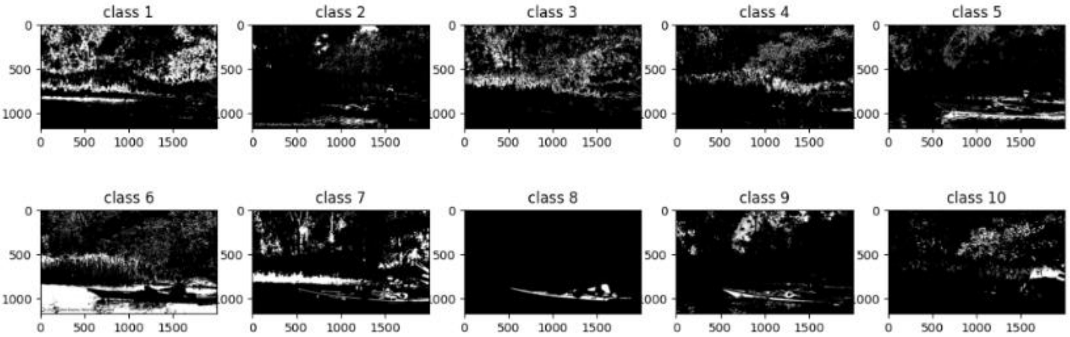
1. Conventional Approach for Segmentation:

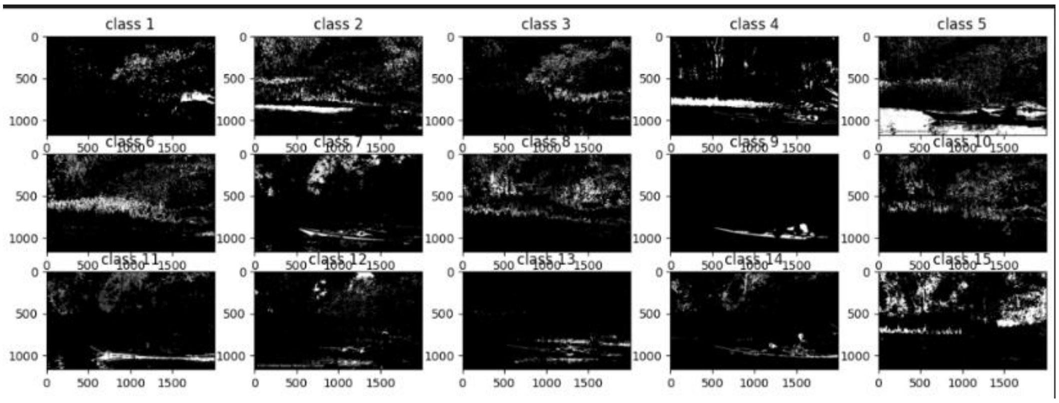
● K-means based segmentation

K-means is a simple method used to complement image segmentation. The formula for distance and the value of k can lead to different segmentation results. Because we recognize that the distance calculated by RGB cannot directly represent the similarity of two colors, we choose to use HSV to calculate the distance. Additionally, we examine different values of k and observe the results as follows.

k = 10

 class 1
class 2
class 3
class 4
class 5
class 6
class 7
class 8
class 9
class 10
0
500
1000
1500

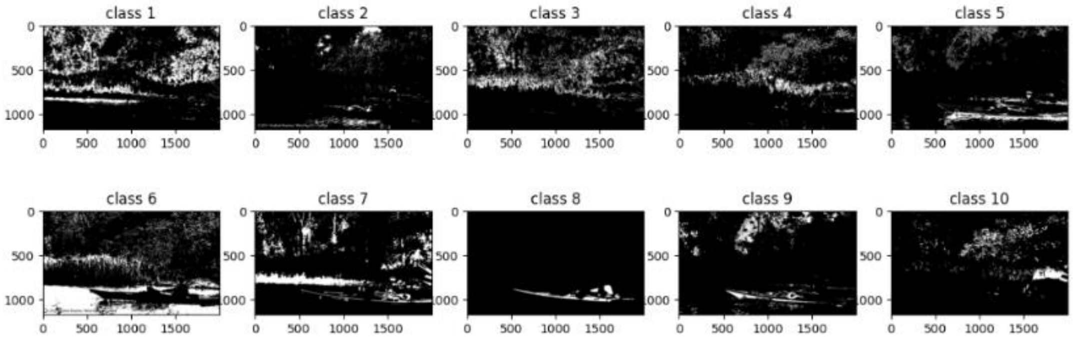
k = 15

 class 1
class 2
class 3
class 4
class 5
class 6
class 7
class 8
class 9
class 10
class 11
class 12
class 13
class 14
class 15
0
500
1000
1500

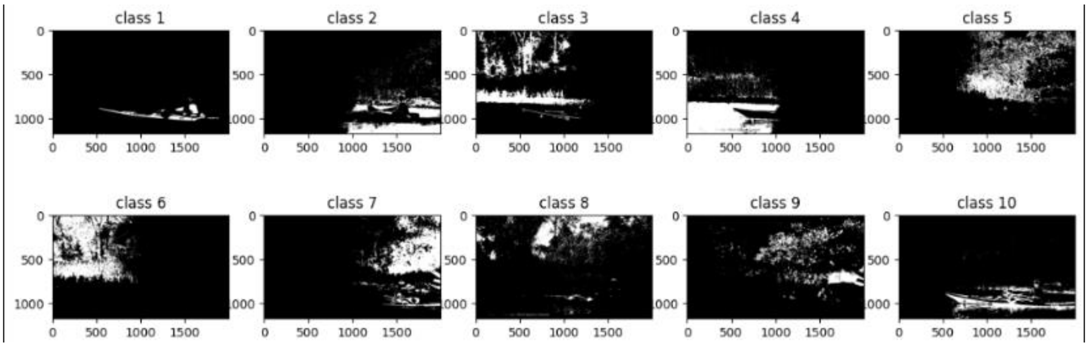
According to the results, the larger segments are almost unaffected by the value of k, while the smaller segments become tinier as k increases. We assume that under the condition that each segment can maintain its own characteristics, larger segments are preferable. Segmentation will be beneficial for subsequent identification. In all subsequent experiments, we consistently choose to use k = 10.

