

BỘ THÔNG TIN VÀ TRUYỀN THÔNG
HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG



BÁO CÁO THỰC TẬP **TỐT NGHIỆP ĐẠI HỌC**

Đề tài: Thực hành phát triển ứng dụng Android trên
camera giám sát

Người hướng dẫn : ThS. LÊ MINH HOÁ
Sinh viên thực hiện : PHAN THỊ TRÀ GIANG
Mã số sinh viên : N19DCPT015
Lớp : D19CQPU01 – N
Khoá : 2019 - 2024
Hệ : ĐẠI HỌC CHÍNH QUY

TP.HCM, tháng 08/2023

BỘ THÔNG TIN VÀ TRUYỀN THÔNG
HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG



BÁO CÁO THỰC TẬP TỐT NGHIỆP ĐẠI HỌC

Đề tài: Thực hành phát triển ứng dụng Android trên
camera giám sát

Người hướng dẫn : ThS. LÊ MINH HOÁ
Sinh viên thực hiện : PHAN THỊ TRÀ GIANG
Mã số sinh viên : N19DCPT015
Lớp : D19CQPU01 – N
Khoá : 2019 - 2024
Hệ : ĐẠI HỌC CHÍNH QUY

TP.HCM, tháng 08/2023

LỜI CẢM ƠN

Sau thời gian học tập và trau dồi kiến thức trên ghế nhà trường, em đã có cơ hội thực tập tại Tổng công ty Công nghiệp Công nghệ cao Viettel (Viettel High Tech). Đây là cơ hội để em áp dụng những kiến thức đã học vào thực tế và tự mình tích lũy thêm nhiều kinh nghiệm quý báu. Sau những nỗ lực của bản thân cùng sự giúp đỡ của mọi người, em đã hoàn thành được báo cáo thực tập tốt nghiệp của mình.

Để đạt được kết quả này, em xin gửi lời cảm ơn đến những người đã trực tiếp và gián tiếp giúp đỡ em. Đầu tiên là thầy cô khoa Công nghệ thông tin 2 đã tạo điều kiện thuận lợi, đặc biệt là giảng viên hướng dẫn ThS. Lê Minh Hoá đã luôn tận tình hướng dẫn, góp ý cho em. Sau là ban lãnh đạo Tổng Công ty Công nghiệp Công nghệ cao Viettel và các anh chị em đồng nghiệp luôn hỗ trợ em hết mình trong công việc, đặc biệt là anh Nguyễn Tuấn Phước, người cố vấn trực tiếp cho em tại đơn vị thực tập đã giúp đỡ em về những kiến thức thực tế trong môi trường làm việc để em hoàn thành tốt chuyên đề thực tập tốt nghiệp này.

Với thái độ nghiêm túc, dù đã hoàn thành chuyên đề với sự nỗ lực cao nhất nhưng sẽ không thể tránh khỏi những thiếu sót do kiến thức còn hạn chế. Em rất mong nhận được những góp ý của mọi người để đề tài được hoàn thiện hơn.

Em xin chân thành cảm ơn!

TP.HCM, ngày 14 tháng 08 năm 2023

Sinh viên thực hiện đề tài

Phan Thị Trà Giang

MỤC LỤC

MỞ ĐẦU.....	1
1. Lý do chọn đề tài.....	1
2. Tình hình nghiên cứu đề tài.....	1
3. Mục đích nghiên cứu	2
4. Đối tượng và phạm vi nghiên cứu	3
5. Phương pháp nghiên cứu	3
6. Đóng góp của đề tài Thực tập tốt nghiệp	3
7. Kết cấu đề tài nghiên cứu.....	4
CHƯƠNG 1: TỔNG QUAN VỀ ĐƠN VỊ THỰC TẬP	5
1.1. Giới thiệu về Tập đoàn Công nghiệp - Viễn thông Quân đội Viettel.....	5
1.2. Giới thiệu về Tổng Công ty Công nghiệp Công nghệ cao Viettel.....	6
1.3. Lĩnh vực kinh doanh của Viettel High Tech	7
1.3.1. Lĩnh vực Quân sự	7
1.3.2. Lĩnh vực Dân sự	8
1.3.3. Lĩnh vực Viễn thông.....	9
1.4. Nội dung thực tập của bản thân tại Viettel High Tech	9
1.5. Chi tiết nội dung thực tập tại VHT	11
1.5.1. Theo dõi dữ liệu bằng Selenium.....	11
1.5.1.a. Tìm hiểu về Selenium.....	11
1.5.1.b. Viết chương trình lấy dữ liệu thời tiết trong 24 giờ	12
1.5.1.c. Kiểm thử.....	13
1.5.2. Làm việc với ADB và viết ứng dụng theo dõi logcat.....	14
1.5.2.a. Android Debug Bridge - Khái niệm và cơ chế hoạt động	14
1.5.2.b. Các tính năng và ưu điểm của Android Debug Bridge	15
1.5.2.c. Ứng dụng của Android Debug Bridge trong phát triển ứng dụng Android.....	15
1.5.2.d. Tìm hiểu về các câu lệnh tương tác với logcat.....	16
1.5.2.e. Ưu điểm và ứng dụng của các câu lệnh trong ADB.....	20
1.5.2.f. Viết ứng dụng theo dõi logcat của các package cần thiết.....	20
1.5.2.g. Tạo mini web server trong ứng dụng	21
1.5.2.h. Tạo service cho ứng dụng chạy ngầm	22

1.5.2.i.	Cài đặt như ứng dụng hệ thống	22
1.5.3.	Làm việc với giao thức SOAP và kiểm thử Onvif	23
1.5.3.a.	Cấu trúc SOAP	23
1.5.3.b.	Tích hợp SOAP vào ứng dụng	23
1.5.3.c.	SOAP Action	23
1.5.3.d.	Ưu điểm và hạn chế của SOAP	24
1.5.3.e.	Kiểm thử Onvif bằng phần mềm Onvif Test Tool	24
CHƯƠNG 2: QUY TRÌNH PHÁT TRIỂN ỨNG DỤNG ANDROID		27
2.1.	Tổng quan về Android	27
2.1.1.	Hệ điều hành	27
2.1.2.	Kiến trúc của Android	27
2.1.3.	Ưu – nhược điểm của Android	29
2.1.3.a.	Ưu điểm	29
2.1.3.b.	Nhược điểm.....	30
2.1.4.	Phát triển ứng dụng Android	30
2.2.	Quy trình phát triển ứng dụng trong doanh nghiệp Viettel High Tech	31
2.2.1.	Thu thập yêu cầu.....	32
2.2.2.	Thiết kế và lập kế hoạch	32
2.2.3.	Phát triển và kiểm thử.....	32
2.2.4.	Triển khai và phân phối	32
2.2.5.	Bảo trì và nâng cấp	33
2.2.6.	Đánh giá và phản hồi	33
CHƯƠNG 3: CHUẨN ONVIF TRÊN CAMERA GIÁM SÁT		34
3.1.	Tổng quan về lĩnh vực giám sát và vai trò của chuẩn ONVIF	34
3.1.1.	Lĩnh vực giám sát	34
3.1.2.	Vai trò của chuẩn ONVIF.....	34
3.2.	Lý thuyết cơ bản về chuẩn ONVIF	34
3.2.1.	Nguyên lý và mục tiêu của chuẩn ONVIF	34
3.2.2.	Kiến trúc tổng quan của chuẩn ONVIF	35
3.2.3.	Các giao thức và tiêu chuẩn liên quan trong chuẩn ONVIF.....	35
3.3.	Các chức năng và tính năng của ONVIF.....	35
3.3.1.	Nhận dạng và xác thực	35

3.3.2.	Tìm kiếm và truy cập.....	36
3.3.3.	Truyền dữ liệu và phát sóng	36
3.3.4.	Quản lý thiết bị và dịch vụ:	36
3.3.5.	Khả năng tích hợp và tương thích	36
3.4.	Triển khai và ứng dụng thực tế của ONVIF	36
3.4.1.	Triển khai chuẩn ONVIF trên các hệ thống giám sát.....	36
3.4.2.	Ví dụ về ứng dụng và áp dụng chuẩn ONVIF.....	37
3.4.3.	Lợi ích và thách thức của việc sử dụng chuẩn ONVIF	37
CHƯƠNG 4: PHÁT TRIỂN ỨNG DỤNG TÍCH HỢP TRÊN CAMERA GIÁM SÁT		38
4.1.	Phát triển ứng dụng Android trên thiết bị camera giám sát.....	38
4.1.1.	Đặt mục tiêu và yêu cầu	38
4.1.2.	Thiết kế và kế hoạch	38
4.1.3.	Phát triển và kiểm thử.....	38
4.1.4.	Triển khai và phân phối	39
4.1.5.	Quản lý và duy trì	39
4.2.	Các ứng dụng tích hợp hệ thống camera giám sát giao thông	39
4.2.1.	Nhận dạng biển số xe.....	39
4.2.2.	Phân tích dữ liệu giao thông	39
4.2.3.	Ghi lại video	40
4.2.4.	Xử lý hình ảnh	40
CHƯƠNG 5: KỸ THUẬT OPTICAL FLOW – THƯ VIỆN OPENCV		41
5.1.	Kỹ thuật Optical Flow	41
5.1.1.	Nguyên lý hoạt động.....	41
5.1.2.	Các phương pháp tính toán giải thuật Optical Flow.....	42
5.1.2.a.	<i>Phương pháp Lucas – Kanade</i>	<i>42</i>
5.1.2.b.	<i>Phương pháp Horn – Schunck.....</i>	<i>44</i>
5.1.2.c.	<i>Phương pháp FarneBack.....</i>	<i>44</i>
5.1.2.d.	<i>Phương pháp Pyramidal.....</i>	<i>45</i>
5.1.2.e.	<i>Phương pháp Dense</i>	<i>46</i>
5.1.2.f.	<i>Phương pháp Deep Learning</i>	<i>47</i>
5.1.3.	Ứng dụng của kỹ thuật Optical Flow	48

5.1.4.	Các thách thức và hạn chế của kỹ thuật Optical Flow	49
5.2.	Thư viện OpenCV và các hàm hỗ trợ tính toán Optical Flow	49
	CHƯƠNG 6: XÂY DỰNG ỨNG DỤNG ĐO LƯỜNG GIAO THÔNG	51
6.1.	Xác định yêu cầu của ứng dụng	51
6.2.	Thiết kế - triển khai logic và xác định các công nghệ sử dụng.....	51
6.2.1.	Thiết kế xác định làn đường	51
6.2.2.	Thiết kế mô hình Optical Flow.....	51
6.2.3.	Xác định các công nghệ sử dụng	52
6.3.	Các bước xây dựng ứng dụng.....	52
6.3.1.	Nhận dữ liệu đầu vào.....	52
6.3.2.	Xử lý hình ảnh và xác định làn đường	55
6.3.3.	Xác định mật độ xe và tốc độ trung bình.....	55
6.3.4.	Phân loại làn đường	57
6.3.5.	Vẽ biểu đồ theo dõi.....	58
6.3.6.	Hiển thị kết quả.....	59
6.4.	Kiểm thử và xác định các hạn chế của ứng dụng	62
6.4.1.	Kiểm thử hiệu suất của mô hình Optical Flow	62
6.4.2.	Kiểm thử độ chính xác trong việc phân loại làn đường	63
6.4.3.	Kiểm thử về tốc độ xử lý	63
6.4.4.	Xác định các hạn chế về môi trường ánh sáng và thời tiết.....	63
6.5.	Các hướng phát triển tiềm năng	65
	KẾT LUẬN	66
	TÀI LIỆU THAM KHẢO.....	68

DANH MỤC CÁC BẢNG, SƠ ĐỒ, HÌNH

Hình 1.1. Logo của Tập đoàn Công nghiệp – Viễn thông Quân đội Viettel.....	5
Hình 1.2. Logo của Tổng Công ty Công nghiệp Công nghệ cao Viettel.....	6
Hình 1.3. Mô hình các sản phẩm của từng lớp trong lĩnh vực Viễn thông của VHT	9
Bảng 1.1. Lịch trình thực tập.....	10
Hình 1.4. Dữ liệu từ file csv	13
Hình 1.5. Biểu đồ thống kê nhiệt độ và độ ẩm trong 24 giờ	14
Bảng 1.1. Lệnh adb cơ bản	16
Bảng 1.2. Lệnh adb làm việc với thông tin thiết bị	17
Bảng 1.3. Lệnh adb làm việc với cấu hình thiết bị.....	17
Bảng 1.4. Lệnh adb dùng để sao lưu và khôi phục.....	18
Bảng 1.5. Lệnh adb để cài đặt và gỡ cài đặt ứng dụng.....	18
Bảng 1.6. Lệnh truy xuất thông tin package.....	19
Bảng 1.7. Lệnh adb thao tác với tệp của thiết bị	19
Bảng 1.8. Lệnh adb truy xuất nhật ký hoạt động của thiết bị.....	19
Bảng 1.9. Lệnh adb làm việc với các quyền của package trong thiết bị	20
Hình 1.6. Hình kết quả cho ứng dụng Logcast.....	22
Hình 1.7. Tool test Onvif tự động.....	26
Hình 1.8. Tool test Onvif thủ công	26
Hình 2.1. Minh họa cấu trúc của Android	28
Hình 2.2. Minh họa sơ lược quy trình phát triển ứng dụng di động.....	32
Hình 5.1. Vector quỹ đạo chuyển động của vật thể trong 5 khung hình liên tiếp.....	41
Hình 5.2. Luồng quang minh họa quỹ đạo chuyển động tính toán theo Lucas-Kanade.....	43
Hình 6.1. Minh họa cho 2 hướng để xác định làn đường.	55
Hình 6.2. Hiển thị kết quả của ứng dụng đối với mẫu thử 1	59
Hình 6.3. Hiển thị kết quả của ứng dụng đối với mẫu thử 2	60
Hình 6.4. Hiển thị kết quả của ứng dụng đối với mẫu thử 3	60
Hình 6.5. Hiển thị kết quả của ứng dụng đối với mẫu thử 3	61
Hình 6.6. Hiển thị kết quả của ứng dụng đối với mẫu thử 4	61
Hình 6.7. Biểu diễn vector flow trong ứng dụng.....	63

KÍ HIỆU CÁC CỤM TỪ VIẾT TẮT

VHT	Viettel High Tech
ADB/adb	Android Debug Bridge
ARM	Avanced RISC Machine
armeabi	ARM Embedded Application Binary Interface
IDE	Integrated development environment
ONVIF	Open Network Video Interface Forum
IP	Internet Protocol
NVR	Network Video Recorder
SOAP	Simple Object Access Protocol
RTSP	Realtime Streaming Protocol

MỞ ĐẦU

1. Lý do chọn đề tài

Trên thế giới ngày nay, công nghệ di động đang phát triển với tốc độ chóng mặt. Điện thoại di động đã trở thành một công cụ không thể thiếu trong cuộc sống hàng ngày của chúng ta. Với sức mạnh tính toán và tính năng ngày càng tăng của điện thoại di động, các ứng dụng giám sát dựa trên nền tảng Android đang trở nên phổ biến hơn bao giờ hết. Điều này đã đặt nền tảng cho việc phát triển ứng dụng Android cho camera giám sát, mang lại nhiều lợi ích quan trọng trong lĩnh vực an ninh và quản lý.

Camera giám sát đã trở thành một phần không thể thiếu trong hệ thống an ninh hiện đại và được áp dụng rộng rãi trong nhiều lĩnh vực như nhà ở, doanh nghiệp, trường học, bệnh viện và các khu công cộng. Sự phát triển ứng dụng Android cho camera giám sát mang lại khả năng cung cấp các tính năng tiện ích và nâng cao sự an toàn và quản lý trong các môi trường này. Các ứng dụng này cho phép người dùng dễ dàng theo dõi và kiểm soát từ xa thông qua điện thoại di động, mang lại sự linh hoạt và thuận tiện cho việc quản lý an ninh.

Thị trường ứng dụng di động đang phát triển mạnh mẽ, và ngành công nghệ an ninh cũng đang tăng trưởng với tốc độ nhanh chóng. Việc phát triển ứng dụng Android cho camera giám sát mở ra cơ hội kinh doanh rộng lớn và có khả năng thu hút sự quan tâm từ các tổ chức và công ty trong lĩnh vực an ninh và công nghệ. Với số lượng người dùng điện thoại di động ngày càng tăng, nhu cầu sử dụng các ứng dụng giám sát thông qua điện thoại di động cũng ngày càng cao. Điều này tạo ra một cơ hội thúc đẩy sự phát triển và tiếp cận thị trường rộng lớn.

Nền tảng Android là một môi trường mở, cho phép tích hợp và tương tác với các công nghệ và thiết bị khác nhau. Phát triển ứng dụng Android cho camera giám sát cho phép tạo ra giao diện tương tác dễ sử dụng, kết nối với các cảm biến và thiết bị ngoại vi, và tận dụng các tính năng của điện thoại di động như GPS, máy ảnh và kết nối mạng. Điều này mở ra những khả năng sáng tạo trong việc xây dựng các tính năng thông minh và cải thiện trải nghiệm người dùng.

Nói tóm lại, việc thực tập phát triển ứng dụng Android cho camera giám sát không chỉ mang lại những kỹ năng và kiến thức cần thiết trong lĩnh vực phát triển ứng dụng di động, mà còn mở ra cơ hội trong một thị trường đầy tiềm năng. Đề tài "Phát triển ứng dụng Android cho camera giám sát" đáng được quan tâm và nghiên cứu để tận dụng các lợi ích của công nghệ di động và camera giám sát trong việc cải thiện an ninh và quản lý trong các môi trường khác nhau.

2. Tình hình nghiên cứu đề tài

❖ Trong nước:

Ở nhiều quốc gia, phát triển ứng dụng Android cho thiết bị camera giám sát đang trở thành một lĩnh vực quan trọng và phát triển nhanh chóng. Do nhu cầu gia tăng về an ninh và giám sát, có sự tăng trưởng đáng kể trong việc sử dụng ứng dụng này để quản lý và giám sát các hệ thống camera.

Các công ty công nghệ trong nước và các nhà phát triển độc lập đang tạo ra nhiều ứng dụng Android cho thiết bị camera giám sát. Các ứng dụng này thường cung cấp các tính năng như xem trực tiếp hình ảnh, quản lý camera từ xa, ghi lại và lưu trữ video, và tích hợp các công nghệ như nhận diện khuôn mặt và phát hiện chuyển động.

❖ Ở nước ngoài:

Cũng giống như trong nước, tại nhiều quốc gia trên thế giới, phát triển ứng dụng Android cho thiết bị camera giám sát đang được quan tâm và phát triển mạnh mẽ. Các công ty công nghệ lớn và nhỏ đều tham gia vào lĩnh vực này.

Các ứng dụng Android cho thiết bị camera giám sát ở nước ngoài thường có các tính năng tương tự như trong nước. Tuy nhiên, tùy thuộc vào quy định về bảo mật và quyền riêng tư của từng quốc gia, một số tính năng có thể khác nhau hoặc được tùy chỉnh để tuân thủ các quy định địa phương.

Ngoài ra, trong vài năm gần đây, có xu hướng kết hợp ứng dụng Android cho thiết bị camera giám sát với các công nghệ mới như trí tuệ nhân tạo, học máy và tính toán đám mây. Điều này giúp cải thiện khả năng phân tích dữ liệu, nhận diện sự cố và tăng cường tính năng an ninh của các hệ thống camera giám sát.

3. Mục đích nghiên cứu

Cải thiện an ninh và giám sát: Mục đích chính của việc phát triển ứng dụng Android cho thiết bị camera giám sát là tăng cường an ninh và giám sát trong các khu vực như nhà ở, công ty, trường học, công cộng, hay các khu vực quan trọng khác. Ứng dụng này giúp người dùng có khả năng xem trực tiếp hình ảnh từ camera, quản lý và điều khiển các thiết bị giám sát từ xa, và ghi lại các sự kiện quan trọng.

Quản lý và điều khiển từ xa: Ứng dụng Android cho thiết bị camera giám sát cung cấp khả năng quản lý và điều khiển từ xa cho người dùng. Người dùng có thể kiểm soát các thiết bị camera, điều chỉnh các cài đặt, và nhận thông báo về các sự kiện quan trọng. Điều này giúp người dùng có thể giám sát và kiểm soát an ninh từ bất kỳ địa điểm nào, sử dụng điện thoại di động hoặc máy tính bảng.

Ghi lại và lưu trữ dữ liệu: Ứng dụng cho phép ghi lại video từ camera giám sát và lưu trữ dữ liệu liên quan. Điều này giúp người dùng có thể xem lại các sự kiện quan trọng, tìm kiếm và phân loại video theo thời gian hoặc sự kiện cụ thể, và duy trì một hệ thống lưu trữ an toàn cho dữ liệu quan trọng.

Tích hợp công nghệ mới: Mục đích nghiên cứu có thể bao gồm việc tích hợp các công nghệ mới như trí tuệ nhân tạo, học máy và tính toán đám mây vào ứng dụng. Điều này giúp cải thiện khả năng phân tích dữ liệu từ camera, nhận diện sự cố và hành vi đáng ngờ, và cung cấp các tính năng mở rộng như nhận diện khuôn mặt, phát hiện chuyển động, hay phân loại sự kiện.

4. Đối tượng và phạm vi nghiên cứu

- ❖ Đối tượng nghiên cứu: quy trình phát triển ứng dụng
- ❖ Phạm vi nghiên cứu: ứng dụng trên camera giám sát

5. Phương pháp nghiên cứu

- ❖ Nghiên cứu thư mục
- ❖ Nghiên cứu thực nghiệm
- ❖ Đánh giá và so sánh

6. Đóng góp của đề tài Thực tập tốt nghiệp

Cải thiện an ninh và giám sát: Phát triển ứng dụng Android cho thiết bị camera giám sát giúp cải thiện khả năng an ninh và giám sát trong các khu vực như nhà ở, công ty, trường học, công cộng và nhiều nơi khác. Việc sử dụng ứng dụng này giúp người dùng có khả năng xem trực tiếp hình ảnh từ camera, quản lý và điều khiển các thiết bị giám sát từ xa, và ghi lại các sự kiện quan trọng.

Quản lý và điều khiển từ xa: Ứng dụng Android cho thiết bị camera giám sát cho phép người dùng quản lý và điều khiển các thiết bị camera từ bất kỳ địa điểm nào. Người dùng có thể kiểm soát các thiết bị, điều chỉnh cài đặt và nhận thông báo về các sự kiện quan trọng. Điều này mang lại sự tiện lợi và linh hoạt trong việc giám sát và kiểm soát an ninh.

Phát triển công nghệ và ứng dụng di động: Nghiên cứu trong lĩnh vực này đóng góp vào sự phát triển và cải tiến công nghệ và ứng dụng di động. Việc tích hợp các công nghệ mới như trí tuệ nhân tạo, học máy và tính toán đám mây trong ứng dụng giúp cải thiện khả năng phân tích dữ liệu, nhận diện sự cố và nâng cao tính năng an ninh của hệ thống camera giám sát.

Tạo ra giải pháp thông minh và tiết kiệm thời gian: Phát triển ứng dụng Android cho thiết bị camera giám sát giúp tạo ra các giải pháp thông minh và tiết kiệm thời gian trong việc quản lý và giám sát an ninh. Việc sử dụng các tính năng như tự động phát hiện chuyển động, cảnh báo và ghi lại các sự kiện quan trọng giúp người dùng tập trung vào các vấn đề quan trọng và tiết kiệm thời gian trong việc xem và xử lý dữ liệu.

7. Kết cấu đề tài nghiên cứu

Chương 1: Tổng quan về Đơn vị thực tập

Giới thiệu về Đơn vị thực tập Viettel High Tech và các lĩnh vực mà đơn vị đang tham gia nghiên cứu, triển khai và kinh doanh

Chương 2: Quy trình phát triển ứng dụng Android

Giới thiệu chung về Android và quy trình phát triển ứng dụng trong các doanh nghiệp tại Việt Nam nói chung và Viettel High Tech nói riêng.

