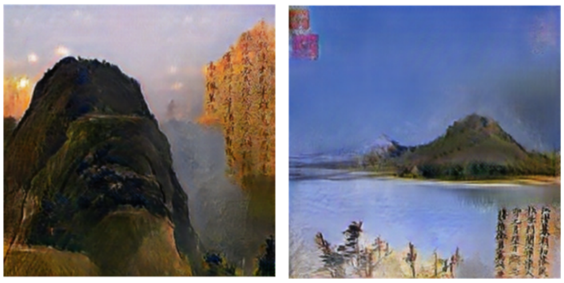

Figure 35: Bad style transfer results caused by an area of text.

Figure 36 shows the comparison of segmentation results before and after style transfer from the DeepLab-V2 model. This is an example of what we would expect for most of the images. However, the quantitative results indicate that there are not enough successful cases like this in our test set. In this example, we can see that the water area is recognized after the style transfer and the mountain is also outlined better.

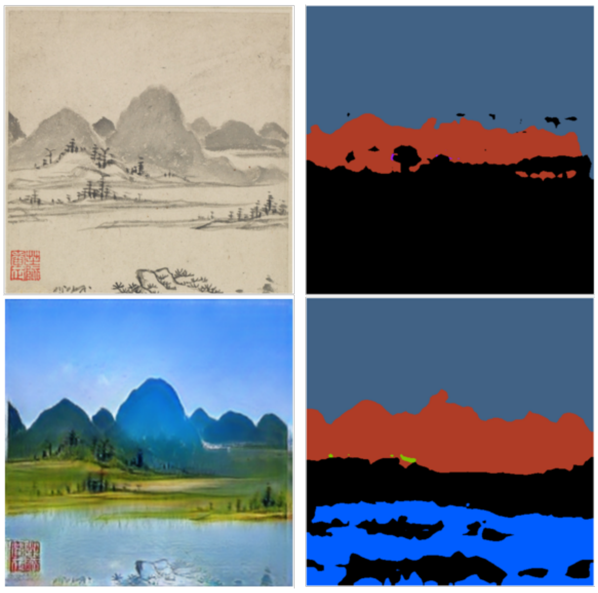

Figure 36: Comparison of segmentation results before and after style transfer from DeepLab-V2 model.

6.2.2 Text Removal

As observed with the style transfer results, the text and stamps which are present in Chinese paintings generate problems for style transfer. Hence, here we test two text detection models, and see which one can frame the text area most accurately. After performing text area detection, we fill the text area with white color and perform image inpainting to fill in the text area with surrounding area pixels.

Figure 37 shows the results we obtained from using MSER to perform the text detection. We can see that the traditional method did not achieve our expectation. Because all the objects including the text are drawn with ink, it is harder to recognize the text area than in natural scene images.

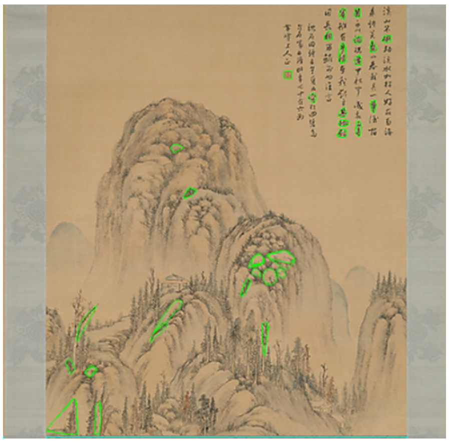

Figure 37: Text detection result obtained from MSER.

After the failure of traditional computer vision methods, we performed the character-region awareness text detection based on neural network. The results are shown in the middle image in each row in Figure 38. We can see that this method can correctly detect the text area. Therefore, we will use this method to continue our study. After the text detection, we performed image inpainting. Because the text tends to be in a blank area, the inpainting can achieve satisfying results. The whole process of text removal is shown in Figure 38.