To enhance the aggregation and increase the size of each segment, we also attempted to incorporate spatial information. As anticipated, we observed that the distribution of each segment became more concentrated, reducing the influence of distant noise.

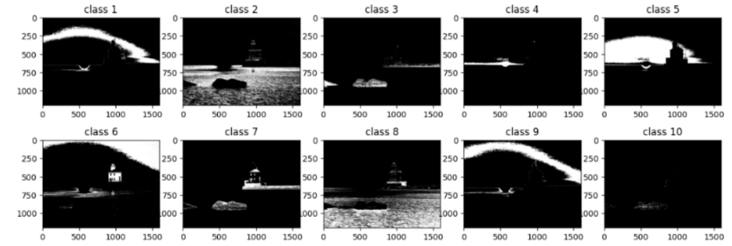
HSV without spatial domain

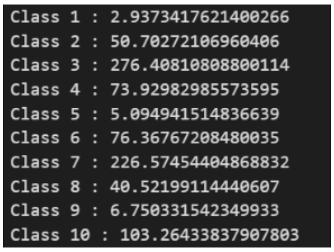


HSV with spatial domain



After achieving a well-defined segmentation, the next step is to identify whether each segmented block corresponds to water. We referred to multiple papers, and one of them, denoted as [1], introduced the concept of hue invariance. The paper asserted that water exhibits hue invariance, meaning its hue variance is relatively low compared to other objects. Consequently, we computed the average hue variance within each segment. However, we encountered challenges as the relationship between water and hue variance varied significantly across different types and variations of water bodies. Consequently, the calculated hue variance values were not consistently indicative of water presence. The results below illustrate this issue, where both Class 2 and Class 8 represent water segments, yet their hue variances do not exhibit substantial differences. Conversely, the variance in the sky's hue is very low, potentially leading to misjudgments.





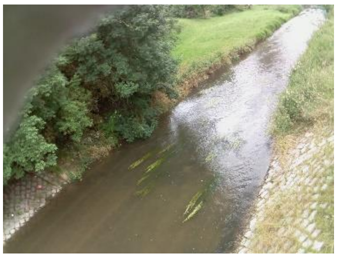
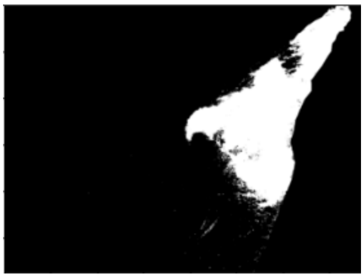
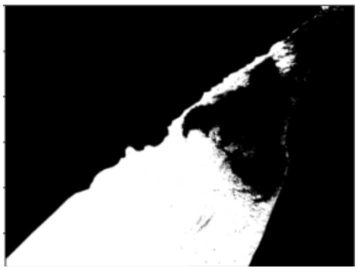
```
Class 1 : 2.9373417621400266
Class 2 : 50.70272106960406
Class 3 : 276.40810808800114
Class 4 : 73.92982985573595
Class 5 : 5.094941514836639
Class 6 : 76.36767208480035
Class 7 : 226.57454404868832
Class 8 : 40.52199114440607
Class 9 : 6.750331542349933
Class 10 : 103.26433837907803
```

In our continued exploration, we investigated Local Binary Patterns (LBP) as a potential texture feature for discerning water body types. LBP is renowned for its ability to identify different water body categories and has been employed successfully in water recognition applications. However, we encountered challenges when applying LBP features to our dataset. Notably, for Class 1, representing a water body, the LBP values fell within the mid-range across all classifications, indicating a lack of distinct texture patterns.

Both hue variance and LBP proved inconclusive in pinpointing water bodies within our dataset. Consequently, we are considering an alternative strategy that involves leveraging statistical data to establish the HSV (Hue, Saturation, Value) ranges indicative of water bodies. This statistical approach aims to provide a more robust framework for confirming the identification of water bodies.

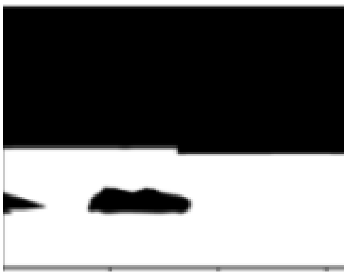
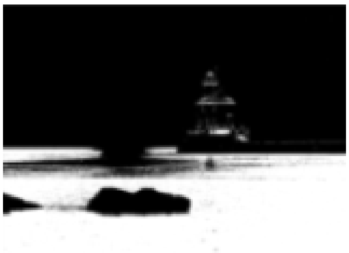
When processing each image in the training dataset, we opted to take the union of the image and the corresponding mask, followed by calculating the average HSV value of the union region as a representative point for the water body space. However, we found that a direct averaging approach might pose some challenges.

