|  |
| --- |
| HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG  **VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG CDIT**  **---------------------------------------** |
|  |
| **TÀO NGỌC CƯỜNG** |
|  |
| **…**  **BÁO CÁO THỬ VIỆC**  **NGƯỜI HƯỚNG DẪN: TS. CAO MINH THẮNG** |
| Hà Nội – 2020 |

# **MỤC LỤC**

[**MỤC LỤC** 2](#_Toc32855214)

[**DANH MỤC CÁC TỪ VIẾT TẮT** 3](#_Toc32855215)

[**DANH MỤC CÁC KÝ HIỆU** 4](#_Toc32855216)

[**DANH MỤC CÁC BẢNG** 5](#_Toc32855217)

[**DANH MỤC CÁC HÌNH VẼ** 6](#_Toc32855218)

[**MỞ ĐẦU** 7](#_Toc32855219)

[**CHƯƠNG 1: NHẬN DẠNG ĐỒNG HỒ NƯỚC** 10](#_Toc32855220)

[**1.1. Mở đầu chương** 10](#_Toc32855221)

[**1.2. Nhận dạng vật thể sử dụng Pattern Recognition** 10](#_Toc32855222)

[**1.3. Nhận dạng vật thể sử dụng các phiên bản của Fast-YOLO** 10](#_Toc32855223)

[**1.4. Nhận dạng đối tượng sử dụng thuật toán AdaBoost** 11](#_Toc32855224)

[**1.5. Nhận dạng đối tượng sử dụng SSD** 11](#_Toc32855225)

[**1.6. Đánh giá các giải pháp cho nhận dạng đồng nước sử dụng bộ dữ liệu của ezWater.** 12](#_Toc32855226)

[**1.7. Kết luận** 12](#_Toc32855227)

[**CHƯƠNG 2: TÌM KIẾM VÙNG CHỈ SỐ ĐỒNG HỒ** 13](#_Toc32855228)

[**2.1. Mở đầu chương** 13](#_Toc32855229)

[**2.2. Sơ lược về một số phương pháp đã biết** 13](#_Toc32855230)

[**2.3. Phân loại cụ thể** 13](#_Toc32855231)

[**2.4. Sử dụng một số phương pháp lọc của xử lý ảnh số** 13](#_Toc32855232)

[**2.5. Mạng MultiNOD** 14](#_Toc32855233)

[**2.6. Sử dụng một hộp giới hạn** 14](#_Toc32855234)

[**CHƯƠNG 3: ĐỌC CHỈ SỐ ĐỒNG HỒ** 15](#_Toc32855235)

[**3.1. Mở đầu chương** 15](#_Toc32855236)

[**3.2. Sử dụng kỹ thuật tầm nhìn máy tính truyền thống** 15](#_Toc32855237)

[**3.3. Sử dụng thư viện Tesseract OCR** 15](#_Toc32855238)

[**3.4. Sử dụng bộ SDK của Anyline** 16](#_Toc32855239)

[**3.5. Sử dụng GOCR** 16](#_Toc32855240)

[**3.6. Sử dụng OpenCV** 16](#_Toc32855241)

[**3.7. Mạng CR-NET** 16](#_Toc32855242)

# **DANH MỤC CÁC TỪ VIẾT TẮT**

# **DANH MỤC CÁC KÝ HIỆU**

# **DANH MỤC CÁC BẢNG**

# **DANH MỤC CÁC HÌNH VẼ**

# **MỞ ĐẦU**

Hiện nay, đồng hồ nước rất phổ biến trong các hộ gia đình và trong công nghiệp để tính toán lượng nước tiêu thụ. Nó được sử dụng bởi các nhà cung cấp nước để tính toán lượng nước tiêu thụ của các hộ gia đình sử dụng dịch vụ cung cấp nước của nhà cung cấp đó. Đồng hồ nước có đa dạng các mẫu mã, từ các loại đồng hồ dạng cơ đo đạc lượng nước chảy qua nó và hiển thị trên màn hình các con số cho đến loại đồng hồ điện tử sử dụng cảm biến điện từ để đo các thông số và hiển thị qua một màn hình LED LCD. Thông thường, theo mỗi chu kỳ, các nhà cung cấp nước sẽ cử các nhân viên đi đến từng hộ gia đình sử dụng dịch vụ nước, đọc và ghi chép chỉ số đồng hồ và mang về nhà máy để tính toán lượng tiền mà khách hàng sẽ phải trả. Trong quá trình đọc và ghi chỉ số một cách thủ công như vậy mất nhiều thời gian và rất dễ xảy ra nhầm lẫn [8]. Để giảm thiểu sai sót cũng như tiết kiệm nhân lực, rất nhiều biện pháp khác nhau đã được đưa ra. Trong đó có các biện pháp tiêu biểu [4] như:

* Phương pháp sử dụng các loại đồng hồ điện tử thông minh: Phương pháp này sử dụng các đồng hồ điện tử hiện đại tiến hành đo chỉ số nước bằng cảm biến và gửi về cho nhà cung cấp nước một cách tự động. Có hai phương pháp tiêu biểu là:
  1. Đồng hồ thông minh sử dụng cảm biến điện từ: Loại đồng hồ này sử dụng cảm biến điện từ để đo chỉ số nước và gửi về cho nhà cung cấp nước một cách tự động.
  2. Đồng hồ thông minh sử dụng cảm biến điện dung: Đây là một loại đồng hồ thông minh được tạo ra bằng cách nhúng thêm một mạch điện vào bên trong màn hình của đồng hồ cơ, mạch này có tác dụng phát hiện ra vị trí mũi tên chỉ số của đồng hồ.

Mặc dù đồng hồ thông minh rất tiện lợi, có thể tự động hóa công việc đo và truyền chỉ số nước về cho nhà cung cấp, tuy nhiên giá thành của những loại đồng hồ này là khá đắt đỏ, hơn nữa việc thay thế hoàn toàn những đồng hồ cơ học đang được sử dụng thành loại đồng hồ này tiêu tốn nhiều tiền bạc và công sức. Ngoài ra, loại đồng hồ này còn phát sinh thêm nhiều vấn đề như vấn đề về bảo mật, môi trường đặt đồng hồ, … và khó sửa chữa khi bị lỗi.

* Sử dụng phương pháp xử lý ảnh để đọc chỉ số đồng hồ: Phương pháp này áp dụng những thành tựu trong lĩnh vực thị giác máy tính về việc phát hiện đối tượng và nhận dạng chữ viết trong điều kiện môi trường tự nhiên dựa trên ảnh chụp. Có hai phương pháp chính thường được sử dụng để đọc chỉ số đồng hồ nước:
  1. Đọc chỉ số đồng hồ nước dựa trên định vị mũi tên chỉ số đồng hồ: Phương pháp này sẽ nhận diện từng chỉ số của đồng hồ nước thông qua mũi tên chỉ số trên mặt đồng hồ. Phương pháp này có một vấn đề đặt ra là khi ảnh được chụp từ nhiều góc khác nhau, thì đồng hồ phải được điều chỉnh lại.
  2. Đọc chỉ số đồng hồ nước dựa trên nhận dạng chỉ số dạng số trên mặt đồng hồ: Để cho dễ đọc thì hầu như các đồng hồ nước đều hiển thị chỉ số đồng hồ dưới dạng một tập hợp của các bộ số từ 0 đến 9. Tuy nhiên, khi chỉ số đồng hồ tăng lên và hình ảnh được chụp có thể chứa hai nửa của các số liền kề nhau và chúng ta phải xác định được số đó.

Đọc chỉ số đồng hồ nước tự động dựa vào phương pháp xử lý ảnh giúp tiết kiệm được công sức của nhân công, giảm thiểu sai sót và ít tốn kém, dễ dàng để sửa đổi và tận dụng được các thiết bị sẵn có như điện thoại thông minh. Tuy nhiên, nó cũng có một số nhược điểm như vấn đề của ảnh chụp phụ thuộc vào độ sáng, độ nhòe, độ nhiễu, độ sạch của đồng hồ, … .

Để phù hợp với ứng dụng hỗ trợ các đơn vị phân phối nước sạch ezWater, trong phạm vi báo cáo này, chúng ta sẽ chỉ tìm hiểu về phương án đọc chỉ số đồng hồ nước dựa trên nhận dạng các bộ số đại diện cho chỉ số đồng hồ.

Cùng với sự phát triển của trí tuệ nhân tạo, cụ thể là lĩnh vực tầm nhìn máy tính, thì hệ thống ARM cũng là một trong số những ứng dụng được quan tâm và phát triển mạnh mẽ. Trong đó thì người ta chia nhiệm vụ xây dựng hệ thống ARM thành 3 nhiệm vụ con chính [12] đó là

1. Nhận dạng đồng hồ

2. Nhận dạng vùng chỉ số

3. Nhận dạng chỉ số chứa trong vùng

Trong đó với mỗi nhiệm vụ lại được sử dụng nhiều phương pháp khác nhau để xử lý, chúng sẽ được đề cập cụ thể trong từng chương bên dưới.

# **CHƯƠNG 1: NHẬN DẠNG ĐỒNG HỒ NƯỚC**

## **1.1. Mở đầu chương**

Trong chương 1 chúng ta sẽ nói về cách nhận dạng đồng hồ nước có trong một bức ảnh. Nhiệm vụ nhận dạng đồng hồ nước nói riêng hay nhận dạng vật thể trong một bức ảnh nói chung là một nhiệm vụ thiết yếu, yêu cầu một độ chính xác lớn, bởi vì chỉ khi có kết quả đúng trong bước này, các bước tiếp theo mới có thể được thực hiện. Nếu kết quả đưa ra sai lệch, rất có thể dẫn đến kết quả của các bước sau bị nhầm lẫn, mà thậm chí dẫn tới kết quả bài toán bị sai gây hậu quả nghiêm trọng.

Trong những mục 1.2 đến 1.5 chúng ta sẽ tiến hành dạo qua từng phương án cụ thể.

# **1.2. Nhận dạng vật thể sử dụng Pattern Recognition**

Pattern Recognition còn được gọi là machine recognition, computer recognition hay automatic recognition giúp cho một máy tính có thể học được cách định danh cho một vật một cách tự động. [7]

Mục đích của Pattern Recognition là để phân loại các đối tượng vật lý, sao cho kết quả càng gần với đối tượng càng tốt với ít lỗi nhất. Về cơ bản, phương pháp này nhận dạng một vật thể bằng cách tính toán một mẫu chung của vật thể bằng phương pháp tính toán độ tương tự giữa các đối tượng mẫu đã được xác định để tạo thành một mẫu tiêu chuẩn, và sau đó sử dụng mẫu tiêu chuẩn đó để đi tìm vật thể có độ tương tự gần nhất với nó. [7]

## **1.3. Nhận dạng vật thể sử dụng các phiên bản của Fast-YOLO**

YOLO là một phiên bản được phát triển dựa trên cơ sở của CNN trong đó thì Fast-YOLO là một phiên bản cải tiến của YOLO, điều chỉnh lại phương pháp phát hiện đối tượng như là một vấn đề hồi quy đơn, trực tiếp từ các pixel hình ảnh đến hộp giới hạn và xác định xác suất của lớp đó. Với phương châm chỉ cần nhìn một lần vào một hình ảnh để dự đoán các đối tượng là gì và đang ở đâu. YOLO nhìn bao quát toàn bộ bức ảnh trong thời gian đào tạo và học về sự xuất hiện của chúng dựa trên thông tin về ngữ cảnh [25] bởi thế nên YOLO rất nhanh chóng, được sử dụng nhiềucho các hệ thống thời gian thực. Nhưng trong bối cảnh phức tạp, YOLO có tỉ lệ chính xác không quá cao [5].

Ngoài Fast-YOLO còn có các phiên bản đáng chú ý của YOLO như YOLOv3 [26] và YOLO9000 [27] là những phiên bản nâng cấp sau này.

## **1.4. Nhận dạng đối tượng sử dụng thuật toán AdaBoost**

Thuật toán Adaboost là phương pháp kết hợp một tập hợp các phân loại yếu để tạo thành một phân loại mạnh (Chen và Youle, 2004) được xây dựng vào năm 1995 bởi Freund và Schapire và được mô tả trong bài giới thiệu của Freund và Schapire vào năm 1999 [15].

Nhận dạng đối tượng bằng thuật toán theo kỹ thuật boosting với thuật toán AdaBoost kết hợp với các bộ phân loại yếu sử dụng các đặc trưng Haar-like đã và đang được coi là một phương pháp hiệu quả, vừa đảm bảo tốc độ nhận dạng, vừa có độ chính xác cao [10].

Phương pháp này đem một hình ảnh chia nhỏ thành các ảnh con, mỗi ảnh được biểu thị bởi một vector đặc trưng. Sau đó, một bộ mặt nạ được đưa vào để trích xuất các đặc trưng này. Với mỗi mặt nạ, tổng cấp xám của các điểm ảnh trong vùng sáng trừ tổng cấp xám trong vùng tối được coi là một đặc trưng. Như vậy, với cách chia thành các ảnh con sẽ có nhiều hơn một triệu các đặc trưng và mỗi đặc trưng được tính toán rất nhanh [10]. Thuật toán được trình bày cụ thể trong các tài liệu [10] và [15].

## **1.5. Nhận dạng đối tượng sử dụng SSD**

Các hệ thống phát hiện đối tượng hiện đại thường có độ chính xác cao, tuy nhiên tốc độ phát hiện lại rất thấp. Ví dụ như Faster-RCNN là mạng được đánh giá có tính chính xác tốt nhất hiện nay nhưng tốc độ chỉ có 7 FPS.

Giải quyết vấn đề tốc độ, mạnh SSD được ra đời. Cách tiếp cận của SSD là dựa trên một mạng tích chập chuyển tiếp tạo ra một bộ sưu tập kích thước cố định của các hộp giới hạn và đánh giá điểm số cho sự hiện diện của các đối tượng xuất hiện trong hộp đó. Theo sau đó là các bước triệt tiêu không tối đa để tạo ra phát hiện cuối cùng. Kết quả của mạng được đánh giá với bộ dữ liệu PASCAL VOC 2007 so với mạng Faster R-CNN và YOLO là tốc độ 59 FPS với độ chính xác 74.3% đối với SSD, 45 FPS với độ chính xác 63.4% đối với mạng YOLO [28].

## **1.6. Đánh giá các giải pháp cho nhận dạng đồng nước sử dụng bộ dữ liệu của ezWater.**

## **1.7. Kết luận**

# **CHƯƠNG 2: TÌM KIẾM VÙNG CHỈ SỐ ĐỒNG HỒ**

## **2.1. Mở đầu chương**

Sau khi đã tìm kiếm được vị trí của đồng hồ, tiếp theo là quá trình xác định vị trí hiển thị của chỉ số bên trong mặt đồng hồ. Thực tế có thể xem chỉ số đồng hồ là một dạng thực thể như mặt đồng hồ, và có thể sử dụng lại các biện pháp của chương trước. Do đó, một vài phần đã được mô tả ở chương trước sẽ chỉ được giới thiệu qua ở chương này.

## **2.2. Sơ lược về một số phương pháp đã biết**

Chúng ta có thể áp dụng lại các phương pháp đã biết đó là sử dụng mạng Fast-YOLO, SSD, sử dụng thuật toán AdaBoost. Khi sử dụng mạng nơ ron, ta hoàn toàn có thể gộp hai chương 1 và hai vào làm một, bởi vì các mạng trên hoàn toàn có thể đào tạo để nhận ra nhiều đối tượng. Ngoài ra, với thuật toán AdaBoost, có thể sử dụng so sánh khớp mẫu như trong tài liệu [15] mô tả.

## **2.3. Phân loại cụ thể**

Phân loại cụ thể được đề xuất trong [6]. Tại đây người ta định hình phần mặt tiền của một xe bus với các vùng nổi bật thành một mẫu chung. Tiếp sau đó, bằng cách cho máy tính học mẫu này, máy tính có thể hiểu được các thành phần trong mẫu. Tuy nhiên, phương pháp này khá hạn chế khi mẫu này được áp dụng cho chỉ một góc độ với một mẫu nhất định. Trong môi trường đồng hồ nước, các mẫu mã đồng hồ có thể khác nhau từ nhỏ đến lớn, và rất khó để khái quát thành mẫu chung như trong bài viết này.