Chương 3: Chuẩn Onvif trên camera giám sát

Trình bày về lĩnh vực an ninh giám sát và vai trò của camera giám sát cũng như chuẩn Onvif trong camera giám sát.

Chương 4: Phát triển ứng dụng tích hợp trên camera giám sát

Trình bày các ứng dụng được tích hợp vào trên camera giám sát thông thường và camera giám sát giao thông (các tính năng trong camera).

Chương 5: Kỹ thuật Optical Flow – Thư viện OpenCV

Tìm hiểu về kỹ thuật Optical Flow và các phương pháp để triển khai kỹ thuật này cũng như sự hỗ trợ của thư viện OpenCV trong việc triển khai này.

Chương 6: Xây dựng ứng dụng phân luồng giao thông

Viết ứng dụng Android xử lý video nhận được từ camera giám sát để phân tích tình hình của các luồng giao thông từ đó đưa ra phán đoán hợp lý trong việc điều khiển đèn tín hiệu giao thông.

Kết luận: Sơ lược lại những gì đã đạt được và các thách thức trong lúc thực hiện đề tài, đồng thời đưa ra hướng mở rộng cho đề tài

CHƯƠNG 1: TỔNG QUAN VỀ ĐƠN VỊ THỰC TẬP

1.1. Giới thiệu về Tập đoàn Công nghiệp - Viễn thông Quân đội Viettel

Tập đoàn Công nghiệp – Viễn thông Quân đội Viettel (Viettel Group), thường được biết đến dưới tên giao dịch Viettel hay Tập đoàn Viettel, là một tập đoàn viễn thông và công nghệ Việt Nam được thành lập vào ngày 1 tháng 6 năm 1989.

Các ngành nghề chính của tập đoàn bao gồm: ngành dịch vụ viễn thông & công nghệ thông tin; ngành nghiên cứu sản xuất thiết bị điện tử viễn thông, ngành công nghiệp quốc phòng, ngành công nghiệp an ninh mạng và ngành cung cấp dịch vụ số. Sản phẩm nổi bật nhất của Viettel hiện nay là mạng di động Viettel Mobile. Công ty thành viên Viettel Telecom của Viettel hiện đang là nhà mạng giữ thị phần lớn nhất trên thị trường dịch vụ viễn thông Việt Nam.

Hiện nay, Viettel đã đầu tư tại 10 thị trường nước ngoài ở 3 châu lục gồm Châu Á, Châu Mỹ và Châu Phi. Năm 2018, Viettel đạt doanh thu 10 tỷ USD (234.500 tỷ VND). Viettel được đánh giá là một trong những công ty viễn thông có tốc độ phát triển nhanh nhất thế giới. Năm 2019, Viettel đã trở thành doanh nghiệp thuộc Top 15 công ty viễn thông lớn nhất thế giới về số thuê bao, Top 40 công ty viễn thông lớn nhất thế giới về doanh thu. Giá trị thương hiệu của Viettel được Brand Finance xác định là 4,3 tỷ USD – thuộc Top 500 thương hiệu lớn nhất trên thế giới, và là thương hiệu giá trị nhất tại Việt Nam.



Hình 1.1. Logo của Tập đoàn Công nghiệp – Viễn thông Quân đội Viettel

Sau hơn 3 thập kỷ nỗ lực hoàn thành mục tiêu phổ cập dịch vụ viễn thông, đưa viễn thông và công nghệ thông tin vào mọi lĩnh vực của cuộc sống ở Việt Nam, Tập đoàn Công nghiệp - Viễn thông Quân đội (Viettel) đặt ra khát vọng trở thành Tập đoàn công nghiệp và công nghệ vươn tầm thế giới. Ở bất cứ giai đoạn nào trên hành trình ấy, lời hứa “sáng tạo vì con người” vẫn còn mãi.

Các công ty thành viên của Viettel Group bao gồm:

Viettel Network	Cam Pha Cement	Viettel Cyberspace
Viettel IDC	Viettel Constructions	Viettel Security
Viettel Consultancy	Viettel Commerce	Viettel Asset
Viettel Global	Viettel Manufacturing	Viettel Sports
VDTC	Viettel Aerospace	Viettel Academy

Viettel Media
Viettel Digital

Viettel Solutions
Viettel Post

Viettel High Tech
Viettel Telecom

1.2. Giới thiệu về Tổng Công ty Công nghiệp Công nghệ cao Viettel

Tổng Công ty Công nghiệp Công nghệ cao Viettel (Viettel High Tech) là đơn vị nghiên cứu sản xuất chủ lực của Viettel trong lĩnh vực quân sự và dân sự, tự tin làm chủ toàn trình từ nghiên cứu, thiết kế, chế tạo, sản xuất, và kinh doanh sản phẩm với tầm nhìn trở thành một công ty công nghệ hiện đại, hàm lượng tri thức cao, có thương hiệu trên thị trường quốc tế.



Hình 1.2. Logo của Tổng Công ty Công nghiệp Công nghệ cao Viettel

Các sản phẩm của Tổng Công ty dựa trên nền tảng công nghệ tiên phong (Pioneer Technologies), kiến tạo nên một hệ thống an ninh quốc phòng tin cậy để bảo vệ vững chắc chủ quyền quốc gia, một mạng lưới kết nối vạn vật thông minh (5G, IoT, AI) để phát triển kinh tế - xã hội.

Với sứ mệnh “Mang lại sự đơn giản cho cuộc sống, góp phần xây dựng xã hội an toàn, nơi vạn vật được kết nối thông minh”, chúng tôi tự tin bước đồng nhịp cùng các tập đoàn công nghệ lớn mạnh trên thế giới.

Tổng doanh thu lũy kế mang lại từ hoạt động nghiên cứu, sản xuất trang thiết bị công nghệ cao đến thời điểm hiện tại đạt trên 1.5 tỷ USD.

Trong Viettel High Tech được chia ra làm 3 khối:

- ❖ Khối Quân sự: Chuyên triển khai các sản phẩm phục vụ cho lĩnh vực quân sự.
- ❖ Khối Dân sự: Chuyên triển khai các sản phẩm phục vụ cho lĩnh vực dân sự (phục vụ cho công đồng, xã hội hoặc người dân)
- ❖ Khối Sáng tạo: Chuyên triển khai các giải pháp độc đáo sáng tạo, dẫn đầu xu hướng công nghệ hiện nay, ví dụ như: IoT, Chipset 5G,...

1.3. Lĩnh vực kinh doanh của Viettel High Tech

1.3.1. Lĩnh vực Quân sự

Hiện tại, VHT nghiên cứu và phát triển các sản phẩm CNQP công nghệ cao theo mô hình tác chiến C5ISR (Command, Control, Computer, Communication, Cyber, Interlligence, Surveillance, Reconnaissance) – là mô hình tác chiến hiện đại được áp dụng ở nhiều nước có nền khoa học quân sự tiên tiến trên thế giới.

Mô hình tác chiến thực hiện 04 nhiệm vụ chính

- ❖ **Nhiệm vụ thu thập thông tin:** Các thiết bị thực hiện trinh sát, thu thập thông tin đầu vào tại chiến trường và truyền về sở chỉ huy Trung tâm.
- ❖ **Nhiệm vụ truyền nhận thông tin, mệnh lệnh:** Bao gồm các sản phẩm thiết bị thông tin liên lạc thế hệ mới, tích hợp công nghệ bảo mật an toàn hàng đầu thế giới.
- ❖ **Nhiệm vụ xử lý thông tin, hỗ trợ chỉ huy điều hành tác chiến:** Các Hệ thống tự động hóa chỉ huy điều khiển phục vụ cho các quân, binh chủng.
- ❖ **Tác chiến không gian mạng:** Các hệ thống tự động hóa chỉ huy điều khiển giúp xử lý, phân tích tình huống và hỗ trợ người chỉ huy ra quyết định nhanh chóng.

Bên cạnh đó, VHT nghiên cứu phát triển các Hệ thống mô hình mô phỏng phục vụ công tác huấn luyện của Quân đội.

Các sản phẩm tiêu biểu của Viettel High Tech trong lĩnh vực quân sự:

- ❖ **RA-ĐA:** Ra-đa 2D, 3D thuộc các thế hệ thứ 2, 3, 4, nhằm thực hiện các nhiệm vụ cảnh giới, phát hiện các mục tiêu trên không, trên biển và chỉ thị cho hỏa lực đáp ứng các nhu cầu khác nhau của khách hàng.
- ❖ **MÁY BAY KHÔNG NGƯỜI LÁI:** Ứng dụng nhiều công nghệ tiên tiến trong lĩnh vực hàng không vũ trụ, công nghệ vật liệu, công nghệ động cơ lực đẩy, điện tử, các cảm biến trinh sát, tình báo, điều khiển và dẫn đường.
- ❖ **TÁC CHIẾN ĐIỆN TỬ:** Thiết bị gây nhiễu thế hệ thứ 2 và hệ thống trinh sát thế hệ thứ 3 có khả năng phát hiện, định vị, bắt bám, phân loại và nhận dạng các nguồn phát tín hiệu thông tin liên lạc, phát xung ra-đa để trang bị cho các lực lượng tác chiến điện tử.
- ❖ **THÔNG TIN LIÊN LẠC:** Thiết bị thông tin vô tuyến - thế hệ thứ 4 và thế hệ thứ 5 ứng dụng trong quân sự và dân dụng đảm bảo liên lạc chính xác, tin cậy.

1.3.2. Lĩnh vực Dân sự

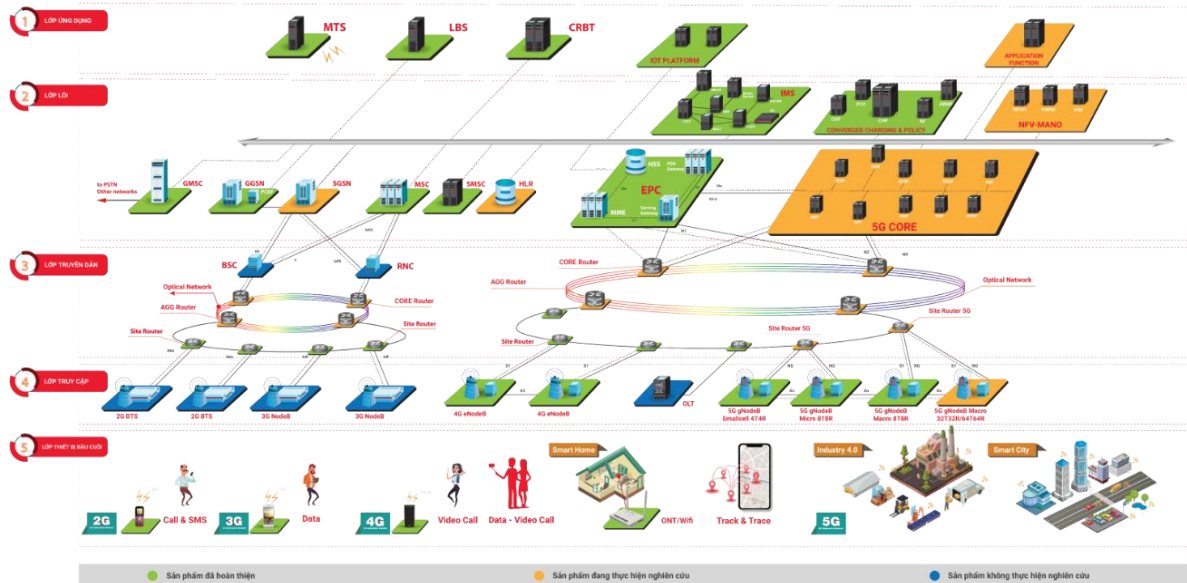
Trong thời đại công nghệ số phát triển vượt bậc, lĩnh vực kinh doanh dân sự đã trở thành một phần không thể thiếu trong cuộc sống hàng ngày của con người. Các công ty công nghệ, như Viettel High Tech - một đơn vị thành viên của tập đoàn Viễn thông Quân đội Việt Nam (Viettel Group), đã chứng tỏ vai trò quan trọng của mình trong việc cung cấp các giải pháp công nghệ thông tin và truyền thông cho cộng đồng dân cư. Bài luận văn này nhằm tìm hiểu sâu hơn về lĩnh vực kinh doanh dân sự của Viettel High Tech, từ các sản phẩm, dịch vụ cho đến những cam kết và đóng góp của công ty cho xã hội.

Viettel High Tech đã nghiên cứu và sản xuất các sản phẩm nhằm đáp ứng dịch và giải pháp cho các nền tảng:

- ❖ **NỀN TẢNG AI CAMERA:** Nền tảng camera trí tuệ nhân tạo là nền tảng hỗ trợ phát triển ứng dụng và giải pháp người dùng cuối dựa trên camera thông minh. Nền tảng hoạt động theo phương thức của một nền tảng sáng tạo, đóng vai trò là nền móng phát triển nên các ứng dụng thông minh cho 5 lĩnh vực trọng tâm: An ninh Căn hộ; Tòa nhà thông minh; Giao thông thông minh; Sản xuất thông minh; Nông nghiệp thông minh.
- ❖ **NỀN TẢNG IOT:** VHT IoT Platform sẽ đơn giản hóa và tối ưu quy trình xây dựng ứng dụng, giải pháp IoT từ thiết kế đến quản trị với giải pháp toàn trình gồm Thiết bị (Module), Bộ công cụ phát triển phần mềm (Software Development Kit-SDK), Nền tảng Back-end (IoT Back-end Platform), Ứng dụng (Application). Là hệ thống quản lý IoT tập trung do VHT cung cấp, cho phép thu thập dữ liệu từ hàng tỷ thiết bị phần cứng và cung cấp dịch vụ quản trị/ giám sát/ phân tích dữ liệu cho các khu chung cư, sân bay, khu công nghiệp,... theo thời gian thực.
- ❖ **CHĂM SÓC SỨC KHỎE:** Viettel High Tech có 2 giải pháp chăm sóc sức khỏe:
 - ✓ **OurHealth** là nền tảng chăm sóc sức khỏe từ xa dành cho cộng đồng, giúp kết nối bác sĩ và bệnh nhân bằng các tính năng theo dõi, tư vấn sức khỏe từ xa; hỗ trợ điều trị và quản lý bệnh.
 - ✓ **Smart Purifier** là máy lọc không khí - cấp không khí tươi giúp cải thiện môi trường sinh sống trong căn hộ gia đình.

1.3.3. Lĩnh vực Viễn thông

Các sản phẩm hạ tầng mạng viễn thông của Viettel High Tech đáp ứng các tiêu chuẩn quốc tế ITU-T, ETSI.



Hình 1.3. Mô hình các sản phẩm của từng lớp trong lĩnh vực Viễn thông của VHT

- ❖ **LỚP THIẾT BỊ ĐẦU CUỐI:** Nhằm phục vụ việc truy cập mạng viễn thông, lớp thiết bị đầu cuối được chúng tôi nghiên cứu và sản xuất gồm các sản phẩm công nghệ cao phục vụ đời sống dân sinh như IOT, Smart City, Điện thoại bảo mật (VIPPHONE)...
- ❖ **LỚP TRUY CẬP VÀ LỚP TRUYỀN DẪN:** Mạng truyền dẫn cung cấp các đường kết nối để chuyển tải lưu lượng giữa các node trong mạng viễn thông. Có 2 loại: Truyền dẫn hữu tuyến và truyền dẫn vô tuyến. Truyền dẫn hữu tuyến sử dụng môi trường truyền dẫn thông qua cáp đồng trục, cáp quang, truyền dẫn vô tuyến sử dụng môi trường truyền dẫn vô tuyến với thiết bị đầu cuối là viba SDH/PDH.
- ❖ **LỚP ỨNG DỤNG VÀ LỚP LỖI:** Là các sản phẩm đảm nhận các nhiệm vụ tính cước, cung cấp các dịch vụ thoại, tin nhắn, video, data, cơ sở dữ liệu người dùng... OCS, mạng lõi 5G, EPC, IMS, MSC, SMSC (lớp lõi) và hệ thống tổng đài nhạc chuông chờ CRBT (lớp ứng dụng).

1.4. Nội dung thực tập của bản thân tại Viettel High Tech

Thời gian thực tập: từ ngày 05/06/2023 đến ngày 11/08/2023

Khối đơn vị thực tập: Khối 3 – Thực tập trung tâm Camera giám sát

Người hướng dẫn: Nguyễn Tuấn Phước

Lịch trình thực tập cụ thể:

Tuần 1 (05/06/2023 – 09/06/2023)	Làm quen với môi trường làm việc, ôn lại những kiến thức cần cho việc thực tập.
Tuần 2 (12/06/2023 – 16/06/2023)	Thực tập với Selenium để lấy dữ liệu trên web page, theo dõi và vẽ biểu đồ thống kê.
Tuần 3 (19/06/2023 – 23/06/2023)	Tham gia vào dự án Camera AI giám sát giao thông, tìm hiểu về Android Debug Bridge, các câu lệnh để tương tác với logcat của thiết bị Camera AI.
Tuần 4 (26/06/2023 – 30/06/2023)	Viết gói ứng dụng Android (apk) để đọc logcat của thiết bị, lọc logcat theo từng package cần theo dõi. Viết chức năng truy cập đến ứng dụng từ web thông qua địa chỉ IP.
Tuần 5 (03/07/2023 – 07/07/2023)	Kiểm thử phần mềm. Tìm hiểu về cách cài đặt ứng dụng như ứng dụng của hệ thống và tiến hành cài đặt.
Tuần 6 (10/07/2023 – 14/07/2023)	Tìm hiểu về tài liệu của Onvif (Profile S). Học cách sử dụng test tool của Onvif.
Tuần 7 (17/07/2023 – 21/07/2023)	Tìm hiểu về giao thức SOAP, gửi request SOAP action để thực hiện kiểm thử phần mềm Onvif server trên Camera AI.
Tuần 8 (24/07/2023 – 28/07/2023)	Viết báo cáo cho kiểm thử Onvif. Viết ứng dụng OSMapp hiển thị offline bản đồ nguồn Open Street Map.
Tuần 9 (31/07/2023 – 04/08/2023)	Viết ứng dụng OSMapp chạy cross-platform bằng React Native (javascript)
Tuần 10 (07/08/2023 – 11/08/2023)	Tìm hiểu về cách thức app Jami truyền tải dữ liệu (đặc biệt là video)

Bảng 1.1. Lịch trình thực tập

1.5. Chi tiết nội dung thực tập tại VHT

1.5.1. Theo dõi dữ liệu bằng Selenium

1.5.1.a. Tìm hiểu về Selenium

Trong thời đại công nghệ ngày nay, phần mềm và ứng dụng di động đóng vai trò quan trọng trong cuộc sống và kinh doanh. Để đảm bảo chất lượng và tin cậy của các sản phẩm này, việc thực hiện kiểm thử là bước không thể thiếu. Tuy nhiên, kiểm thử thủ công đòi hỏi nhiều thời gian, công sức và có thể dẫn đến sự thiếu chính xác. Do đó, cần sự hỗ trợ của các công cụ tự động hóa test để tối ưu quá trình này. Trong bài luận văn này, chúng ta sẽ tìm hiểu về Selenium - một công cụ tự động hóa test mạnh mẽ và phổ biến.

❖ Định nghĩa Selenium

Selenium là một bộ công cụ mã nguồn mở dùng để tự động hóa các thao tác trên các ứng dụng web. Được giới thiệu lần đầu tiên bởi Jason Huggins vào năm 2004, Selenium đã nhanh chóng trở thành một trong những công cụ tự động hóa test phổ biến nhất trên thị trường. Ban đầu, Selenium chỉ hỗ trợ việc viết các test script bằng ngôn ngữ JavaScript, nhưng sau đó, nó đã được mở rộng để hỗ trợ nhiều ngôn ngữ lập trình khác nhau như Java, Python, C#, Ruby, và nhiều ngôn ngữ khác.

❖ Các thành phần chính của Selenium

Selenium IDE (Integrated Development Environment): Đây là một công cụ dựa trên trình duyệt dùng để ghi lại, chỉnh sửa và gỡ lỗi các test script một cách trực quan. Selenium IDE là lựa chọn lý tưởng cho các nhà kiểm thử mới bắt đầu với tự động hóa test.

Selenium WebDriver: Đây là thành phần mạnh mẽ của Selenium, cung cấp API để tương tác với các trình duyệt web. WebDriver cho phép viết các test script bằng nhiều ngôn ngữ lập trình khác nhau và hỗ trợ tự động hóa test trên nhiều nền tảng trình duyệt, bao gồm Chrome, Firefox, Safari, Edge, và nhiều trình duyệt khác.

Selenium Grid: Đây là một công cụ cho phép chạy các test script đồng thời trên nhiều trình duyệt và hệ điều hành khác nhau, tạo ra sự phân tán và tối ưu hiệu suất tự động hóa test.

❖ Ưu điểm của Selenium

Đa nền tảng và đa trình duyệt: Selenium hỗ trợ việc tự động hóa test trên nhiều nền tảng hệ điều hành khác nhau và các trình duyệt phổ biến. Điều

này giúp cho việc đảm bảo tương thích và kiểm thử trên nhiều môi trường một cách hiệu quả.

Ngôn ngữ lập trình đa dạng: Selenium cho phép chọn ngôn ngữ lập trình ưa thích để viết test script, giúp tăng tính linh hoạt và thuận tiện cho nhóm kiểm thử.

Tương thích với các công cụ và frameworks khác: Selenium có thể tích hợp dễ dàng với các công cụ và frameworks khác, như TestNG, JUnit, Maven, và nhiều công cụ tự động hóa test khác, giúp tối ưu hóa quá trình kiểm thử.

❖ Ứng dụng của Selenium trong Tự động hóa Test

Selenium được sử dụng rộng rãi trong việc tự động hóa test cho các ứng dụng web và di động. Các ứng dụng của Selenium bao gồm:

- ✓ Kiểm thử chấp nhận (Acceptance Testing): Selenium được sử dụng để đảm bảo rằng ứng dụng web hoạt động đúng theo yêu cầu kỹ thuật và yêu cầu của người dùng.
- ✓ Kiểm thử tích hợp (Integration Testing): Selenium có thể tích hợp với các công cụ và frameworks khác để kiểm tra tích hợp giữa các thành phần trong ứng dụng.
- ✓ Kiểm thử hiệu suất (Performance Testing): Selenium Grid cho phép chạy đồng thời các test script trên nhiều trình duyệt, giúp kiểm tra hiệu suất và tải của ứng dụng.
- ✓ Kiểm thử hộp trắng (White-box Testing): Selenium hỗ trợ kiểm thử các đoạn mã và các chức năng nội bộ của ứng dụng web.

Selenium là một công cụ tự động hóa test mạnh mẽ và linh hoạt, giúp tăng cường đáng kể hiệu quả kiểm thử cho các ứng dụng web và di động. Nhờ vào tích hợp đa ngôn ngữ, đa nền tảng và đa trình duyệt, Selenium trở thành một lựa chọn hàng đầu cho các nhà phát triển và kiểm thử để đảm bảo chất lượng và tin cậy của các sản phẩm công nghệ. Sự tiến bộ của Selenium và cộng đồng mã nguồn mở đằng sau nó đã và tiếp tục đóng góp không nhỏ cho sự phát triển của ngành kiểm thử phần mềm.

1.5.1.b. *Viết chương trình lấy dữ liệu thời tiết trong 24 giờ*

Mục đích của chương trình: Lấy dữ liệu nhiệt độ và độ ẩm từ trang web [Current Local Time in Ho Chi Minh, Vietnam \(timeanddate.com\)](http://Current Local Time in Ho Chi Minh, Vietnam (timeanddate.com)) trong vòng 24 giờ và lưu dữ liệu vào file csv, vẽ biểu đồ thống kê từ dữ liệu trong file đó.