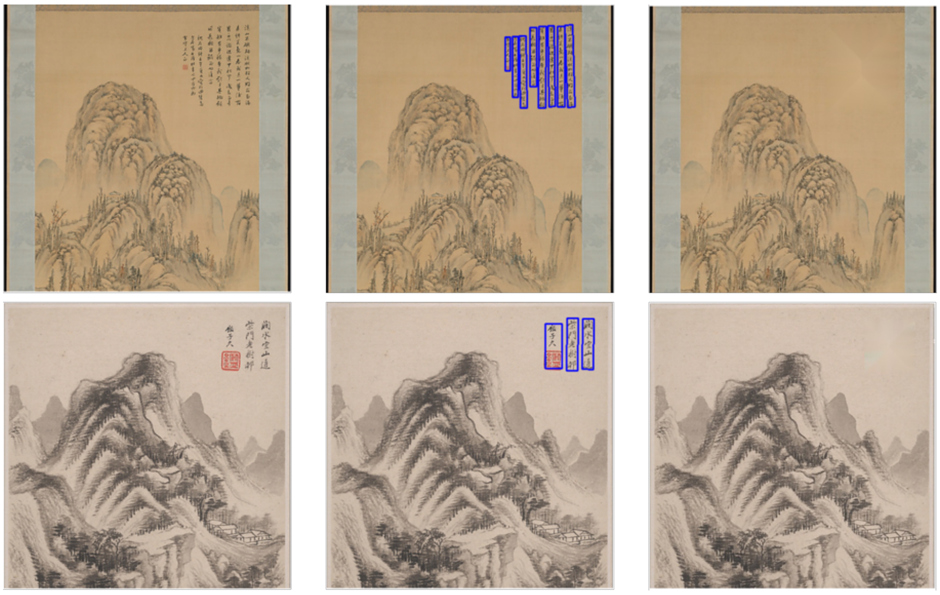

Figure 38: Text removal process.

After the text removal, we perform the style transfer again on the same painting set without text in the paintings. Figure 39 shows the comparison of style transfer results before and after text removal. We can see that in the example, without the text, the water area is transferred with consistent color and texture.

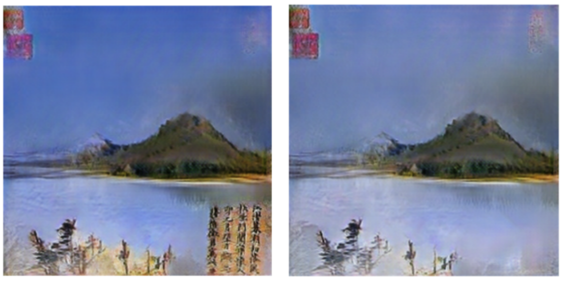

Figure 39: Comparison of style transfer results before and after text removal.

Table 5 shows the results from different models on images converted to photographs after text removal. Comparing to the results on style-transferred paintings before text removal for all three models, the score for sky and tree class are improved. For mountains, DeepLab-V2 and DeepLab-V3 both improved by 1%. We can see that overall, text removal helps all models with the performance compared with performance after style transfer without text removal. Comparing to the baseline, the results for sky and water classes are improved while the other two classes show decreased scores.

Model	Sky	Tree	Water	Mountain
Unet	0.43	0.26	0.17	0.21
DeepLab-V2	0.29	0.23	0.17	0.28
DeepLab-V3	**0.52**	0.23	0.25	0.19
Unet (baseline)	0.29	**0.31**	0.07	0.35
DeepLab-V2 (baseline)	0.31	0.28	0.16	0.24
DeepLab-V3 (baseline)	0.44	0.30	**0.26**	**0.40**

Table 5: Results from Unet and DeepLab after text removal and style transfer.

