

VIỆN NGHIÊN CỨU DỮ LIỆU LỚN



BÁO CÁO DỰ ÁN CUỐI KHÓA

Đo mực nước trạm thủy văn sử dụng camera

Người hướng dẫn: TS. Phan Hải Hồng

Người thực hiện: Nguyễn Văn Sơn (Trưởng nhóm)

Phạm Ngọc Đông

Đoàn Quang Khải

Nguyễn Thế Hiển

Hà Nội, 16/01/2020

Lời cảm ơn

Nhóm chúng em xin chân thành gửi lời cảm ơn đến cô Phan Hải Hồng vì đã định hướng, chỉ dẫn, giúp đỡ chúng em để có thể thực hiện tốt đề tài được giao. Nhóm cũng gửi lời cảm ơn tới VinBigData, AI Academy vì đã tổ chức, tài trợ cho chương trình phát triển nhân lực AI, mang đến cho chúng em những kiến thức về AI rất thực tiễn, bổ ích và là bước đệm quan trọng trong sự nghiệp của chúng em nói riêng, công cuộc phát triển nguồn nhân lực AI cho Việt Nam nói chung.

Tóm tắt nội dung

Trong báo project này, nhóm sẽ giải quyết bài toán đo mực nước thủy văn dựa vào dữ liệu camera có chứa hình ảnh khu vực nước có thước đo. Nhóm đề xuất phương án thực hiện gồm 2 giai đoạn lần lượt là phát hiện ra vị trí thước trong ảnh, cách phát hiện là sử dụng 1 mạng deep learning, tiếp đến là trích xuất ra được thông tin trên thước sử dụng các phương pháp xử lý ảnh truyền thống. Kết quả đạt được yêu cầu của bài toán là đưa ra được thông tin đo mực nước, tuy nhiên với dữ liệu nhóm nhận được thì project mang tính thực hành, vận dụng kiến thức đã học trong khóa học, còn tính ứng dụng của phương pháp được nêu ra trong báo cáo này thì chưa cao. Qua project này, các thành viên trong nhóm đã nắm được cơ bản quy trình để làm một sản phẩm về thị giác máy tính, cũng như thấy các vấn đề mà mô hình deep learning gặp phải khi được ứng dụng vào bài toán thực tế.

MỤC LỤC

CHƯƠNG 1. KHÁI QUÁT BÀI TOÁN	6
1.1 Giới thiệu chung	6
1.2 Quy trình giải quyết bài toán.....	6
1.2.1 Phương pháp thực hiện.....	6
1.2.2 Phương pháp đánh giá	7
CHƯƠNG 2. SỬ DỤNG MẠNG YOLOv3 PHÁT HIỆN VÙNG CÓ THUỐC TRONG ẢNH.....	8
2.1 Giới thiệu mạng YOLOv3.....	8
2.1.1 Kiến trúc mạng YOLOv3	8
2.1.2 Output của YOLOv3	9
2.2 Tiền xử lý dữ liệu	10
2.2.1 Loại ảnh ban đêm và phân chia dữ liệu.....	10
2.2.2 Data augmentation.....	10
2.3 Phát hiện vùng có thuốc trong ảnh.....	10
2.3.1 Huấn luyện mô hình YOLOv3	10
2.3.2 Nhận diện thuốc trong ảnh	11
CHƯƠNG 3. TRÍCH XUẤT MỰC NƯỚC TỪ THUỐC ĐO	13
3.1 Khảo sát histogram.....	13
3.2 Cải thiện chất lượng ảnh nhị phân và trích xuất thông tin mực nước	15
3.3 Đánh giá kết quả trích xuất thông tin mực nước.....	16
3.4 Áp dụng trích thông tin mực nước cho video	17
CHƯƠNG 4. KẾT LUẬN	18
4.1 Kết luận	18
4.2 Hướng phát triển	18