Taking one image as an example, the water body is evidently divided into two parts, namely the darker clear water area and the brighter reflective area. A direct average of the entire region could yield colors not present in either area, leading to inaccuracies in the statistical representation of water body features. Therefore, our solution involves applying k=2 K-means clustering to the intersected region, calculating the average HSV values for each category, and incorporating both averages into the water body features. This method contributes to a more accurate capture of distinct features in different water body regions.

Original image	Segment 1	Segment 2
		

Considering potential discrepancies between the average HSV values obtained through K-Means segmentation and the ideal values, we further explored the implementation results during simulated testing for additional insights. Initially, we applied K-Means segmentation to the images and identified the segment with the maximum intersection with the mask as the representative water body. Subsequently, we calculated the average HSV values for all pixels within this segment and incorporated them into the water body features.

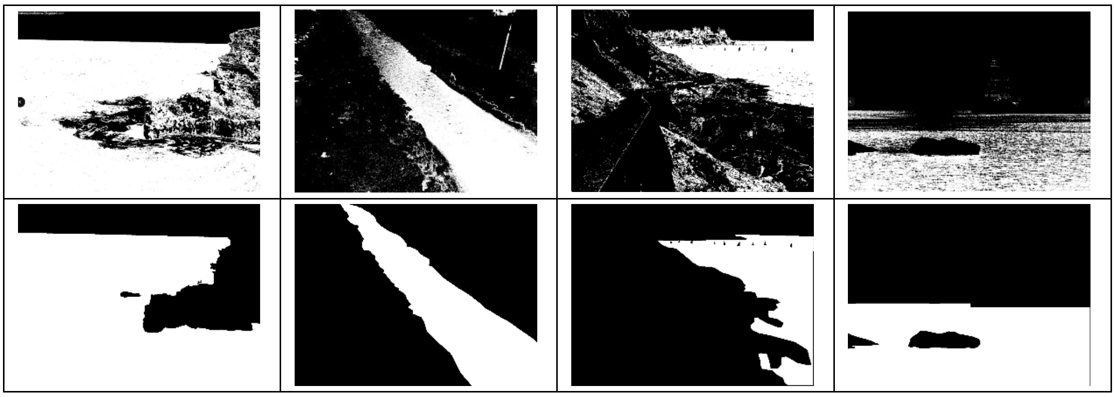
While this approach might marginally reduce the accuracy of water body features, given the inherent ambiguity in K-Means segmentation, this reduction in accuracy aims to enhance recognition tolerance. We believe that such experimentation proves beneficial for the overall results.

Original image	Original Mask	Max Intersection Category
		

After collecting the water body HSV features, the next step involves determining whether a target segment corresponds to water. We represent the collected HSV values as points in a three-dimensional space, specifically the HSV color space. Subsequently, we compute the minimum convex hull that encompasses all these points.

Upon obtaining the average HSV for the target segment, we place this calculated HSV point in the same spatial configuration and check if it lies within the convex hull. If the point falls inside the convex hull, it indicates similarity to the statistical data of past water bodies, leading to a classification of the segment as a water body. Conversely, if the point is outside the convex hull, the segment is deemed non-aqueous. This methodology leverages spatial relationships in the HSV color space to determine the resemblance of a target segment to previously observed water body statistics.

Here are the results obtained from our K-Means segmentation. It is evident that the water bodies are generally well-segmented and included. As for the small white dots generated by K-Means, we will attempt to eliminate them using alternative methods, and the specific procedures for handling them will be elaborated in the subsequent content.



Gaussian Mixture Model based segmentation:

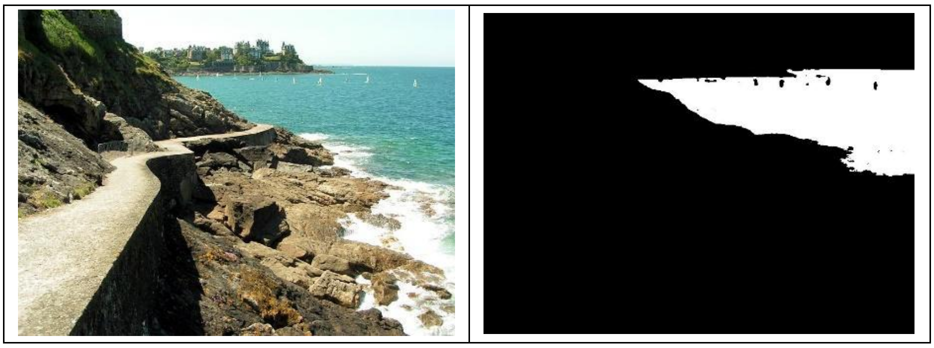
Gaussian Mixture Model is a cluster-based segmentation method similar to the K-means method. Unlike the K-means algorithm, GMM states that the distribution of image pixels stems from N Gaussian distribution random variables. By this assumption, GMM become a more complex and flexible segmentation tool than K-means.

In order to use GMM algorithm to segment the image, we first need to decide in what domain, with what coordinates should we perform GMM algorithm. While GMM is said to be conducted on HSV domain in the PowerPoint in the presentation video, we actually also tried different domain to perform GMM method. Common domains like RGB, RGB+XY, HSV+XY etc. are also tested to see the segmentation results of the image. By observing whether a specific set of clusters can be combined to form a good coverage of water body in the image, we find that the performance of this cluster-based segmentation varies from case to case. For example, RGB domain GMM may result in bad segmentation when the color of the water body is close to that of the background object. If the raw clustering result is not good enough, the performance of the final label will be limited.

After segmenting the image, we need to choose which segments includes the water body in the image. This is where we use some of the characteristic of the water to select the desired clusters. As [1] states, the Hue variance of the water pixels in their nearby region is relatively small compared to other parts of the image. Thus, we utilize this property as an indicator for water cluster. First, we calculate the mean Hue variance of the pixels included in the clusters. Next, we use a predefined threshold to select the clusters. If a cluster has a mean Hue variance less than the threshold value, the cluster is classified as water segment and the pixels in the cluster will be added to the label. In all, the process of water segmentation by GMM method is as follow:

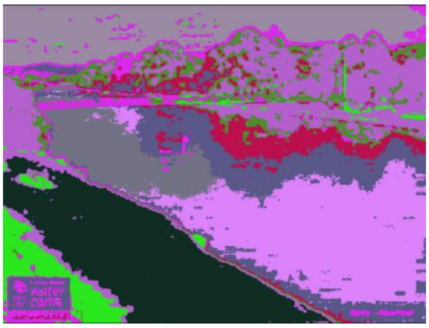
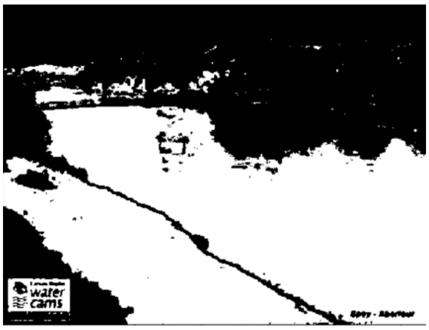
1. Decide a domain you want your GMM to operate on (RGB, HSV, x-y, …)
2. Perform GMM algorithm on the specified domain
3. Calculate the mean Hue variance of each cluster
4. Collect the pixels in the clusters that have a mean Hue variance lower than the threshold and add them to the final label

One of the resulted labelling:



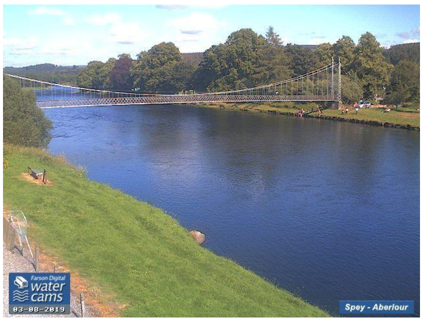
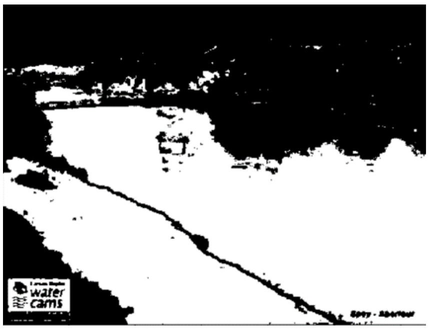
However, there are still problems when using this method:

1. Imperfect clustering: When using K-mean or GMM method on common domain like RGB or HSV, we often see that some of the clusters contain not only part of the water body in the image but also other irrelevant part of the image (trees, houses, …). When these clusters appear, we are forced to face the dilemma: If we select the clusters into the final label, the labelled region will contain the non-water part of the image; if we don't select these regions into the final label, we can not fully cover all the water body in the image. In short, the imperfect clustering limit the maximum performance we can obtain by using these clustering-based methods.

clustering result	final label
	

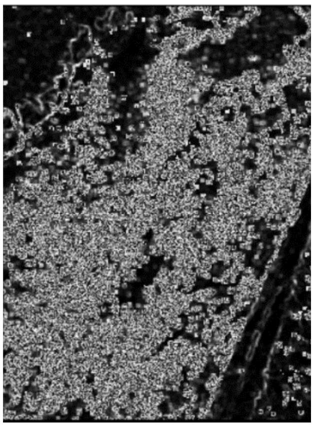
Possible solution: This imperfect clustering comes from the fact that the domain we perform GMM on cannot separate the distribution of water from that of non-water part. One of the solutions to this problem can be choosing a domain that is more complex and have features that can distinguish water from other non-water objects or background (like using texture features etc.). This way, the clustering-based methods are able to form clusters that purely contain water and increase the upper limit of the segmentation IOU.

2. Flat surfaces: In some of the images, surfaces like lawn or clear sky may exist. These surfaces also have the characteristic of low Hue variance. When doing thresholding, these clusters may incorrectly be included in the final label, creating a result of mis-labeling.

Image	Final label
	

Possible solution: One of the solutions to this problem can be adding other features to select the water body clusters. If these flat surfaces are different from the water body on these addition features, we can select exclusively the water part of the image and reduce the error labeling of the process.

3. Threshold selection: Often, the water body in the image is not still but has waves on it. These waves will increase the mean Hue variance of the water segments. As the threshold to select the clusters of water fixed at the beginning of the process, the true water clusters may be eliminated from the final label.
   Possible improvement: The result may be improved when finer GMM is conducted so that water with waves is separated from the still water parts. While we still cannot correctly label the waves of water, the still water body can be correctly selected into the final label and improve the overall IOU of the segmentation.

Image	Hue variance
	

Conclusion: While this method may produce relatively good results on some of the image, its performance greatly depends on the environment of the image. The overly simple algorithm results in poor labeling when the images processed don't met our expectation. Despite having weak points when segmenting, many advanced skills or features can be added to this algorithm to improve the overall performance of the method.

2. Deep learning for Segmentation:

Model:

- U-Net: Convolutional Networks for Biomedical Image Segmentation.
- Uformer: U-shape network with Swin-Transformer architecture.

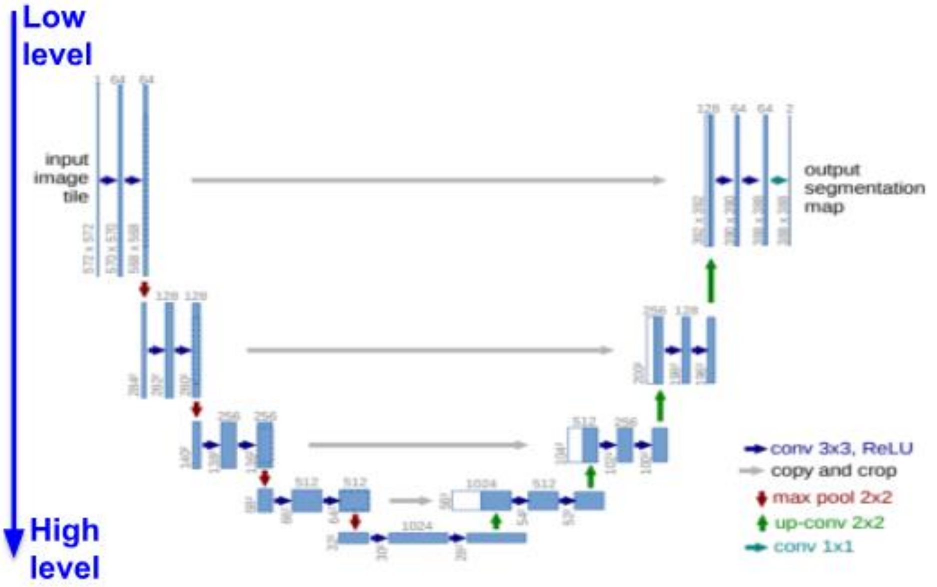
Why we choose these two model architectures?

1. UNet:

UNet is a famous architecture since 2015, which was used to segment the Biomedical image. UNet's structure has the following features which make it popular in doing segmentation task:

1. Auto Encoder-Decoder structure
2. Multiple levels of receptive field
3. Simple structure to integrate different building blocks

For the first point, encoder part is believed to learn the input features and the decoder part needs to integrate the information from encoder and skip connection to decode the context information of the input image and then complete the segmentation output map. For the second point, with the image down sample mechanism, the feature map of each level has different receptive field. Hence, the model can learn the latent information from low level (texture, angle, color…) to high level (context, object…). For the third point, the structure of the UNet is the basic building block with down sample and up sample and skip connection modules. This makes it flexible to construct different building block which can enhance the model ability.



2. Uformer:

Uformer is one of the UNet structure with Swin-transformer layer as basic building block. Since 2022, Swin transformer reach SOTA in many computer vision benchmarks. Swin transformer takes advantage of the CNN (locality) and Transformer (Long term dependency).

Uformer is first proposed to deal with image restoration task such as image denoising and image deblurring, which reach the SOTA on image denoising task in 2022. We modify the model structure to meet the segmentation requirement, furthermore, we reduce the model size and simplify the architecture to make the fare comparation with vanilla UNet.

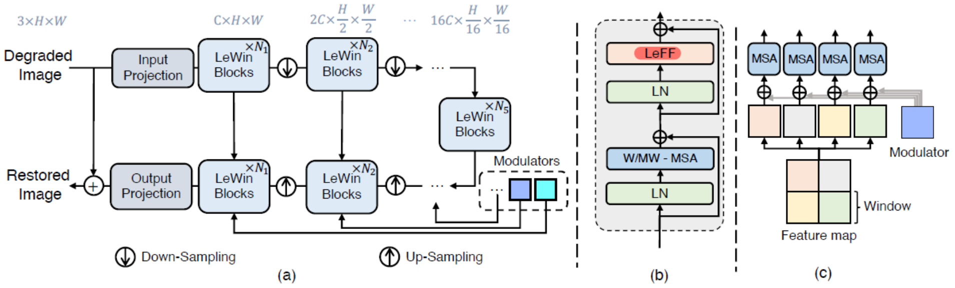
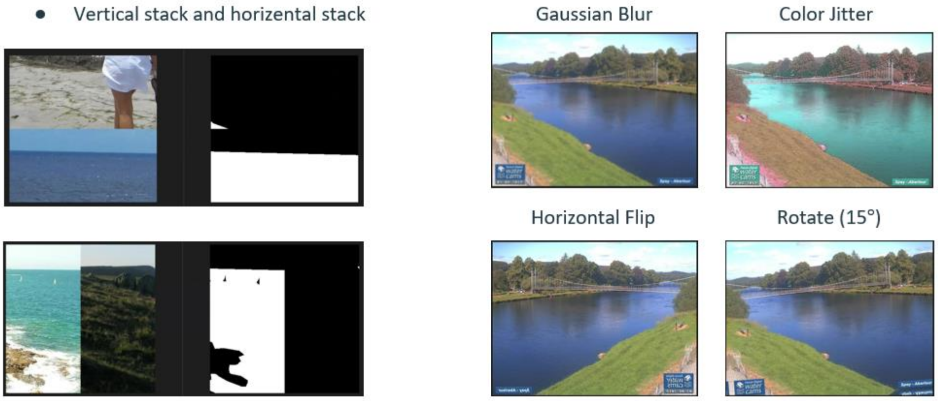


Figure 2. (a) Overview of Uformer. (b) LeWin Transformer block. (c) Illustration of how the modulators modulate the W-MSAs in each LeWin Transformer block which is named MW-MSA in (b).