Cách thức thực hiện:

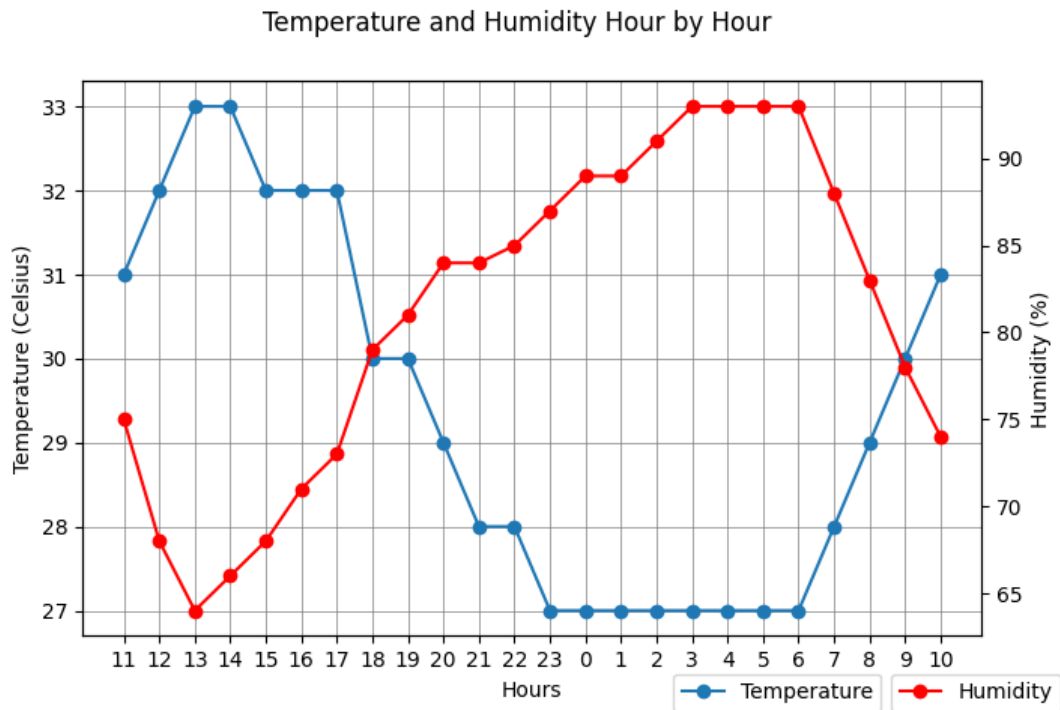
- ❖ Sử dụng module webdriver của thư viện selenium để truy cập đến URL: [Hourly forecast for Ho Chi Minh, Vietnam \(timeanddate.com\)](https://timeanddate.com/Hourly-forecast/Ho-Chi-Minh-Vietnam) bằng Chrome.
- ❖ Trong module webdriver của thư viện selenium, import module By nhằm sử dụng phương thức find_elements bằng XPath để tìm element của html chứa các dữ liệu: giờ hiện tại, nhiệt độ và độ ẩm.
- ❖ Để tìm được XPath, truy cập đến URL nói trên, mở dev tool của trình duyệt (F12) và chỉ hướng đến element chứa dữ liệu cần lấy rồi copy XPath của nó.
- ❖ Sử dụng thuộc tính text để lấy dữ liệu trong element đó, cắt số liệu cần trong chuỗi mà thuộc tính text trả về và ép kiểu nó về integer để chuẩn bị dữ liệu vẽ biểu đồ.
- ❖ Vẽ biểu đồ đường giữa nhiệt độ và thời gian cho y axis bên trái bằng hàm *plot*, làm tương tự cho phía bên phải với độ ẩm và thời gian.
- ❖ Vẽ thêm các đường lưới bằng hàm *grid*

1.5.1.c. Kiểm thử

Chạy chương trình 24 giờ và thu được kết quả vào file csv như sau:

Hour (h) ▾	Temperature (°C) ▾	Humidity (%) ▾			
12	32	68	11	27	87
13	33	64	12	27	89
14	33	66	13	27	89
15	32	68	14	27	91
16	32	71	15	27	93
17	32	73	16	27	93
18	30	79	17	27	93
19	30	81	18	27	93
20	29	84	19	28	88
21	28	84	20	29	83
10	28	85	21	30	78
			10	31	74
			11	31	72

Hình 1.4. Dữ liệu từ file csv



Hình 1.5. Biểu đồ thống kê nhiệt độ và độ ẩm trong 24 giờ

1.5.2. Làm việc với ADB và viết ứng dụng theo dõi logcat

Trong thời đại công nghệ thông tin phát triển mạnh mẽ hiện nay, việc phát triển ứng dụng di động đã trở nên ngày càng phổ biến. Đối với hệ điều hành Android, việc sử dụng Android Debug Bridge (ADB) đã trở thành một phần không thể thiếu trong quá trình phát triển ứng dụng. ADB là một công cụ mạnh mẽ cho phép các nhà phát triển giao tiếp và tương tác với các thiết bị Android từ máy tính cá nhân thông qua dòng lệnh. Bài luận văn này tập trung nghiên cứu về ADB, những tính năng, ưu điểm và vai trò của công cụ này trong việc phát triển ứng dụng Android.

1.5.2.a. Android Debug Bridge - Khái niệm và cơ chế hoạt động

ADB là một công cụ dòng lệnh được tích hợp trong Android Software Development Kit (SDK), cung cấp môi trường phát triển cho việc xây dựng ứng dụng Android. Được phát triển bởi Google, ADB cho phép nhà phát triển thực hiện nhiều thao tác trên thiết bị Android như cài đặt ứng dụng, gỡ lỗi, sao lưu dữ liệu, tải và tải lên tệp tin, và thậm chí truy cập vào giao diện dòng lệnh (shell) của thiết bị.

ADB hoạt động dưới cơ chế client-server với ba thành phần chính:

- ❖ Máy khách (Client): Chạy trên máy tính cá nhân của nhà phát triển. Đây là nơi mà nhà phát triển nhập các lệnh ADB và gửi đến máy chủ (server).

- ❖ Tiến trình nền (adb): Là một tiến trình chạy dưới dạng nền (background) trên mỗi thiết bị Android. Tiến trình này thực thi các lệnh được gửi từ máy chủ.
- ❖ Máy chủ (Server): Chạy trên máy tính cá nhân của nhà phát triển. Máy chủ quản lý việc gửi lệnh từ máy khách đến tiến trình adb trên thiết bị và chuyển trả kết quả cho máy khách.

1.5.2.b. Các tính năng và ưu điểm của Android Debug Bridge

- ❖ Cài đặt ứng dụng và gỡ lỗi: ADB cho phép nhà phát triển cài đặt và gỡ lỗi ứng dụng trực tiếp trên thiết bị Android từ máy tính, giúp tiết kiệm thời gian và tăng hiệu quả trong quá trình phát triển ứng dụng.
- ❖ Truy cập vào shell của thiết bị: ADB cung cấp một giao diện dòng lệnh trực tiếp đến thiết bị Android, cho phép nhà phát triển thực hiện các lệnh tùy chỉnh và kiểm tra trạng thái hệ thống.
- ❖ Sao lưu và khôi phục dữ liệu: ADB cho phép sao lưu và khôi phục dữ liệu từ thiết bị Android, giúp bảo vệ thông tin quan trọng và dễ dàng di chuyển dữ liệu giữa các thiết bị.
- ❖ Kiểm tra hoạt động mạng: ADB cho phép nhà phát triển kiểm tra trạng thái kết nối mạng trên thiết bị Android, giúp phát hiện và khắc phục sự cố kết nối.
- ❖ Tương tác với cảm biến và phần cứng: ADB cung cấp các lệnh để truy cập và kiểm tra thông tin từ các cảm biến và phần cứng trên thiết bị Android, giúp xác định các vấn đề liên quan đến phần cứng.

1.5.2.c. Ứng dụng của Android Debug Bridge trong phát triển ứng dụng Android

ADB đóng vai trò quan trọng trong việc phát triển ứng dụng Android và cung cấp nhiều lợi ích cho nhà phát triển. Điển hình, ADB giúp nhà phát triển:

- ❖ Sửa lỗi và gỡ rối: Dễ dàng theo dõi lỗi và gỡ rối ứng dụng trên thiết bị Android từ máy tính cá nhân, giúp tối ưu hóa hiệu suất ứng dụng và cải thiện trải nghiệm người dùng.
- ❖ Kiểm tra tương tác giao diện: Giúp nhà phát triển thử nghiệm và kiểm tra tương tác giao diện trên nhiều thiết bị Android khác nhau, đảm bảo tính tương thích và đa dạng hóa trong trải nghiệm người dùng.
- ❖ Tối ưu hóa hiệu năng ứng dụng: Kiểm tra hiệu năng và sử dụng tài nguyên trên thiết bị Android để tối ưu hóa ứng dụng và tăng tốc độ phản hồi.
- ❖ Điều khiển từ xa: Cho phép nhà phát triển kiểm soát và thao tác với thiết bị Android từ xa, hỗ trợ việc kiểm tra ứng dụng trên các thiết bị không gắn trực tiếp vào máy tính phát triển.

Trên cơ sở nghiên cứu về Android Debug Bridge (ADB) trong luận văn này, chúng ta nhận thấy tầm quan trọng và đóng góp quan trọng của công cụ này trong quá trình phát triển ứng dụng Android. ADB cung cấp một cách linh hoạt và hiệu quả để giao tiếp với thiết bị Android, giúp nhà phát triển dễ dàng thực hiện các tác vụ cài đặt, gỡ lỗi và kiểm tra hiệu năng. Sự tích hợp của ADB trong Android SDK Platform Tools cũng đảm bảo việc sử dụng công cụ này một cách tiện lợi và hợp lý trong quá trình phát triển ứng dụng Android.

1.5.2.d. Tìm hiểu về các câu lệnh tương tác với logcat

adb devices	liệt kê các thiết bị được kết nối
adb root	khởi động lại adb với quyền root
adb start-server	khởi động máy chủ
adb kill-server	dừng máy chủ
adb reboot	khởi động lại thiết bị
adb devices -l	danh sách thiết bị theo sản phẩm / kiểu máy
adb shell	khởi động thiết bị đầu cuối nền
exit	thoát khỏi thiết bị đầu cuối nền
adb help	liệt kê tất cả các lệnh
adb -s <tên thiết bị> <lệnh>	chuyển hướng lệnh đến thiết bị cụ thể
adb -d <lệnh>	chỉ thị lệnh tới chỉ thiết bị USB được đính kèm
adb -e <lệnh>	chỉ thị lệnh tới chỉ trình giả lập được đính kèm

Bảng 1.1. Lệnh adb cơ bản

adb get-state	in trạng thái thiết bị
adb shell dumpsys package <package_name>	Lấy các thông tin của package, trong đó có các quyền đã được cấp
adb shell netstat	danh sách kết nối TCP
adb shell pwd	in thư mục làm việc hiện tại
adb shell dumpsys battery	tình trạng pin
adb shell pm list features	liệt kê các chức năng của điện thoại
adb shell service list	liệt kê tất cả các dịch vụ
adb shell dumpsys activity <package>/<activity>	thông tin hoạt động
adb shell ps	in trạng thái tiến trình
adb shell wm size	hiển thị độ phân giải màn hình hiện tại

Bảng 1.2. Lệnh adb làm việc với thông tin thiết bị

adb shell dumpsys battery set level <n>	thay đổi mức pin từ 0 đến 100
adb shell dumpsys battery set status <n>	thay đổi trạng thái pin thành không xác định, sạc, xả, không sạc hoặc đầy
adb shell dumpsys battery reset	đặt lại pin
adb shell dumpsys battery set usb <n>	thay đổi trạng thái kết nối USB ON hoặc OFF
adb shell wm size WxH	đặt độ phân giải thành WxH

Bảng 1.3. Lệnh adb làm việc với cấu hình thiết bị

adb reboot-recovery	khởi động lại thiết bị vào chế độ khôi phục
adb reboot fastboot	khởi động lại thiết bị vào chế độ fastboot
adb shell screencap -p “/path/to/screenshot.png”	chụp ảnh màn hình thiết bị
adb shell screenrecord “/path/to/record.mp4”	quay màn hình thiết bị
adb backup -apk -all -f backup.ab	sao lưu cài đặt và ứng dụng
adb backup -apk -shared -all -f backup.ab	sao lưu cài đặt, ứng dụng và bộ nhớ trong dùng chung
adb backup -apk -nosystem -all -f backup.ab	chỉ sao lưu các ứng dụng không thuộc hệ thống
adb restore backup.ab	khôi phục sao lưu trước đó
adb shell am start -t image/* -a android.intent.action.VIEW	mở thư viện

Bảng 1.4. Lệnh adb dùng để sao lưu và khôi phục

adb shell install <apk>	cài đặt ứng dụng
adb shell install <đường dẫn>	cài đặt ứng dụng từ đường dẫn điện thoại
adb shell install -r <đường dẫn>	cài đặt ứng dụng từ đường dẫn điện thoại
adb shell uninstall <tên package>	gỡ cài đặt ứng dụng

Bảng 1.5. Lệnh adb để cài đặt và gỡ cài đặt ứng dụng

adb shell list packages	danh sách các gói
-------------------------	-------------------

adb shell list packages -r	danh sách các gói + đường dẫn đến apks
adb shell list packages -3	liệt kê tên gói của bên thứ ba
adb shell list packages -s	chỉ liệt kê gói hệ thống
adb shell list packages -u	liệt kê tên gói + đã gỡ cài đặt
adb shell dumpsys package packages	thông tin của tất cả các ứng dụng
adb shell dump <tên>	danh sách thông tin trên một gói
adb shell path <package>	đường dẫn đến tệp apk

Bảng 1.6. Lệnh truy xuất thông tin package

adb push <đường dẫn nguồn> <đường dẫn đích>	sao chép file vào thiết bị
adb pull <đường dẫn đích> <đường dẫn nguồn>	sao chép file từ thiết bị
adb shell ls	danh sách nội dung thư mục
adb shell ls -s	in kích thước của mỗi tệp
adb shell ls -R	liệt kê đệ quy các thư mục con

Bảng 1.7. Lệnh adb thao tác với tệp của thiết bị

adb logcat [tùy chọn] [bộ lọc]	xem nhật ký thiết bị
adb bugreport	in báo cáo lỗi

Bảng 1.8. Lệnh adb truy xuất nhật ký hoạt động của thiết bị

adb shell permissions groups	danh sách định nghĩa nhóm quyền
------------------------------	---------------------------------

adb shell list permissions -g -r	danh sách quyền chi tiết
----------------------------------	--------------------------

Bảng 1.9. Lệnh adb làm việc với các quyền của package trong thiết bị

1.5.2.e. Ưu điểm và ứng dụng của các câu lệnh trong ADB

Hiệu quả và linh hoạt: Các câu lệnh trong ADB giúp nhà phát triển thực hiện các tác vụ cần thiết trên thiết bị Android một cách hiệu quả và nhanh chóng từ máy tính cá nhân, giảm thiểu thời gian và công sức.

Gỡ rối và kiểm tra: Các câu lệnh như adb logcat cung cấp thông tin chi tiết về hoạt động của ứng dụng và hệ thống, giúp nhà phát triển tìm kiếm và sửa chữa lỗi một cách dễ dàng.

Kiểm tra hiệu năng: ADB cho phép nhà phát triển kiểm tra hiệu năng ứng dụng trên thiết bị Android, từ đó tối ưu hóa ứng dụng để đảm bảo hoạt động mượt mà và tiết kiệm tài nguyên.

Hỗ trợ đa thiết bị: Các câu lệnh trong ADB cho phép nhà phát triển tương tác với nhiều thiết bị Android khác nhau, đảm bảo tính đa dạng và tương thích của ứng dụng trên các thiết bị khác nhau.

1.5.2.f. Viết ứng dụng theo dõi logcat của các package cần thiết

- ❖ Xác định các quyền cần thiết cho gói ứng dụng:
 - ✓ android.permission.READ_LOGS
 - ✓ android.permission.INTERNET
 - ✓ android.permission.ACCESS_WIFI_STATE
 - ✓ android.permission.RECEIVE_BOOT_COMPLETED
 - ✓ android.permission.FOREGROUND_SERVICE
- ❖ Tạo class dữ liệu

Các log trong logcat có cấu trúc như sau:

08-16 (*date*) 10:34:31.567 (*time*) 4219 (*PID*) 4219 (*TID*) E (*tag*) ActivityThread (*log name*): Failed to find provider info for com.google.android.gsf.gservices (*message*).

Vậy nên tạo lớp LogItem với các thuộc tính như trên để chứa các dòng log nhận được.

- ❖ Tạo class để tiến hành đọc logcat
- Lấy PID package cần theo dõi

Bởi vì mỗi logcat thuộc một package sẽ chứa PID của package đó, nên để lấy được logcat của từng package riêng lẻ thì cần biết PID của package.

Tạo một Thread riêng chạy “adb logcat” để bắt được các dòng log có log name là ActivityManager và có chứa “Start proc” kèm theo “for activity”/”for service”/”for broadcast”, bởi vì những dòng log này thông báo

PID của các package đang chạy trong thiết bị. Cần theo dõi package nào thì lọc theo tên package đó. Rồi lưu vào một Map tương ứng với tên package và PID của nó.

Lấy logcat cần quan tâm của package sau:

```
package:com.securityandsafetythings.datetimecontrol
package:com.securityandsafetythings.messagebroker
package:com.securityandsafetythings.appmanager.app
package:com.securityandsafetythings.firmware
package:com.securityandsafetythings.io
package:com.securityandsafetythings.videopipeline
package:com.securityandsafetythings.networkcontrol
package:com.securityandsafetythings.adbauthorization
package:com.securityandsafetythings.gateway
package:com.securityandsafetythings.devicemanagement
package:com.securityandsafetythings.appsourceproxy
package:com.securityandsafetythings.deviceid
package:com.securityandsafetythings.cloudconnector.app
package:com.securityandsafetythings.crashreporter.app
package:com.securityandsafetythings.health
package:com.securityandsafetythings.webserver
package:com.securityandsafetythings.deviceapp
package:com.securityandsafetythings.userdb
package:com.securityandsafetythings.event
package:com.securityandsafetythings.wificonnect
package:com.securityandsafetythings.media
package:com.securityandsafetythings.onvif
package:com.securityandsafetythings.webui
```

Tạo một Thread tiếp theo để chạy “adb logcat” và lọc theo PID đã biết của package. Lọc kèm theo các dấu hiệu cần chú ý như: tag ERROR, crash, err, error, die.

1.5.2.g. Tạo mini web server trong ứng dụng

Tạo mini server cho ứng dụng bằng thư viện NanoHTTDP. Trước tiên, tạo một class thừa kế thư viện NanoHTTDP. Trong class này định nghĩa 3 thuộc tính có kiểu dữ liệu Map là htmlData, csvData, changedMap. Trong đó, htmlData để chứa dữ liệu hiển thị lên web theo từng package, csvData để chứa dữ liệu của từng package trong 24h để người dùng có thể download xuống, changeMap để ghi nhận giá trị khi web thay đổi dữ liệu.

Sau đó ghi đè phương thức `serve()` của thư viện NanoHTTPD để tạo giao diện cho web và nhận dữ liệu từ `htmlData` để hiển thị. Tạo thêm button chứa hàm chức năng tải dữ liệu nhận được từ thuộc tính `csvData`.

1.5.2.h. Tạo service cho ứng dụng chạy ngầm

Từ class đọc logcat khi này, kế thừa lớp Service của android, định nghĩa service trong tệp `AndroidManifest.xml`. Tạo receiver để nhận tín hiệu khi ứng dụng được khởi động thì bắt đầu service chạy ngầm, định nghĩa receiver trong tệp `AndroidManifest.xml` kèm theo các action:

`android.intent.action.PACKAGE_ADDED`
`android.intent.action.PACKAGE_INSTALL`
`android.intent.action.BOOT_COMPLETED`
`android.intent.action.ACTION_BOOT_COMPLETED`
`android.intent.action.REBOOT`
`android.intent.action.QUICKBOOT_POWERON`
`android.intent.action.ACTION_SHUTDOWN`

1.5.2.i. Cài đặt như ứng dụng hệ thống

Sử dụng `apksigner` để ký chứng thực là ứng dụng hệ thống android sử dụng public key của Google. Rồi cài đặt ứng dụng với lệnh của adb và grant quyền đầy đủ.

COM.SECURITYANDSAFETYTHINGS.WEBUI

Date	Time	PID	Level	Name	Message
See All					

COM.SECURITYANDSAFETYTHINGS.FIRMWARE

Date	Time	PID	Level	Name	Message
See All					

COM.SECURITYANDSAFETYTHINGS.ADBAUTHORIZATION

Date	Time	PID	Level	Name	Message
See All					

COM.SECURITYANDSAFETYTHINGS.CRASHREPORTER.APP

Date	Time	PID	Level	Name	Message
07-10-2023	16:46:45	4289	D	MainService	Handling crash event with tag=system_app_crash
See All					

COM.SECURITYANDSAFETYTHINGS.DEVICEID

Date	Time	PID	Level	Name	Message
See All					

COM.SECURITYANDSAFETYTHINGS.ONVIF

Date	Time	PID	Level	Name	Message
See All					

COM.SECURITYANDSAFETYTHINGS.CLOUDCONNECTOR.APP

Date	Time	PID	Level	Name	Message
07-10-2023	16:48:26	2956	E	AbstractHttpRequestClient	Not authorized..
07-10-2023	16:47:56	2956	E	AbstractHttpRequestClient	Not authorized..
07-10-2023	16:47:25	2956	E	AbstractHttpRequestClient	Not authorized..
07-10-2023	16:46:55	2956	E	AbstractHttpRequestClient	Not authorized..
07-10-2023	16:46:47	2956	D	CrashEventHandler	Handling crash event
See All					

Hình 1.6. Hình kết quả cho ứng dụng Logcast

1.5.3. *Làm việc với giao thức SOAP và kiểm thử Onvif*

Trong thời đại kỹ thuật số ngày nay, việc kết nối và trao đổi thông tin giữa các ứng dụng và hệ thống đã trở nên cực kỳ quan trọng. Để giải quyết vấn đề này, giao thức SOAP (Simple Object Access Protocol) đã ra đời và trở thành một phần không thể thiếu trong các hệ thống ứng dụng phức tạp. SOAP là một giao thức cơ sở cho việc trao đổi thông tin giữa các ứng dụng dựa trên web service, nơi mà sự giao tiếp giữa các ứng dụng được thực hiện bằng cách gửi và nhận các thông điệp dưới dạng tin nhắn XML.

1.5.3.a. *Cấu trúc SOAP*

Một thông điệp SOAP được biểu diễn dưới dạng XML và chứa các phần quan trọng sau:

- ❖ Phần tiêu đề (Header): Chứa thông tin bổ sung liên quan đến quá trình trao đổi dữ liệu như thông tin xác thực, mã hóa, hay quản lý phiên.
- ❖ Phần thân (Body): Chứa các tham số và giá trị dữ liệu gửi đi hoặc nhận về.
- ❖ Phần phụ (Fault): Chứa thông tin về lỗi nếu có bất kỳ vấn đề gì xảy ra trong quá trình trao đổi.

1.5.3.b. *Tích hợp SOAP vào ứng dụng*

Để tích hợp giao thức SOAP vào ứng dụng, các bước chính như sau:

- ❖ Xác định dịch vụ web: Xác định các chức năng mà ứng dụng cần trao đổi thông tin và cung cấp chúng dưới dạng dịch vụ web.
- ❖ Xây dựng WSDL (Web Services Description Language): WSDL là một tệp mô tả dịch vụ web, định nghĩa các tham số, phương thức và thông tin cần thiết để tạo một yêu cầu hợp lệ.
- ❖ Tạo yêu cầu SOAP: Tạo một yêu cầu SOAP dựa trên WSDL, bao gồm cấu trúc chính xác và các tham số cần thiết.
- ❖ Gửi yêu cầu: Gửi yêu cầu SOAP đến dịch vụ web thông qua giao thức mạng như HTTP.
- ❖ Xử lý phản hồi: Nhận phản hồi từ dịch vụ web, phân tích nó và xử lý dữ liệu nhận được.