DANH MỤC HÌNH VẼ

Hình 1-1 Ảnh cắt từ video dữ liệu nhận được	6
Hình 1-2 Quy trình giải quyết bài toán	7
Hình 2-1 Sơ đồ kiến trúc mạng YOLOv3.....	8
Hình 2-2 Các layer trong mạng darknet-53.	9
Hình 2-3: loss trong quá trình huấn luyện	11
Hình 2-4: Kết quả nhận diện thước trong ảnh bằng weights đã được huấn luyện	12
Hình 3-1 Thước đo được phát hiện trong ảnh.....	13
Hình 3-2 Histogram của ảnh	13
Hình 3-3 Ảnh nhị phân sau khi phân ngưỡng (5-10-15-20-25-30)	14
Hình 3-4 Các dấu mốc dung để đếm và đo mực nước.....	15
Hình 3-5 Ảnh nhị phân sau khi đã qua bộ lọc Gaussian.....	16

DANH MỤC BẢNG BIỂU

Bảng 3-1 Một số công thức ngưỡng cơ bản.....	14
Bảng 3-2: Công thức bộ lọc Gaussian 1D, 2D	16
Bảng 3-3 Kết quả thử nghiệm khử nhiễu với các ngưỡng khác nhau	17

CHƯƠNG 1. KHÁI QUÁT BÀI TOÁN

1.1 Giới thiệu chung

Việc theo đo đạc để theo dõi mực nước của hệ thống thủy lợi đem lại nhiều lợi ích trong thực tế. Cảm biến áp suất, phao nổi, cảm biến radar và kỹ thuật siêu âm là các phương pháp được sử dụng nhiều trong việc đo mực nước. Nhược điểm của các phương pháp trên là độ chính xác kém, việc kiểm tra phải thực hiện bởi con người khi có điều kiện môi trường thay đổi.

Sử dụng camera để theo dõi, đưa ra kết quả đo đạc mực nước là một cách tiếp cận mới, tuy nhiên lại đem đến hiệu quả cao và đang được ưa chuộng trong thời gian gần đây.

Trong khuôn khổ project này, nhóm được giao cho dữ liệu là video thu được từ một camera đặt ở khu vực hồ nước thủy văn Phú Lâm. Nhiệm vụ của nhóm là trích xuất ra được thông tin mực nước từ đó.



Hình 1-1 Ảnh cắt từ video dữ liệu nhận được

1.2 Quy trình giải quyết bài toán

1.2.1 Phương pháp thực hiện

Quan sát dữ liệu, có thể thấy việc đo mực nước ở đây sẽ chuyển về bài toán trích xuất được thông tin trên thước đo từ ảnh. Để làm điều đó, nhóm đã tìm hiểu về các bài toán tương tự đã được giải quyết cũng như tìm hiểu về các bài toán liên quan đến phần thị giác máy tính. Với dữ liệu nhận được là video, nhóm thực hiện tách video thành tập hợp các ảnh và giải quyết bài toán trên từng ảnh sau đó cải tiến tốc độ để có thể triển khai thời gian thực trên video.

Với mỗi ảnh, bài toán đo mực nước được nhóm chia thành 2 giai đoạn lần lượt là:

- Giai đoạn 1: Phát hiện ra vùng có thước đo trong ảnh, sử dụng kỹ thuật phát hiện đối tượng bằng deep learning.
- Giai đoạn 2: Từ vùng có thước đo, trích xuất thông tin vạch chia trên thước sử dụng các kỹ thuật xử lý ảnh.



Hình 1-2 Quy trình giải quyết bài toán

1.2.2 Phương pháp đánh giá

Video sau khi tách thành các ảnh, sẽ được các thành viên trong nhóm phân công nhau gán nhãn là mực nước tại thời điểm đó.

Sau đó tập hợp bộ dữ liệu ảnh sẽ được chia thành 2 phần huấn luyện và kiểm thử theo tỷ lệ 8:2.

Việc đánh giá được thực hiện trên bộ kiểm thử dựa trên độ đo RMSE:

$$RMSE = \sqrt{\sum_{t=1}^n \frac{(\hat{y}_t - y_t)^2}{n}}$$

Trong đó:

- \hat{y}_t là kết quả đo mực nước từ ảnh thứ t .
- y_t là mực nước thực tế từ ảnh thứ t .