## **2.4. Sử dụng một số phương pháp lọc của xử lý ảnh số**

Trong tài liệu [12], một quy trình xử lý ảnh đã được đưa ra để trích xuất vùng chỉ số của đồng hồ nước. Trong quy trình gồm có các bước như sau:

1. Chuyển ảnh về dạng đa mức xám

2. Lọc homormophic để tăng độ tương phản, làm rõ nét các vùng của ảnh.

3. Biến đổi hình thái Top Hat được sử dụng, làm nổi lên chi tiết của ảnh.

4. Biến đổi ảnh về nhị phân với phương thức Otsu, ảnh được chia làm hai màu đen và trắng.

5. Biến đổi hình thái Closing nhằm trích xuất viền của các khu vực.

6. Vùng chỉ số được cắt rời và được nhị phân hóa bởi phương thức Otsu

Phương pháp này tuy dễ dàng để thực hiện, tuy nhiên với nhiều loại đồng hồ nước khác nhau, mỗi một loại lại cần một quá trình xử lý riêng biệt, và với ảnh chụp đồng hồ có chất lượng khác nhau, áp dụng một quy trình duy nhất để xử lý có thể dẫn đến các kết quả sai lệch. Một số quy trình tương tự có trong [8] và [7].

## **2.5. Mạng MultiNOD**

Mạng multiNOD được đề xuất tại [2] nhằm tìm kiếm vùng chỉ số của đồng hồ khí gas. Điều thú vị ở mạng này, đó là với mỗi node trong mạng, chúng ta có thể sử dụng một thuật toán học máy để làm node cho mạng, hoặc thậm chí có thể cấu hình mỗi node mạng là một thuật toán khác nhau. Ngoài ra, thuật toán cũng áp dụng nhiều phương pháp lọc khác nhau để trích xuất đặc trưng đầu vào chứ không sử dụng một loại nhất định. Kết quả của mạng rất tốt, với bộ dữ liệu đa dạng có nhiều ảnh bị vỡ, nhòe nhưng cho kết quả nhận dạng khá tốt, lên tới 95%.

## **2.6. Sử dụng một hộp giới hạn**

Phương pháp này thường được xây dựng ở ứng dụng di động, bằng cách tạo ra một hộp giới hạn hình chữ nhật, chúng ta yêu cầu nhân viên chốt nước phải đặt chính xác vùng có chứa chỉ số vào trong vùng giới hạn. Phương án này đã được thực hiện và được bán như một sản phẩm công nghiệp bởi Anyline [18].

# **CHƯƠNG 3: ĐỌC CHỈ SỐ ĐỒNG HỒ**

## **3.1. Mở đầu chương**

Cuối cùng là đọc chỉ số của đồng hồ nước và chuyển đổi nó thành dạng chữ. Đây là một nhiệm vụ OCR (Optical Character Recognition) là một trong những nhiệm vụ thị giác máy tính được giải quyết sớm nhất vì một số khía cạnh nó không đòi hỏi sử dụng học sâu. Do đó, có những OCR khác nhau đã được triển khai kể từ năm 1914 trước khi có sự bùng nổ của học sâu vào năm 2012 [22]. Thông qua phần 3.2 đến 3.8, chúng ta sẽ cùng đi qua một vài biện pháp tiêu biểu.

## **3.2. Sử dụng kỹ thuật tầm nhìn máy tính truyền thống**

Có nhiều ví dụ chúng ta có thể tìm thấy ở trên mạng, ví dụ như Adrian Rosebrook có một số hướng dẫn trên trang web của anh ấy, tiêu biểu như hướng dẫn về đọc số tài khoản ngân hàng tại [23] và [24].