6.2.3 Elastic Data Augmentation

Mountains are one of the main objects in ancient Chinese landscapes. Therefore, we wanted to further improve the accuracy of the mountain class. We took the DeepLab-V3 model as the model to experiment with because it achieved the highest IoU score in the first test in Section 6.1.2 on Chinese paintings. Comparing Chinese landscapes and Bob Ross paintings, we found that the shape of mountains in Chinese paintings are more diverse than in Bob Ross paintings. For example, Figure 40, shows a Bob Ross painting and two Chinese paintings that are examples of the respective datasets. The style of Bob Ross paintings which are used as training data is not very diverse while in Chinese ancient landscapes, although the color and texture are alike, we can see that there exist different shapes of mountains. We gave only two examples here. We found that the model performs better when the shapes of mountains are similar to those in Bob Ross paintings (such as the one in the middle Figure 40). When the shape of mountains appears different, such as the sharp shape in the painting on the left, the model can not recognize and outline it successfully.

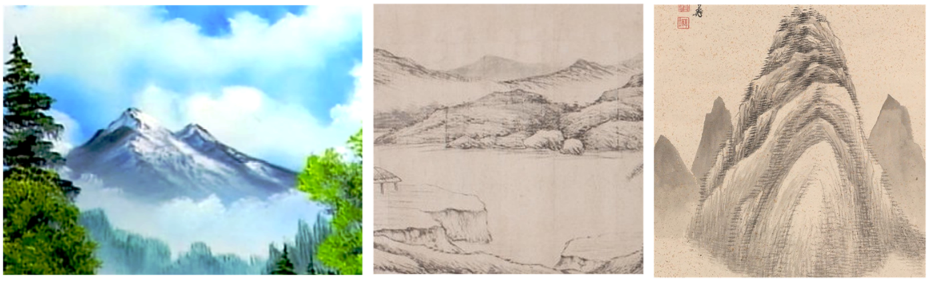

Figure 40: Difference between the shape of mountains in Chinese paintings and Bob Ross paintings.

In order to enrich our training data specifically in terms of the shape of mountains, we added elastic augmentation into the training process. We tested the DeepLab-V3 model on the original Chinese landscape paintings and the results are shown in Table 6. We can see that the mountain class obtained a 1% improvement like we wanted. However, the other classes IoU scores deteriorated.

Sky	Tree	Water	Mountain
0.30	0.29	0.10	0.42

Table 6: Results from Deeplab-V3 model (trained on the Bob Ross dataset) after adding elastic data augmentation.

In conclusion, the combination of text removal and style transfer mainly benefits the water and sky classes due to the additional information added to help distinguish between the two classes.

Adding elastic segmentation improves the accuracy in the mountain class for the reason that it enriches the training data with more diverse shapes of mountains. The style transfer worsens the performance of the mountain class mainly because it sometimes eliminates the mountains in the distant which tent to have less strokes and simpler color. These mountains sometimes get blended in with the sky during the style transfer. The elastic augmentation helps with the mountain class but also causes deterioration in other classes because other classes have relatively fixed and regular shapes. Enriching the training data with different shapes and scales of objects from these classes may confuse the model and worsen the features it learnt for these classes.

6.3 Statistical analysis for Chinese landscapes

In this subsection, we are going to show some statistical results we obtained from our manually annotated test set that contains 100 Chinese landscape paintings.

6.3.1 Composition Proportion

Table 7 shows some statistical results about the average proportions each object takes up in the paintings. The 'Mean' row shows the average proportions each class takes in sample dataset. We can see that the mountain class takes up the biggest proportion and the tree class takes up the smallest proportion. The 'Variance' row shows how stable each class is in terms of the proportions taken in the paintings. We can see that the smallest variance is from tree class.

Statistical Indicators	Mountain	Sky	Water	Tree
Mean	0.31	0.23	0.16	0.14
Variance	0.024	0.017	0.017	0.013

Table 7: Statistical results regarding the proportion of pixels each class takes in the paintings.

6.3.2 Object location

By checking statistically the location information of classes, we will be able to discover some layout patterns in the paintings. For example, during the study, we often found in the paintings that mountains appear mostly above the water. We can verify those observations by performing statistical analysis for the object location. Those patterns can not only help us understand the paintings, but also can be used in future research such as generating Chinese landscape paintings.

In order to find the pattern of the layout in Chinese landscapes paintings, for each class, We will overlay all the segmentation masks that only contain this class. This way, we will find the location where each object appears the most. We first singled out each class and eliminated other classes in the mask. By overlaying all the masks one by one and combining the nine grids we can find in which part of painting each object occurs the most.