CHƯƠNG 2. SỬ DỤNG MẠNG YOLOv3 PHÁT HIỆN VÙNG CÓ THUỐC TRONG ẢNH

2.1 Giới thiệu mạng YOLOv3

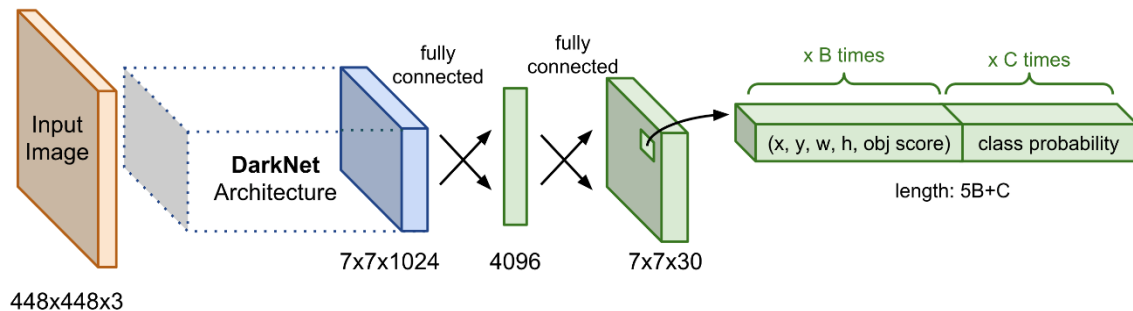
YOLO là thuật toán object detection nên mục tiêu của mô hình không chỉ là dự báo nhãn cho vật thể như các bài toán classification mà nó còn xác định location của vật thể [1]. Do đó YOLO có thể phát hiện được nhiều vật thể có nhãn khác nhau trong một bức ảnh thay vì chỉ phân loại duy nhất một nhãn cho một bức ảnh.

Sở dĩ YOLO có thể phát hiện được nhiều vật thể trên một bức ảnh như vậy là vì thuật toán có những cơ chế rất đặc biệt mà chúng ta sẽ tìm hiểu bên dưới.

2.1.1 Kiến trúc mạng YOLOv3

Kiến trúc YOLO bao gồm: base network là các mạng convolution làm nhiệm vụ trích xuất đặc trưng. Phần phía sau là những Extra Layers được áp dụng để phát hiện vật thể trên feature map của base network [1].

Base network của YOLO sử dụng chủ yếu là các convolutional layer và các fully connected layer. Các kiến trúc YOLO cũng khá đa dạng và có thể tùy biến thành các version cho nhiều input shape khác nhau.



Hình 2-1 Sơ đồ kiến trúc mạng YOLOv3

Thành phần Darknet Architecture được gọi là base network có tác dụng trích xuất đặc trưng. Output của base network là một feature map có kích thước $7 \times 7 \times 1024$ sẽ được sử dụng làm input cho các Extra layers có tác dụng dự đoán nhãn và tọa độ bounding box của vật thể.

Trong YOLO version 3 tác giả áp dụng một mạng feature extractor là darknet-53 [2]. Mạng này gồm 53 convolutional layers kết nối liên tiếp, mỗi layer được theo sau bởi một batch normalization và một activation Leaky Relu. Để giảm kích thước của output sau mỗi convolution layer, tác giả down sample bằng các filter với kích thước là 2. Mạng này có tác dụng giảm thiểu số lượng tham số cho mô hình.

	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1x	Convolutional	32	1 × 1	128 × 128
	Convolutional	64	3 × 3	
	Residual			
	Residual			
	Convolutional	128	3 × 3 / 2	64 × 64
2x	Convolutional	64	1 × 1	64 × 64
	Convolutional	128	3 × 3	
	Residual			
	Residual			
	Convolutional	256	3 × 3 / 2	32 × 32
8x	Convolutional	128	1 × 1	32 × 32
	Convolutional	256	3 × 3	
	Residual			
	Residual			
	Convolutional	512	3 × 3 / 2	16 × 16
8x	Convolutional	256	1 × 1	16 × 16
	Convolutional	512	3 × 3	
	Residual			
	Residual			
	Convolutional	1024	3 × 3 / 2	8 × 8
4x	Convolutional	512	1 × 1	8 × 8
	Convolutional	1024	3 × 3	
	Residual			
	Residual			
	Avgpool		Global	
	Connected		1000	
	Softmax			

Hình 2-2 Các layer trong mạng darknet-53.