Cụ thể, kỹ thuật này có cách tiếp cận như sau:

1. Áp dụng bộ lọc để tách ký tự khỏi nền

2. Tìm kiếm đường viền để nhận dạng từng ký tự một

3. Áp dụng phân loại hình ảnh để xác định các ký tự

Tuy nhiên, việc phát hiện đường viền là khó khăn để khái quát hóa, nó đòi hỏi nhiều tính thủ công, do đó không khả thi trong hầu hết các vấn đề và nó khó để nhận dạng khi các ký tự được xếp gần nhau [22].

## **3.3. Sử dụng thư viện Tesseract OCR**

Tesseract được phát triển tại các phòng thí nghiệm ở Hewlett-Packard, Bristol và tại Hewlett-Packard Co, Greeley Colorado giữa năm 1985 và 1994, với vài thay đổi vào năm 1996 để có thể sử dụng được trên Windows. Vào năm 2005, Tesseract đã được công bố mã nguồn bởi HP. Vào khoảng năm 2006 nó đã được phát triển bởi Google [20].

Tesseract có thể kết hợp với nhiều ngôn ngữ khác nhau và là một bộ nhật dạng ký tự khá mạnh mẽ và ngày nay vẫn được sử dụng nhiều. Tuy vậy, kết quả nhận dạng 332 chữ số trong một tài liệu [2] chỉ là 58,73%.

## **3.4. Sử dụng bộ SDK của Anyline**

Anyline cung cấp một bộ công cụ cho ứng dụng hoặc trang web, cho phép quét và xử lý các ký tự chữ và số. Một vài ứng dụng tiêu biểu có thể sử dụng Anyline như: quét tài liệu, quét giấy phép lái xe, quét chứng minh nhân dân, đọc chỉ số đồng hồ, … .[18]

Anyline cung cấp các bộ công cụ hỗ trợ phát triển ứng dụng trên các nền tảng như: Javascript, API, Android và IOS, Windows, … [18]. Tuy nhiên, với mỗi nền tảng, chúng ta đều phải trả phí để mua bộ công cụ cho nền tảng đó.

## **3.5. Sử dụng GOCR**

GOCR là một chương trình OCR, được phát triển dưới giấy phép dành cho cộng đồng GNU, nó chuyển chữ viết trong hình ảnh thành dạng text. Joerg Schulenburg đã bắt đầu chương trình và lãnh đạo nhóm các nhà phát triển trên SF, và sau năm 2010 vẫn quản lý các gói ở mức cơ bản. [21]

GOCR không được nhắc đến nhiều, tuy nhiên chúng ta vẫn có thể tìm thấy một số hướng dẫn như trong [29] và ví dụ thông qua trang chủ của nó [21].

## **3.6. Sử dụng OpenCV**

OpenCV (Open Source Computer Vision Library) là một thư viện mã nguồn mở cung cấp hàng trăm thuật toán trong lĩnh vực thị giác máy tính. Ngoài các thuật toán để xử lý ảnh, openCV còn cung cấp các phương pháp cho lĩnh vực OCR như sử dụng mạng CNN, KNN, EAST. OpenCV hỗ trợ hầu hết các ngôn ngữ hiện đại như Java, C#, C++, Python, Javascript, … Trang chủ của OpenCV tại [19].

## **3.7. Mạng CR-NET**

CR-NET là một mô hình dựa trên nền tảng của YOLO được đề xuất để phát hiện và nhận dạng cho biển số xe. Mô hình này gồm mười một lớp YOLO đầu tiên và bốn lớp tích chập được thêm vào để cải thiện khả năng phi tuyến tính. Trong tài liệu [9], một CR-NET được sử dụng để đọc chỉ số đồng hồ nước, kết quả nhận dạng chính xác xấp xỉ 94.13% với tốc độ 475 FPS.

