

DẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH



ĐỒ ÁN MÔN HỌC THIẾT KẾ LUẬN LÝ

Đếm vật thể trên băng chuyền qua Camera

GVHD: Trần Thanh Bình.

Lớp: L02 - Nhóm: 3

Sinh viên thực hiện: Lê Duy Hào – 2011142.

Nguyễn Tuyết Vy – 2012458.

Phạm Thị Hiền – 2011203.

THÀNH PHỐ HỒ CHÍ MINH, THÁNG 2/2023



Nội dung

1	Tổng quan đồ án	2
1.1	Giới thiệu đề tài	2
1.2	Yêu cầu chức năng	2
2	Công nghệ sử dụng	3
2.1	Teachable Machine	3
2.1.1	Teachable Machine là gì?	3
2.1.2	Tại sao chọn Teachable Machine?	3
2.1.3	Teachable Machine hoạt động như thế nào?	3
2.2	Flask	4
2.2.1	Flask là gì ?	4
2.2.2	Tại sao nên chọn Flask ?	4
2.2.3	Những lợi ích của Flask	5
2.2.4	Những tính năng của Flask	5
3	Thiết kế hệ thống	6
3.1	Dụng cụ và linh kiện sử dụng	6
3.1.1	Băng chuyền	6
3.1.2	Camera(B) 5MP Cho Raspberry Pi	6
3.1.3	Raspberry Pi 4 Model B 4GB	7
3.1.4	Màn Hình LCD Cảm Ứng HDMI Raspberry Pi	8
3.2	Nhận diện vật thể	9
3.2.1	Vật thể nhận diện	9
3.2.2	Xử lý dữ liệu	9
4	Kết quả hiện thực	12
5	Tổng kết	13
5.1	Điểm đạt được	13
5.2	Điểm chưa đạt	14
6	Thành viên & Phân công công việc	15



1 Tổng quan đồ án

1.1 Giới thiệu đề tài

Tại các hộ kinh doanh, doanh nghiệp sản xuất hàng hóa với số lượng lớn, trong quy trình sản xuất, việc quản lý sản phẩm thường xoay quanh các vấn đề như: thông tin, số lượng, phân loại, đóng gói, vận chuyển sản phẩm,...

Để giải quyết các vấn đề trên, các nhà sản xuất thường sử dụng đến "hệ thống băng tải dầm sản phẩm" khi kết thúc quá trình sản xuất để đo lường số lượng sản phẩm, năng suất, cũng như dựa vào đó để có kế hoạch sản xuất và đánh giá mức độ chất lượng thật chính xác.

Với đồ án này, nhóm đã chọn đề tài “Dém vật thể trên băng chuyền thông qua Camera” để hiện thực lại hệ thống băng tải trên với quy mô nhỏ, nhóm sẽ thực hiện nhận diện các đồ dùng học tập trên băng chuyền thông qua camera gắn với Raspberry Pi4 và phân loại chúng ở cuối băng chuyền bằng động cơ Servo.

1.2 Yêu cầu chức năng

#	Yêu cầu	Mô tả
1	Băng chuyền chắc chắn	Tốc độ của động cơ băng chuyền ổn định (không quá nhanh hay quá chậm) để đảm bảo camera ghi được hình ảnh đủ rõ, băng chuyền chịu được tải tối đa là 300g, vật thể nằm ổn định trên băng chuyền trong quá trình di chuyển đến cuối băng chuyền.
2	Camera ghi hình rõ vật thể	Hình ảnh thu được với tốc độ màn trập tối thiểu 1/30, độ phân giải tối thiểu là 720p
3	Hệ thống nhận diện được vật thể với độ chính xác cao	Tổng số lượng vật thể được nhận diện đúng loại $\geq 80\%$



2 Công nghệ sử dụng

2.1 Teachable Machine

2.1.1 Teachable Machine là gì?

- Teachable Machine là công cụ nền web của Google Creative Lab + PAIR kết hợp cùng studio thiết kế Stoj và Use All Five cho phép mọi người tạo ra các mô hình học máy (machine learning model) một cách đơn giản và dễ dàng. Ngoài ra model còn có thể được xuất ra để sử dụng cho website, web app, app điện thoại,...
- Teachable machine sử dụng TensorFlow.js để tiến hành training ngay trên browser của người dùng. Nghĩa là dữ liệu không hề gửi đến server của google mà sử dụng chính máy tính để training. Người dùng có thể train máy tính để nhận diện hình ảnh, âm thanh, dán người láy từ những hình ảnh có sẵn hay từ chính webcam của người dùng.
 - **TensorFlow.js** là một thư viện cho machine learning trong Javascript để train và chạy model người dùng làm trên website hay app.
 - TensorFlow.js có thể chạy những model có sẵn, train lại model hoặc xây dựng và huấn luyện các mô hình trực tiếp bằng JavaScript bằng API linh hoạt và trực quan.

2.1.2 Tại sao chọn Teachable Machine?

- **Dễ sử dụng:** Teachable Machine cung cấp cho người dùng trang web dễ tương tác và có tutorial cho từng bước để có thể tạo ra một model hoàn chỉnh.
- **Tích hợp sẵn:** Người dùng có thể trực tiếp sử dụng Teachable Machine trên web để tạo ra model mà không cần thêm bất kỳ thư viện nào.
- **Linh hoạt:** Model được tạo ra là TensorFlow.js, do đó người dùng có thể xuất chúng ra và sử dụng tại bất kỳ đâu chạy Javascript như Glitch, P5.js, Node.js,... Ngoài ra người dùng còn có thể xuất model ra với những format khác để sử dụng trong những môi trường và nền tảng khác như Coral, Arduino,...
- **Tùy chỉnh:** Teachable Machine cung cấp cho người dùng frame code sẵn và người dùng có thể tải xuống máy tính và tùy chỉnh cho phù hợp với nhu cầu của mình.

2.1.3 Teachable Machine hoạt động như thế nào?

- Sử dụng Teachable Machine gồm 3 bước cơ bản: thu thập mẫu dữ liệu, train và xuất model. 3 bước này đều có thể làm được ngay trên web.
- Những mô hình này sử dụng một kỹ thuật gọi là transfer learning. Có một pretrained neural network và khi tạo các class của riêng mình, có thể hình dung các lớp này sẽ trở thành lớp hoặc bước cuối cùng của neural network. Cụ thể, cả model hình ảnh và tư thế đều học hỏi từ các mô hình mobilenet được đào tạo trước và mô hình âm thanh được xây dựng trên Speech Command Recognizer.



2.2 Flask

2.2.1 Flask là gì ?

- Flask là một web framework, một Python module được ứng dụng trong thiết kế website đơn giản và tạo lập các ứng dụng cho những trang web lớn và phức tạp. Flask là một microframework không bao gồm ORM nên mang lại khả năng kết nối với các tiện ích mở rộng.
 - **Web framework** là các framework ứng dụng web, đại diện cho một tập hợp các thư viện và module cho phép các nhà phát triển ứng dụng web viết ứng dụng mà không cần quan tâm về các chi tiết cấp thấp như giao thức (protocol), quản lý luồng (thread management), v.v.
- Flask dựa trên bộ công cụ Werkzeug WSGI và template engine Jinja2. Hiện framework này đã trở thành công cụ chính trong phát triển ứng dụng web của Python
 - **Werkzeug**: Werkzeug là một bộ công cụ WSGI thực hiện các yêu cầu, đối tượng phản hồi và các chức năng tiện ích. Điều này cho phép một web framework được xây dựng trên đó. Flask framework sử dụng Werkzeug làm một trong những cơ sở của nó.
 - **WSGI**: Giao diện cổng máy chủ web (Web Server Gateway Interface, WSGI) đã được sử dụng như một tiêu chuẩn để phát triển ứng dụng web Python. WSGI là đặc điểm kỹ thuật của giao diện chung giữa máy chủ web và ứng dụng web.
 - **jinja2**: jinja2 là một template engine phổ biến cho Python. Hệ thống web template kết hợp một template với một nguồn dữ liệu cụ thể để hiển thị một trang web động. Điều này cho phép bạn chuyển các biến Python vào các HTML template.
- Nó được thiết kế để giữ cho cốt lõi của ứng dụng đơn giản và có thể mở rộng (scalable). Thay vì sử dụng một abstraction layer để hỗ trợ cơ sở dữ liệu, Flask hỗ trợ các phần extensions để thêm các khả năng như vậy vào ứng dụng.

2.2.2 Tại sao nên chọn Flask ?

- **Tùy biến linh hoạt**: Với Flask Python, lập trình viên có thể sử dụng thư viện, công cụ hoặc các cơ chế trong lập trình, phát triển ứng dụng. Ngoài ra, framework này còn hỗ trợ lập trình viên trong sử dụng tiện ích mở rộng nhằm tích hợp thêm tính năng cho ứng dụng.
- **Là micro web framework**: Flask Python không cần công cụ hoặc thư viện cụ thể, mang đến lõi chức năng tối giản nhưng có thể mở rộng cho các ứng dụng web. Nhiều thành phần tiện ích như xác thực biểu mẫu, tích hợp CSDL, email,... có thể đưa vào web bất cứ lúc nào.
- **Dễ cài đặt, triển khai**: Cách cài đặt Flask Python và các thao tác của nó khá dễ dàng. Chỉ cần vài dòng lệnh là có thể xây dựng các ứng dụng đơn giản.
- **Là lựa chọn thông minh**: Framework này hỗ trợ xây dựng các API, web services, các ứng dụng web application cỡ vừa và nhỏ.
- **Hỗ trợ tập trung xây dựng các ý tưởng và mục tiêu riêng**: Flask Python cho phép tập trung chủ yếu vào khâu xác định các ý tưởng, mục tiêu riêng cho ứng dụng web nhờ vào các đặc điểm như linh hoạt, tối giản, định tuyến dễ dàng, dễ mở rộng,...



- **Được sử dụng phổ biến:** Flask Python được sử dụng trong phát triển khá nhiều ứng dụng nổi tiếng như Pinterest, LinkedIn, Netflix,... vì phổ biến nên người sử dụng Flask Python đã là một cộng đồng lớn, việc tìm kiếm nguồn tham khảo trở nên rất dễ dàng.

2.2.3 Những lợi ích của Flask

- Tốc độ hoạt động cực nhanh
- Có khả năng hỗ trợ NoQuery
- Trình duyệt được nhúng sẵn trình gõ rối
- Sử dụng các mã ngắn, đơn giản của Python. Cho phép biên dịch mô-đun, thư viện, giúp việc lập trình nhanh chóng, dễ dàng hơn và không cần gõ code bậc thấp
- Ngăn chặn các rủi ro về bảo mật khi lập trình web do ít phụ thuộc vào bên thứ ba.
- Có khả năng kiểm soát mọi vấn đề khi dùng Flask.
- Có khả năng tương thích với các công cụ nhờ vào Google.
- Thiết kế gọn nhẹ và sở hữu tính modular

2.2.4 Những tính năng của Flask

- Phát triển máy chủ, các trình gõ lỗi
- Có API đặc đáo và mạch lạc, Jinja2 templates, RESTful request dispatch, Full WSGI compliant, ORM-agnostic.
- Hỗ trợ kiểm thử đơn vị, bảo mật cho cookie, xử lý HTTP request



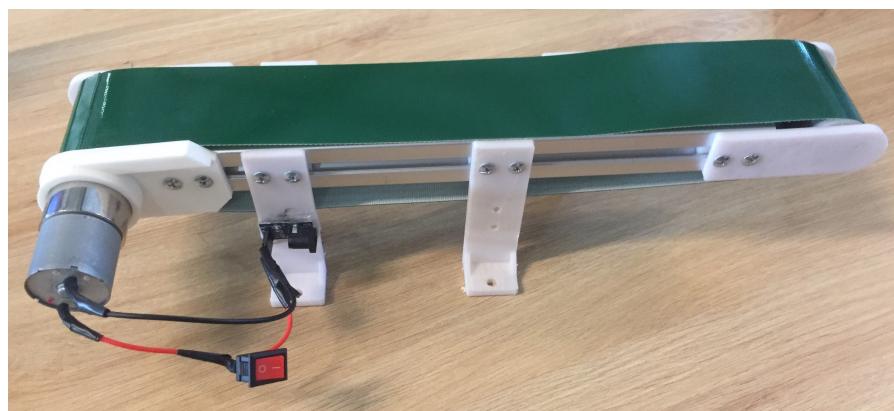
3 Thiết kế hệ thống

3.1 Dụng cụ và linh kiện sử dụng

3.1.1 Băng chuyền

- Phân loại: băng tải PVC
- Kích thước: 50cm x 6cm x 10cm
- Động cơ: động cơ giảm tốc DC 24V