2.1.2 Output của YOLOv3

Output của mô hình YOLO là một véc tơ sẽ bao gồm các thành phần:

$$y^T = [p_0, \underbrace{\langle t_x, t_y, t_w, t_h \rangle}_{\text{bounding box}}, \underbrace{\langle p_1, p_2, \dots, p_c \rangle}_{\text{scores of c classes}}]$$

Trong đó: [3]

- p_0 là xác suất dự báo vật thể xuất hiện trong bounding box.
- $\langle t_x, t_y, t_w, t_h \rangle$ giúp xác định bounding box. Trong đó t_x, t_y là tọa độ tâm và t_w, t_h là kích thước rộng, dài của bounding box.
- $\langle p_1, p_2, \dots, p_c \rangle$ là véc tơ phân phối xác suất dự báo của các classes.

2.2 Tiền xử lý dữ liệu

2.2.1 Loại ảnh ban đêm và phân chia dữ liệu

Dữ liệu video được tách thành tập hợp các ảnh, nhóm sẽ giải quyết bài toán phát hiện thước đo trên từng ảnh. Tuy nhiên, vì camera không có chức năng quay ban đêm nên ảnh ban đêm đều không thể sử dụng được. Trong phạm vi môn học, nhóm sẽ loại bỏ những ảnh trong khoảng thời gian trước 5 giờ 30 và sau 18 giờ.

Những ảnh ban ngày được giữ lại và chia thành 2 phần là huấn luyện và kiểm thử theo tỷ lệ 9:1.

2.2.2 Data augmentation

Camera được đặt cố định, vị trí thước đo không thay đổi, sự thay đổi đáng kể ở trong video đến từ việc ánh sáng và mực nước thay đổi theo từng khung giờ trong ngày. Do đó các ảnh được tách ra có ít sự khác nhau, không đa dạng, nếu trực tiếp sử dụng bộ dữ liệu ảnh đó để huấn luyện mô hình thì mô hình sẽ không có tính tổng quát hóa, đặc biệt là khi chuyển góc quay camera ra góc khác thì mô hình không thể phát hiện ra được thước. Vấn đề dữ liệu thiếu sự tổng quát, không đa dạng là một vấn đề thường gặp trong học máy. Trong lĩnh vực thị giác máy tính, khi việc thu thập dữ liệu đa dạng là không khả thi, người ta có một phương pháp để làm tăng tính đa dạng của dữ liệu đó chính là học tăng cường (data augmentation) sử dụng các biến đổi tiền xử lý hình ảnh đầu vào. Đây là phương pháp hiệu quả nhằm thay đổi dữ liệu tập huấn luyện, giúp dữ liệu phong phú và tổng quát hơn, mang lại hiệu quả ứng dụng thực tế cao. Nhóm sẽ áp dụng các phương pháp tăng cường ảnh lên bộ dữ liệu huấn luyện.

2.3 Phát hiện vùng có thước trong ảnh

2.3.1 Huấn luyện mô hình YOLOv3

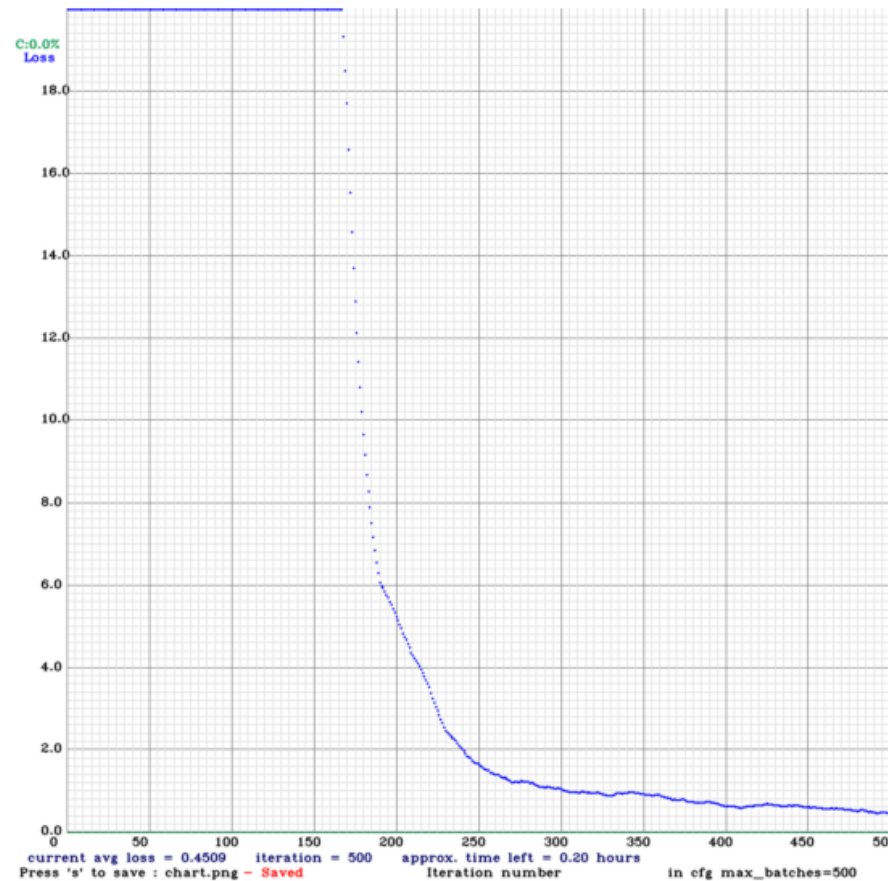
Trong bài báo cáo này, nhóm cấu hình mạng YOLOv3 với các thông số:

- batch = 64
- subdivisions = 16
- max_batch = 500
- steps = 400, 450

Do dự án chỉ nhận diện 1 đối tượng là thước đo thủy văn (classes = 1) trong 3 lớp YOLO nên thông số filters = $(1+5)*3 = 18$

Để huấn luyện mạng ta sử dụng câu lệnh sau:

```
!./darknet detector train data/obj.data cfg/yolov3_custom.cfg darknet53.conv.74 -dont_show
```



Hình 2-3: loss trong quá trình huấn luyện

2.3.2 Nhận diện thước trong ảnh

Để nhận diện thước ta sử dụng câu lệnh:

```
!./darknet detector test data/obj.data cfg/yolov3_custom2.cfg /content/darknet/backup/yolov3_custom_final.weights /content/darknet/data/test/img_20201201_34.jpg #-thresh 0.5
```



Hình 2-4: Kết quả nhận diện thước trong ảnh bằng weights đã được huấn luyện

CHƯƠNG 3. TRÍCH XUẤT MỨC NƯỚC TỪ THUỐC ĐO

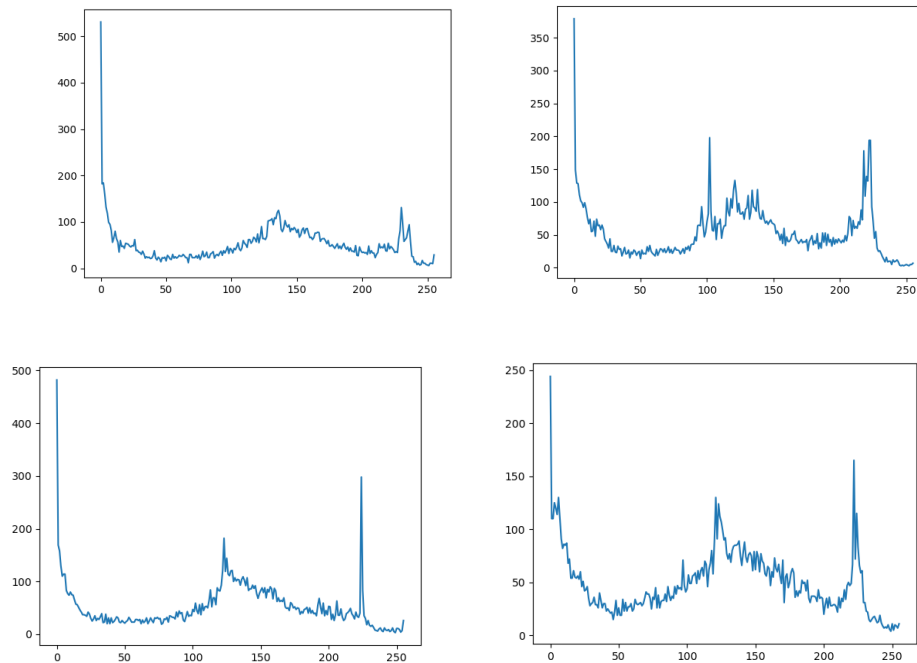
3.1 Khảo sát histogram

Sau khi kết thúc quá trình phát hiện thước đo, ta thu được tọa độ vùng có thước, từ đây nhóm sẽ sử dụng các kỹ thuật xử lý ảnh truyền thống để có thể lấy được thông tin từ thước ở vùng này.



Hình 3-1 Thước đo được phát hiện trong ảnh

Quan sát dữ liệu có thể thấy, để lấy được thông tin độ sâu thì ta chỉ cần đếm được số vạch trên thước. Để khảo sát rõ hơn về tần xuất thống kê số lần xuất hiện các mức sáng trong ảnh, nhóm sử dụng histogram.



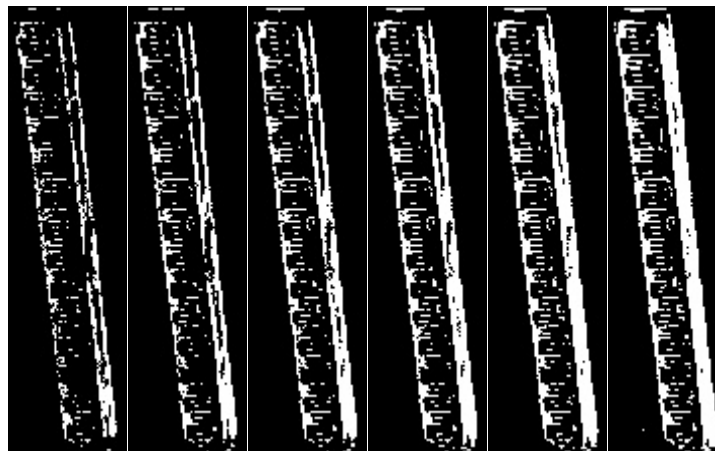
Hình 3-2 Histogram của ảnh

Điểm ảnh tập trung nhiều ở khu vực 0-30. Khu vực này thể hiện màu tối, cụ thể trong bức ảnh có thể thấy vùng tối xuất hiện nhiều nhất là vùng vạch đánh dấu trên thước. Rất tự nhiên, ta có thể nghĩ đến việc đặt ngưỡng cho ảnh, với kỳ vọng sẽ phân tách được phần vạch đánh dấu trên thước với phần còn lại của ảnh. [4]

Binary	$dst(x, y) = \begin{cases} maxval & \text{if } src(x, y) > thresh \\ 0 & \text{otherwise} \end{cases}$
Inverted Binary	$dst(x, y) = \begin{cases} 0 & \text{if } src(x, y) > thresh \\ maxval & \text{otherwise} \end{cases}$
Truncated	$dst(x, y) = \begin{cases} threshold & \text{if } src(x, y) > thresh \\ src(x, y) & \text{otherwise} \end{cases}$
To Zero	$dst(x, y) = \begin{cases} src(x, y) & \text{if } src(x, y) > thresh \\ 0 & \text{otherwise} \end{cases}$
To Zero Inverted	$dst(x, y) = \begin{cases} 0 & \text{if } src(x, y) > thresh \\ src(x, y) & \text{otherwise} \end{cases}$

Bảng 3-1 Một số công thức ngưỡng cơ bản

Từ việc khảo sát histogram, nhóm thử các phương pháp và chọn các ngưỡng khác nhau, phương pháp tách được thước với môi trường xung quanh mà vẫn giữ được nhiều thông tin trên thước nhất mà nhóm đã thử nghiệm là phương pháp Inverted Binary.



Hình 3-3 Ảnh nhị phân sau khi phân ngưỡng (5-10-15-20-25-30)

Với mức ngưỡng là 30, tức là các điểm ảnh có mức sáng nhỏ hơn 30 sẽ mang màu trắng trong ảnh nhị phân ở trên, ứng với phần đồ thị phần bên trái của histogram. Đúng như kỳ vọng, ảnh nhị phân thu được ở ngưỡng 30 mang lại kết quả tốt khi giữ lại được phần đánh dấu các mốc trên thước đo, nhóm kỳ vọng có thể phân được rõ vạch chia đơn vị, tuy nhiên do chất lượng ảnh không tốt, vạch chia đơn vị rất nhỏ và không rõ nên nhóm tạm thời chấp nhận đếm dấu mốc trên thước sau khi phân ngưỡng để làm kết quả cho độ sâu mực nước.