The outputs of overlaying all the masks for the mountain, water, tree and sky classes are shown in Figure 41, Figure 42, Figure 43 and Figure 44. The brighter the color is, the more overlap there is. Looking at the most frequently overlaid area for each class, other than the tree class, other classes show a horizontal distribution within in a certain range. This pattern indicates that trees tend to appear in the paintings in a less consistent way. First of all, the trees do not take up a whole and complete area because there are a lot of gaps between trees and between branches. Secondly, trees tend to be scattered across the land or the mountains, which also makes the overlaid area less consistent. We can see in Figure 41 that mountains appear most frequently in the center of the paintings horizontally. For water, we can see in Figure 42 that it mostly appears at the bottom of the painting. For sky, as expected, it appears most often at the top of the painting. By observing the blank area in Figure 42 and Figure 44 for water and sky respectively, the black area in one image is approximately the area in which the other classes appear the most frequently. This observation matches with the impression for Chinese landscapes where the background of the painting tends to be water fading into the sky with a blurry or sometimes non-exist boundary. Also, the layout pattern for the water and mountains verify our observation that mountains are often drawn above water area.

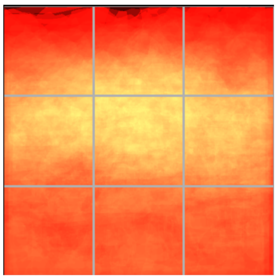

Figure 41: The layout pattern for mountains.

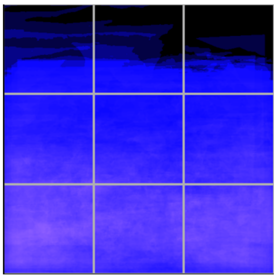

Figure 42: The layout pattern for water.

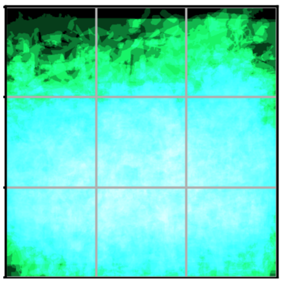

Figure 43: The layout pattern for trees.

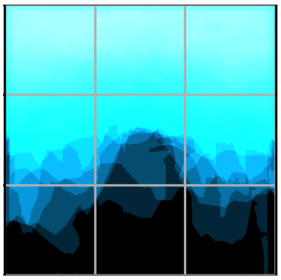

Figure 44: The layout pattern for sky.

7 Discussion

In this section, we will discuss the advantages and disadvantages of each model and how these advantages and disadvantages affect the results. Then, we will discuss how each method we proposed affects the results.

Looking at the results in Table 3, with two models both trained on Bob Ross paintings and applied to Chinese landscape paintings, DeepLab-V3 achieves better quantitative results overall.

Comparing the results obtained from Unet and DeepLab-V3, we observed that the boundaries in DeepLab-V3 predictions are more smooth and accurate than those in the Unet results. One of the examples is shown in Figure 45. The reason why DeepLab-V3 outperforms Unet in terms of producing more accurate boundary information is the use of the Atrous convolution in the downsampling phase. As introduced in Section 5.1, the DeepLab model replaced the downsampling filters with Atrous convolutions, which results in denser feature map extraction. With a higher resolution feature map, DeepLab-V3 can make more accurate predictions in the boundary area when compared to Unet.

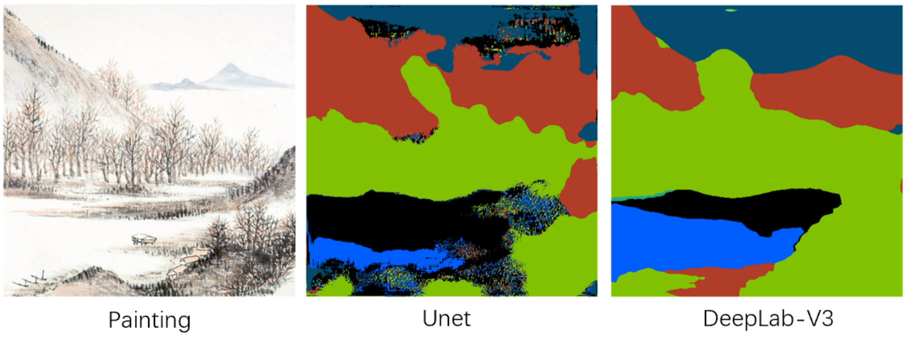

Figure 45: Comparison of results obtained from Unet and DeepLab-V3 (a).