Hình ảnh thực tế:



Hình 1: Băng chuyền

3.1.2 Camera(B) 5MP Cho Raspberry Pi

- Cảm biến 5 megapixel OV5647
- Điều chỉnh khoảng cách lấy nét
- Thông số kỹ thuật camera
- Kích thước CCD: 1 / 4inch
- Khẩu độ (F): 2.0
- Tiêu cự: có thể điều chỉnh, tối đa 6mm
- Góc nhìn (đường chéo): 60,6 độ
- Cảm biến độ phân giải tốt nhất: 1080p
- Kích thước: 32mm x32mm



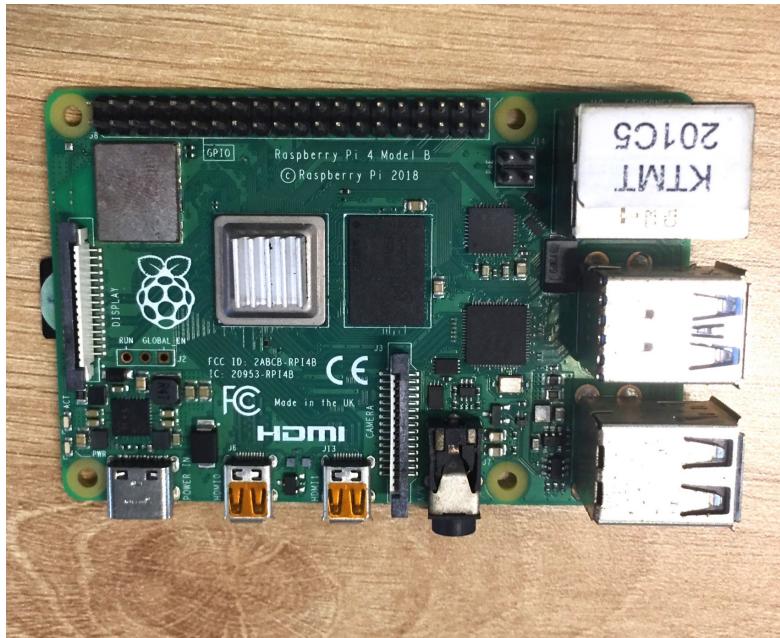
Hình ảnh thực tế:



Hình 2: Camera Raspberry Pi 5MP

3.1.3 Raspberry Pi 4 Model B 4GB

- Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
- RAM: 4GB LPDDR4-2400 SDRAM
- Wifi chuẩn 2.4 GHz và 5.0 GHz IEEE 802.11ac. Bluetooth 5.0, BLE
- Cổng mạng Gigabit Ethernet
- 2 cổng USB 3.0 và 2 cổng USB 2.0
- Hỗ trợ 2 cổng ra màn hình chuẩn Micro HDMI với độ phân giải lên tới 4K



Hình 3: Raspberry Pi 4

3.1.4 Màn Hình LCD Cảm Ứng HDMI Raspberry Pi

- Model: Waveshare 7inch Capacitive Touch Screen LCD (C), 1024×600, HDMI, IPS, Low Power
- Kích thước đường chéo: 7inch
- Giao tiếp: HDMI
- Độ phân giải: 1024x600pixels
- Loại cảm ứng: Điện dung
- Điện áp cấp: 5Vdc
- Chế độ hiển thị: Transmissive
- Kích thước: 164.9x124.27mm
- Nhiệt độ hiển thị: -10 → +60°C

Hình ảnh thực tế:



Hình 4: Màn hình LCD cảm ứng HDMI

3.2 Nhận diện vật thể

3.2.1 Vật thể nhận diện

Vật thể nhận diện gồm:

- Gấu bông nhỏ
- Hộp giấy
- Chai lọ

Tập dữ liệu dùng để nhận diện: [Link Google Drive](#)

3.2.2 Xử lý dữ liệu

Thiết lập Raspberry Pi 4

Gắn camera vào Raspberry Pi và tiến hành Enable Camera và test thử hình ảnh trong terminal. Nếu camera hiện lên thì camera đã được enable.

```
1 libcamera-hello
```

Nhóm sử dụng PiCamera2 để mở camera, dùng Teachable Machine với Tensorflow Lite, model Coral, Quantized.

