

**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI**  
**PHÂN HIỆU TẠI TP. HỒ CHÍ MINH**  
**BỘ MÔN CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO ