DIP
Final Project

TEAM NUMBER：9
TEAM MEMBER：3115511022 邱政岡 3115911023 燕新城 3125810010 孟羽真

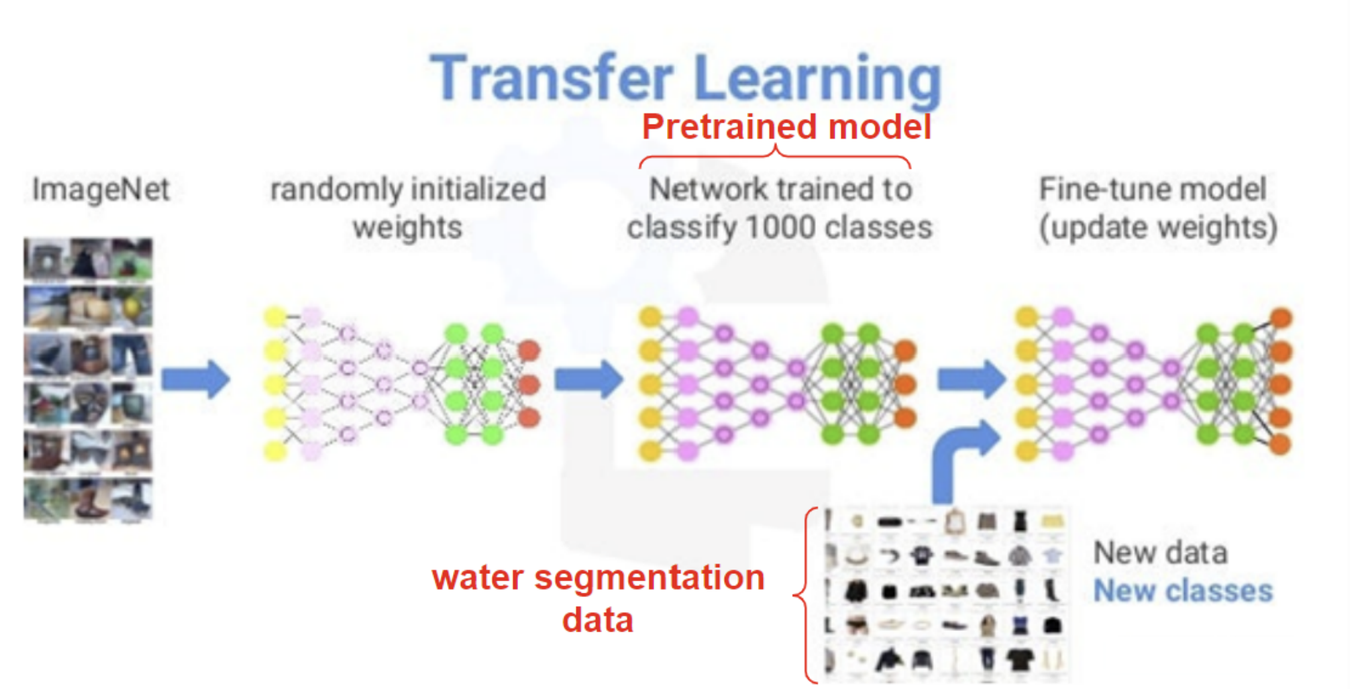
DL method

01. Pretrained model + Data augmentation

Pretrained model

- Pretrain model已在大量的影像資料上面做過一些任務的訓練(預訓練)，如imagenet, imagenet+5k等等，其預訓練後的模型本身具備一定的捕捉feature的能力

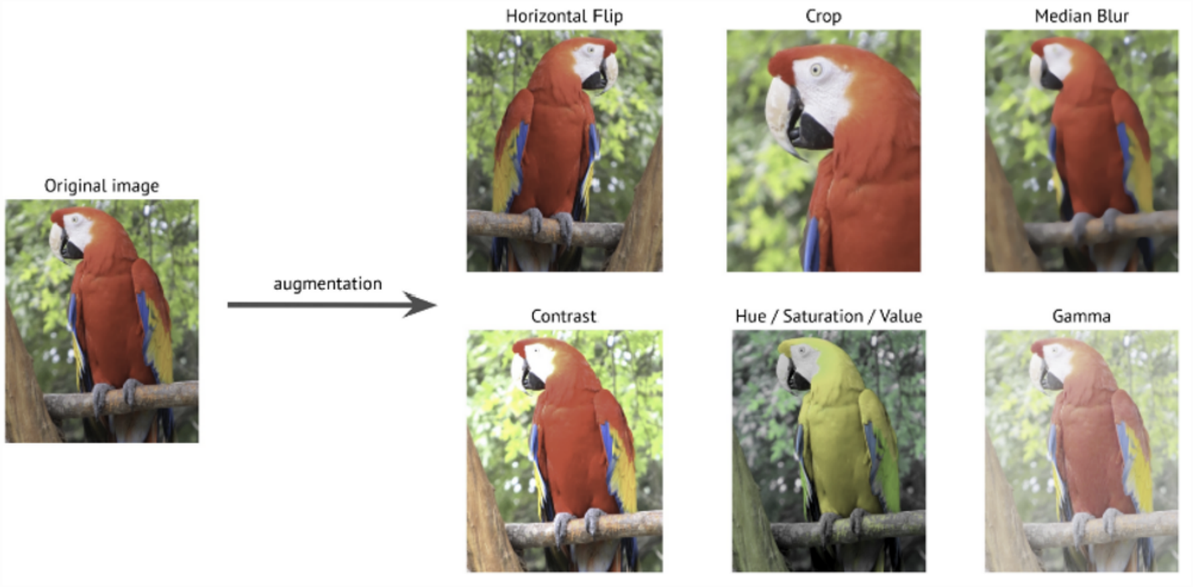
- 再透過fine-tuning方式套用到此次water segmentation的task上，即使在資料較少的情況下，預訓練模型也能實現良好的效能。



Data augmentation

- 為了提高模型的效能，我們需要提供更多的 training data，以使模型能夠辨識更多的特徵，從而做出最終決定。

- 資料增強（Data Augmentation）為從現有數據中套用一些變化產生新的數據，進而增加可以用來訓練的資料量，它能夠有效提高模型的準確度並節省時間金錢，為一個有效解決資料不足的方法。



Pretrained model + Data augmentation

- data augmentaion method

```python
# data augmentation
def get_training_augmentation():
    train_transform = [

        albu.HorizontalFlip(p=0.5),

        albu.ShiftScaleRotate(scale_limit=0.5, rotate_limit=0, shift_limit=0.1, p=1, border_mode=0),

        albu.PadIfNeeded(min_height=320, min_width=320, always_apply=True, border_mode=0),
        albu.RandomCrop(height=320, width=320, always_apply=True),

        albu.IAAAdditiveGaussianNoise(p=0.2),
        albu.IAAPerspective(p=0.5),

        albu.OneOf(
            [
                albu.CLAHE(p=1),
                albu.RandomBrightness(p=1),
                albu.RandomGamma(p=1),
            ],
            p=0.9,
        ),

        albu.OneOf(
            [
                albu.IAASharpen(p=1),
                albu.Blur(blur_limit=3, p=1),
                albu.MotionBlur(blur_limit=3, p=1),
            ],
            p=0.9,
        ),

        albu.OneOf(
            [
                albu.RandomContrast(p=1),
                albu.HueSaturationValue(p=1),
            ],
            p=0.9,
        ),
    ]
    return albu.Compose(train_transform)
```

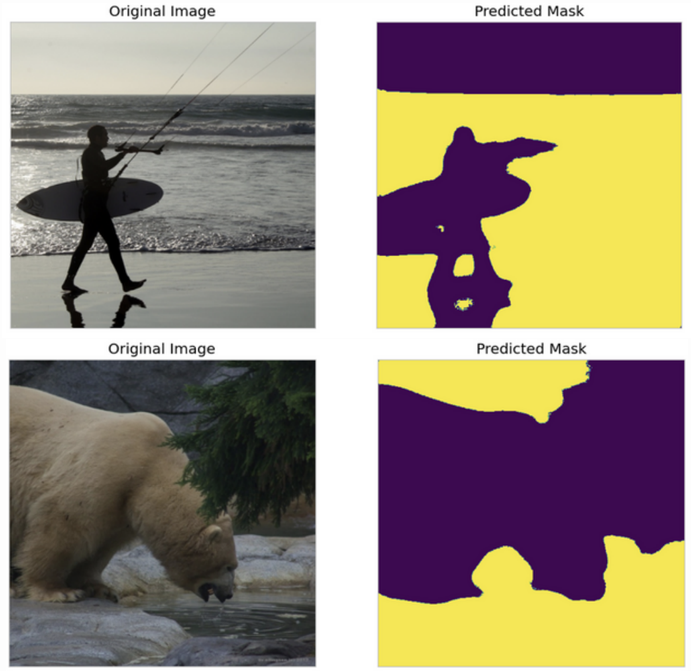
- Experiment with data augmentation in combination with different pretrained models
  - 45 for training, 10 for validation, 5 for testing

pretrained backbone	resnet50	resnet101	dpn68b	dpn92	efficientnet-b2	efficientnet-b3
UNet++	0.47	0.396	0.64	0.57	0.64	0.54
DeepLabV3+	0.53	0.495	0.498	0.54	0.59	0.41

Testing result on UNet++ architecture with dpn68b pretrained backbone

- 從下面兩張預測結果可以看到此方法在water segmentation效果還不錯，也可以清楚的區分水和浪花

- 但從下面這兩張預測結果可以發現與水較相近顏色的區塊會造成模型混淆，容易也將其誤判成水的一部分



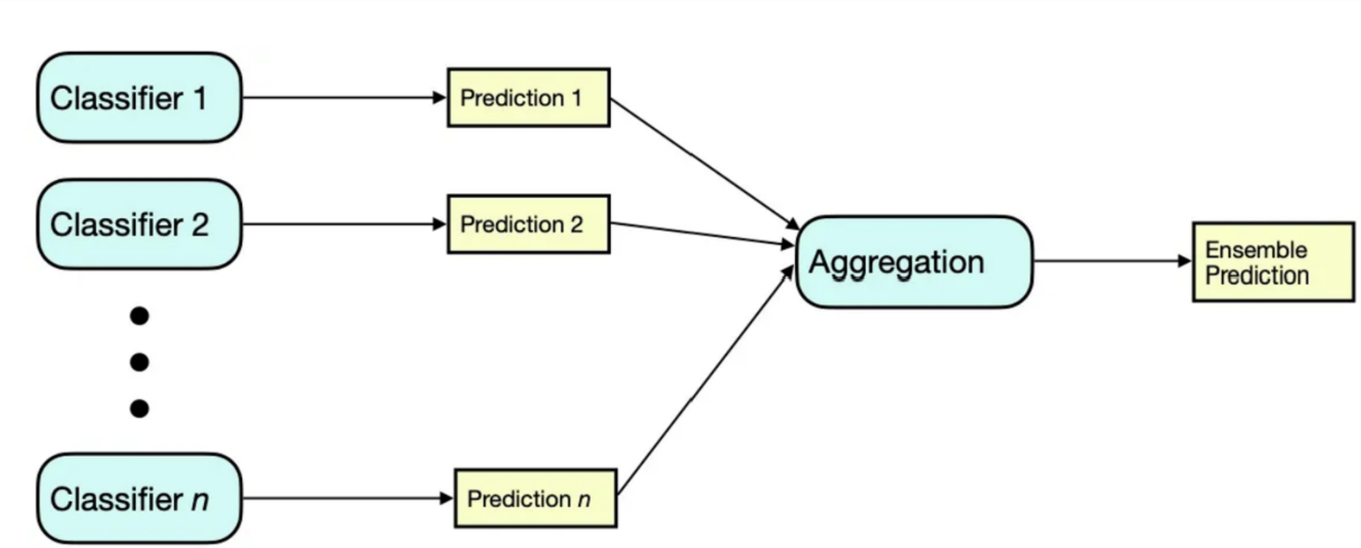
DL method

02. Ensemble learning

Ensemble learning

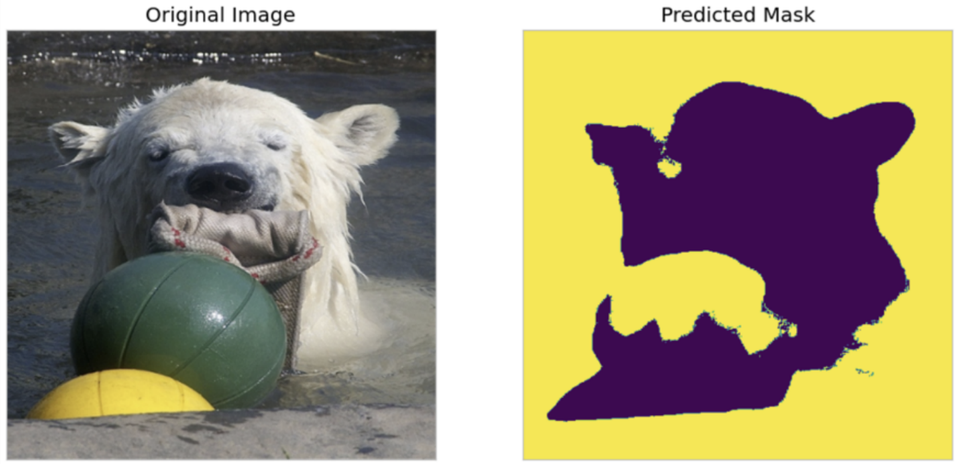
- 透過結合多個模型的預測結果來提高整體的預測效能，此次採用類似bagging的方法，結合多個 Pretrained model，再將這些模型的輸出做 average 來當作最後的 output

- 採用於先前 Testing average IOU score 分數最高的 3 個模型(UNet++ with dpn68b / dpn92 / efficientnet-b2 backbone) 來做 ensemble learning



Experimental results compare single model and Ensemble Learning

- 從下面 testing image 中可以看出利用多個模型去做 segmentaion 的效果，會比使用單一 pretrained model fine-tuning 後來的好，外圍輪廓的地方分得更好的同時，也比較能辨認出球的部分。



[UNet++with dpn68b]

[UNet++ with dpn68b / dpn92 / efficientnet-b2]

Cluster based segmentation

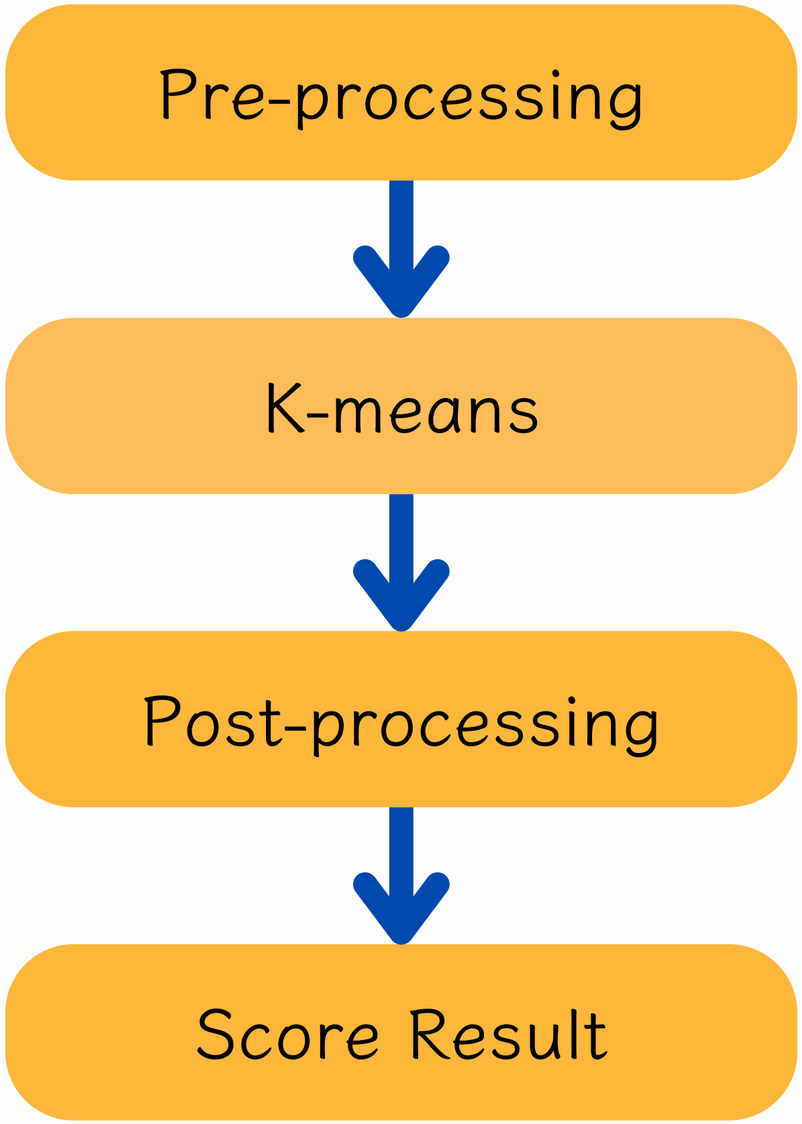
01. Method Choosing
02. Process Flow
03. Pre-processing
04. K-means
05. Post-processing
06. Result

邱政岡、燕新城

Method Chosing

	Pros.	Cons.
Boundary based	Effective for specific shapes	Fail at low gradient boudary
Otsu's Method	Automatic threshold selection	Applicable to bimodal distributions only
Mean-shift algorithm	Non-parametric	High computational complexity
Cluster-based Segmentation K-means	1. Better at low gradient boudary 2. Strong adaptability	

Process Flow



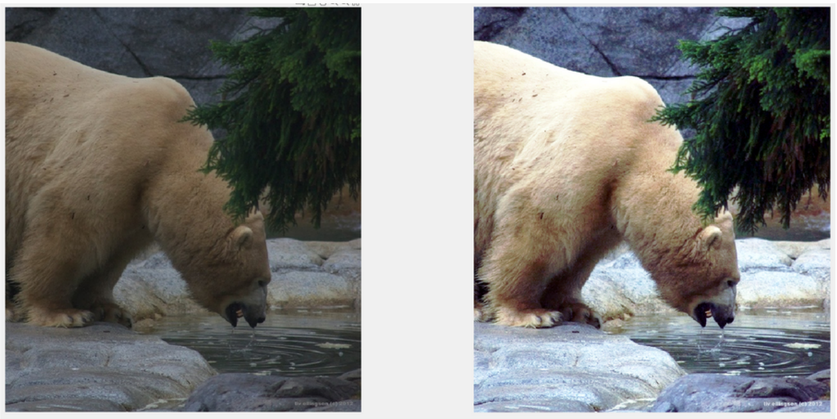
- Histogram equalization
- Color temperature balance
- Local Laplacian filtering

- First K-means for details
- Second K-means for merge

- Select segments and merge
- Closing of opening

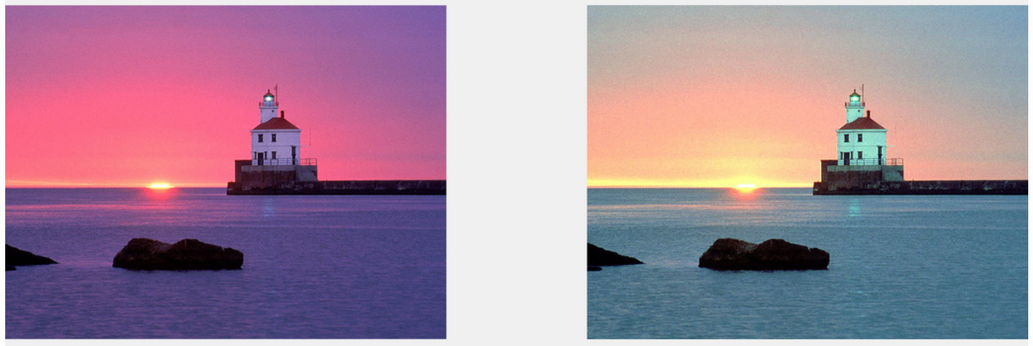
Pre-processing

- Histogram equlization



- Local Laplacian filtering

- Color temperature balance



1. max of mean_r, mean_g, mean_b
2. pixels * max_mean / current_mean

K-means

- First K-means: Up to 1000 Cluster, precision
- Second K-means: 10 Cluster, merge

For the first time of K-means:

- Iter through of Pts

Criteria:

- Distance
- Color

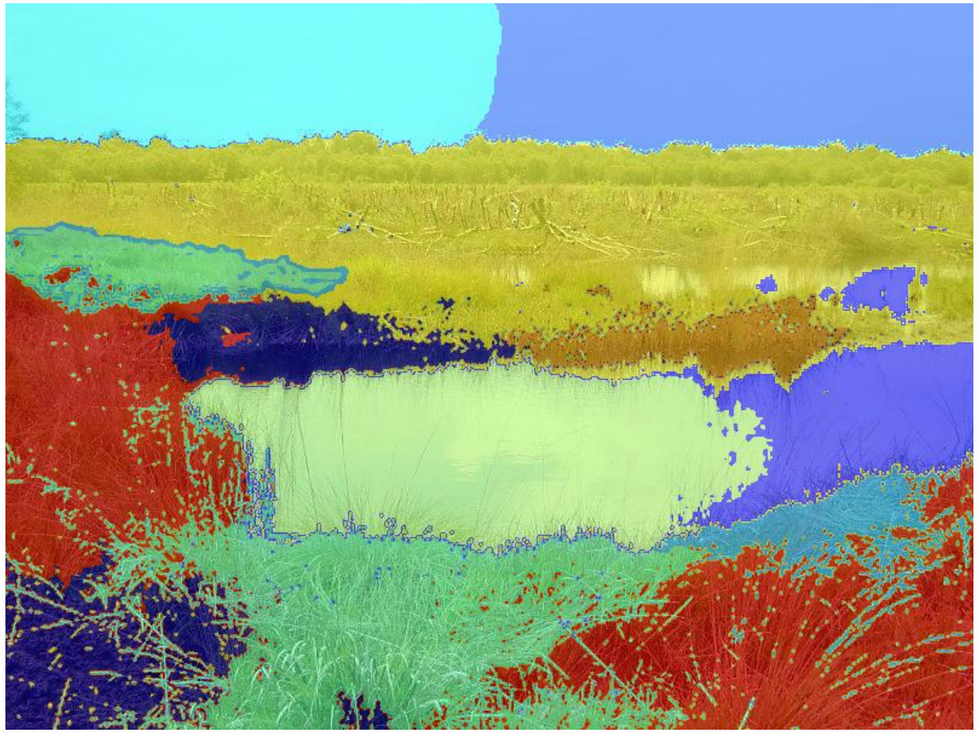
Same Cluster?

Criteria:

- Mean Distance
- Mean Color

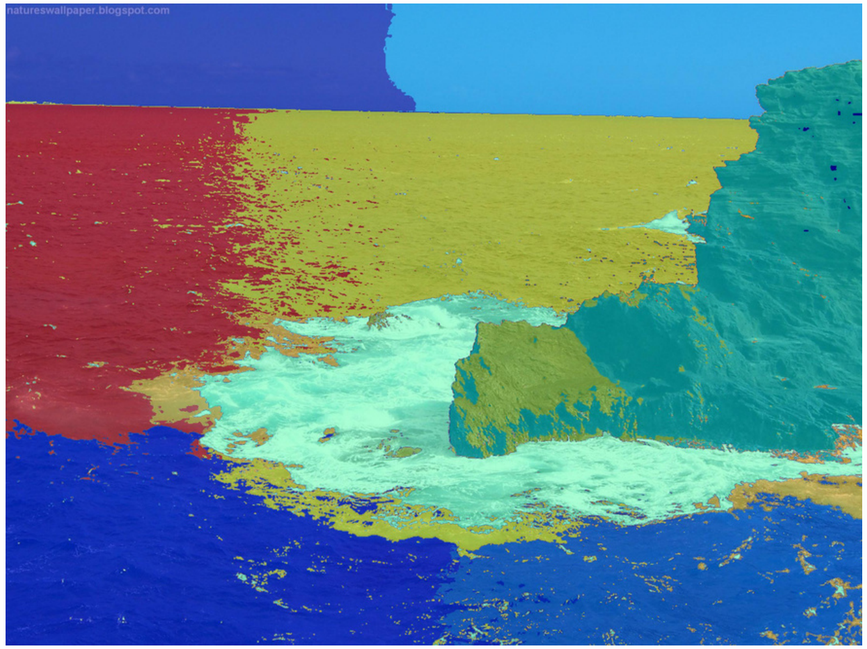
Second K-means

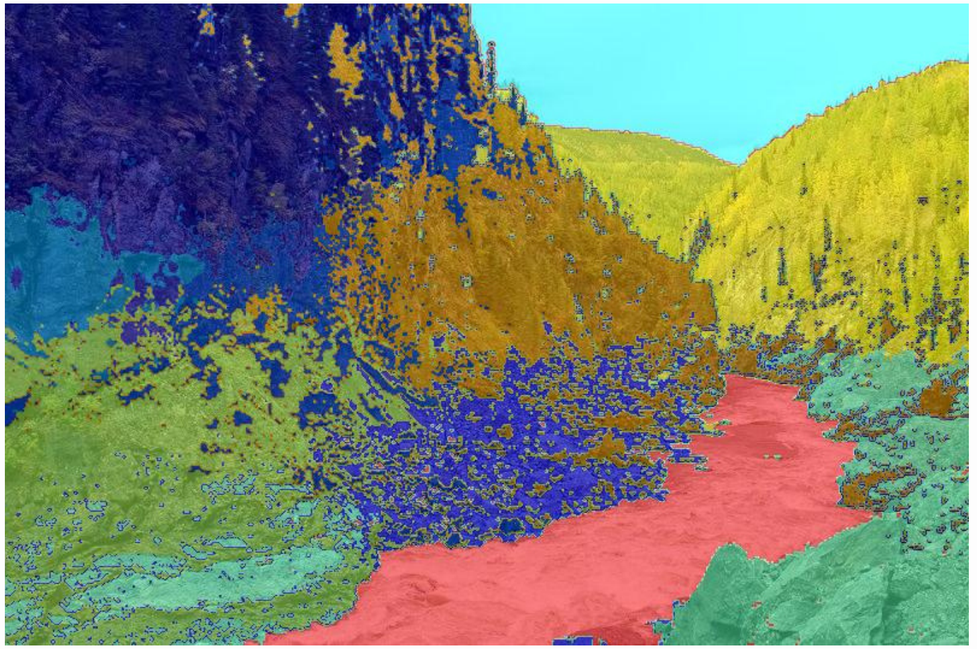
After first clustering, if the number of clusters is greater than 10. We apply a second K-means clustering by Matlab image processing toolbox function to reduce the cluster number into 10.



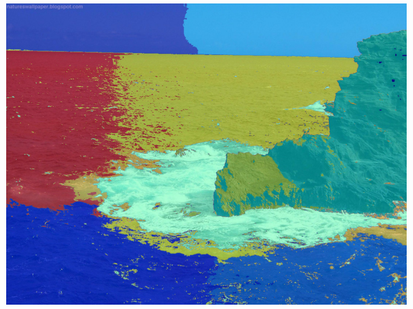
Post-processing

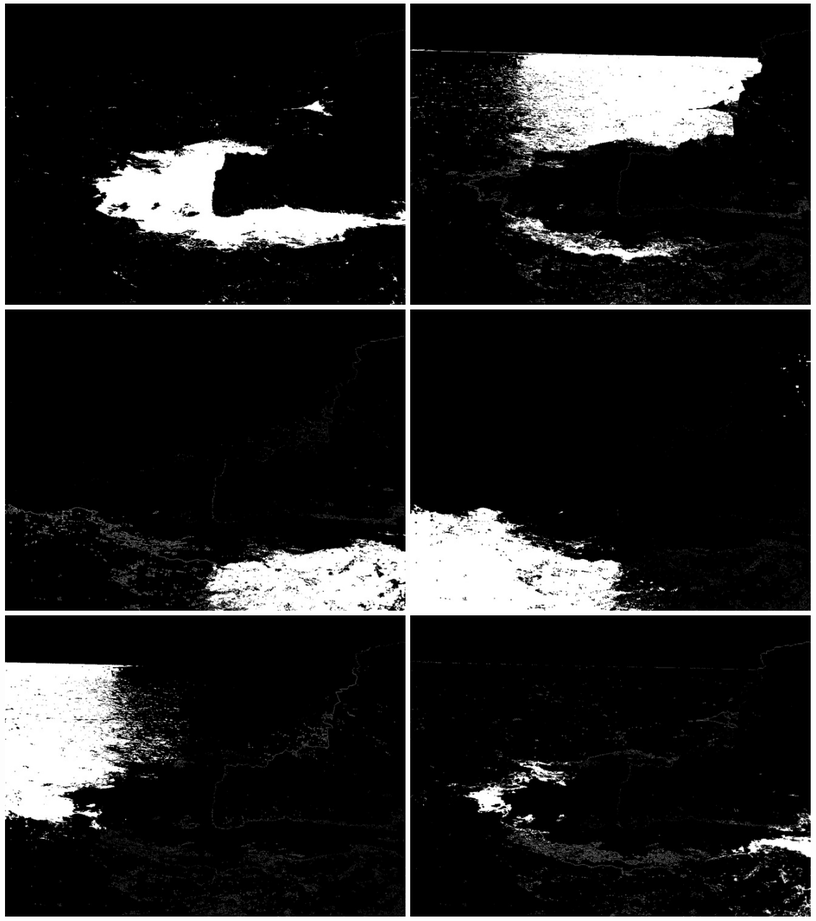
After K-means clustering, the image is split into at most 10 segments. The water may contain one or more segments. We should manually select segments that are water.

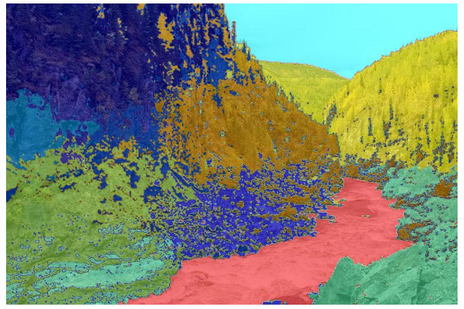




Post-processing

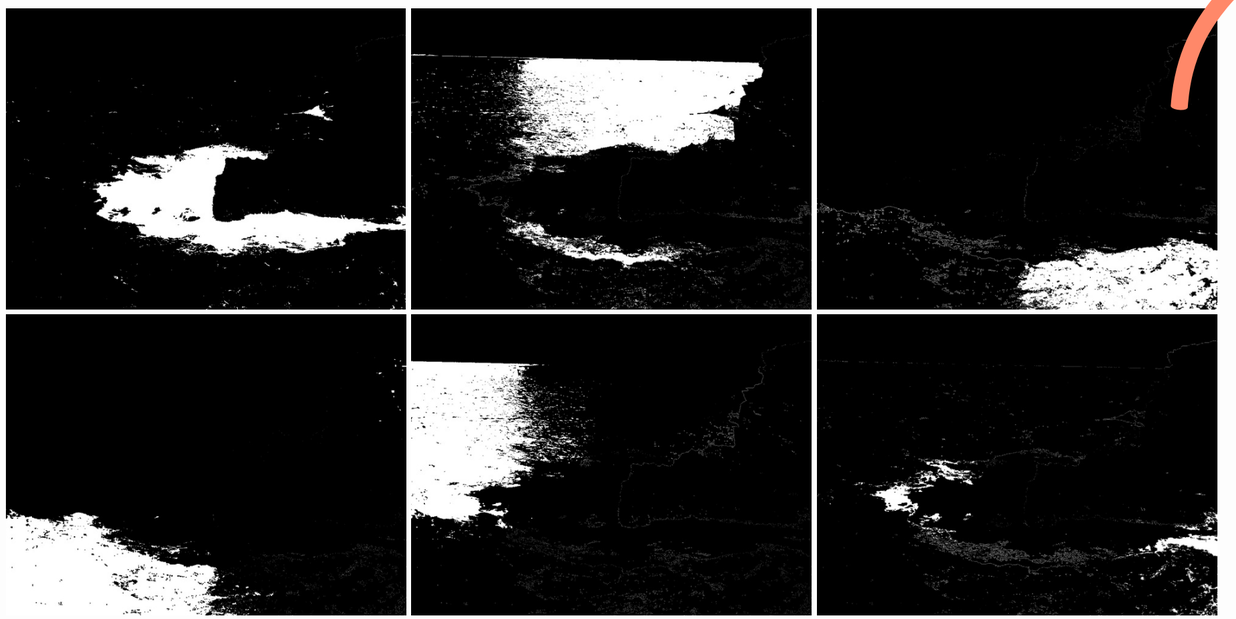






Post-processing

Merge the selected segments and apply closing of opening.



Result Analysis

Because color is an important feature of water, the water can be segmented by our processing.
The average IoU acheive 71.59%.

```
Average IoU: 0.7159382239404538
```

Result Analysis

Because the major feature of our clustering is color. Water in some specific scenes is hard to be segmented.

- reflection
- shadow

Testing Result

The average IoU in testing data

- DL method: 75.57%
- Conventional method: 73.60%
- Combined (use DL in reflection scenes): 83.97%

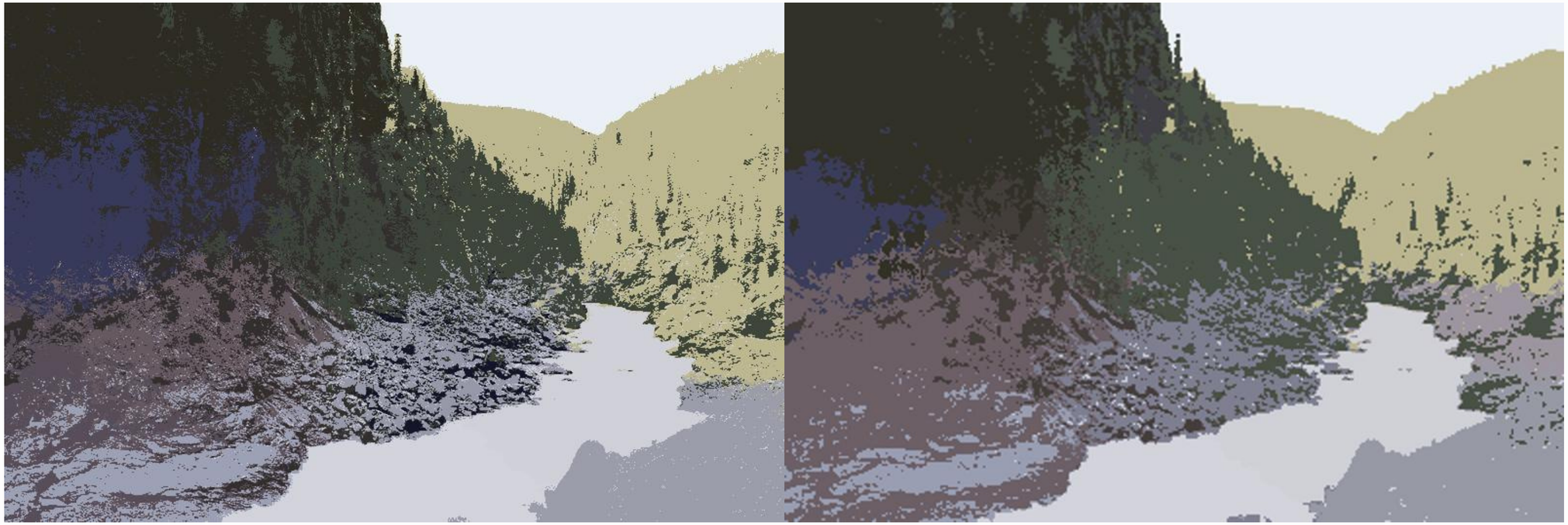
```
IoU for pair 1: 0.16073242597150553
IoU for pair 2: 0.7848913589991728
IoU for pair 3: 0.8182985476963321
IoU for pair 4: 0.3020579441878946
IoU for pair 5: 0.8558345176061898
IoU for pair 6: 0.9449319727891157
IoU for pair 7: 0.9862122200188983
IoU for pair 8: 0.8945885788245621
IoU for pair 9: 0.7476103165872143
IoU for pair 10: 0.9260863905985608
IoU for pair 11: 0.9313146746301827
IoU for pair 12: 0.7157820047485195
Average IoU: 0.7556950793881789
```

```
IoU for pair 1: 0.9073157632255342
IoU for pair 2: 0.8762106731150603
IoU for pair 3: 0.9184150134182739
IoU for pair 4: 0.7128757290264692
IoU for pair 5: 0.740198416151771
IoU for pair 6: 0.8705160430256342
IoU for pair 7: 0.4693018164215066
IoU for pair 8: 0.7328974962246912
IoU for pair 9: 0.15527658825599813
IoU for pair 10: 0.7914210574045525
IoU for pair 11: 0.8452900993872585
IoU for pair 12: 0.8123593411791689
Average IoU: 0.73600065030696599
```

```
IoU for pair 1: 0.9073157632255342
IoU for pair 2: 0.8762106731150603
IoU for pair 3: 0.9184150134182739
IoU for pair 4: 0.7128757290264692
IoU for pair 5: 0.740198416151771
IoU for pair 6: 0.8705160430256342
IoU for pair 7: 0.9862122200188983
IoU for pair 8: 0.7328974962246912
IoU for pair 9: 0.7476103165872143
IoU for pair 10: 0.9260863905985608
IoU for pair 11: 0.8452900993872585
IoU for pair 12: 0.8123593411791689
Average IoU: 0.8396656251632112
```

Appendix

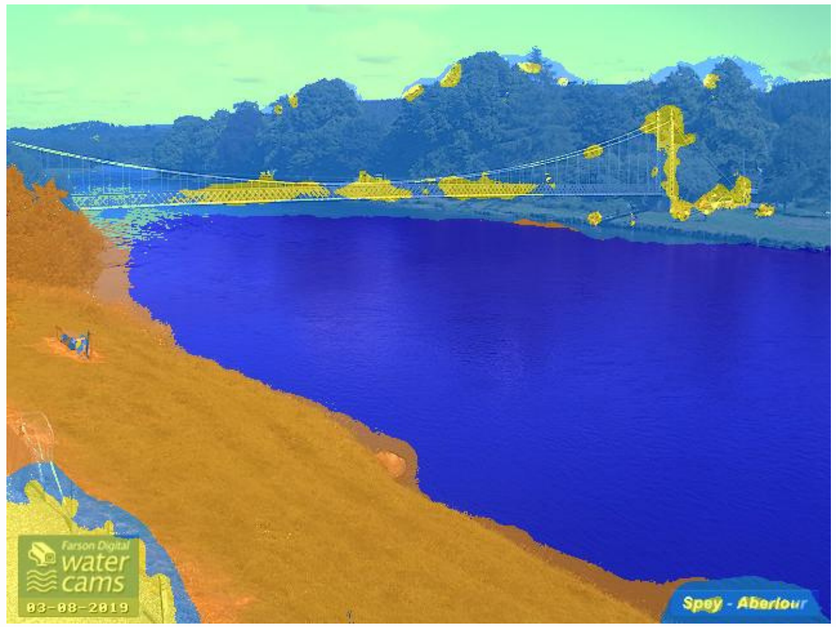
To reduce the computing time and match the demo time limitation. We resize the image to 0.5 before clustering.



Appendix

We also try other features, like textural features. But it is only efficient in some specific scenes.

- large area wave
- large object with obvious texture



Thanks!