We also observed that there are less noisy color blocks in the results from the DeepLab-V3 model compared to the Unet model, which can also be reflected in Figure 45. In DeepLab-V2, by using CRF as a post-processing step, the little noisy predictions can be assigned to nearby classes resulting in cleaner predictions. However, in DeepLab-V3, they removed the CRF post-processing step while still achieving a better performance than DeepLab-V2. One possible reason is that, as mentioned in Section 5.1.2, the augmented version of ASPP engages image-level feature maps, which not only provides more valid training weights, but also helps the model to understand where the object is, and helps to segment the object more completely. As for Unet, using a few skip connections to connect the high-level features and low-level features did not achieve satisfying segmentation results.

Figure 46 shows the comparison of Unet and DeepLab-V3 in terms of their performance on multi-scale images. In the painting on the first row, the mountain is in the distance, and the whole mountain is contained in the picture. The painting on the second row shows a painting with the mountain in a different scale. The shape and scale of the mountain in the first painting is more similar to the paintings in the training set. Therefore, we can see that on the first row, both models can approximately recognize the mountain. However, in the painting on the second row, we can clearly see that DeepLab-V3 performs better than Unet on the mountain that is more in the foreground of the painting, appearing bigger. This shows that DeepLab-V3 is better at recognizing objects with different scales. This is due to the Atrous Spatial Pyramid Pooling (ASPP) applied in DeepLab model, which is introduced in Section 5.1. The qualitative results here show the advantage of this feature intuitively.

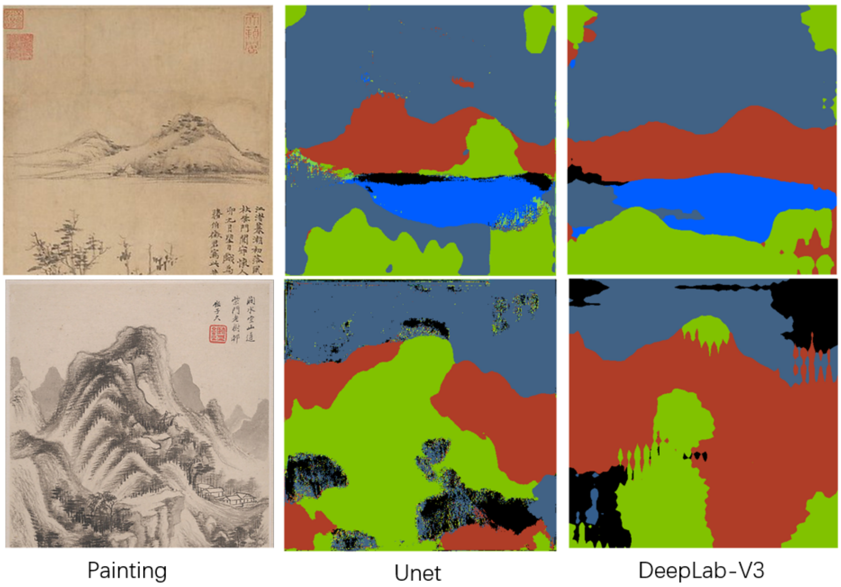

Figure 46: Comparison of results obtained from Unet and DeepLab-V3 (b).

We furthermore experimented with three methods to improve the performance. Although none of the three methods can improve the IoU score for all the classes, each method has their own impact on a different class. This is due to the diversity of the challenges in the task.