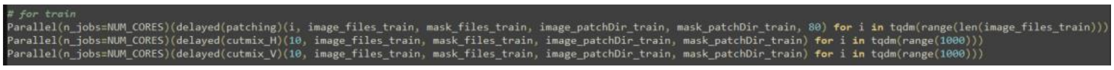
Dataset:

- Slicing 60 images into numbers of 384x384 size patches as training data.
- Random cut two images into 384x192 patches and mix them together in horizontal or vertical directions.
- Data augmentations for training such as GaussianBlur, Rotate, HorizontalFlip, Colorjitter.

Since the dataset for training is limited to 60 images, we want to enlarge the training data to make the deep learning method result more effectively. Hence, we choose to cut the origin images into many 384x384 patches. Due to the limited computation resource, we choose 384 rather than 512. As for the lower resolution such as 224 or 256, the IOU results on the testing set do not outperform the 384 one.



Finally, we augment the training dataset to 24800 images.



```
# for train
Parallel(n_jobs=NUM_CORES)(delayed(patching)(i, image_files_train, mask_files_train, image_patchDir_train, mask_patchDir_train, 80) for i in tqdm(range(len(image_files_train)))
Parallel(n_jobs=NUM_CORES)(delayed(cutmix_H)(10, image_files_train, mask_files_train, image_patchDir_train, mask_patchDir_train) for i in tqdm(range(1000)))
Parallel(n_jobs=NUM_CORES)(delayed(cutmix_V)(10, image_files_train, mask_files_train, image_patchDir_train, mask_patchDir_train) for i in tqdm(range(1000)))
```

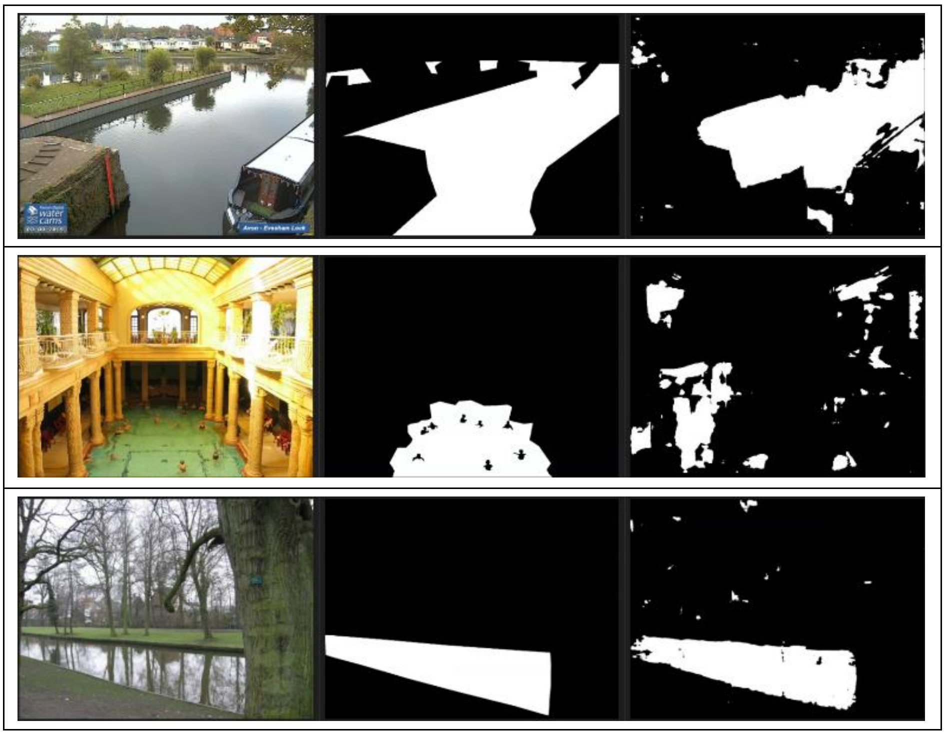
Experiment:

The following table is the IOU performance of the UNet and Uformer trained on the 256x256 or 384x384 patch sizes testing result.