3.2 Cải thiện chất lượng ảnh nhị phân và trích xuất thông tin mực nước

Ảnh nhị phân đã phân nào tách được các thông tin của thước đo với môi trường xung quanh, tuy nhiên với ý tưởng là đếm dấu mốc trên thước thì chưa thể thực hiện được ngay, cần phải tách rõ được dấu mốc trên thước với các đối tượng khác chẳng hạn như nhiễu và các vạch chia đơn vị không rõ ràng để thực hiện việc đếm chính xác hơn. Để thực hiện việc đó, nhóm tiến hành sử dụng các phương pháp làm trơn ảnh và khử nhiễu trước khi tách ngưỡng.



Hình 3-4 Các dấu mốc dung để đếm và đo mực nước

Có nhiều phương pháp làm trơn ảnh để khử nhiễu, tuy nhiên thư viện opencv cung cấp 4 kiểu phổ biến nhất:

- Trung bình (Average)
- Bộ lọc Gaussian
- Bộ lọc Median
- Bộ lọc Biateral

Thực hiện thử từng phương pháp với các bộ tham số khác nhau trên 10 ảnh có vùng chứa thước, kết quả cuối cùng, bộ lọc Gaussian cho ảnh nhị phân sau khi lấy ngưỡng có thể phân tách rõ nhất các dấu mốc trên thước đo.

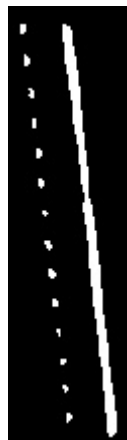
Bộ lọc Gaussian áp dụng lên ảnh bằng phép tích chập của hàm Gaussian [5] với công thức

1 chiều	2 chiều
$G(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}$	$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$

Bảng 3-2: Công thức bộ lọc Gaussian 1D, 2D

Trong đó:

- x, y là các giá trị tương ứng trong kernel
- σ là độ lệch chuẩn



Hình 3-5 Ảnh nhị phân sau khi đã qua bộ lọc Gaussian

Kết quả đáng ngạc nhiên khi chỉ sử dụng 2 phương pháp cơ bản là bộ lọc Gaussian và dung ngưỡng, nhóm đã có thể tách tách được ra dấu mốc trên thước đo. Công việc bây giờ là đếm số chấm trên ảnh nhị phân bằng cách các khu vực các điểm ảnh có cùng màu và liên kề nhau, rồi trừ đi các phần có diện tích lớn chính là phần gây thước (vùng trắng dài trải dọc ở thước). Với ảnh nhị phân ở hình 3.5, sau khi chạy thuật toán đếm số chấm, kết quả là 14, đúng với số chấm trắng xuất hiện. Từ kết quả này ta chỉ cần có thông số cụ thể của thước đo thì có thể suy ra được độ sâu của mực nước hiện tại.

3.3 Đánh giá kết quả trích xuất thông tin mực nước

Kết quả được đánh giá bằng độ đo RMSE trên tập ảnh test gồm 52 ảnh các khu vực chứa thước đo được cắt ra từ video sau đó được nhóm gán nhãn số mốc trên thước.

<div>Ngưỡng</div> <div>Khử nhiễu</div>	5	10	15	20	25	30
Trung bình	11.732	7.062	2.987	2.699	0.931	0.772
Gaussian	10.489	4.387	3.207	2.413	0.898	0.746
Median	3.608	5.457	4.618	2.983	1.641	1.028
Bilateral	4.863	7.207	8.058	4.206	3.001	4.363

Bảng 3-3 Kết quả thử nghiệm khử nhiễu với các ngưỡng khác nhau

Từ kết quả thử nghiệm trên, để trích xuất thông tin mực nước, nhóm sử dụng phương pháp có RMSE thấp nhất:

- Bộ lọc Gaussian được sử dụng với kernel = (7,7)
- Inverted Binary với thresh = 30

3.4 Áp dụng trích thông tin mực nước cho video

Quá trình thực hiện bên trên là giải quyết bài toán lấy thông tin mực nước đo tại mỗi ảnh riêng biệt, là bài toán con của bài toán lấy thông tin mực nước từ video. Trường hợp lý tưởng nhất là hệ thống của nhóm sẽ hoạt động với độ chính xác 100%, như vậy thì thông tin mực nước lấy từ 1 ảnh tại thời điểm khung hình đó xuất hiện trong video sẽ cho ta thông tin chính xác. Tuy nhiên, hệ thống thực tế thì không thể hoạt động với độ chính xác 100%, sai số có thể xuất hiện tại giai đoạn phát hiện thước hoặc giai đoạn xử lý ảnh để trích xuất thông tin thước. Do đó, với dữ liệu video nhóm nhận được, số khung hình trên 1 giây là 8 (8 FPS), điều đó tương ứng với trong 1 giây, sẽ có 8 kết quả mực nước được trả về và mực nước được xác định trong 1 giây đó sẽ là mực nước được trả về nhiều nhất.