1.5.3.c. *SOAP Action*

SOAP Action là một phần của tiêu đề (Header) trong thông điệp SOAP và đóng vai trò quan trọng trong việc xác định mục tiêu của yêu cầu SOAP. Nó cho phép dịch vụ web xác định phương thức hoặc hành động cụ thể mà yêu cầu này

đang hướng tới. Bằng cách xác định SOAP Action, người nhận có thể hiểu được mục đích của yêu cầu và thực hiện xử lý tương ứng.

Tuy nhiên, SOAP Action không bắt buộc và có thể được bỏ qua trong một số trường hợp. Trong trường hợp không có SOAP Action, phương thức mục tiêu sẽ phải được xác định thông qua các phần khác của thông điệp, ví dụ như tên phương thức nằm trong thân (Body) của thông điệp.

1.5.3.d. Ưu điểm và hạn chế của SOAP

❖ Ưu điểm:

Độ tin cậy cao: SOAP hỗ trợ các cơ chế xác thực và mã hóa dữ liệu, giúp tăng cường tính bảo mật và tin cậy trong quá trình trao đổi thông tin.

Độc lập với ngôn ngữ: Vì SOAP sử dụng XML để biểu diễn thông điệp, nó độc lập với ngôn ngữ lập trình, giúp các ứng dụng có thể tương tác với nhau một cách dễ dàng.

❖ Hạn chế:

Tính phức tạp: SOAP sử dụng cú pháp XML phong phú, dẫn đến thông điệp có thể trở nên phức tạp và lớn hơn so với các giao thức khác nhẹ nhàng hơn.

Hiệu suất: Vì thông điệp SOAP dựa trên XML, quá trình mã hóa và giải mã có thể tốn nhiều thời gian và tài nguyên, ảnh hưởng đến hiệu suất của hệ thống.

SOAP đóng vai trò quan trọng trong việc xây dựng và triển khai các dịch vụ web, cung cấp cơ chế trao đổi thông tin đáng tin cậy và độc lập với ngôn ngữ. Mặc dù có những hạn chế về hiệu suất và phức tạp, nhưng với sự phát triển của công nghệ, SOAP vẫn tiếp tục đóng vai trò quan trọng trong việc kết nối và tích hợp các hệ thống ứng dụng phức tạp trong thời đại số hiện nay.

1.5.3.e. Kiểm thử Onvif bằng phần mềm Onvif Test Tool

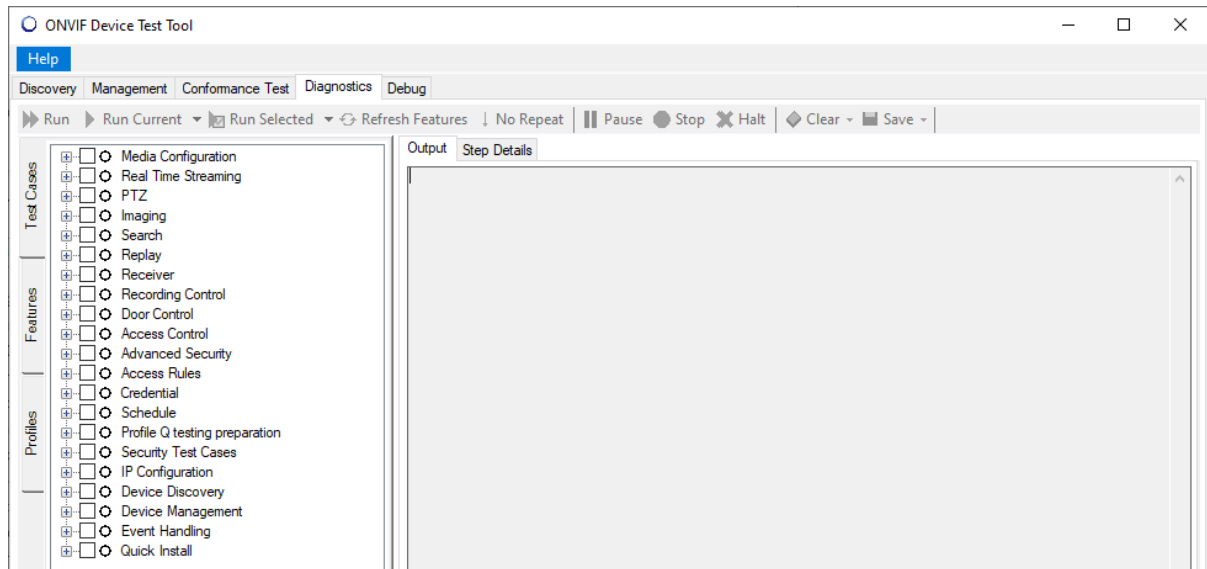
Kiểm thử theo chuẩn của profile S, gồm các tiêu chí sau:

GetCapabilities	GetHostname
GetWsdUrl	SetHostname
GetDiscoveryMode	GetDNS
SetDiscoveryMode	SetDNS
GetScopes	GetNetworkInterfaces
SetScopes	SetNetworkInterfaces
AddScopes	GetNetworkProtocols
RemoveScopes	SetNetworkProtocols

GetNetworkDefaultGateway	AddVideoEncoderConfiguration
SetNetworkDefaultGateway	RemoveVideoEncoderConfiguration
GetDeviceInformation	SetVideoEncoderConfiguration
GetSystemDateAndTime	GetCompatibleVideoEncoderConfigura
SetSystemDateAndTime	-tions
SetSystemFactoryDefault	GetVideoEncoderConfigurationOptions
Reboot	GetGuaranteedNumberOfVideoEncode
GetUsers	-rInstances
CreateUsers	GetProfiles
DeleteUsers	GetProfile
SetUser	CreateProfile
Notify	DeleteProfile
Subscribe	GetVideoSources
Renew	GetVideoSourceConfiguration
Unsubscribe	GetVideoSourceConfigurations
SetSynchronizationPoint	AddVideoSourceConfiguration
CreatePullPointSubscription	RemoveVideoSourceConfiguration
PullMessages	SetVideoSourceConfiguration
GetEventProperties	GetCompatibleVideoSourceConfigurati
TopicFilter	-ons
MessageContentFilter	GetVideoSourceConfigurationOptions
GetProfiles	GetMetadataConfiguration
GetStreamUri	GetMetadataConfigurations
Media Streaming using RTSP	AddMetadataConfiguration
MJPEG Media streaming using RTSP	RemoveMetadataConfiguration
Media Streaming using RTSP - JPEG	SetMetadataConfiguration
RTP header extension	GetCompatibleMetadataConfigurations
GetVideoEncoderConfiguration	GetMetadataConfigurationOption
GetVideoEncoderConfigurations	

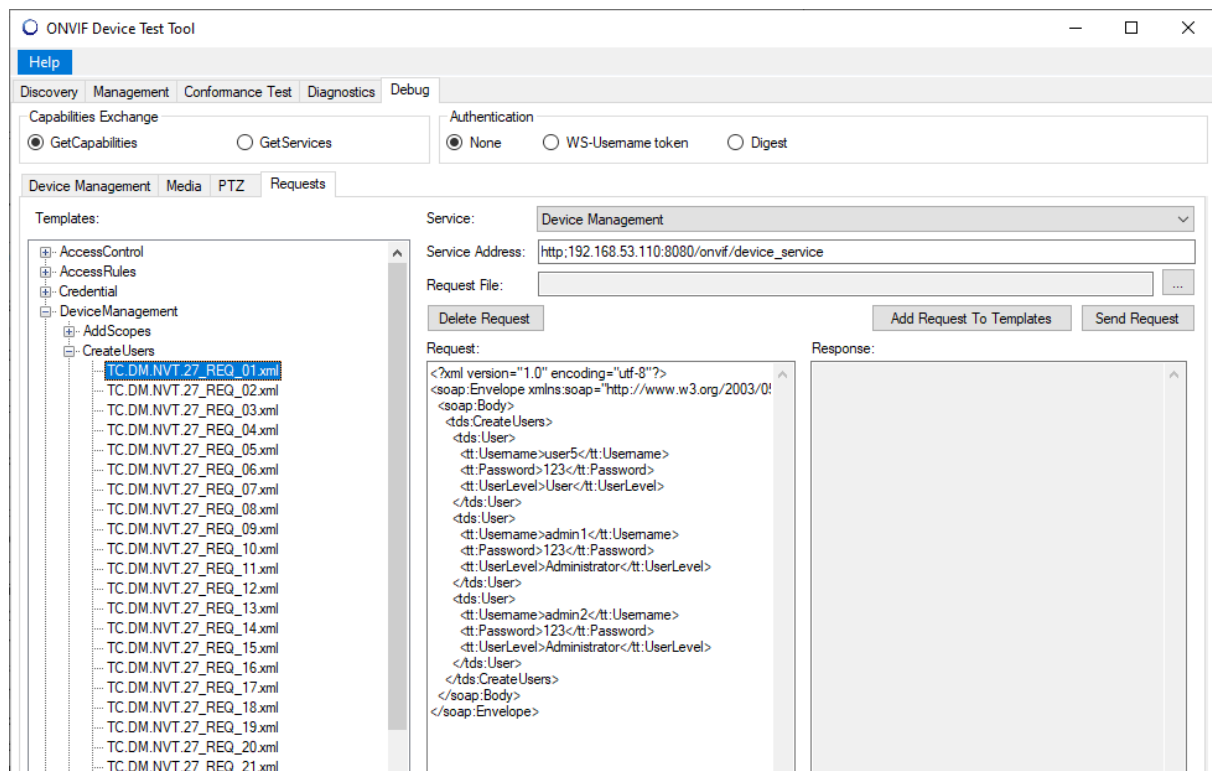
Đầu tiên dò tìm thiết bị để biết URI của thiết bị sau đó vào tab Diagnostic | Debug.

❖ Test tự động với Diagnostic Tab



Hình 1.7. Tool test Onvif tự động

❖ Test thủ công với Debug Tab



Hình 1.8. Tool test Onvif thủ công

Các test tool sẽ gửi request chứa các SOAP action đến thiết bị và trả về tình trạng của request kèm phản hồi của SOAP.

CHƯƠNG 2: QUY TRÌNH PHÁT TRIỂN ỨNG DỤNG ANDROID

2.1. Tổng quan về Android

Trong thời đại kỹ thuật số ngày nay, ứng dụng di động đóng vai trò ngày càng quan trọng trong cuộc sống hàng ngày của chúng ta. Và trong lĩnh vực phát triển ứng dụng di động, Android đã trở thành một nền tảng nổi tiếng và phổ biến.

2.1.1. Hệ điều hành

Android là hệ điều hành dựa trên nền tảng Linux được thiết kế chủ yếu để tương tác với thiết bị di động, như điện thoại thông minh và máy tính bảng. Hệ điều hành này đã phát triển rất nhiều trong 15 năm qua, bắt đầu từ điện thoại đen trắng đến các điện thoại thông minh hay các máy tính mini thời nay.

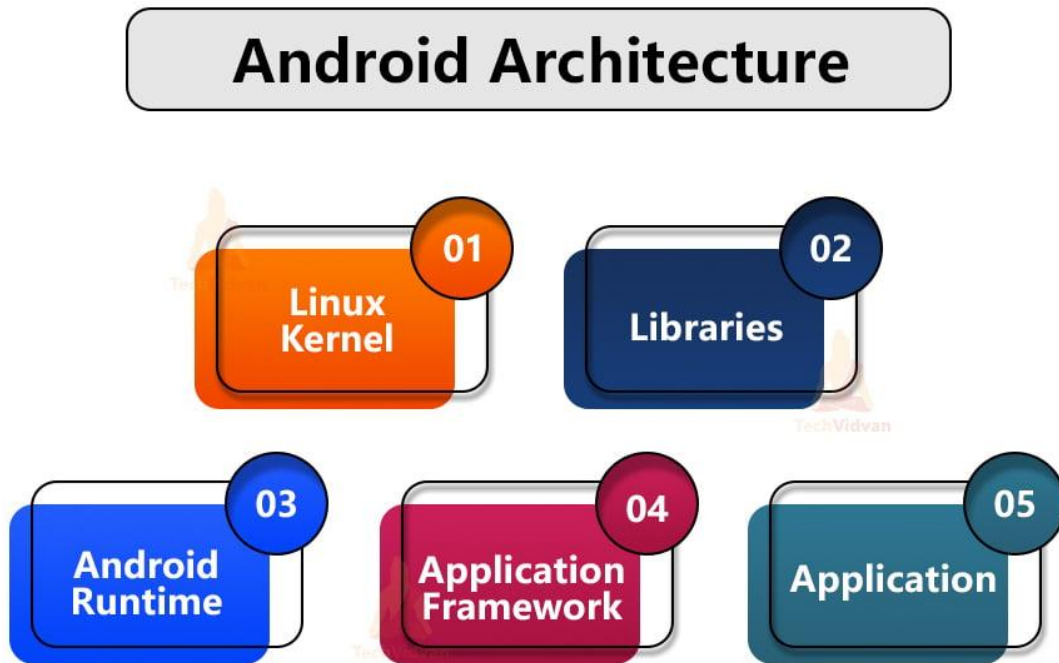
Android là một hệ điều hành mạnh mẽ và hỗ trợ lượng lớn các ứng dụng trên điện thoại thông minh. Các ứng dụng này ngày càng tiện lợi và hữu ích hơn cho người dùng. Phần cứng hỗ trợ cho các phần mềm Android được dựa trên kiến trúc nền tảng của ARM (Advanced RISC Machine), bên cạnh đó còn có armeabi (phiên bản cũ của ARM), x86, x86_64 (đối với android 11+). Android là hệ điều hành mã nguồn mở, điều này có nghĩa là nó miễn phí và bất kỳ ai cũng có thể sử dụng. Android đã có hơn một triệu ứng dụng khả dụng có thể giúp người dùng quản lý cuộc sống của họ bằng cách này hay cách khác và chi phí thấp trên thị trường là một lý do khiến Android trở nên phổ biến hơn.

Android được phát hành lần đầu tiên vào năm 2008 và nhanh chóng trở thành một trong những hệ điều hành di động phổ biến nhất trên thế giới. Điểm mạnh của Android nằm ở sự linh hoạt và tùy biến cao, cho phép người dùng tùy chỉnh giao diện và các ứng dụng theo sở thích cá nhân.

2.1.2. Kiến trúc của Android

Kiến trúc của hệ điều hành Android đóng vai trò quan trọng trong việc cung cấp một nền tảng mạnh mẽ và linh hoạt cho phát triển ứng dụng di động. Kiến trúc Android được thiết kế để đáp ứng các yêu cầu đa dạng của người dùng và nhà phát triển, từ giao diện người dùng cho đến cấu trúc hệ thống.

Kiến trúc Android dựa trên mô hình ngăn xếp (stack) phần mềm. Nó được chia thành các tầng khác nhau, mỗi tầng có nhiệm vụ và chức năng riêng biệt. Tầng cấp cao nhất của kiến trúc Android là ứng dụng người dùng, trong khi tầng cấp thấp nhất là phần cứng.



Hình 2.1. Minh họa cấu trúc của Android

Ứng dụng người dùng: Tầng này chứa các ứng dụng và giao diện người dùng mà người dùng cuối có thể tương tác trực tiếp. Các ứng dụng này bao gồm các ứng dụng điện thoại, trò chơi, ứng dụng xem video, ứng dụng mạng xã hội và nhiều ứng dụng khác. Giao diện người dùng Android được thiết kế để cung cấp trải nghiệm tốt cho người dùng, bao gồm các thành phần như thanh trạng thái, hộp ứng dụng và các biểu tượng chức năng.

Khung ứng dụng: Tầng này cung cấp các dịch vụ và API (Application Programming Interface) cho việc phát triển ứng dụng Android. Nó bao gồm các thành phần như Activity Manager (quản lý hoạt động), Package Manager (quản lý gói ứng dụng), Window Manager (quản lý cửa sổ) và các dịch vụ khác như Notification Manager (quản lý thông báo). Khung ứng dụng giúp các nhà phát triển xây dựng và quản lý ứng dụng một cách dễ dàng và hiệu quả.

Runtime Android: Tầng này chứa môi trường thực thi ứng dụng Android. Trước đây, Android sử dụng máy ảo Dalvik (Dalvik Virtual Machine - DVM) để thực thi mã Java. Tuy nhiên, từ phiên bản Android 5.0, Google đã chuyển sang sử dụng ART (Android Runtime) - một môi trường thực thi mã mới cải tiến hiệu suất và tốc độ. ART chuyển mã Java thành mã máy trực tiếp trước khi chạy ứng dụng, giúp tăng cường hiệu năng và tiết kiệm tài nguyên hệ thống.

Thư viện Android: Tầng này chứa các thư viện cung cấp các chức năng và công cụ hỗ trợ cho việc phát triển ứng dụng Android. Các thư viện này bao gồm các thành phần như thư viện đồ họa 2D/3D (OpenGL ES), thư viện đa phương tiện

(Media Framework), thư viện định vị (Location Framework) và nhiều thư viện khác. Nhờ vào các thư viện này, các nhà phát triển có thể tận dụng các chức năng phức tạp mà không cần xây dựng từ đầu.

Hạ tầng nhân (Kernel): Tầng cuối cùng trong kiến trúc Android là hạ tầng nhân, được dựa trên Linux. Hạ tầng nhân đóng vai trò quan trọng trong việc quản lý phần cứng và tương tác với các thành phần phần cứng của thiết bị. Nó cung cấp giao tiếp giữa phần mềm và phần cứng, bao gồm quản lý bộ nhớ, quản lý tài nguyên và truyền thông dữ liệu.

2.1.3. Ưu – nhược điểm của Android

Android, là một hệ điều hành di động phổ biến, đã tạo ra sự cách mạng trong ngành công nghiệp di động và mang lại nhiều lợi ích cho người dùng và nhà phát triển. Tuy nhiên, nó cũng đặt ra một số thách thức và hạn chế. Bằng cách nắm bắt và hiểu rõ các ưu và nhược điểm của Android, người dùng và nhà phát triển có thể đưa ra quyết định thông thái về việc sử dụng và phát triển ứng dụng trên nền tảng này.

2.1.3.a. Ưu điểm

❖ Linh hoạt và tùy biến

Android cung cấp một môi trường linh hoạt và tùy biến, cho phép người dùng tùy chỉnh giao diện và cài đặt theo sở thích cá nhân. Người dùng có thể chọn từ nhiều ứng dụng và giao diện người dùng khác nhau, tạo ra trải nghiệm cá nhân hóa.

❖ Hệ sinh thái ứng dụng phong phú

Google Play Store, cửa hàng ứng dụng chính thức của Android, cung cấp hàng triệu ứng dụng và trò chơi khác nhau. Điều này mang lại sự đa dạng và lựa chọn cho người dùng, đồng thời tạo ra cơ hội kinh doanh cho nhà phát triển ứng dụng.

❖ Tích hợp dịch vụ Google

Android tích hợp một loạt các dịch vụ của Google như Google Search, Google Maps và Google Drive. Điều này mang lại sự liên kết với các dịch vụ trực tuyến và tiện ích cao, giúp người dùng truy cập và quản lý dữ liệu dễ dàng.

❖ Đa nền tảng và đa thiết bị

Android hỗ trợ trên nhiều thiết bị di động khác nhau, bao gồm điện thoại thông minh, máy tính bảng, đồng hồ thông minh và TV thông minh. Điều này

tạo ra một hệ sinh thái đa dạng và cho phép người dùng trải nghiệm Android trên nhiều thiết bị khác nhau.

2.1.3.b. *Nhược điểm*

❖ Sự phân mảnh

Vì Android chạy trên nhiều thiết bị khác nhau từ nhiều nhà sản xuất, sự phân mảnh là một vấn đề phổ biến. Mỗi nhà sản xuất có thể tùy chỉnh giao diện và cài đặt riêng, gây ra sự không đồng nhất trong trải nghiệm người dùng và khó khăn trong việc phát triển ứng dụng đa nền tảng.

❖ Bảo mật

Mặc dù Android đã nâng cao bảo mật, nhưng vẫn có nguy cơ mở và lỗ hổng bảo mật. Với số lượng lớn thiết bị và ứng dụng Android trên thị trường, việc bảo vệ chống lại mã độc và các cuộc tấn công là một thách thức liên tục.

❖ Hiệu suất

Một số thiết bị Android có thể trải qua vấn đề về hiệu suất, đặc biệt là khi chạy các ứng dụng phức tạp hoặc trên các phiên bản Android cũ hơn. Điều này có thể gây ra sự chậm trễ và không ổn định trong trải nghiệm người dùng.

❖ Cập nhật hệ điều hành

Sự phân mảnh của Android cũng gây ra vấn đề về cập nhật hệ điều hành. Mỗi nhà sản xuất và nhà mạng có quyền quyết định việc phát hành cập nhật, dẫn đến sự chậm trễ hoặc không có cập nhật cho các thiết bị Android, điều này ảnh hưởng đến việc sử dụng các tính năng và bảo mật mới nhất của hệ điều hành.

2.1.4. *Phát triển ứng dụng Android*

Phát triển ứng dụng Android đã trở thành một lĩnh vực quan trọng và đầy triển vọng trong lĩnh vực công nghệ thông tin. Với sự phổ biến ngày càng tăng của thiết bị di động, việc tạo ra các ứng dụng Android đáp ứng nhu cầu của người dùng đã trở thành một mục tiêu quan trọng của các nhà phát triển phần mềm.

Phát triển ứng dụng Android cung cấp cho các nhà phát triển một nền tảng mạnh mẽ để xây dựng các ứng dụng đa dạng, từ ứng dụng di động cho công việc hàng ngày cho đến trò chơi và ứng dụng giải trí. Android sử dụng ngôn ngữ lập trình Java và Kotlin, đem lại sự linh hoạt và tiện lợi cho các nhà phát triển trong việc xây dựng và triển khai ứng dụng.

Một yếu tố quan trọng của phát triển ứng dụng Android là sự tích hợp với các dịch vụ của Google. Với việc tích hợp dịch vụ như Google Maps, Google Drive và Google Search, các nhà phát triển có thể tận dụng sức mạnh của các dịch vụ này để tạo ra những ứng dụng phong phú và đa chức năng.

Một trong những ưu điểm của phát triển ứng dụng Android là sự phổ biến và sự đa dạng của cộng đồng phát triển. Cộng đồng này cung cấp tài liệu phong phú, hỗ trợ kỹ thuật và các nguồn tài nguyên khác nhau để giúp các nhà phát triển nắm bắt kiến thức và kỹ năng cần thiết để xây dựng ứng dụng Android chất lượng cao.

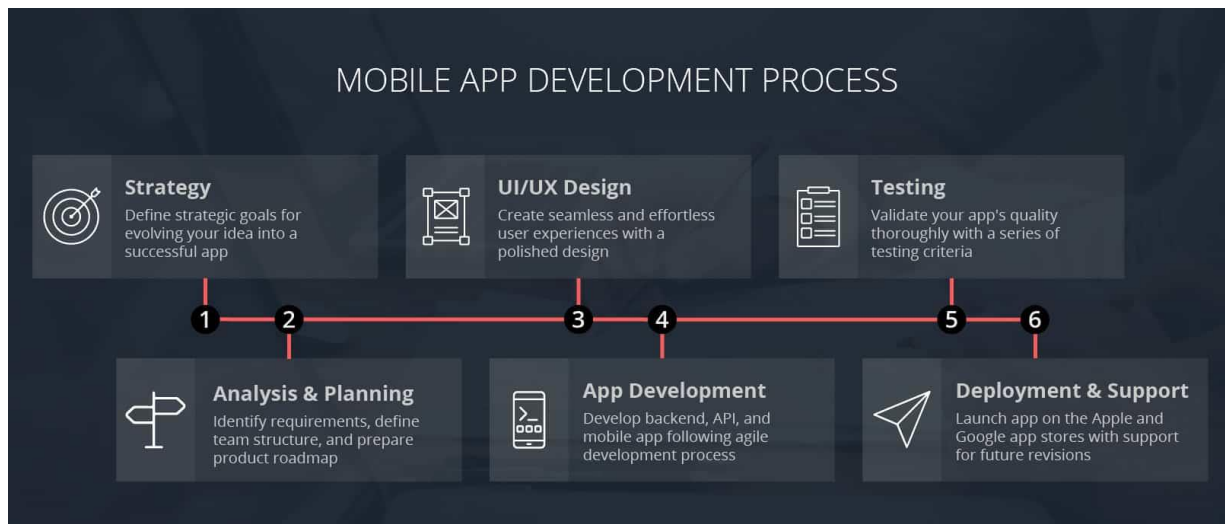
Để phát triển ứng dụng Android, các nhà phát triển sử dụng Android Studio – môi trường phát triển tích hợp (IDE) chính thức của Google cho phát triển ứng dụng Android. Android Studio cung cấp các công cụ mạnh mẽ để phát triển, thử nghiệm và triển khai ứng dụng Android. Nó bao gồm trình biên dịch, trình gỡ lỗi và giao diện người dùng thân thiện để giúp các nhà phát triển tạo ra ứng dụng chất lượng cao và tối ưu hóa hiệu suất.

Khi phát triển ứng dụng Android, quá trình kiểm thử và sửa lỗi là rất quan trọng. Các nhà phát triển cần đảm bảo rằng ứng dụng hoạt động một cách ổn định trên nhiều thiết bị và phiên bản Android khác nhau. Đồng thời, việc đảm bảo bảo mật và bảo vệ dữ liệu cá nhân của người dùng cũng là một yếu tố quan trọng trong quá trình phát triển ứng dụng Android.

Cuối cùng, khi phát triển ứng dụng Android, việc tối ưu hóa hiệu suất và trải nghiệm người dùng là yếu tố quan trọng. Với sự cạnh tranh khốc liệt trên thị trường ứng dụng di động, ứng dụng cần có hiệu suất tốt và giao diện người dùng hấp dẫn để thu hút và giữ chân người dùng.

2.2. Quy trình phát triển ứng dụng trong doanh nghiệp Viettel High Tech

Tổng công ty Công nghiệp Công nghệ cao Viettel là một trong những công ty hàng đầu trong lĩnh vực công nghệ thông tin và truyền thông tại Việt Nam. Với mục tiêu đem lại các giải pháp công nghệ tiên tiến và đột phá, Viettel đã thiết lập một quy trình phát triển ứng dụng hiệu quả để đáp ứng nhu cầu ngày càng tăng về ứng dụng di động.



Hình 2.2. Minh họa sơ lược quy trình phát triển ứng dụng di động

2.2.1. Thu thập yêu cầu

Quy trình phát triển ứng dụng bắt đầu bằng việc thu thập yêu cầu từ khách hàng hoặc bộ phận nội bộ của Viettel. Các yêu cầu này bao gồm các tính năng và chức năng cần thiết của ứng dụng, giao diện người dùng, hiệu năng, bảo mật và các yêu cầu khác. Quá trình này thường liên tục và đòi hỏi sự tương tác chặt chẽ giữa đội ngũ phát triển và khách hàng.

2.2.2. Thiết kế và lập kế hoạch

Sau khi thu thập yêu cầu, đội ngũ phát triển ứng dụng trong Viettel sẽ tiến hành thiết kế và lập kế hoạch. Các giai đoạn thiết kế bao gồm thiết kế giao diện người dùng, cơ sở dữ liệu, kiến trúc hệ thống và các thành phần chức năng khác. Kế hoạch phát triển cụ thể bao gồm lịch trình, phân công công việc và các milestone quan trọng trong quá trình phát triển.

2.2.3. Phát triển và kiểm thử

Sau khi hoàn thành thiết kế và lập kế hoạch, quy trình phát triển chuyển sang giai đoạn phát triển và kiểm thử. Đội ngũ phát triển trong Viettel sử dụng các công nghệ và công cụ phát triển tiên tiến để triển khai ứng dụng. Các kiểm thử bao gồm kiểm thử chức năng, kiểm thử đơn vị, kiểm thử hệ thống và kiểm thử hiệu năng. Quá trình kiểm thử giúp đảm bảo ứng dụng hoạt động một cách ổn định và đáp ứng yêu cầu của khách hàng.

2.2.4. Triển khai và phân phối

Sau khi ứng dụng đã hoàn thành và kiểm thử, quy trình phát triển tiếp tục với giai đoạn triển khai và phân phối. Đội ngũ phát triển sẽ triển khai ứng dụng trên môi

trường sản xuất hoặc môi trường thử nghiệm, tùy thuộc vào yêu cầu và kế hoạch. Quá trình triển khai bao gồm cài đặt và cấu hình hệ thống, chuẩn bị cơ sở hạ tầng và đảm bảo sự tương thích với các thành phần khác.

2.2.5. Bảo trì và nâng cấp

Sau khi ứng dụng đã được triển khai, quy trình phát triển tiếp tục với giai đoạn bảo trì và nâng cấp. Đội ngũ phát triển sẽ theo dõi và giám sát hoạt động của ứng dụng, xử lý sự cố và triển khai các bản vá và cập nhật để cải thiện tính ổn định và bảo mật. Ngoài ra, việc nâng cấp ứng dụng để đáp ứng nhu cầu mới và thay đổi của khách hàng cũng được thực hiện trong giai đoạn này.

2.2.6. Đánh giá và phản hồi

Cuối cùng, sau quá trình triển khai và bảo trì, quy trình phát triển ứng dụng trong Viettel bao gồm giai đoạn đánh giá và phản hồi. Đánh giá giúp đánh giá hiệu suất của ứng dụng và đảm bảo rằng các yêu cầu ban đầu đã được đáp ứng. Phản hồi từ khách hàng và người dùng được thu thập để cải thiện và cung cấp phiên bản ứng dụng tốt hơn trong tương lai.

CHƯƠNG 3: CHUẨN ONVIF TRÊN CAMERA GIÁM SÁT

3.1. Tổng quan về lĩnh vực giám sát và vai trò của chuẩn ONVIF

3.1.1. Lĩnh vực giám sát

Trong xã hội hiện đại, giám sát an ninh đóng một vai trò cực kỳ quan trọng trong việc bảo vệ an toàn và bảo mật của cá nhân, tài sản và các khu vực công cộng. Các hệ thống giám sát an ninh, đặc biệt là camera giám sát, đã trở thành công cụ chính để ghi lại và giám sát các hoạt động, nhận diện sự cố và hỗ trợ quản lý an ninh.

Camera giám sát đóng vai trò quan trọng trong việc thu thập dữ liệu hình ảnh và video từ môi trường xung quanh. Chúng được sử dụng rộng rãi trong các lĩnh vực như bảo vệ an ninh, giám sát giao thông, quản lý tài sản và quản lý môi trường. Tuy nhiên, các camera giám sát thường được sản xuất bởi nhiều nhà sản xuất khác nhau, với các đặc điểm riêng biệt và giao thức giao tiếp khác nhau.

3.1.2. Vai trò của chuẩn ONVIF

Trong các môi trường phức tạp, nhu cầu tích hợp và tương tác giữa các hệ thống giám sát là cực kỳ quan trọng. Điều này cho phép các hệ thống hoạt động cùng nhau, truyền dữ liệu và chia sẻ thông tin để tạo ra một hệ thống giám sát toàn diện. Tuy nhiên, việc tích hợp giữa các hệ thống giám sát từ các nhà sản xuất khác nhau có thể gặp nhiều khó khăn do sự không tương thích về giao thức, giao diện và cấu trúc dữ liệu.

Để giải quyết vấn đề không tương thích giữa các hệ thống giám sát, tổ chức ONVIF (Open Network Video Interface Forum) đã phát triển và giới thiệu chuẩn ONVIF. ONVIF là một chuẩn giao tiếp mở, đa nhà sản xuất và đa nhà cung cấp dựa trên IP (Internet Protocol) để thống nhất giao tiếp giữa các thiết bị giám sát, bao gồm camera giám sát, đầu ghi hình (NVR) và các ứng dụng quản lý giám sát.

3.2. Lý thuyết cơ bản về chuẩn ONVIF

3.2.1. Nguyên lý và mục tiêu của chuẩn ONVIF

Chuẩn ONVIF (Open Network Video Interface Forum) có mục tiêu chính là thống nhất giao tiếp giữa các thiết bị giám sát từ các nhà sản xuất khác nhau. Điều này giúp đảm bảo tính tương thích, tích hợp và tương tác giữa các hệ thống giám sát. ONVIF cho phép các thiết bị giám sát gửi và nhận dữ liệu, điều khiển và truy cập từ xa, cũng như quản lý các thiết bị và dịch vụ giám sát.

3.2.2. *Kiến trúc tổng quan của chuẩn ONVIF*

Chuẩn ONVIF xây dựng một kiến trúc tổng quan để thực hiện việc thống nhất giao tiếp giữa các thiết bị giám sát. Kiến trúc này bao gồm các thành phần chính sau đây:

- ❖ Thành phần Devices: Đại diện cho các thiết bị giám sát như camera, đầu ghi hình (NVR) và các thiết bị khác.
- ❖ Thành phần Network Video Transmitter (NVT): Chịu trách nhiệm gửi dữ liệu video từ camera hoặc các nguồn video khác.
- ❖ Thành phần Network Video Recorder (NVR): Lưu trữ và quản lý dữ liệu video từ các nguồn video.
- ❖ Thành phần Client: Đại diện cho ứng dụng hoặc hệ thống quản lý giám sát, cho phép người dùng truy cập và điều khiển các thiết bị giám sát.
- ❖ Thành phần Event: Quản lý và xử lý các sự kiện liên quan đến giám sát như phát hiện chuyển động, báo động, và hơn thế nữa.

3.2.3. *Các giao thức và tiêu chuẩn liên quan trong chuẩn ONVIF*

Chuẩn ONVIF sử dụng các giao thức và tiêu chuẩn liên quan để thực hiện việc thống nhất giao tiếp giữa các thiết bị giám sát. Các giao thức và tiêu chuẩn quan trọng trong chuẩn ONVIF bao gồm:

- ❖ Simple Object Access Protocol (SOAP): Giao thức truyền thông dựa trên XML (Extensible Markup Language) cho việc trao đổi dữ liệu giữa các thiết bị và ứng dụng.
- ❖ WS-Addressing: Tiêu chuẩn địa chỉ Web dựa trên SOAP cho việc định danh và định vị các thông điệp.
- ❖ WS-Security: Tiêu chuẩn bảo mật Web dựa trên SOAP cho việc đảm bảo tính toàn vẹn, xác thực và phân quyền trong giao tiếp.
- ❖ Real-Time Streaming Protocol (RTSP): Giao thức truyền phát video và âm thanh thời gian thực qua mạng.
- ❖ Profiles: ONVIF định nghĩa các Profiles (Hồ sơ) để đảm bảo tính tương thích và tương tác giữa các thiết bị. Mỗi Profile xác định một tập hợp các tính năng và giao thức cần thiết cho các ứng dụng cụ thể.

3.3. *Các chức năng và tính năng của ONVIF*

3.3.1. *Nhận dạng và xác thực*

ONVIF cung cấp chức năng nhận dạng và xác thực giữa các thiết bị giám sát. Điều này đảm bảo rằng các thiết bị có thể xác định và xác thực lẫn nhau trong một

hệ thống tích hợp. ONVIF sử dụng các tiêu chuẩn và giao thức bảo mật để đảm bảo tính toàn vẹn và xác thực của dữ liệu truyền qua mạng.

3.3.2. *Tìm kiếm và truy cập*

ONVIF hỗ trợ chức năng tìm kiếm và truy cập vào các thiết bị giám sát từ xa. Thông qua giao thức WS-Discovery, các thiết bị có thể tự động được tìm thấy và liệt kê trong mạng. Người dùng có thể dễ dàng truy cập vào các thiết bị được tìm thấy thông qua giao diện ứng dụng hoặc hệ thống quản lý giám sát.

3.3.3. *Truyền dữ liệu và phát sóng*

ONVIF hỗ trợ việc truyền dữ liệu video và âm thanh từ các thiết bị giám sát. Sử dụng giao thức RTSP, người dùng có thể truy cập và xem video trực tiếp từ camera hoặc NVR. Điều này cho phép người dùng giám sát và theo dõi các vùng quan trọng từ xa thông qua mạng.

3.3.4. *Quản lý thiết bị và dịch vụ:*

ONVIF cung cấp các chức năng quản lý thiết bị và dịch vụ giám sát. Điều này bao gồm khả năng quản lý và cấu hình các thiết bị giám sát, như điều chỉnh cài đặt, thay đổi độ phân giải, và xác định các quy tắc giám sát. Ngoài ra, ONVIF cũng hỗ trợ quản lý các dịch vụ giám sát, bao gồm xác thực người dùng, cấp quyền truy cập, và xác định các chính sách an ninh.

3.3.5. *Khả năng tích hợp và tương thích*

Một trong những điểm mạnh của ONVIF là khả năng tích hợp và tương thích với các hệ thống và thiết bị giám sát từ các nhà sản xuất khác nhau. Chuẩn ONVIF đảm bảo rằng các thiết bị tương thích có thể hoạt động cùng nhau và chia sẻ dữ liệu một cách dễ dàng. Điều này giúp tạo ra một hệ thống giám sát toàn diện và linh hoạt.

3.4. *Triển khai và ứng dụng thực tế của ONVIF*

3.4.1. *Triển khai chuẩn ONVIF trên các hệ thống giám sát*

Triển khai chuẩn ONVIF trên các hệ thống giám sát đòi hỏi các bước thực hiện cụ thể. Điều này bao gồm việc xác định các thiết bị và ứng dụng hỗ trợ chuẩn ONVIF, cấu hình các thiết bị để hỗ trợ chuẩn này, và kết nối và tích hợp các thiết bị vào hệ thống giám sát tổng thể. Các bước triển khai này đòi hỏi hiểu biết về các giao thức, giao diện và cấu trúc dữ liệu của chuẩn ONVIF.

3.4.2. Ví dụ về ứng dụng và áp dụng chuẩn ONVIF

Chuẩn ONVIF có thể được áp dụng trong nhiều môi trường và ứng dụng thực tế. Ví dụ, trong lĩnh vực giám sát an ninh, chuẩn ONVIF cho phép tích hợp các camera giám sát từ các nhà sản xuất khác nhau vào cùng một hệ thống quản lý giám sát. Điều này giúp quản lý và kiểm soát nhiều khu vực và thiết bị từ một giao diện duy nhất.

Một ứng dụng khác của chuẩn ONVIF là trong hệ thống giám sát giao thông. Các camera giám sát giao thông có thể sử dụng chuẩn ONVIF để truyền dữ liệu và cung cấp thông tin về tình trạng giao thông và an ninh đến các trung tâm điều khiển. Điều này giúp cải thiện việc quản lý giao thông, giảm thiểu tai nạn và cải thiện an ninh đô thị.

Ngoài ra, chuẩn ONVIF cũng có thể được áp dụng trong các hệ thống giám sát tại chỗ, như trong các tòa nhà, trung tâm thương mại, bệnh viện và trường học. Bằng cách tích hợp các thiết bị giám sát từ các nhà sản xuất khác nhau thông qua chuẩn ONVIF, người dùng có thể quản lý và giám sát các khu vực quan trọng một cách dễ dàng và hiệu quả.

3.4.3. Lợi ích và thách thức của việc sử dụng chuẩn ONVIF

Việc sử dụng chuẩn ONVIF mang lại nhiều lợi ích cho các hệ thống giám sát. Đầu tiên, nó cho phép tích hợp và tương tác giữa các thiết bị giám sát từ các nhà sản xuất khác nhau, đảm bảo tính tương thích và tích hợp linh hoạt. Thứ hai, chuẩn ONVIF giúp đơn giản hóa việc quản lý và kiểm soát các thiết bị giám sát từ một giao diện duy nhất. Thứ ba, ONVIF cung cấp các chức năng và tính năng tiêu chuẩn để đảm bảo tính bảo mật, xác thực và quản lý dữ liệu.

Tuy nhiên, việc triển khai và sử dụng chuẩn ONVIF cũng đặt ra một số thách thức. Một trong những thách thức chính là sự đảm bảo tính tương thích và tương thích ngược giữa các phiên bản ONVIF và các thiết bị giám sát. Hơn nữa, việc cấu hình và tích hợp các thiết bị giám sát có thể đòi hỏi kiến thức kỹ thuật và kỹ năng chuyên môn.

CHƯƠNG 4: PHÁT TRIỂN ỨNG DỤNG TÍCH HỢP TRÊN CAMERA GIÁM SÁT

Ứng dụng Android tích hợp vào camera giám sát đã trở thành một công cụ quan trọng và tiện ích để nâng cao an ninh và giám sát trong các môi trường như nhà ở, công ty, trường học và các khu vực công cộng. Việc phát triển một ứng dụng Android cho camera giám sát không chỉ giải quyết những nhu cầu cơ bản của người dùng như xem trực tiếp hình ảnh và quản lý từ xa, mà còn mở ra rất nhiều tiềm năng để phát triển các chức năng mở rộng. Trong luận văn này, chúng ta sẽ tìm hiểu về các chức năng tiềm năng mà ứng dụng Android có thể tích hợp vào camera giám sát để tăng cường hiệu quả an ninh và giám sát.

4.1. Phát triển ứng dụng Android trên thiết bị camera giám sát

Việc sử dụng công nghệ để giám sát giao thông đã trở thành một yếu tố quan trọng trong việc tăng cường an toàn và hiệu quả giao thông. Doanh nghiệp Viettel, một tập đoàn viễn thông hàng đầu tại Việt Nam, đã phát triển một dòng sản phẩm thiết bị camera giám sát giao thông dựa trên nền tảng Android. Quá trình phát triển ứng dụng Android trên thiết bị này đòi hỏi một quy trình công phu và cẩn thận để đảm bảo rằng ứng dụng đáp ứng yêu cầu chất lượng và đáp ứng nhu cầu của người dùng.

4.1.1. Đặt mục tiêu và yêu cầu

Quy trình phát triển ứng dụng Android trên thiết bị camera giám sát giao thông của Viettel bắt đầu bằng việc đặt ra mục tiêu và yêu cầu rõ ràng cho ứng dụng. Các yêu cầu bao gồm khả năng quay phim và chụp ảnh chất lượng cao, khả năng phân tích và nhận dạng biển số xe, gửi dữ liệu và cảnh báo liên tục và khả năng kết nối và quản lý từ xa.

4.1.2. Thiết kế và kế hoạch

Sau khi xác định mục tiêu và yêu cầu, quy trình phát triển ứng dụng Android trên thiết bị camera giám sát giao thông của Viettel tiến hành thiết kế và lập kế hoạch cho ứng dụng. Các nhà phát triển và nhà thiết kế tạo ra bản thiết kế giao diện người dùng, kiến trúc ứng dụng và lập kế hoạch triển khai. Đồng thời, các yếu tố quan trọng khác như tích hợp với hệ thống camera, cơ chế lưu trữ dữ liệu và bảo mật cũng được xem xét trong quá trình thiết kế.

4.1.3. Phát triển và kiểm thử

Quá trình phát triển ứng dụng Android trên thiết bị camera giám sát giao thông của Viettel tiếp tục với việc xây dựng và kiểm thử ứng dụng. Nhóm phát triển

sử dụng các công cụ và công nghệ phát triển hiện đại để tạo ra ứng dụng theo thiết kế và yêu cầu đã được xác định trước đó. Các quy trình kiểm thử chặt chẽ được thực hiện để đảm bảo rằng ứng dụng hoạt động một cách ổn định, chất lượng hình ảnh và tính năng phân tích biển số xe đáp ứng yêu cầu và không có lỗi.

4.1.4. Triển khai và phân phối

Khi ứng dụng đã được phát triển và kiểm thử hoàn tất, quy trình phát triển ứng dụng Android trên thiết bị camera giám sát giao thông của Viettel tiếp tục với giai đoạn triển khai và phân phối. Ứng dụng được cài đặt và tích hợp trên các thiết bị camera giám sát giao thông của Viettel và triển khai trên các điểm giám sát giao thông. Các quy trình phân phối chính thức như việc cài đặt, cấu hình và kiểm tra hệ thống cũng được thực hiện để đảm bảo tính tương thích và hiệu suất của ứng dụng.

4.1.5. Quản lý và duy trì

Sau khi ứng dụng được triển khai, quy trình phát triển ứng dụng Android trên thiết bị camera giám sát giao thông của Viettel không kết thúc ở đây. Doanh nghiệp Viettel tiếp tục quản lý và duy trì ứng dụng, bao gồm cập nhật, vá lỗi, theo dõi hiệu suất và đáp ứng các yêu cầu và phản hồi từ người dùng.

4.2. Các ứng dụng tích hợp hệ thống camera giám sát giao thông

Ứng dụng Android tích hợp camera giám sát giao thông đóng vai trò quan trọng trong việc nâng cao hiệu quả quản lý giao thông và đảm bảo an toàn cho người tham gia giao thông. Phát triển các chức năng cho ứng dụng này là mục tiêu quan trọng nhằm tăng cường khả năng giám sát và quản lý thông qua camera giám sát giao thông.

4.2.1. Nhận dạng biển số xe

Chức năng nhận dạng biển số xe là một trong những chức năng quan trọng trong ứng dụng Android tích hợp camera giám sát giao thông. Nó cho phép hệ thống nhận dạng và ghi lại thông tin từ biển số xe trong khung hình từ camera. Chức năng này giúp cung cấp dữ liệu về vi phạm giao thông, theo dõi lưu thông và tăng cường an ninh trong việc kiểm soát phương tiện di chuyển trên đường.

4.2.2. Phân tích dữ liệu giao thông

Chức năng phân tích dữ liệu giao thông là một yếu tố quan trọng để cải thiện quá trình quản lý giao thông. Ứng dụng Android tích hợp camera giám sát giao thông có thể sử dụng các thuật toán xử lý hình ảnh và phân tích dữ liệu để đếm số lượng phương tiện, phân loại phương tiện hoặc phát hiện các hành vi vi phạm giao

thông. Thông tin này giúp cải thiện độ chính xác và hiệu suất của quá trình quản lý giao thông.

4.2.3. Ghi lại video

Chức năng ghi lại video cho phép ứng dụng Android tích hợp camera giám sát giao thông ghi lại các sự kiện giao thông và lưu trữ chúng. Việc này cung cấp dữ liệu và bằng chứng cho việc phân tích, đánh giá và xử lý vi phạm giao thông, đồng thời cung cấp tài liệu tham khảo cho việc giải quyết tranh chấp về vi phạm giao thông.

4.2.4. Xử lý hình ảnh

Chức năng xử lý hình ảnh là một phần quan trọng của ứng dụng Android tích hợp camera giám sát giao thông. Các thuật toán xử lý hình ảnh có thể được áp dụng để phát hiện vật thể, nhận dạng biển báo giao thông, phát hiện chuyển động và thậm chí dự đoán tình huống giao thông nguy hiểm. Chức năng này cung cấp thông tin hữu ích và hỗ trợ quá trình giám sát và quản lý giao thông.