Mở camera:

```
1 cv2.startWindowThread()
2 cap = Picamera2()
3 cap.configure(cap.create_preview_configuration(main={"format":
4     'RGB888', "size": (640, 480)}))
5 cap.start()
```

Nhận diện bằng 2 file .tflite và labels.txt:



```
1      # the TFLite converted to be used with edgetpu
2 modelPath = 'model.tflite'
3
4 # The path to labels.txt that was downloaded with your model
5 labelPath = 'labels.txt'
6
7 # This function takes in a TFLite Interpreter and Image, and
8 # returns classifications
9 def classifyImage(interpreter, image):
10    size = common.input_size(interpreter)
11    #//size = (1, 480, 640, 4)
12    common.set_input(interpreter, cv2.resize(image, size, fx=0, fy
13 =0,
14                               interpolation=cv2.INTER_CUBIC)
15 )
16 #common.set_input(interpreter, cv2.resize(image, size))
17 interpreter.invoke()
18 return classify.get_classes(interpreter)
```

Đếm vật thể

```
1      #Count object, reduce noise
2          a[0] = a[1]
3          a[1] = a[2]
4          a[2] = a[3]
5          a[3] = a[4]
6          #a[4] = a[5]
7          #a[5] = a[6]
8          #a[6] = a[7]
9          a[4] = results[0].id
10         if a[0] == a[1] and a[1] == a[2] and a[2] == a[3] and a[3]
11 == a[4]:== a[7]:
12             if(a[5] != a[4]):
13                 a[5] = a[4]
14             if(a[4] != 3):
15                 global count
16                 count = count + 1
17
18 print(count)
```

Sau đó xuất kết quả ra biến results bằng tại hàm main() và hiển thị màn hình bằng lệnh imshow() của openCV:

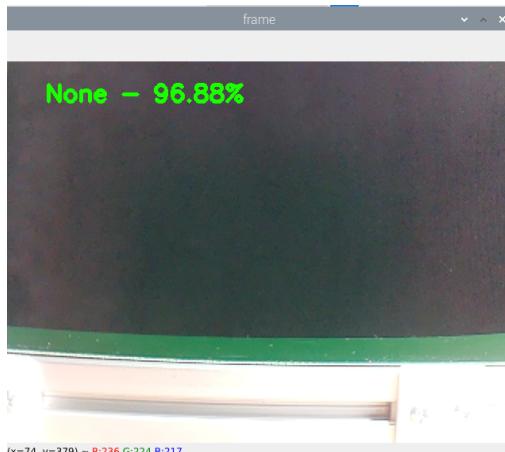
```
1      while True:
2          frame = cap.capture_array()
3          #//if not ret:
4          #//      break
5
6          # Flip image so it matches the training input
7          frame = cv2.flip(frame, 1)
8
9          # Classify and display image
```



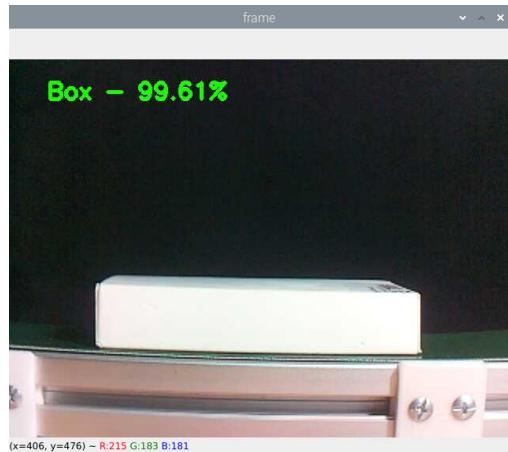
```
10     results = classifyImage(interpreter, frame)
11 #Count object
12 #number = countObject(f'{labels[results[0].id]}', number)
13 # Text
14 text = f'{labels[results[0].id]} - {round(results[0].score
15 *100,2)}%'
16 org = (30, 50)
17 fontFace = cv2.FONT_HERSHEY_SIMPLEX
18 fontScale = 0.8
19 color = (0, 255, 25)
20 lineType = cv2.LINE_4
21
22 img_text = cv2.putText(frame, text, org, fontFace,
23 fontScale, color, lineType)
24
25 #Text 2
26 text2 = f'Number of objects: {count}'
27 org2 = (30, 100)
28 fontFace2 = cv2.FONT_HERSHEY_SIMPLEX
29
30 img_text = cv2.putText(img_text, text2, org2, fontFace2,
31 fontScale, color, lineType)
32
33 #show screen
34 cv2.imshow('frame', img_text)
35 print(f'Label: {labels[results[0].id]}, Score: {results
36 [0].score}')
37 if cv2.waitKey(1) & 0xFF == ord('q'):
38     break
39
40 cv2.destroyAllWindows()
41
42 if __name__ == '__main__':
43     main()
```