Style transfer mostly benefits the sky and water classes. Among the three models we tested, two of them (Unet and DeepLab-V3) showed improvements in the sky class. In addition, two of them (Unet and DeepLab-V2) showed improvements in the water class. This indicates that by performing style transfer in Chinese landscapes paintings, additional information is added to the image. This makes it easier to distinguish between these two classes, therefore improving the IoU score.

Adding text removal before performing the style transfer overall improves the results we obtained from style transfer because it enhances the outputs of style transfer by removing inconsistent parts from the images. Also, another reason why it benefits almost all classes is that the models always assign the text area to a certain class which increases the union area of prediction and ground truth while not increasing the intersection. Therefore, after removing the text area, the IoU score for the class this area should improve and the class which the model wrongly assigns to this area also improves. Since most of the texts are in the sky area, this method benefits the sky class the most and actually achieves the highest IoU score in sky class of 52% from the DeepLab-V3 model.

Adding elastic augmentation to the training procedure proves to be an effective method for the mountain class. For mountains, the diversity in shape is the biggest challenge. Therefore, we performed elastic augmentation as an alternative way of enriching the training data.

Given our observations above, for future study, we can adjust the framework proposed in this work to improve the performance for all the classes together. First of all, we can see that DeepLab-V3 shows great potential for achieving higher accuracy if given more training data. Therefore, we should keep DeepLab-V3 as the prediction model. In terms of the elimination of mountains in the distance while performing style transfer, we can try to add a pre-processing step to sharpen the images before performing style transfer. For the mountains, we would like to add more diverse shapes without deforming other objects. To achieve this, we can use synthetic training data set. The synthesis process can be performed as follows: First, take some cropped mountain components from the manually annotated Chinese landscape paintings. Then create a simple general background with only the land or water and sky. Then imitate the elastic data augmentation step by deforming the mountains randomly and put one mountain on each background. This way, we enrich the diversity of the shapes of mountains while keeping the remaining part untouched. Figure 47 shows the workflow of this adjusted method.

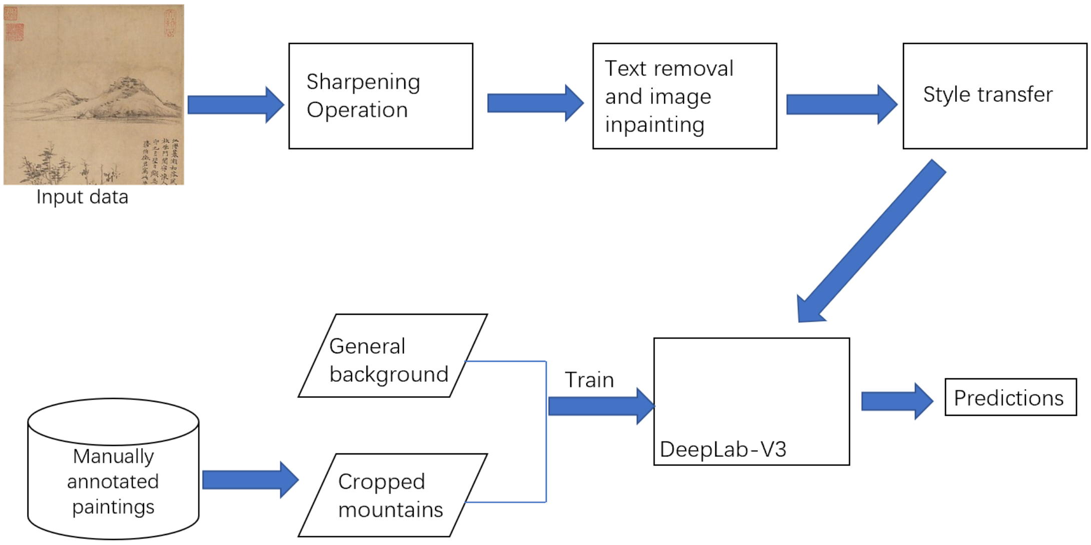

Figure 47: Adjusted pipeline for improving all classes at once.