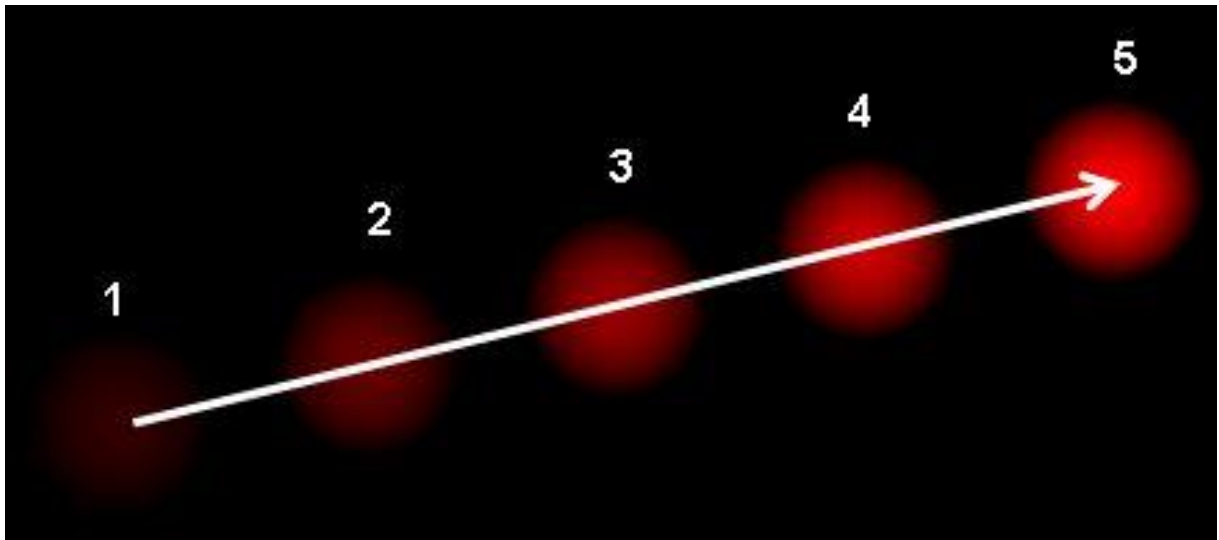
CHƯƠNG 5: KỸ THUẬT OPTICAL FLOW – THƯ VIỆN OPENCV

5.1. Kỹ thuật Optical Flow

Optical Flow là một giải thuật phân tích hình ảnh được sử dụng để tính toán sự di chuyển của các điểm ảnh trong một chuỗi hình ảnh. Giải thuật này cung cấp thông tin về vận tốc và hướng di chuyển của các điểm ảnh trong một hình ảnh hoặc video, cho phép phát hiện và theo dõi các vật thể di chuyển trong không gian 2D. Các phương pháp tính toán Optical Flow được sử dụng rộng rãi trong lĩnh vực thị giác máy tính và thị giác máy bay, và có nhiều ứng dụng trong thực tế như giám sát an ninh, điều khiển máy bay không người lái và phát hiện chuyển động trong y tế.

5.1.1. Nguyên lý hoạt động

Nguyên lý hoạt động của kỹ thuật Optical Flow dựa trên giả định rằng các điểm ảnh trong một chuỗi hình ảnh di chuyển theo một quy luật nào đó giữa các khung hình liên tiếp. Mục tiêu của Optical Flow là ước lượng và phân tích chuyển động của các điểm ảnh trong chuỗi hình ảnh đó.



Hình 5.1. Vector quỹ đạo chuyển động của vật thể trong 5 khung hình liên tiếp

Ý tưởng chính của Optical Flow là sử dụng sự thay đổi vị trí của các điểm ảnh trên một chuỗi hình ảnh để xác định vector chuyển động tương ứng cho mỗi điểm. Vector chuyển động này thể hiện hướng và tốc độ di chuyển của điểm ảnh tại mỗi khung hình. Các vector chuyển động này tạo thành một trường vector chuyển động (motion vector field), gọi là Optical Flow field, mô tả chuyển động của các điểm trong cảnh.

Để ước lượng Optical Flow, các phương pháp thường sử dụng các giả định về độ mịn (smoothness) và độ nhất quán (coherence) của Optical Flow. Giả định về

độ mịn cho rằng các điểm gần nhau trong không gian ảnh và thời gian thường có Optical Flow tương tự nhau. Giả định về độ nhất quán cho rằng chuyển động của các điểm ảnh trong một vùng nhỏ xung quanh điểm đang xét có thể được xấp xỉ bằng một phép biến đổi tuyến tính.

5.1.2. Các phương pháp tính toán giải thuật Optical Flow

Có nhiều phương pháp tính toán Optical Flow đã được phát triển trong suốt nhiều năm qua, nhằm đảm bảo tính chính xác và hiệu suất của quá trình ước lượng chuyển động. Dưới đây là một số phương pháp quan trọng và phổ biến nhất để tính toán Optical Flow.

5.1.2.a. Phương pháp Lucas – Kanade

Phương pháp Lucas-Kanade là một phương pháp cổ điển và phổ biến để tính toán Optical Flow. Được giới thiệu bởi Bruce D. Lucas và Takeo Kanade vào năm 1981, phương pháp này giả định rằng chuyển động của các điểm ảnh trong một vùng nhỏ xung quanh điểm đang xét có thể được xấp xỉ bằng một phép biến đổi tuyến tính.

Nguyên lý hoạt động của phương pháp Lucas-Kanade dựa trên việc ước lượng các tham số của phép biến đổi tạo ra chuyển động tốt nhất giữa hai khung hình liên tiếp. Quá trình tính toán được thực hiện như sau:

- ❖ Đầu tiên, một vùng cửa sổ (window) có kích thước nhỏ được chọn xung quanh mỗi điểm ảnh trong khung thứ nhất. Vùng cửa sổ này có thể là một hình vuông hoặc một hình chữ nhật.
- ❖ Tiếp theo, phương pháp sử dụng một giả định quan trọng, đó là chuyển động của các điểm ảnh trong vùng cửa sổ có thể được xấp xỉ bằng một phép biến đổi tuyến tính. Giả định này đồng nghĩa với việc giả sử rằng sự thay đổi của các điểm ảnh trong vùng cửa sổ chỉ phụ thuộc vào đạo hàm không gian (spatial gradient) và đạo hàm thời gian (temporal gradient) của các điểm ảnh này.
- ❖ Tiếp theo, phương pháp sử dụng các đạo hàm không gian và đạo hàm thời gian của các điểm ảnh trong vùng cửa sổ để xây dựng ma trận Jacobian và vector độ lệch (displacement vector). Ma trận Jacobian đại diện cho đạo hàm không gian của các điểm ảnh, trong khi vector độ lệch biểu thị sự chênh lệch giữa hai khung hình.
- ❖ Quá trình tối ưu hóa được thực hiện bằng cách giải hệ phương trình tuyến tính, trong đó ma trận Jacobian được nhân với vector độ lệch và cho kết quả là vector chuyển động (motion vector). Vector chuyển động này thể hiện hướng và tốc độ di chuyển của điểm ảnh tại mỗi khung hình.

Phương pháp Lucas-Kanade lặp lại quá trình trên cho tất cả các điểm ảnh trong ảnh hoặc trong một số điểm được chọn. Kết quả là một trường vector chuyển động (motion vector field) biểu thị chuyển động của các điểm ảnh trong cảnh.

Phương pháp Lucas-Kanade có một số ưu điểm. Đầu tiên, nó đơn giản và dễ hiểu. Thứ hai, phương pháp này tốt cho các vùng nhỏ và có chuyển động tương đối nhỏ, vì nó giả định chuyển động là tuyến tính và thường cho kết quả tốt trong các tình huống như theo dõi các điểm đặc trưng trong video.

Tuy nhiên, phương pháp Lucas-Kanade cũng có một số hạn chế. Đầu tiên, nó không thể xử lý chuyển động lớn hoặc chuyển động không tuyến tính. Thứ hai, phương pháp này dễ bị ảnh hưởng bởi sự biến đổi ánh sáng mạnh hoặc nhiễu. Do đó, để tăng độ chính xác và ổn định, nhiều biến thể và cải tiến của phương pháp Lucas-Kanade đã được đề xuất, như việc sử dụng vùng cửa sổ biến đổi (warped window) để điều chỉnh sự thay đổi ánh sáng và tăng cường giảm nhiễu.

$$\text{Công thức: } \sum [\nabla I(x, y) \nabla I(x, y)^T] [u \ v]^T = -\sum [\nabla I(x, y) \nabla I(x, y)^T] \nabla I(x, y)^T$$

Phương pháp:

- ❖ Chọn các điểm quan sát ban đầu.
- ❖ Tính toán gradient của khung hình đầu tiên.
- ❖ Tính toán gradient tại mỗi điểm quan sát.
- ❖ Xác định ma trận A và vector b trong công thức tính toán optical flow.
- ❖ Giải phương trình $Ax = b$ để tìm vector chuyển động (u, v) tương ứng với mỗi điểm quan sát.



Hình 5.2. Luồng quang minh họa quỹ đạo chuyển động tính toán theo Lucas-Kanade

5.1.2.b. Phương pháp Horn – Schunck

Phương pháp Horn-Schunck là một phương pháp quan trọng và cổ điển để tính toán Optical Flow. Được đề xuất bởi Brian Horn và Brian Schunck vào năm 1981, phương pháp này đã đóng góp quan trọng vào lĩnh vực xử lý ảnh và thị giác máy tính.

Phương pháp Horn-Schunck giả định rằng Optical Flow trong toàn bộ ảnh là đồng nhất và mịn. Để tính toán Optical Flow, phương pháp này xây dựng một hàm mục tiêu dựa trên giả định về độ mịn và độ nhất quán của Optical Flow trong không gian ảnh và thời gian. Hàm mục tiêu này bao gồm hai thành phần chính: thành phần đạo hàm (derivative) và thành phần độ mịn (smoothness).

Thành phần đạo hàm trong hàm mục tiêu đo lường sự khác biệt giữa gradient của khung thứ nhất và khung thứ hai của chuỗi hình ảnh. Nó xác định sự biến thiên không gian của Optical Flow. Thành phần này được tính bằng cách sử dụng phép tích chập với gradient của các khung hình.

Thành phần độ mịn trong hàm mục tiêu xác định sự nhất quán của Optical Flow trong không gian ảnh và thời gian. Nó giả định rằng chuyển động của các điểm ảnh gần nhau trong không gian ảnh và thời gian thường tương tự nhau. Để đạt được tính nhất quán này, phương pháp Horn-Schunck sử dụng phương pháp giảm thiểu bình phương tổng của gradient của Optical Flow.

Quá trình ước lượng Optical Flow trong phương pháp Horn-Schunck là một quá trình tối ưu hóa. Bằng cách tối thiểu hóa hàm mục tiêu, phương pháp tìm kiếm Optical Flow tối ưu nhất. Một thuật toán tối ưu được sử dụng để tìm ra giá trị của Optical Flow mà làm giảm hàm mục tiêu xuống mức tối thiểu.

Công thức: $\nabla I(x, y) \nabla I(x, y)^T \nabla u \nabla v = -\nabla I(x, y) \nabla I(x, y)^T \nabla I(x, y)$

Phương pháp:

- ❖ Chọn các điểm quan sát ban đầu.
- ❖ Tính toán gradient của tất cả các khung hình.
- ❖ Khởi tạo giá trị ban đầu cho các thành phần chuyển động u và v .
- ❖ Lặp lại các bước sau đến khi hội tụ:
 - Tính toán gradient của u và v .
 - Cập nhật u và v bằng cách giải phương trình $\nabla I \nabla I^T \nabla u + \nabla I \nabla I^T \nabla v = -\nabla I \nabla I^T \nabla I$.

5.1.2.c. Phương pháp FarneBack

Phương pháp Farneback là một phương pháp phổ biến và hiệu quả để tính toán Optical Flow. Được đề xuất bởi Gunnar Farneback vào năm 2003, phương

pháp này dựa trên ý tưởng sử dụng thông tin về biến thiên không gian và thời gian để ước lượng chuyển động.

Phương pháp Farneback sử dụng ảnh xám thay vì ảnh màu để giảm khối lượng tính toán và tăng tốc độ. Đầu tiên, nó tính toán một biểu diễn không gian-tần số của các khung hình bằng cách áp dụng biến đổi Fourier cho mỗi khung hình. Quá trình này giúp xác định các biến thiên tần số trong không gian và thời gian.

Tiếp theo, phương pháp Farneback sử dụng phương pháp Polynomial Expansion để xấp xỉ các giá trị biến đổi trong không gian và thời gian. Điều này cho phép ước lượng các thông số chuyển động tại mỗi vùng trong ảnh. Sau đó, phương pháp sử dụng phép chia để tính toán Optical Flow dựa trên sự tương quan giữa các giá trị biến đổi của các vùng tương ứng trên hai khung hình.

Phương pháp Farneback cung cấp kết quả Optical Flow một cách nhanh chóng và ổn định, đặc biệt là trong các trường hợp chuyển động đơn giản hoặc chuyển động nhất quán trong ảnh. Nó cũng có khả năng xử lý các biến đổi ánh sáng mạnh và khả năng ước lượng các chuyển động lớn.

5.1.2.d. *Phương pháp Pyramidal*

Phương pháp Pyramidal là một phương pháp quan trọng trong tính toán Optical Flow. Ý tưởng chính của phương pháp Pyramidal là sử dụng các khung hình có độ phân giải khác nhau để ước lượng Optical Flow ở nhiều mức độ chi tiết khác nhau. Phương pháp này được áp dụng để giảm khối lượng tính toán và cải thiện độ chính xác của quá trình ước lượng chuyển động.

Quá trình tính toán Optical Flow theo phương pháp Pyramidal bắt đầu bằng việc tạo ra các pyramid từ khung hình ban đầu. Mỗi pyramid là một chuỗi các khung hình có độ phân giải khác nhau, thường được tạo ra bằng cách lấy mẫu (downsampling) hoặc làm mờ (blurring) khung hình gốc để giảm độ phân giải. Việc tạo ra các pyramid này giúp chúng ta có cái nhìn tổng quan về chuyển động trên các mức độ chi tiết khác nhau.

Sau khi có các pyramid, quá trình ước lượng Optical Flow được thực hiện từ trên xuống dưới qua các mức độ chi tiết thấp hơn. Đầu tiên, quá trình ước lượng được thực hiện trên khung hình có độ phân giải cao nhất, nơi chúng ta mong đợi tìm ra các đặc trưng chuyển động chi tiết. Sau đó, kết quả Optical Flow từ mức độ cao nhất được sử dụng làm đầu vào để ước lượng Optical Flow tại mức độ chi tiết thấp hơn. Quá trình này tiếp tục cho đến khi đạt được mức độ chi tiết thấp nhất trong pyramid.

Khi ước lượng Optical Flow trên mỗi mức độ, phương pháp Pyramidal thường sử dụng các phương pháp ước lượng Optical Flow khác, chẳng hạn như phương pháp Lucas-Kanade hay Horn-Schunck. Với sự hỗ trợ từ các pyramid,

các phương pháp này có thể tạo ra kết quả chính xác hơn và ổn định hơn so với việc áp dụng chúng trực tiếp lên khung hình gốc.

Một lợi ích quan trọng của phương pháp Pyramidal là nó giảm khối lượng tính toán. Bằng cách làm giảm độ phân giải của khung hình, chúng ta chỉ cần tính toán Optical Flow trên các pyramid có kích thước nhỏ hơn. Điều này giúp tiết kiệm thời gian và tài nguyên tính toán. Ngoài ra, việc sử dụng các pyramid cũng giúp phương pháp Pyramidal trở nên ổn định hơn đối với các biến đổi ánh sáng mạnh và chuyển động lớn trong ảnh.

Tuy phương pháp Pyramidal có những ưu điểm vượt trội, nhưng nó cũng có một số hạn chế. Việc tạo ra các pyramid và tính toán Optical Flow trên từng mức độ yêu cầu thêm thời gian và bộ nhớ. Do đó, trong các ứng dụng thời gian thực hoặc có yêu cầu tốc độ xử lý cao, việc sử dụng phương pháp Pyramidal có thể không phù hợp..

5.1.2.e. *Phương pháp Dense*

Phương pháp Dense (dày đặc) là một phương pháp tính toán Optical Flow cho mỗi điểm ảnh trong khung hình thay vì chỉ ước lượng cho một số điểm ảnh đại diện. Điều này tạo ra một Optical Flow field hoàn chỉnh, cho phép theo dõi chuyển động chi tiết của các điểm ảnh trong cảnh.

Trong phương pháp Dense, mỗi điểm ảnh trong khung hình đầu vào sẽ có một vector chuyển động tương ứng trong Optical Flow field. Để tính toán Optical Flow dense, các phương pháp thường sử dụng mô hình dự đoán hoặc tối ưu hóa.

Một trong những phương pháp phổ biến để tính toán Optical Flow dense là phương pháp Lucas-Kanade pyramidal. Đầu tiên, các khung hình đầu vào được giảm độ phân giải theo một loạt các mức độ khác nhau, gọi là pyramid levels. Việc giảm độ phân giải giúp tăng tốc độ tính toán và giảm ảnh hưởng của nhiễu.

Tiếp theo, phương pháp Lucas-Kanade được áp dụng cho từng pyramid level theo thứ tự từ cao đến thấp. Tại mỗi pyramid level, phương pháp Lucas-Kanade tính toán vector chuyển động cho từng điểm ảnh trong vùng xem xét xung quanh. Việc tính toán này dựa trên giả định về sự tuyến tính của chuyển động.

Các vector chuyển động tại các pyramid level thấp hơn sẽ được truyền lên pyramid level cao hơn để cải thiện độ chính xác và độ mịn của Optical Flow field. Quá trình truyền ngược này giúp khắc phục vấn đề về chuyển động lớn và cải thiện khả năng ước lượng chuyển động chi tiết.

Một ưu điểm của phương pháp Dense là nó cung cấp một bức tranh toàn cảnh về chuyển động trong cảnh, cho phép phân tích chi tiết và phát hiện các

vùng có chuyển động đặc biệt. Nó hữu ích trong các ứng dụng như theo dõi đối tượng, phát hiện sự thay đổi trong cảnh, hay tái tạo chuyển động trong hình ảnh.

Tuy nhiên, phương pháp Dense cũng có nhược điểm là yêu cầu khối lượng tính toán lớn, đặc biệt đối với ảnh có độ phân giải cao. Ngoài ra, phương pháp này có thể bị ảnh hưởng bởi nhiễu và các vấn đề về độ mịn của cảnh.

5.1.2.f. *Phương pháp Deep Learning*

Phương pháp Deep Learning đã có một sự đóng góp quan trọng cho tính toán Optical Flow trong thị giác máy tính. Deep Learning là một lĩnh vực trong Machine Learning dựa trên cấu trúc mạng neural sâu để học và hiểu các đặc trưng phức tạp từ dữ liệu.

Trong việc tính toán Optical Flow, các phương pháp dựa trên Deep Learning sử dụng mạng neural sâu để học cách ước lượng chuyển động từ dữ liệu huấn luyện. Thay vì dựa vào các phương trình giả định hoặc thuật toán tối ưu như các phương pháp truyền thống, Deep Learning cho phép mô hình tự động học các mẫu chuyển động từ các cặp ảnh huấn luyện.

Các mạng neural sâu được thiết kế để nhận đầu vào là cặp ảnh và tạo ra đầu ra là Optical Flow tương ứng. Kiến trúc mạng thường sử dụng Convolutional Neural Networks (CNN) hoặc Recurrent Neural Networks (RNN) để khám phá và học các đặc trưng cục bộ và không gian của các khung hình.

Quá trình huấn luyện mạng neural sâu trong tính toán Optical Flow thường yêu cầu một tập dữ liệu lớn gồm các cặp ảnh huấn luyện và Optical Flow ground truth. Trong quá trình huấn luyện, mô hình cố gắng điều chỉnh các trọng số mạng neural để giảm sai số giữa Optical Flow được ước lượng và Optical Flow thực tế. Quá trình này thường sử dụng các phương pháp tối ưu gradient descent và các hàm mất mát (loss function) như Mean Squared Error (MSE) để đo lường sai số.

Một số kiến trúc mạng neural phổ biến trong tính toán Optical Flow bao gồm FlowNet, PWC-Net (Pyramid Network), và SpyNet. Các mô hình này tận dụng lợi thế của mạng neural sâu trong việc học các đặc trưng hiệu quả từ dữ liệu hình ảnh và thực hiện ước lượng Optical Flow nhanh và chính xác.

Một trong những ưu điểm của phương pháp Deep Learning trong tính toán Optical Flow là khả năng xử lý các tình huống phức tạp như biến đổi ánh sáng mạnh, đối tượng che khuất, hoặc chuyển động nhanh. Đồng thời, phương pháp này có thể tự động học và tìm ra các đặc trưng chuyển động phức tạp mà các phương pháp truyền thống khó có thể đạt được.

Tuy nhiên, để huấn luyện mạng neural sâu cho tính toán Optical Flow, cần có khối lượng dữ liệu lớn và tài nguyên tính toán phức tạp. Quá trình huấn luyện có thể yêu cầu nhiều ngày hoặc thậm chí tuần để hoàn thành trên các GPU mạnh.

5.1.3. Ứng dụng của kỹ thuật Optical Flow

Theo dõi chuyển động và đo tốc độ: Một trong những ứng dụng quan trọng của Optical Flow là theo dõi chuyển động và đo tốc độ của các đối tượng trong video. Với việc ước lượng Optical Flow, chúng ta có thể xác định hướng và tốc độ di chuyển của các vật thể trong video. Điều này có thể được áp dụng trong việc giám sát an ninh, theo dõi giao thông, phân tích hành vi di chuyển, hoặc theo dõi vật thể trong các hệ thống tự động.

Điều khiển và theo dõi robot: Optical Flow cũng có thể được sử dụng trong điều khiển và theo dõi robot. Bằng cách theo dõi chuyển động của robot thông qua việc ước lượng Optical Flow, chúng ta có thể cải thiện khả năng định vị và điều khiển chính xác của robot trong môi trường động.

Tự động lái xe: Trong lĩnh vực xe tự động, Optical Flow có vai trò quan trọng trong việc phát hiện và theo dõi chuyển động của các đối tượng như xe, người đi bộ, hoặc đèn giao thông. Bằng cách sử dụng Optical Flow, hệ thống xe tự động có thể xác định tốc độ và hướng di chuyển của các đối tượng xung quanh, từ đó đưa ra quyết định phù hợp để đảm bảo an toàn giao thông.

Xử lý hình ảnh y tế: Trong lĩnh vực y học, Optical Flow được áp dụng để phân tích chuyển động trong hình ảnh y tế. Ví dụ, trong quá trình quan sát hình ảnh chụp cộng hưởng từ (MRI), Optical Flow có thể được sử dụng để theo dõi và đo lường chuyển động của các cơ quan và mô trong cơ thể, từ đó giúp trong việc chẩn đoán và theo dõi bệnh lý.

Xử lý video và truyền thông đa phương tiện: Optical Flow cũng có ứng dụng trong xử lý video và truyền thông đa phương tiện. Việc ước lượng Optical Flow giúp cho việc trích xuất các đặc trưng chuyển động, tạo hiệu ứng hình ảnh động, hoặc nén video. Nó cũng có thể được sử dụng trong phân tích và nén video để giảm kích thước dữ liệu và tăng hiệu suất truyền thông.

Thực tế ảo và trò chơi điện tử: Optical Flow cung cấp cơ sở cho việc phát triển các ứng dụng thực tế ảo và trò chơi điện tử. Bằng cách ước lượng và mô phỏng chuyển động của người dùng, Optical Flow cho phép tạo ra trải nghiệm thực tế và chân thực hơn trong các ứng dụng thực tế ảo và trò chơi.

Với khả năng ước lượng chuyển động chính xác và theo dõi đối tượng, Optical Flow đóng vai trò quan trọng trong việc hiểu và tận dụng thông tin từ chuỗi hình ảnh, mở ra cánh cửa cho nhiều ứng dụng sáng tạo và tiềm năng trong tương lai.