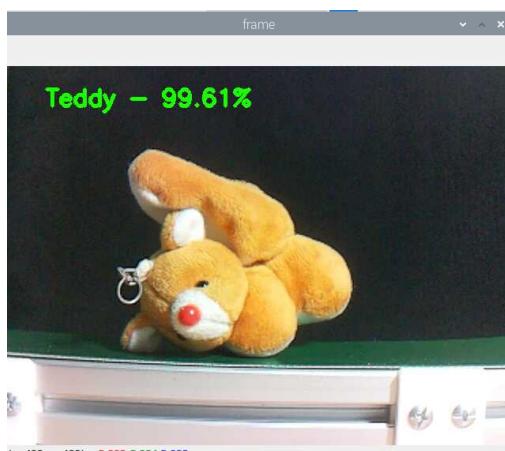
4 Kết quả hiện thực



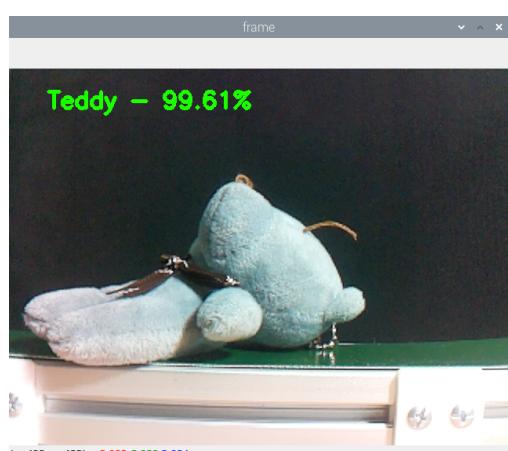
(a) Nhận diện băng chuyền không có vật thể



(b) Nhận diện hộp giấy



(a) Nhận diện gấu bông



(b) Nhận diện gấu bông



Hình 7: Nhận diện chai lọ



Hình 8: Đếm vật thể

5 Tổng kết

5.1 Điểm đạt được

- Băng chuyền chắc chắn, chạy với tốc độ ổn định



- Camera ghi hình rõ nét vật thể
- Nhận diện vật thể với độ chính xác cao >95%
- Độ trễ khung hình khi chạy model thấp <1s
- Đếm được vật thể đi qua băng chuyền tương đối chính xác
- Sửa được các lỗi của hệ thống để model chạy nhận diện nhanh và ổn định

5.2 Điểm chưa đạt

- Chưa có phần phân loại vật thể ở đầu cuối băng chuyền
- Chưa build được Web để tương tác với người dùng
- Số lượng vật thể còn ít
- Việc đếm có thể bị sai lệch khi nhận diện sai
- Còn bị vướng nhiều các lỗi phần cứng và cài đặt: Raspberry Pi 4, setup Camera, cài đặt thư viện,...



6 Thành viên & Phân công công việc

STT	Họ và tên	MSSV	Công việc	Hoàn thành
1	Nguyễn Tuyết Vy	2012458	Setup hệ thống, tìm hiểu về Flask, viết báo cáo	70%
2	Phạm Thị Hiền	2011203	Setup hệ thống, thiết lập và hiện thực việc nhận diện trên Raspberry, viết báo cáo	100%
3	Lê Duy Hào	2011142	Tìm model nhận diện và Flask, hiện thực việc nhận diện trên Raspberry	100%



Tham khảo

- [1] [Install OpenCV 4.5 on Raspberry 64 OS](#)
- [2] [tf.lite.Interpreter](#)
- [3] [V4L2 Drivers](#)
- [4] [RPi Camera \(B\)](#)
- [5] [The PiCamera2 Document](#)
- [6] [Raspberry Pi 3 vs Raspberry Pi 4 Performance with TensorFlow, TF Lite, Coral USB Accelerator](#)
- [7] [Raspberry Bulleye Camera Document](#)