In summary, style transfer is not the cure for all the classes. By observing where the models tend to fail during the experiments, we found that different classes have their own challenges. The combination of text removal and style transfer helps to improve the performance in the sky and water class because it makes the two classes more recognizable by adding color and texture to them. However, this method did not show any advantage for the mountain class. In Chinese ancient landscape paintings, the mountains in the distance often appear to have very light color and simple strokes. After the style transfer, mountains in the distance sometimes got blended into the sky therefore leading to the deterioration in the performance of the model. Through observation of the results from the DeepLab-V3 model which achieves highest IoU score in mountain class, we found that that model can actually recognize the mountains correctly when the shape is similar to the mountains in training data which are mostly not very steep. Therefore, adding elastic augmentation is an effective method of helping the model recognize mountains in different shapes.

8 Conclusion

In this work, we explored three widely used segmentation models, Unet, DeepLab-V2 and DeepLab-V3 and tested their performance on Chinese ancient landscapes paintings. To test the baseline performance, we first tested the DeepLab-V2 model trained on the 164k COCO-stuff dataset, which contains a large amount of images. However, these images are very different from Chinese paintings. In order to improve the performance, we further trained the DeepLab-V3 and Unet models on more similar and focused training data, which consists of Bob Ross paintings and some mountain and tree images from the COCO-stuff dataset. With the trained model, we continued with experimenting different methods on the Chinese paintings including style transfer, data augmentation and text removal. We followed the experiments with a discussion of the comparison of the used models for Chinese landscapes. In order to support further work and also provide more insights of Chinese landscapes, we performed statistical analysis on a small test set with manually annotated segmentation masks. We did not include the painting generation part in this study because the synthesis of images needed a lot of computational resources and we did not have enough time and resources for this task.

To conclude, after exploring three models (Unet, DeepLab-V2 and DeepLab-V3) on semantic segmentation task on Chinese ancient landscape paintings, we found that by directly applying models on Chinese paintings, DeepLab-V3 shows the most potential to achieve higher accuracy if given more training data. This also shows that this model can still recognize the objects when the training data and test data have different styles, which offers insights for other computer vision tasks where matching training data is not available. When observing qualitative results, DeepLab-V3 also shows better performance in terms of multi-sclae object recognition and producing less noisy segmentation masks.

Various methods were applied to improve the performance. Different methods show advantages in different classes. The combination of text removal and style transfer mainly benefits water and sky classes due to the additional information added to help distinguish between the two classes. Adding elastic segmentation improves the accuracy in mountain class for the reason that it enriches the training data with more diverse shapes of mountains.

In the end, based on our annotated test set, the statistical analysis shows some patterns of objects' most frequent locations. The mountains appear most frequently in the middle area of the paintings while trees and water appear mainly in the lower bottom area.

For future research, first of all, increasing the test set with annotations would help with observing more significant quantitative results while experimenting with different methods. In our study, due to the limited resources, we only had 100 manually annotated paintings to be used as the test set. Furthermore, in terms of style transfer, different cycle GAN models can be explored such as pixel2pixel [12]. Also, using training data with higher resolution may have an effect on producing better style transfer results.

References

[1] Bruno Artacho and Andreas Savakis. Waterfall atrous spatial pooling architecture for efficient semantic segmentation. *Sensors*, 19(24), 2019.

[2] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(12):2481–2495, 2017.

[3] Youngmin Baek, Bado Lee, Dongyoon Han, Sangdoo Yun, and Hwalsuk Lee. Character region awareness for text detection. *CoRR*, abs/1904.01941, 2019.

[4] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Coco-stuff: Thing and stuff classes in context. In *Computer vision and pattern recognition (CVPR), 2018 IEEE conference on*. IEEE, 2018.

[5] Alex J Champandard. Semantic style transfer and turning two-bit doodles into fine artworks. *arXiv preprint arXiv:1603.01768*, 2016.

[6] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *CoRR*, abs/1606.00915, 2016.

[7] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *CoRR*, abs/1706.05587, 2017.

[8] DeepMind's. Generative adversarial networks (gans). *Medium*, 2019.