5.1.4. Các thách thức và hạn chế của kỹ thuật Optical Flow

Mặc dù kỹ thuật optical flow đã được sử dụng rộng rãi trong các ứng dụng thị giác máy tính, tuy nhiên nó vẫn đối mặt với nhiều thách thức và hạn chế. Sau đây là một số thách thức và hạn chế của kỹ thuật optical flow:

Độ chính xác: Một trong những thách thức lớn nhất của kỹ thuật optical flow là tính chính xác của kết quả. Khi các điểm trong ảnh chuyển động phức tạp hoặc mất mát thông tin trong quá trình tính toán, kết quả của phương pháp optical flow có thể không chính xác, dẫn đến sai lệch trong ứng dụng thực tế.

Nhiều: Nhiều là một vấn đề phổ biến trong quá trình tính toán optical flow. Nhiều có thể xuất hiện trong ảnh ban đầu hoặc được tạo ra trong quá trình tính toán, và có thể dẫn đến kết quả không chính xác.

Thay đổi ánh sáng: Khi ánh sáng trong môi trường thay đổi, các điểm trong ảnh có thể thay đổi mà không liên quan đến chuyển động. Điều này có thể làm giảm độ chính xác của phương pháp optical flow.

Chuyển động phức tạp: Khi chuyển động trong ảnh phức tạp, như chuyển động xoáy hoặc chuyển động vòng quanh, kỹ thuật optical flow có thể không hoạt động hiệu quả, dẫn đến kết quả không chính xác.

Tốc độ tính toán: Tính toán optical flow đòi hỏi nhiều tài nguyên tính toán, đặc biệt là khi tính toán trên ảnh độ phân giải cao hoặc trong thời gian thực. Do đó, tốc độ tính toán có thể là một hạn chế khi áp dụng kỹ thuật optical flow.

Không đồng nhất: Kỹ thuật optical flow giả định rằng chuyển động của các điểm trong ảnh đồng nhất. Tuy nhiên, khi chuyển động không đồng nhất, kỹ thuật optical flow có thể không hoạt động hiệu quả.

5.2. Thư viện OpenCV và các hàm hỗ trợ tính toán Optical Flow

OpenCV (Open Source Computer Vision Library) là một thư viện mã nguồn mở phổ biến trong lĩnh vực xử lý ảnh và thị giác máy tính. Nó cung cấp một bộ các công cụ và thuật toán tiên tiến để thực hiện các nhiệm vụ xử lý ảnh, nhận dạng đối tượng, theo dõi chuyển động và nhiều ứng dụng khác. OpenCV hỗ trợ nhiều ngôn ngữ lập trình như C++, Python và Java, và nó được sử dụng rộng rãi trong cả nghiên cứu và phát triển ứng dụng thực tế.

OpenCV cung cấp một loạt các tính năng và chức năng để làm việc với hình ảnh và video. Điều này bao gồm khả năng đọc và ghi các định dạng tệp hình ảnh và video khác nhau, xử lý hình ảnh, trích xuất đặc trưng, nhận dạng đối tượng, phân loại và nhận dạng khuôn mặt, và rất nhiều tính năng khác. Trong số đó, tính toán Optical Flow là một tính năng quan trọng và hữu ích trong lĩnh vực nhận dạng chuyển động, theo dõi đối tượng và xe tự động.

Trong OpenCV, có một số hàm quan trọng để tính toán Optical Flow. Dưới đây là một số hàm phổ biến để ước lượng Optical Flow trong OpenCV:

- ❖ ***calcOpticalFlowFarneback***: Hàm này tính toán Optical Flow sử dụng phương pháp Farneback. Nó nhận đầu vào là hai khung hình liên tiếp và tạo ra Optical Flow field tương ứng. Hàm này cho phép tùy chỉnh các tham số như độ phân giải, số lượng mức trong pyramid, và các thông số của phương pháp Farneback.
- ❖ ***calcOpticalFlowPyrLK***: Hàm này sử dụng phương pháp Lucas-Kanade để tính toán Optical Flow. Nó nhận đầu vào là hai khung hình liên tiếp và một tập hợp các điểm ảnh ban đầu, và tạo ra Optical Flow vector tương ứng cho mỗi điểm. Hàm này cũng cho phép tùy chỉnh các tham số như kích thước cửa sổ và số lượng mức pyramid.
- ❖ ***calcOpticalFlowSparseToDense***: Hàm này tính toán Optical Flow từ các điểm thưa thớt đến dạng mật độ. Nó nhận đầu vào là hai khung hình liên tiếp và một tập hợp các điểm ảnh ban đầu, và tạo ra Optical Flow field hoàn chỉnh. Hàm này cho phép tùy chỉnh các tham số như kích thước cửa sổ, số lượng mức pyramid và các thông số của phương pháp tính toán.
- ❖ ***calcOpticalFlowDualTVL1***: Hàm này tính toán Optical Flow sử dụng phương pháp Dual TV-L1. Nó nhận đầu vào là hai khung hình liên tiếp và tạo ra Optical Flow field tương ứng. Hàm này cung cấp các tham số để tùy chỉnh độ mịn và độ nhất quán của Optical Flow.

Các hàm trên đều cung cấp các tham số để điều chỉnh quá trình tính toán Optical Flow và đạt được độ chính xác và hiệu suất tốt trong nhiều tình huống. Kết quả của các hàm này thường là một Optical Flow field, biểu diễn thông tin về hướng và tốc độ chuyển động của các điểm ảnh trong cảnh.

OpenCV cũng cung cấp các công cụ và hàm hỗ trợ khác để xử lý và hiển thị Optical Flow, bao gồm việc vẽ các vector chuyển động, tạo ra hình ảnh màu biểu diễn Optical Flow, hoặc tính toán thống kê từ dữ liệu Optical Flow.

CHƯƠNG 6: XÂY DỰNG ỨNG DỤNG ĐO LƯỜNG GIAO THÔNG

6.1. Xác định yêu cầu của ứng dụng

Trước tiên, tiến hành định rõ các yêu cầu cơ bản mà ứng dụng phải đáp ứng. Yêu cầu này bao gồm những yếu tố quan trọng cần có trong ứng dụng để đảm bảo hoạt động đáp ứng được mục tiêu và đáng tin cậy.

Một trong những yêu cầu cơ bản là khả năng xác định các làn đường trong hình ảnh giao thông. Sử dụng phương pháp Edge detection và Hough transform để phát hiện các đường viền và xác định các làn đường hiện diện trên hình ảnh. Đồng thời, yêu cầu về tính chính xác trong việc phân loại làn đường cũng là một điểm quan trọng. Nhờ đó sẽ đánh giá hiệu suất của mô hình phân loại và điều chỉnh các tham số để đạt được kết quả tốt nhất.

Một yêu cầu quan trọng khác là khả năng xác định mật độ xe và tốc độ trung bình trong mỗi làn đường. Áp dụng kỹ thuật Optical Flow (Farneback hoặc Lucas-Kanade) để tính toán các vector flow, từ đó suy ra mật độ xe và tính toán tốc độ trung bình của xe trong mỗi làn đường. Điều này sẽ giúp đánh giá hiệu quả hoạt động giao thông trong từng làn đường và từ đó đưa ra phán đoán làn đó có đông xe hay không.

6.2. Thiết kế - triển khai logic và xác định các công nghệ sử dụng

Sau khi xác định các yêu cầu của ứng dụng, tiến hành thiết kế tổng quan và triển khai logic cũng như các công nghệ sử dụng trong ứng dụng. Điều này là quy trình quan trọng trong việc xây dựng ứng dụng hiệu quả và đáp ứng yêu cầu đã đề ra.

6.2.1. Thiết kế xác định làn đường

Trước tiên sử dụng phương pháp Edge detection để xác định các đường viền trong hình ảnh. Sau đó, chúng ta sẽ áp dụng Hough transform để xác định các đường thẳng phổ biến trên hình ảnh, từ đó tạo ra các đoạn thẳng đại diện cho các làn đường. Việc này sẽ giúp xác định các làn đường hiện diện và đánh dấu chúng trên hình ảnh giao thông.

6.2.2. Thiết kế mô hình Optical Flow

Từ các làn đường đã được xác định ở phần thiết kế xác định làn đường, tiến hành triển khai mô hình Optical Flow (có thể sử dụng Farneback hoặc Lucas-Kanade) để xác định mật độ xe và tốc độ trung bình của xe trong mỗi làn đường. Mô hình này sẽ tính toán các vector flow, từ đó có thể suy ra các thông tin về mật độ xe, lượng vector càng nhiều thì mật độ xe càng đông và tính toán tốc độ trung bình trong mỗi làn đường.

6.2.3. Xác định các công nghệ sử dụng

Ứng dụng sẽ được lập trình trên thiết bị, đồng nghĩa với việc mã nguồn sẽ được viết và phát triển trực tiếp trên thiết bị, cụ thể ở đây là camera AI giám sát giao thông. Tuy nhiên, với sự hạn chế về điều kiện cơ sở vật chất, ứng dụng này sẽ được mô phỏng lại trên thiết bị điện thoại có camera chạy android 10 trở lên.

IDE: Android Studio Electric Eel | 2022.1.1

Thư viện: OpenCV4

Kỹ thuật ứng dụng: Edge Detection, Hough Transform, Optical Flow

6.3. Các bước xây dựng ứng dụng

Sau khi hoàn thành thiết kế và triển khai các công nghệ, chúng ta sẽ tiến hành xây dựng ứng dụng phân luồng giao thông. Các bước xây dựng ứng dụng sẽ bao gồm:

6.3.1. Nhận dữ liệu đầu vào

Ứng dụng sẽ nhận dữ liệu từ camera của thiết bị. Để thực hiện điều này, sử dụng lớp CameraBridgeViewBase của org.opencv.android. Trong phần giao diện (*activity_main.xml*) gọi đến lớp *org.opencv.android.JavaCameraView* để hiển thị hình ảnh nhận được từ camera lên màn hình.

```
<org.opencv.android.JavaCameraView
    android:id="@+id/camera_view"
    android:layout_width="match_parent"
    android:layout_height="250dp"
    android:layout_marginHorizontal="8dp"
    android:layout_marginVertical="8dp" />
```

Đọc từng frame bằng cách gọi hàm setCvCameraViewListener và override lại phương thức onCameraFrame để lưu từng frame bằng lớp Mat của thư viện *org.opencv.core.Mat*, tùy chỉnh chế độ lọc cho từng frame, ở đây sẽ chuyển sang chế độ xám để thực hiện xử lý Edge Detection và xử lý kỹ thuật Optical Flow còn chế độ màu để hiển thị lên giao diện.

```
cameraBridgeViewBase.setCvCameraViewListener(new
CameraBridgeViewBase.CvCameraViewListener2() {
    @Override
    public void onCameraViewStarted(int width, int height) {}
    @Override
    public void onCameraViewStopped() {}
    @Override
    public Mat onCameraFrame(CameraBridgeViewBase.CvCameraViewFrame inputFrame) {
        Mat gray = inputFrame.gray();
        Mat rgba = inputFrame.rgba();
        return rgba;
    }
});
```

Để sử dụng được camera của thiết bị thì cần được cấp quyền, trước khai báo quyền sử dụng trong file ***AndroidManifest.xml***.

```
<uses-permission android:name="android.permission.CAMERA" />
```

Sau đó kiểm tra quyền đã được cấp hay chưa, nếu chưa được cấp thì yêu cầu lại lần nữa.

```
void getPermission() {
    // Check for camera permission at runtime (for Android 6.0 and higher)
    if (ContextCompat.checkSelfPermission(this,
        android.Manifest.permission.CAMERA)
        != PackageManager.PERMISSION_GRANTED) {
        ActivityCompat.requestPermissions(this,
            new String[]{Manifest.permission.CAMERA},
            CAMERA_REQUEST_CODE);
    } else {
        // Permission already granted, start camera
        cameraBridgeViewBase.enableView();
    }
}

@Override
public void onRequestPermissionsResult(int requestCode, @NonNull String[]
permissions, @NonNull int[] grantResults) {
    if (requestCode == CAMERA_REQUEST_CODE) {
        if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
            // Camera permission granted, start camera
            cameraBridgeViewBase.enableView();
        } else {
            // Camera permission denied, handle this case (e.g., show an
            explanation to the user)
            getPermission();
        }
    }
}
```

Thư viện CameraBridgeViewBase của OpenCV cần được chỉnh sửa chút, bởi vì khi hiển thị thì khung hình bị lệch tọa độ, thay code của hàm deliverAndFrame như sau:

```
protected void deliverAndDrawFrame(CvCameraViewFrame frame) {
    Mat modified;

    if (mListener != null) {
        modified = mListener.onCameraFrame(frame);
    } else {
        modified = frame.rgba();
    }
}
```

```

        boolean bmpValid = true;
        if (modified != null) {
            try {
                Utils.matToBitmap(modified, mCacheBitmap);
            } catch (Exception e) {
                Log.e(TAG, "Mat type: " + modified);
                Log.e(TAG, "Bitmap type: " + mCacheBitmap.getWidth() + "*" +
mCacheBitmap.getHeight());
                Log.e(TAG, "Utils.matToBitmap() throws an exception: " +
e.getMessage());
                bmpValid = false;
            }
        }

        if (bmpValid && mCacheBitmap != null) {
            Canvas canvas = getHolder().lockCanvas();
            if (canvas != null) {
                canvas.drawColor(0, android.graphics.PorterDuff.Mode.CLEAR);
                if (BuildConfig.DEBUG) {
                    Log.d(TAG, "mStretch value: " + mScale);
                }

                // Start of the fix
                Matrix matrix = new Matrix();
                matrix.preTranslate( ( canvas.getWidth() - mCacheBitmap.getWidth() )
/ 2f, ( canvas.getHeight() - mCacheBitmap.getHeight() ) / 2f );
                matrix.postRotate( 90f, ( canvas.getWidth() ) / 2f,
canvas.getHeight() / 2f );
                float scale = (float) canvas.getWidth() / (float)
mCacheBitmap.getHeight();
                matrix.postScale(scale, scale, canvas.getWidth() / 2f ,
canvas.getHeight() / 2f );
                canvas.drawBitmap( mCacheBitmap, matrix, null );

                // Back to original OpenCV code
                if (mFpsMeter != null) {
                    mFpsMeter.measure();
                    mFpsMeter.draw(canvas, 20, 30);
                }

                getHolder().unlockCanvasAndPost(canvas);
            }
        }
    }
}

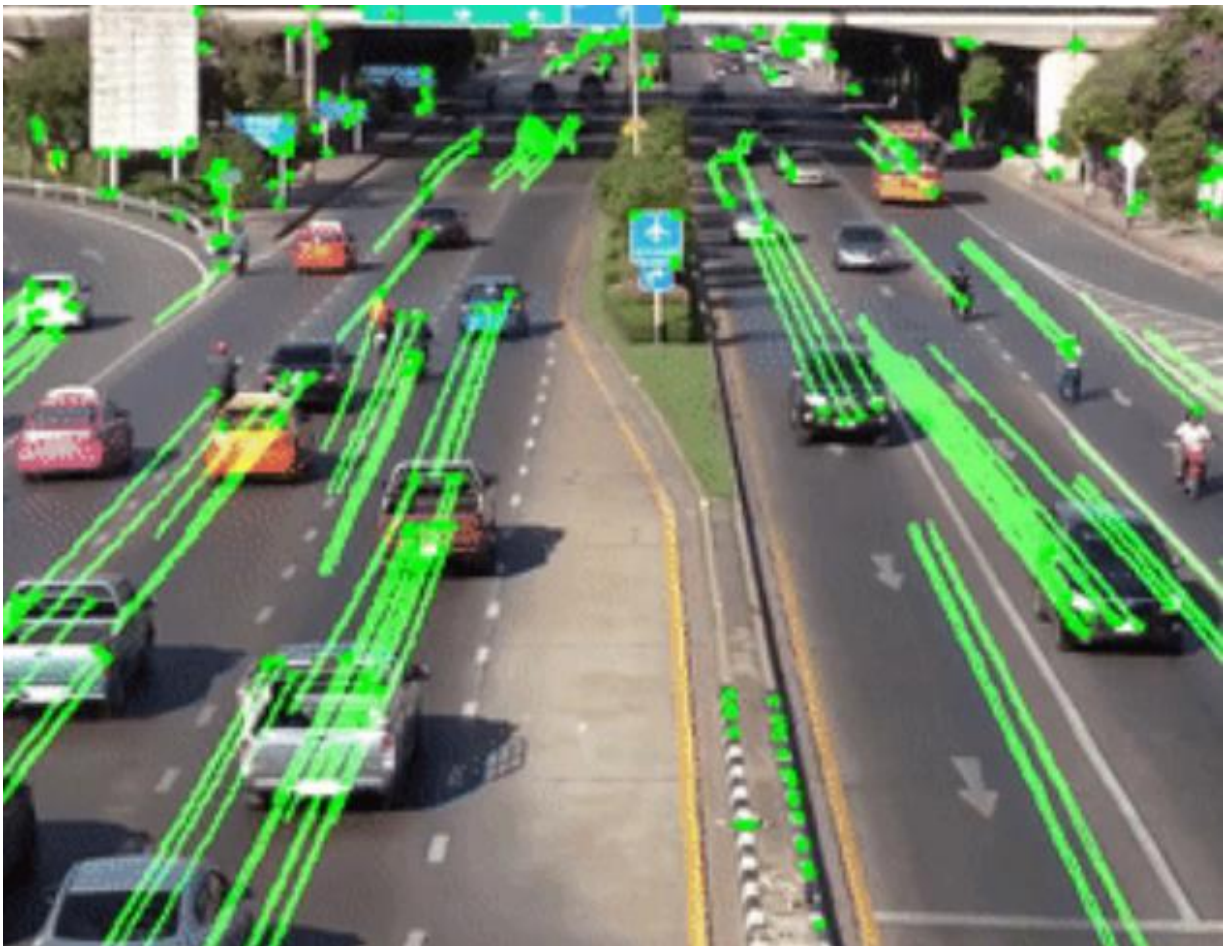
```


6.3.2. Xử lý hình ảnh và xác định làn đường

Đầu tiên tạo lớp TrafficDensityMeasurement để chứa các thông tin xử lý kỹ thuật Optical Flow, bao gồm frame trước, frame sau và vector flow. Trong lớp này tạo các phương thức *calOpticalFlowVector* để tính vector flow theo Farneback.

```
public Mat calOpticalFlowVector() {
    // Calculate optical flow using Farneback method
    Video.calcOpticalFlowFarneback(prevFrame, nextFrame, vectorFlow, PYR_SCALE,
    LEVELS, WIN_SIZE, ITERATIONS, POLYGON_EDGE, POLY_SIGMA, FLAGS);
    return vectorFlow;
}
```

Từ vector flow tính được, xác định làn đường bằng cách xác định hướng của vector, đối với đề tài, sẽ thực hiện xác định 2 hướng là hướng lên và hướng xuống. Đối với hướng lên sẽ là làn đường bên trái, còn hướng xuống sẽ là làn bên phải.



Hình 6.1. Minh họa cho 2 hướng để xác định làn đường.

6.3.3. Xác định mật độ xe và tốc độ trung bình

Ứng dụng sẽ áp dụng kỹ thuật Optical Flow để tính toán mật độ xe và tốc độ trung bình của xe trong mỗi làn đường.

Từ vector tính được ở phương thức trên, tính toán ra góc của vector, nếu vector thuộc nửa trên của vòng tròn lượng giác thì vector được xếp vào loại mũi tên

hướng lên (thuộc làn đường bên trái), còn nếu thuộc nửa dưới của vòng tròn lượng giác thì vector được xếp vào loại mũi tên hướng xuống (thuộc làn đường bên phải).

Đối với vận tốc trung bình (velocity) thì tính toán bằng tính độ lớn của các vector đã được phân loại, nếu vector có độ lớn hơn 10 thì nhận, sau đó chia cho số frame đã quét được.

Độ lớn của vector được tính bằng công thức của tiên đề Euclid:

$$\sqrt{u^2 + v^2}$$

```
public List<Double> calDensityAndVelocity(int frameCount) {
    // Calculate vector flow statistics
    List<Double> densityAndVelocity = new ArrayList<>();
    double firstDensity = 0;
    double secondDensity = 0;
    double firstMagnitudeAVG = 0;
    double secondMagnitudeAVG = 0;

    for (int y = 0; y < vectorFlow.rows(); y++) {
        for (int x = 0; x < vectorFlow.cols(); x++) {
            double[] flowVector = vectorFlow.get(y, x);

            double angle = Math.atan2(flowVector[0], flowVector[1]);
            if (angle < 0) {
                angle += 2 * Math.PI;
            }

            // Calculate vector value
            double magnitude = Math.sqrt(flowVector[0] * flowVector[0] +
            flowVector[1] * flowVector[1]);

            // Classify vector direction and average magnitude
            if (magnitude >= 0) {
                if (angle >= 0.25 * Math.PI && angle < 0.75 * Math.PI) {
                    firstDensity++;
                    firstMagnitudeAVG += magnitude;
                } else if (angle >= 1.25 * Math.PI && angle < 1.75 * Math.PI) {
                    secondDensity++;
                    secondMagnitudeAVG += magnitude;
                }
            }
        }
    }

    long totalLanePixels;
    if (firstDensity == 0 || secondDensity == 0) {
        totalLanePixels = vectorFlow.total();
        firstMagnitudeAVG /= firstDensity == 0 ? firstDensity : secondDensity;
    }
}
```

```

        firstDensity /= totalLanePixels;
        // Lưu kết quả
        densityAndVelocity.add(1.0);
        densityAndVelocity.add(firstDensity * 100);
        densityAndVelocity.add(firstMagnitudeAVG);
    } else {
        totalLanePixels = vectorFlow.total() / 2;
        firstMagnitudeAVG /= firstDensity;
        secondMagnitudeAVG /= secondDensity;
        firstDensity /= totalLanePixels;
        secondDensity /= totalLanePixels;
        // Lưu kết quả
        densityAndVelocity.add(2.0);
        densityAndVelocity.add(firstDensity * 100);
        densityAndVelocity.add(secondDensity * 100);
        densityAndVelocity.add(firstMagnitudeAVG);
        densityAndVelocity.add(secondMagnitudeAVG);
    }

    return densityAndVelocity;
}

```

Hàm trên sẽ tính cho những đoạn đường có 1 làn hoặc 2 làn. Đối với 2 làn trở lên thì sẽ mở rộng dần theo vòng tròn lượng giác.