**TÀI LIỆU THAM KHẢO**

1. Mehdi Chouiten, Peter Schaeffer (2014) “*Vision based mobile Gas-Meter Reading*”
2. Angelo Nodari, Ignazio Gallo (2011) “*A Multi-Neural Network Approach to Image Detection and Segmentation of Gas Meter Counter*”
3. Jaco Marais, Reza Malekian, Ning Ye, and Ruchuan Wang (2016) “*A Review of the Topologies Used in Smart Water Meter Networks: A Wireless Sensor Network Application*”
4. Rayson Laroca, Evair Severo, Luiz A. Zanlorensi, Luiz S. Oliveira, Gabriel Resende
5. Goncalves, William Robson Schwartz and David Menotti “*A Robust Real-time Automatic License Plate Recognition Based on the YOLO Detector*”
6. Claudio Guida, Dario Comanducci, Carlo Colombo “*Automatic bus line number localization and recognition on mobile phones: A computer vision aid for the visually impaired*”
7. Kaicheng Xie (2007) “*Automatic utility meter reading*”
8. Lamiaa A.Elrefaei, Asrar Bajaber, Summayyah Natheir, Nada AbuSanab, Marwa Bazi (2015) “*Automatic Electricity Meter Reading Based on Image Processing*”
9. Rayson Laroca, Victor Barroso, Matheus A.Diniz, Gabriel R.Goncalves, William Robson Schwartz, David Menotti (2019) “*Convolutional Neural Networks for Automatic Meter Reading*”
10. Phạm Hồng Ngự (2009) “*Nhận dạng đối tượng sử dụng thuật toán Adaboost*”
11. Ricardo Ocampo-Vega and Gildardo Sanchez-Ante (2014) “*Automatic Reading of Electro-mechanical Utility Meters*”
12. N Jawas and Indrianto (2017) “*Image based automatic water meter reader”*
13. Chunshan Li, Yukun Su, Rui Yuan, Dianhui Chu and Jinhui Zhu (2017) “*Light-weight Spliced Convolution Network based Automatic Water Meter Reading in Smart City*”
14. Shinu Nelson and K. Vetrivelselvi (2014) “*Modelling and Analysis of Hybird Water Meter using Arrow Pointer Sensor*”
15. Pascal Ouma Nyapoto (2013) “*A text recognition system for reading meters*”
16. Gabriel R.Goncalves, Matheus A. Diniz, Rayson Laroca, David Menotti, William Robson Schwartz “*Real-time Automatic License Plate Recognition Through Deep Multi-Task Networks*”
17. Yunze Gao, Chaoyang Zhao, Jinqiao Wang and Hanqing Lu “*Automatic Watermeter Digit Recognition on Mobile Devices*”
18. Anyline “[*https://anyline.com*/](https://anyline.com/)”
19. Opencv “[*https://opencv.org/*](https://opencv.org/)”
20. Tesseract “[*https://github.com/tesseract-ocr/tesseract*](https://github.com/tesseract-ocr/tesseract)”
21. GOCR “[*http://jocr.sourceforge.net/*](http://jocr.sourceforge.net/)”
22. Gidi Shperber “*A gentle introduction to OCR -* [*https://towardsdatascience.com/a-gentle-introduction-to-ocr-ee1469a201aa*](https://towardsdatascience.com/a-gentle-introduction-to-ocr-ee1469a201aa)”
23. Adrian Rosebrock “*Credit card OCR with OpenCV and Python -* [*https://www.pyimagesearch.com/2017/07/17/credit-card-ocr-with-opencv-and-python/*](https://www.pyimagesearch.com/2017/07/17/credit-card-ocr-with-opencv-and-python/)”
24. Adrian Rosebrock “*Bank check OCR with OpenCV and Python (Part I) -* [*https://www.pyimagesearch.com/2017/07/24/bank-check-ocr-with-opencv-and-python-part-i/*](https://www.pyimagesearch.com/2017/07/24/bank-check-ocr-with-opencv-and-python-part-i/)”
25. Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi “*You Only Look Once: Unified, Real-time Object Detection*”
26. Joseph Redmon, Ali Farhadi “*YOLOv3: An Incremental Improvement*”
27. Joseph Redmon, Ali Farhadi “YOLO9000: Better, Faster, Stronger”
28. Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christan Szegedy, Scott Reed, Cheng-Yang Fu, Alexander C. Berg “*SSD: Sing le Shot MultiBox Detector*”
29. Joerg Schulenburg (2005) “*GOCR Optical Character Recognition*”