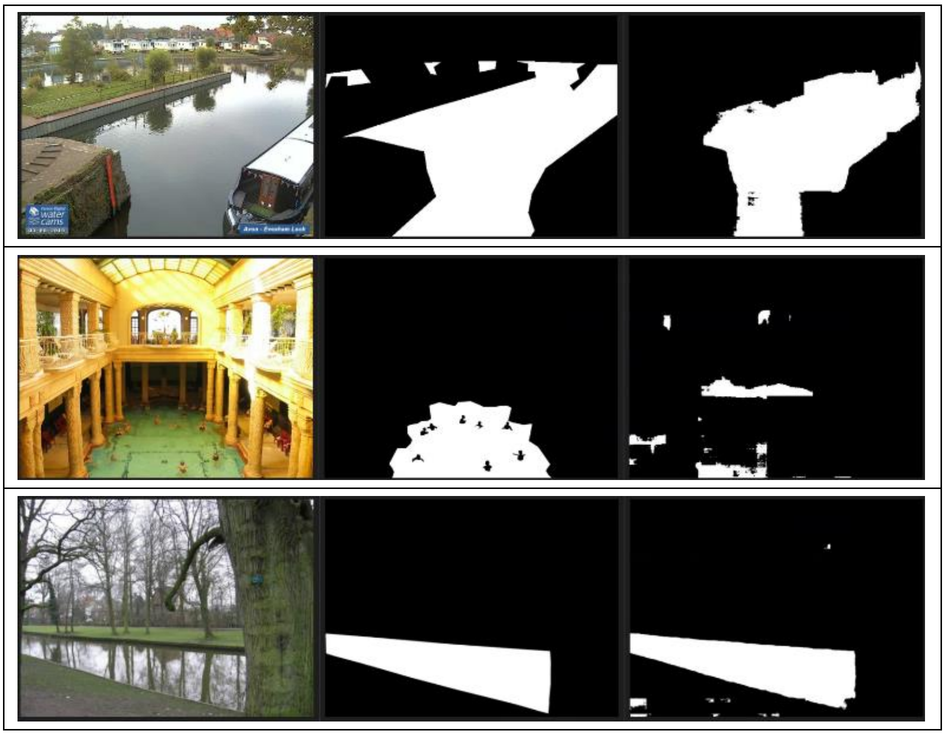
Patch size	256	384
UNet	0.587	0.651

Uformer	0.645	0.702

UNet testing result:



Uformer testing result:



The limitation of the DL method is apparently. If we test on the image that has not be seen or similar to the training data like the second testing data above, the inference result will not perform well.

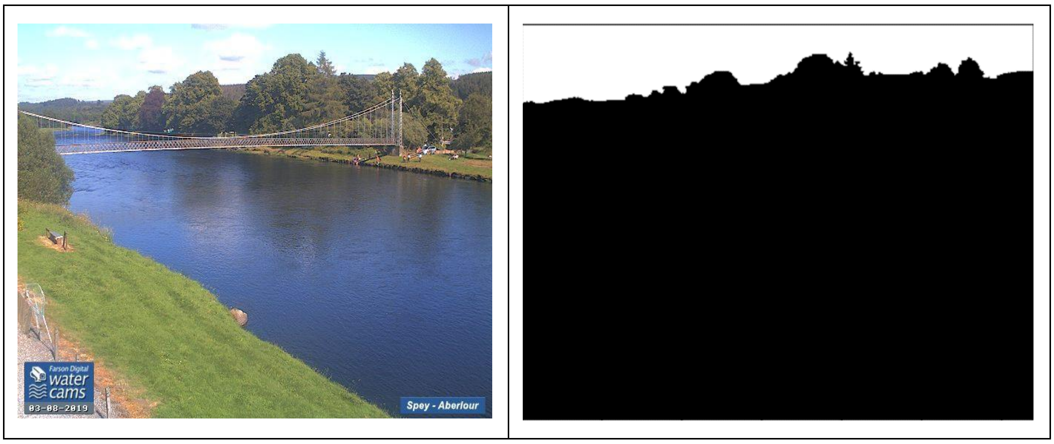
3. Post-processing:

- Sky elimination:

During the implementation and testing of our segmentation methods, we found that part of the sky will be mis-labelled as water. This is because that sky and water body often have some common feature in specific environment. For example, clear sky and still water both have a flat texture. Methods which consider the texture feature of the image may be unable to distinguish between sky and water. To deal with this problem, we try to segment the sky in the image and remove it from the label of the previous methods. The method of finding the sky segment is based on the assumptions that (1) pixels in the sky has relatively high brightness (or lightness) among the entire image; (2) sky are situated at the top of the image; (3) there exists a border (objects like trees or buildings) between the segment of water and segment of sky in the image. With these assumptions, [1] derive the process to obtain the segmentation of sky as follow:

1. Calculate the lightness of each pixel in the image.
2. Use thresholding method to separate the image to high-lightness part and low-lightness part.
3. Acquire seed(marker) from the high-lightness pixels on the top side of the image.
4. Use reconstruction by dilation method (where the mask are the pixels in high-lightness part of the image) to generate the sky segmentation.

By testing result, we observe that this algorithm can actually capture the sky in the image.



However, this method has some weak points:

1. Dark environment: When that weather or time condition in the image makes the entire environment darker than we expected, the predefined and fixed threshold may not be able to distinguish the sky from other part of the image.

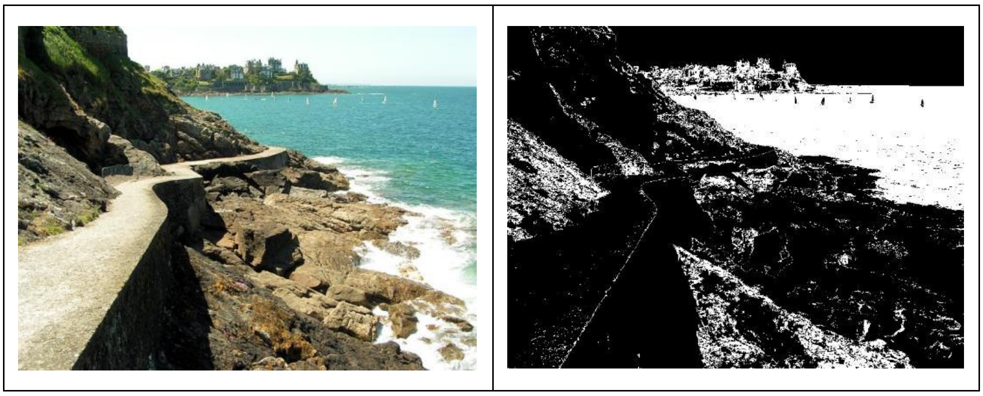
   Possible improvement: We can use a dynamic threshold that changes according to overall lightness condition of the image rather than a fixed one. This way, the adaptive threshold might help us find out the relative brighter spots that includes the sky segment. This solution, nonetheless, can’t deal with the image that has light sources other than the sun in the environment.