6.3.4. Phân loại làn đường

Dựa trên thông tin mật độ xe và tốc độ trung bình, ứng dụng sẽ phân loại làn đường có đông xe hay không. Nếu mật độ cao và tốc độ trung bình thấp thì làn đường đang kẹt và ngược lại.

```

private boolean isDense(double density, double velocity) {
    if (density >= DENSITY_THRESHOLD) {
        return !(velocity >= VELOCITY_THRESHOLD);
    } else {
        return velocity >= VELOCITY_THRESHOLD;
    }
}

```

```

if (isDense(densityAndVelocity.get(0), densityAndVelocity.get(2))) {
    upLaneResult.setText("This lane is dense");
} else {
    upLaneResult.setText("This lane is not dense");
}

if (isDense(densityAndVelocity.get(1), densityAndVelocity.get(3))) {
    downLaneResult.setText("This lane is dense");
}

```

```
} else {
    downLaneResult.setText("This lane is not dense");
}
```

6.3.5. Vẽ biểu đồ theo dõi

Khởi tạo biểu đồ và cài đặt các thông số thuộc tính cho nó. Biểu đồ ở đây sẽ là biểu đồ kết hợp, dạng đường dành cho density và dạng line cho velocity, cập nhật mỗi 20 frame một lần.

```
private void setupCombinedChart(CombinedChart combinedChart) {
    combinedChart.getDescription().setEnabled(false);
    combinedChart.setBackgroundColor(Color.WHITE);
    combinedChart.setDrawGridBackground(false);
    combinedChart.setDrawBarShadow(false);
    combinedChart.setHighlightFullBarEnabled(false);

    // Enable horizontal scrolling
    combinedChart.setScaleEnabled(true);
    combinedChart.setDragEnabled(true);

    // Set the visible range of data
    combinedChart.setVisibleXRangeMinimum(100);
    combinedChart.setVisibleXRangeMaximum(400);

    XAxis xAxis = combinedChart.getXAxis();
    xAxis.setPosition(XAxis.XAxisPosition.BOTTOM);
    xAxis.setGranularity(20f);

    combinedChart.getAxisRight().setEnabled(false);
}
```

Truyền dữ liệu vào để vẽ biểu đồ, cài đặt style cho biểu đồ.

```
private void drawChart(List<BarEntry> densityData, List<Entry> velocityData,
    CombinedChart combinedChart) {
    BarDataSet barDataSet = new BarDataSet(densityData, "Density");
    barDataSet.setColor(Color.BLUE);

    LineDataSet lineDataSet = new LineDataSet(velocityData, "Velocity");
    lineDataSet.setColor(Color.RED);
    lineDataSet.setLineWidth(10f);
    lineDataSet.setCircleColor(Color.RED);
    lineDataSet.setCircleRadius(4f);

    BarData barData = new BarData(barDataSet);
    LineData lineData = new LineData(lineDataSet);

    CombinedData combinedData = new CombinedData();
}
```

```
combinedData.setData(barData);
combinedData.setData(lineData);

combinedChart.setData(combinedData);
combinedChart.invalidate();
}
```

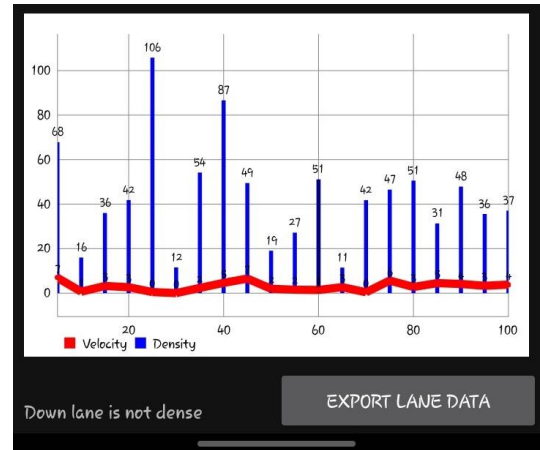
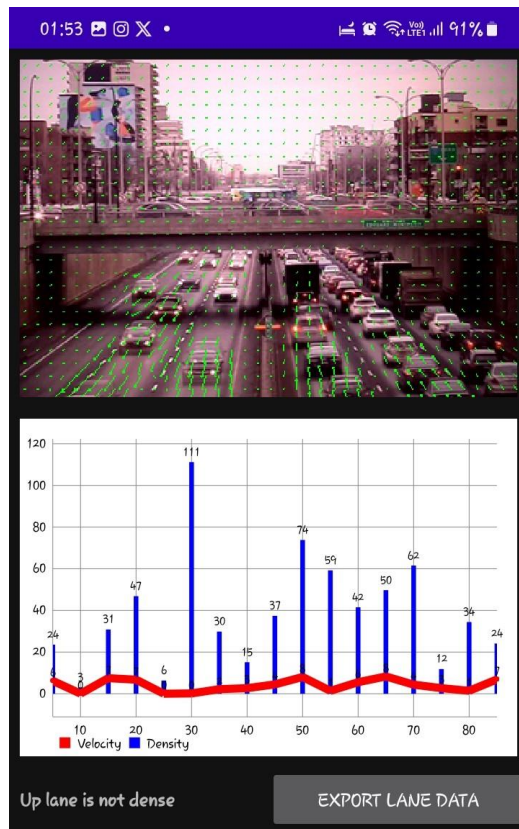
6.3.6. *Hiển thị kết quả*

Kết quả xử lý sẽ được hiển thị trực quan trên giao diện người dùng, giúp người dùng dễ dàng theo dõi tình trạng giao thông trên từng làn đường.

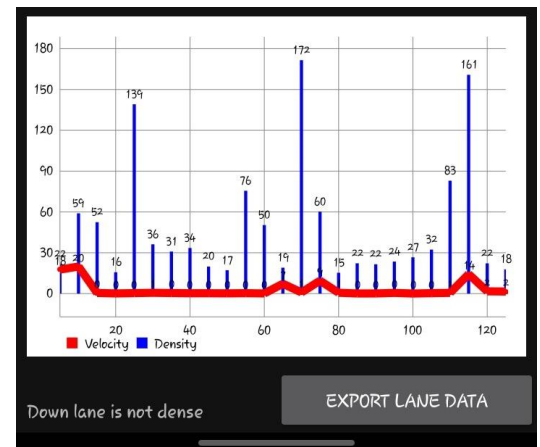
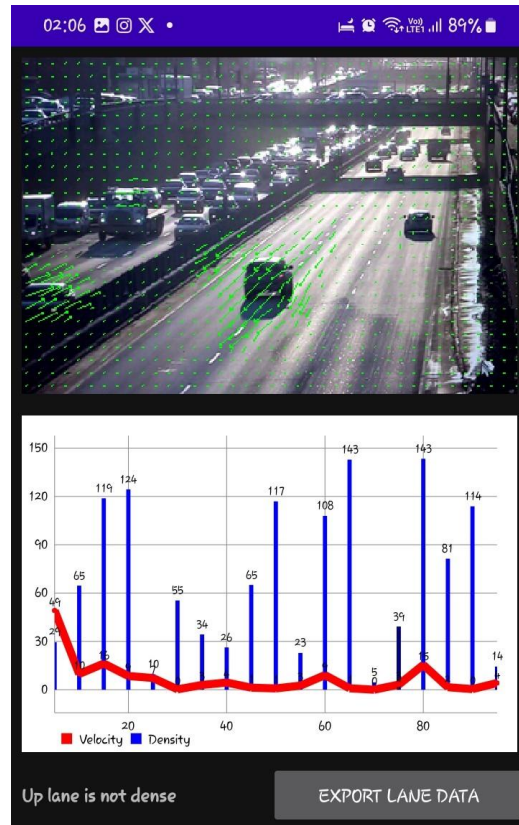
Bởi vì mục đích là triển khai ứng dụng trên thiết bị camera, mà thiết bị camera thì không có màn hình để hiển thị nên trong thực tế có thể tạo web page để truy cập đến thông tin thông qua địa chỉ IP của thiết bị. Tuy nhiên, đối với phạm vi của đề tài, sẽ cho hiển thị lên trong ứng dụng luôn.



Hình 6.2. Hiển thị kết quả của ứng dụng đối với mẫu thử 1



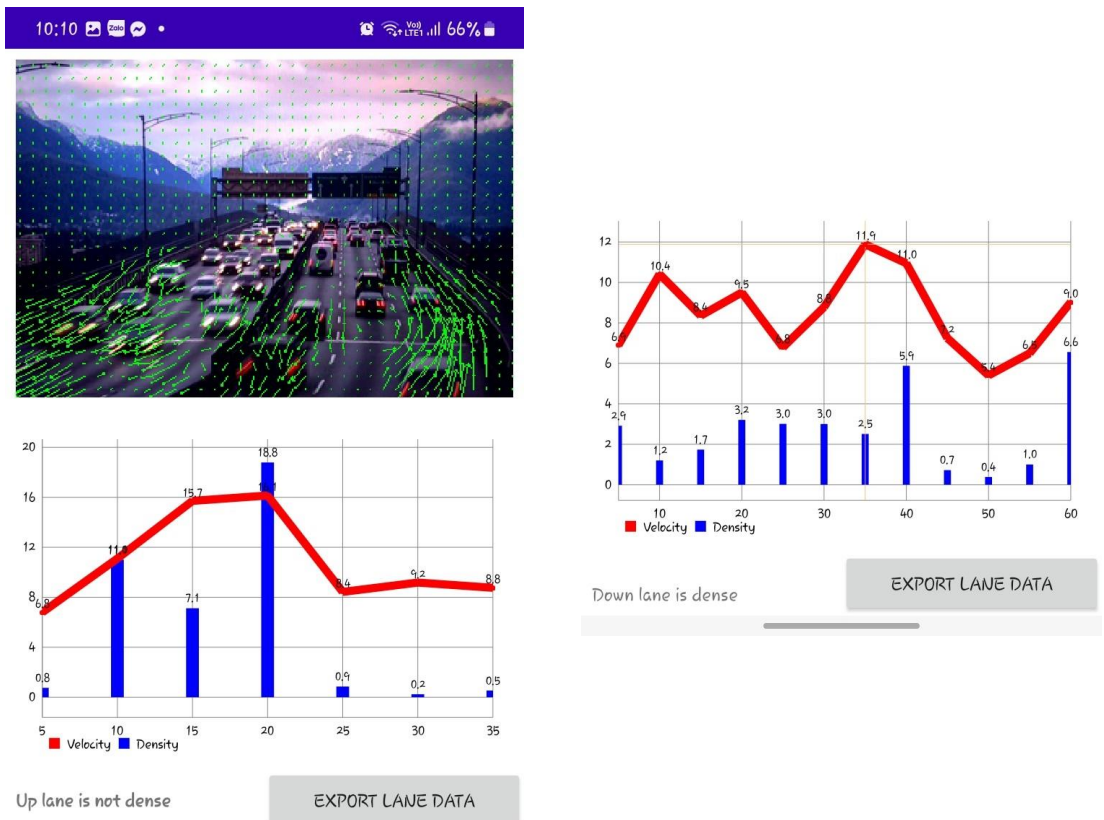
Hình 6.3. Hiển thị kết quả của ứng dụng đối với mẫu thử 2



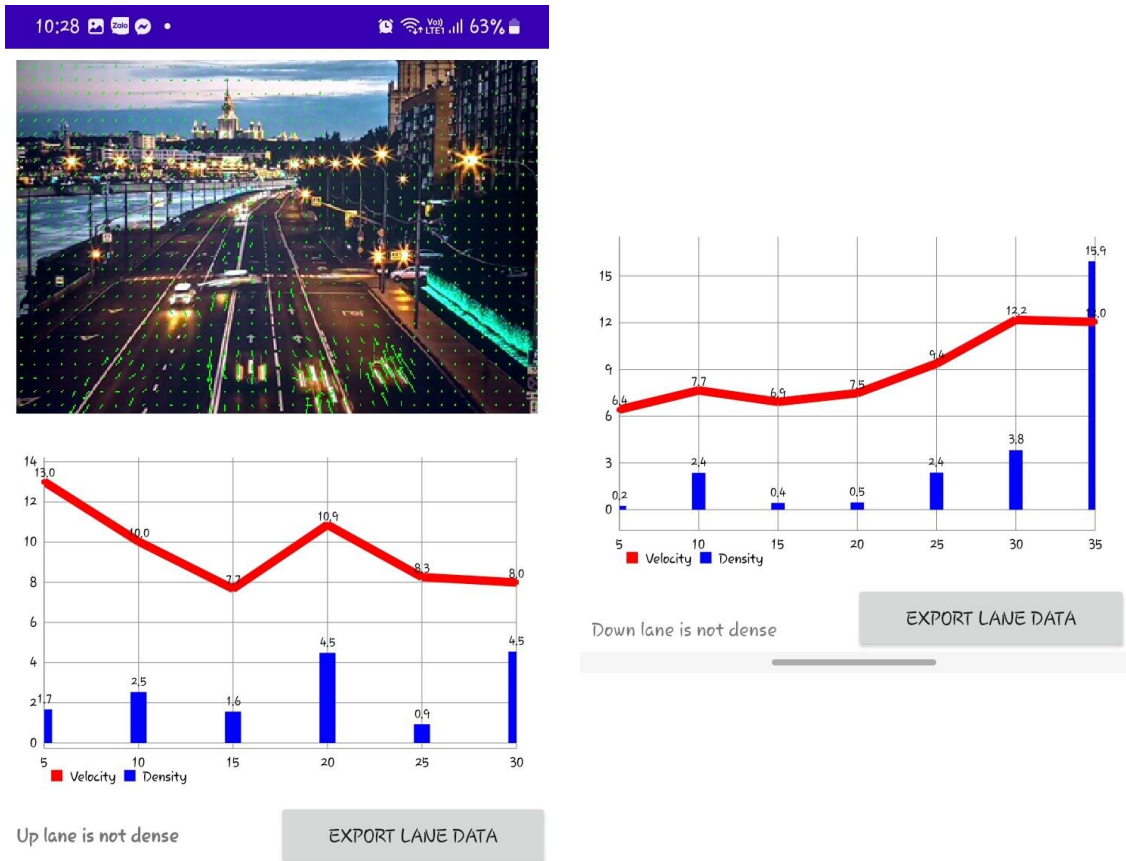
Hình 6.4. Hiển thị kết quả của ứng dụng đối với mẫu thử 3



Hình 6.5. Hiển thị kết quả của ứng dụng đối với mẫu thử 3



Hình 6.6. Hiển thị kết quả của ứng dụng đối với mẫu thử 4



Hình 6.7. Hiển thị kết quả của ứng dụng với mẫu thử 5

6.4. Kiểm thử và xác định các hạn chế của ứng dụng

Sau khi đã xây dựng ứng dụng, tiến hành đánh giá hiệu quả và xác định các hạn chế của nó. Điều này rất quan trọng để hiểu rõ tầm quan trọng của ứng dụng và các hạn chế cần được giải quyết để nâng cao hiệu năng và khả năng sử dụng.

6.4.1. Kiểm thử hiệu suất của mô hình Optical Flow

Chạy kiểm thử với 5 mẫu video khác nhau và đánh giá bằng mắt thường. Vẽ vector flow lên hình ảnh để xác nhận độ chính xác của mô hình Optical Flow.



Hình 6.7. Biểu diễn vector flow trong ứng dụng

6.4.2. Kiểm thử độ chính xác trong việc phân loại làn đường

Bởi vì ngay từ đầu đã xác định chỉ có 2 làn đường là hướng lên và hướng xuống, nên kiểm thử bằng các xác định nếu bắt được cả 2 loại vector thì coi như chính xác.

6.4.3. Kiểm thử về tốc độ xử lý

- ❖ Ứng dụng không có dấu hiệu giật, lag.
- ❖ Đọc dữ liệu nhận được nhanh chóng hiển thị cùng lúc với dữ liệu đầu vào
- ❖ Các vector hiển thị trong video liên tục
- ❖ Không có độ trễ.

6.4.4. Xác định các hạn chế về môi trường ánh sáng và thời tiết

- ❖ Hạn chế về ánh sáng yếu

Trong điều kiện ánh sáng yếu hoặc thiếu ánh sáng, việc thu thập và xử lý hình ảnh có thể gặp khó khăn, làm giảm độ chính xác của kỹ thuật Optical Flow. Kết quả xác định làn đường và mật độ xe có thể không đáng tin cậy.

Giải pháp: Sử dụng các camera có khả năng chống nhiễu và hỗ trợ chế độ chụp hình trong điều kiện ánh sáng yếu (chẳng hạn như các camera có chế

độ Low Light hoặc sử dụng công nghệ IR - hồng ngoại). Ngoài ra, có thể cân nhắc sử dụng các nguồn ánh sáng bổ sung để cải thiện điều kiện ánh sáng trong khu vực giám sát.

❖ Hạn chế về ánh sáng mạnh và chói

Ánh sáng mạnh hoặc chói có thể gây hiện tượng lóa, làm mất thông tin quan trọng trong hình ảnh. Điều này có thể làm giảm tính ổn định và độ chính xác của kỹ thuật Optical Flow.

Giải pháp: Có thể sử dụng các bộ lọc hoặc kỹ thuật giảm thiểu hiện tượng lóa để làm giảm tác động của ánh sáng mạnh và chói lên hình ảnh thu được. Ngoài ra, cần xem xét vị trí đặt camera sao cho tránh ánh sáng trực tiếp và hạn chế tối đa hiện tượng lóa.

❖ Hạn chế về thời tiết

Trong điều kiện thời tiết xấu như mưa, tuyết, sương mù hay bụi bẩn, hình ảnh thu được có thể bị mờ hoặc che mờ, làm giảm độ chính xác của kỹ thuật Optical Flow.

Giải pháp: Sử dụng các camera có khả năng chống thấm nước, chống bụi và hỗ trợ công nghệ loại bỏ sương mù để giảm thiểu tác động của thời tiết xấu. Cần có các biện pháp bảo vệ và làm sạch camera đều đặn để đảm bảo hình ảnh thu được luôn rõ nét.

❖ Hạn chế về tốc độ khung hình (frame rate)

Trong điều kiện ánh sáng yếu hoặc thời tiết xấu, camera có thể giảm tốc độ khung hình để tăng độ sáng hoặc giảm nhiễu. Điều này làm giảm hiệu suất của kỹ thuật Optical Flow và ảnh hưởng đến xác định làn đường và mật độ xe.

Giải pháp: Cần chọn camera có khả năng hoạt động ổn định và giữ nguyên tốc độ khung hình trong điều kiện ánh sáng yếu hoặc thời tiết xấu. Đồng thời, có thể cân nhắc sử dụng các thuật toán tối ưu hóa để xử lý hình ảnh với tốc độ khung hình thấp mà vẫn đảm bảo tính chính xác của kết quả.

❖ Hạn chế về sự che khuất

Trong môi trường giao thông, có thể xảy ra hiện tượng các vật thể (như xe, cây cối) che khuất một phần của hình ảnh, làm giảm khả năng xác định chính xác các làn đường và mật độ xe.

Giải pháp: Nên cân nhắc vị trí đặt camera sao cho tránh các vật thể có khả năng che khuất quan trọng trong khung hình. Ngoài ra, có thể sử dụng các kỹ thuật xử lý hình ảnh như mờ (blurring) hay loại bỏ các vật thể không cần

thiết khác để giảm tác động của sự che khuất lên kết quả xác định làn đường và mật độ xe.

6.5. Các hướng phát triển tiềm năng

Các hướng phát triển này có thể liên quan đến việc sử dụng các mô hình học sâu phức tạp hơn, kết hợp nhiều kỹ thuật xử lý hình ảnh để cải thiện độ chính xác, hoặc phát triển ứng dụng trên các nền tảng công nghệ mới.

Những hướng phát triển tiềm năng này sẽ giúp ứng dụng ngày càng hoàn thiện và hữu ích hơn trong việc giải quyết vấn đề ùn tắc giao thông và nâng cao chất lượng cuộc sống cho người dân. Việc xây dựng ứng dụng phân luồng giao thông dựa trên đo lường mật độ xe và tốc độ trung bình là một bước tiến quan trọng trong công nghệ giao thông thông minh và có tiềm năng ứng dụng rộng rãi trong thực tế.

KẾT LUẬN

Trong Chương 1, em đã giới thiệu về Đơn vị thực tập Viettel High Tech và những lĩnh vực mà đơn vị đang tham gia nghiên cứu, triển khai và kinh doanh. Việc thực tập tại đơn vị này đã cung cấp cho em cơ hội tiếp xúc với công nghệ tiên tiến và kinh nghiệm thực tế trong việc phát triển ứng dụng Android trên camera giám sát.

Trong Chương 2, em đã tìm hiểu quy trình phát triển ứng dụng Android, nhất là trong bối cảnh doanh nghiệp tại Việt Nam nói chung và Viettel High Tech nói riêng. Việc làm rõ quy trình này đã giúp em hiểu rõ hơn về các bước cần thực hiện trong việc phát triển một ứng dụng Android chất lượng và ổn định.

Chương 3 đã trình bày về lĩnh vực an ninh giám sát và vai trò của camera giám sát trong việc giám sát an ninh và giảm thiểu các rủi ro. Ngoài ra, chuẩn Onvif trong camera giám sát cũng được giới thiệu, giúp hiểu rõ hơn về cách tích hợp và tương tác với các thiết bị camera giám sát.

Trong Chương 4, em đã tìm hiểu về các ứng dụng được tích hợp trên camera giám sát thông thường và camera giám sát giao thông. Những tính năng trong camera đã giúp cải thiện hiệu quả của việc giám sát và đóng vai trò quan trọng trong việc đảm bảo an toàn giao thông.

Chương 5 tập trung vào kỹ thuật Optical Flow và thư viện OpenCV. Điều này đã cung cấp cho em những kiến thức cần thiết để xây dựng ứng dụng phân luồng giao thông, cho phép phân tích tình hình của các luồng giao thông và đưa ra phán đoán hợp lý trong việc điều khiển đèn tín hiệu giao thông.

Cuối cùng, trong Chương 6, em đã xây dựng thành công một ứng dụng Android nhằm giải quyết vấn đề phân luồng giao thông thông qua việc xử lý video từ camera giám sát. Ứng dụng này có tiềm năng ứng dụng rộng rãi trong việc cải thiện tình hình giao thông, giảm thiểu ùn tắc và tai nạn giao thông.

Tuy nhiên, việc thực hiện đề tài cũng đã đối diện với một số thách thức, như khả năng tích hợp và tương thích giữa các hệ thống camera giám sát khác nhau, độ chính xác của việc phân tích dữ liệu giao thông, và tối ưu hóa hiệu suất ứng dụng.

Để phát triển ứng dụng này và ứng dụng các công nghệ trong lĩnh vực camera giám sát, em có thể tập trung vào nâng cao khả năng tích hợp và tương tác với nhiều hệ thống camera khác nhau. Cũng như tăng cường sử dụng các phương pháp học máy và trí tuệ nhân tạo để cải thiện độ chính xác và hiệu suất của ứng dụng. Ngoài ra, cần đảm bảo việc bảo mật thông tin trong quá trình giám sát và truyền tải dữ liệu.

Trong tương lai, đề tài này có thể được mở rộng để áp dụng cho nhiều lĩnh vực khác nhau, không chỉ giám sát giao thông mà còn trong các ngành công nghiệp, an ninh, y tế và nhiều ứng dụng khác. Điều này sẽ tạo ra nhiều cơ hội mới trong việc

ứng dụng công nghệ camera giám sát vào thực tiễn và góp phần nâng cao chất lượng cuộc sống và hiệu quả công việc.

Tổng kết lại, đề tài đã đạt được những kết quả đáng kể trong việc phát triển ứng dụng Android trên camera giám sát và đã làm rõ những tiềm năng và hạn chế của đề tài. Hy vọng rằng những nỗ lực và kiến thức trong đề tài này sẽ góp phần vào sự phát triển và ứng dụng của công nghệ camera giám sát trong tương lai.

TÀI LIỆU THAM KHẢO

- Nagar, T. (2023, July 13). Mobile App Development Process (2023). *Dev Technosys*.
<https://devtechnosys.com/insights/mobile-app-development-process/>
- Profiles, Add-ons and Specifications - ONVIF*. (2023, July 18). ONVIF.
<https://www.onvif.org/profiles-add-ons-specifications/>
- Lin, C. (2023). Introduction to Motion Estimation with Optical Flow. *Nanonets AI & Machine Learning Blog*. <https://nanonets.com/blog/optical-flow/>
- OpenCV: Optical Flow*. (n.d.). https://docs.opencv.org/3.4/d4/dee/tutorial_optical_flow.html
- Ho, H. W., De Croon, G., & Chu, Q. (2017). Distance and velocity estimation using optical flow from a monocular camera. *International Journal of Micro Air Vehicles*, 9(3), 198–208. <https://doi.org/10.1177/1756829317695566>
- Estimate object velocities - Simulink*. (n.d.).
<https://www.mathworks.com/help/vision/ref/opticalflow.html>
- Compute Optical Flow Velocities - MATLAB & Simulink*. (n.d.).
<https://www.mathworks.com/help/vision/ug/compute-optical-flow-velocities.html>