CHƯƠNG 4. KẾT LUẬN

4.1 Kết luận

Trong phạm vi môn học thị giác máy tính, nhóm chúng em đã giải quyết được bài toán đo mực nước thủy văn từ dữ liệu video nhận được. Hướng giải quyết cho bài toán mà nhóm sử dụng là dùng mô hình học sâu để phát hiện ra vùng có thước rồi sử dụng các kỹ thuật xử lý ảnh để lấy được thông tin mực nước trong vùng có chứa thước. Trong việc phát hiện thước đo, nhóm sử dụng pre-train của mô hình yolo sau đó tùy chỉnh lại để thu được mô hình có khả năng phát hiện ra thước đo thủy văn trong video, total loss = 0.000225. Sau khi có tọa độ vùng có thước, nhóm sử dụng bộ lọc Gaussian và chọn ngưỡng để biến ảnh thành ảnh nhị phân, từ đó thu được số dấu mốc đơn vị trên thước đo và đưa ra kết quả mực nước cho từng ảnh. Cuối cùng, kết hợp kết quả mực nước trong từng ảnh để thu được mực nước từ video tương ứng với các khoảng thời gian. Với quy trình như trên, sai số xuất hiện ở cả 2 giai đoạn, đặc biệt là phần trích xuất thông tin từ thước, việc dùng xử lý ảnh truyền thống gặp phải vấn đề là không tổng quát hóa được bài toán với các kiểu khung hình hay góc quay khác, ngoài ra việc chỉ đếm số chấm trên thước để đo mực nước thì sai số rất lớn vì mỗi chấm cách nhau khá xa, không đo được chính xác theo đơn vị nhỏ hơn.

Dù còn nhiều vấn đề cần giải quyết khi áp dụng hệ thống vào thực tế, tuy nhiên trong phạm vi môn học thị giác máy tính, để đạt được những kết quả như trên, nhóm đã tìm hiểu, học hỏi được thêm về các kiến thức liên quan đến thị giác máy tính như là phát hiện vật thể hay là sử dụng được các công cụ xử lý ảnh truyền thống, những kiến thức đó tuy cơ bản nhưng lại là cơ sở để phát triển, giải quyết rất nhiều bài toán phức tạp khác trong lĩnh vực thị giác máy tính.

4.2 Hướng phát triển

Để có thể áp dụng vào thực tế, nhóm sẽ phải nâng độ chính xác của phần phát hiện thước lên bằng cách thử nghiệm thêm nhiều kỹ thuật và phương pháp phát hiện vật thể khác, ngoài ra việc thu thập thêm dữ liệu để huấn luyện cũng rất quan trọng trong việc nâng độ chính xác của mô hình. Về phần trích xuất thông tin trên thước, nhóm sẽ hướng đến việc trích xuất thông tin vạch chia trên thước để kết quả thu được có độ chính xác cao hơn, ngoài ra nhóm cũng sẽ thử áp dụng mô hình học sâu thay vì xử lý ảnh truyền thống để bài toán có thể tổng quát hóa.

TÀI LIỆU THAM KHẢO

- [1] P. Đ. Khánh,
"<https://phamdinhhkhanh.github.io/2020/03/09/DarknetAlgorithm.html>," [Online].
- [2] MachineLearningCoban, "<https://forum.machinelearningcoban.com/t/object-detection-yolo/503>," [Online].
- [3] Joseph Redmon, Ali Farhadi, "YOLOv3: An Incremental Improvement".
- [4] OpenCV, "https://docs.opencv.org/3.4/db/d8e/tutorial_threshold.html," [Online].
- [5] L. für, "<http://campar.in.tum.de/Chair/HaukeHeibelGaussianDerivatives>," [Online].