2. Connected sky and water segments: When there is no border between the sky and the water in the image, this method may see the water body as a part of the sky due to the fact that water body sometimes also has a relatively high lightness in the image. When this happened, correct water body labels may be eliminated from the final labeling result.

   Compensation we make: After eliminates the sky segment from the raw labeling result, we will conduct a reconstruction of dilation using the sky-eliminated image as marker and raw labeling result as the mask. As long as some of the water body is not eliminated, we can regenerate the eliminated water body as well as the mis-labelled sky in the image (if not using early stop). This compensation form a trade-off between mis-label of sky segment and the miss label of the water body.

- Noise spots elimination:

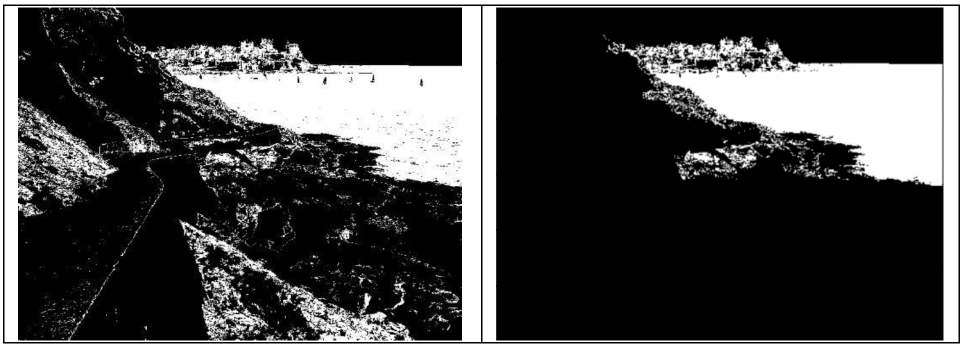
During the segmentation, we often find white or black noise spots appear in the result labeling, just like the below image.



To eliminates this noise, morphological operations are conducted to the image to reach an IOU enhancement effect. The operations we do are:

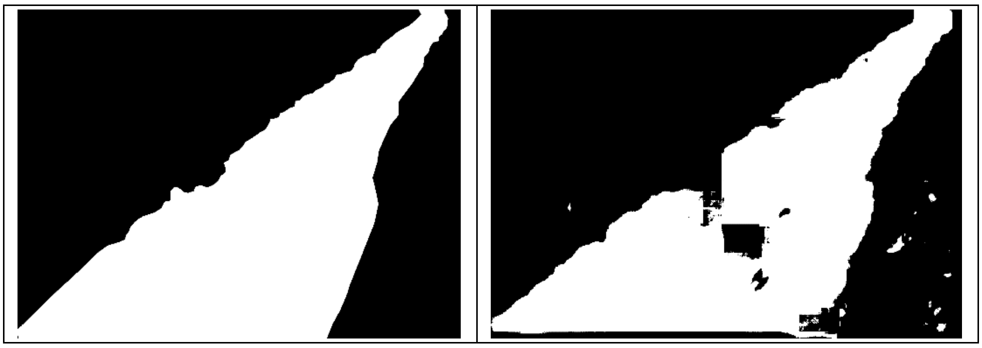
1. Geodesic erosion + reconstruction by dilation => eliminates white spots.
2. Geodesic dilation + reconstruction by erosion => eliminates black spots.

With these process, some of the noise spots can be eliminated



This morphology post-processing also has its limit:

1. Unknown optimal structure element: To conduct dilation or erosion operation, we need to first decide the structure elements used. With larger structure elements is used, larger noise spots can be eliminated with increase possibility to mis-delete the correct but small water body segment. When using smaller structure elements, the effectiveness of eliminating noise spot will be decrease. The optimal structure elements vary from case to case and is hard to determine.
2. Unknown optimal process: Similar to deciding the structure elements to use, how many times we should conduct erosion/dilation in geodesic erosion/dilation process or should we stop early during the reconstruction steps are factors that affects the final post-processed result. While there may be optimal process to conduct, it is difficult for us to find out such a process.
3. Cannot fixed open corruption: When the noise regions are no closed, the reconstruction step will regenerate these noises when no early-stop is conducted (inward corrupting black noise)



Conclusion: While these post processing methods has limited performance, sometimes even decrease the final IOU of the segmentation, it is effective in specific cases and can result in a great improve in the performance of the segmentation process.

4. Reference

[1] Yu, J., Lin, Y., Zhu, Y., Xu, W., Hou, D., Huang, P., & Zhang, G. (2020). Segmentation of River Scenes Based on Water Surface Reflection Mechanism. Applied Sciences, 10(7), 2471. https://doi.org/10.3390/app10072471