[9] Serena Yeung Fei-Fei Li, Justin Johnsond. Lecture 11:detection and segmentation. 2017.

[10] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2414–2423, 2016.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *CoRR*, abs/1406.4729, 2014.

[12] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. *CoRR*, abs/1611.07004, 2016.

[13] Victor H. Jimenez-Arredondo, Jonathan Cepeda-Negrete, and Raul E. Sanchez-Yanez. Multilevel color transfer on images for providing an artistic sight of the world. *IEEE Access*, 5:15390–15399, 2017.

[14] Jonghoon Jin, Aysegul Dundar, and Eugenio Culurciello. Flattened convolutional neural networks for feedforward acceleration. *arXiv preprint arXiv:1412.5474*, 2014.

[15] Levent Karacan, Zeynep Akata, Aykut Erdem, and Erkut Erdem. Learning to generate images of outdoor scenes from attributes and semantic layouts. *CoRR*, abs/1612.00215, 2016.

[16] Uma B. Karanje, R. Dagade, and S. Shiravale. Maximally stable extremal region approach for accurate text detection in natural scene images. 2016.

[17] Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton. Imagenet classification with deep convolutional neural networks. *Neural Information Processing Systems*, 25, 01 2012.

[18] Jinning Li and Yexiang Xue. Scribble-to-painting transformation with multi-task generative adversarial networks. In *IJCAI*, pages 5916–5922, 2019.

[19] Daoyu Lin, Yang Wang, Guangluan Xu, Jun Yu Li, and K. Fu. Transform a simple sketch to a chinese painting by a multiscale deep neural network. *Algorithms*, 11:4, 2018.

[20] Yusuke Niitani, Toru Ogawa, Shunta Saito, and Masaki Saito. Chainercv: a library for deep learning in computer vision. In *ACM Multimedia*, 2017.

[21] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2337–2346, 2019.

[22] Adam Paszke, Abhishek Chaurasia, Sangpil Kim, and Eugenio Culurciello. Enet: A deep neural network architecture for real-time semantic segmentation. *arXiv preprint arXiv:1606.02147*, 2016.

[23] T. Qiao, W. Zhang, M. Zhang, Z. Ma, and D. Xu. Ancient painting to natural image: A new solution for painting processing. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 521–530, 2019.

[24] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing.

[25] Patsorn Sangkloy, Jingwan Lu, Chen Fang, Fisher Yu, and James Hays. Scribbler: Controlling deep image synthesis with sketch and color. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5400–5409, 2017.

[26] E. Shelhamer, J. Long, and T. Darrell. Fully convolutional networks for semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(4):640–651, 2017.

[27] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[28] Wei Ren Tan, Chee Seng Chan, Hernán E Aguirre, and Kiyoshi Tanaka. Artgan: Artwork synthesis with conditional categorical gans. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 3760–3764. IEEE, 2017.

[29] Alexandru Telea. An image inpainting technique based on the fast marching method. *Journal of Graphics Tools*, 9, 01 2004.

[30] Ekin Tiu. Metrics to evaluate your semantic segmentation model. medium, towards data science. *IEEE Access*, 2020.

[31] Yuan Wang, Weibo Zhang, and Peng Chen. Chinastyle: A mask-aware generative adversarial network for chinese traditional image translation. In *SIGGRAPH Asia 2019 Technical Briefs*, SA '19, page 5–8, New York, NY, USA, 2019. Association for Computing Machinery.

[32] Alice Xue. End-to-end chinese landscape painting creation using generative adversarial networks. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 3863–3871, January 2021.

[33] Jordan Yaniv, Yael Newman, and Ariel Shamir. The face of art: Landmark detection and geometric style in portraits. *ACM Trans. Graph.*, 38(4), July 2019.

[34] Cong Yao, Xiang Bai, Wenyu Liu, Yi Ma, and Zhuowen Tu. Detecting texts of arbitrary orientations in natural images. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1083–1090, 2012.

[35] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Free-form image inpainting with gated convolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4471–4480, 2019.

[36] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *CoRR*, abs/1703.10593, 2017.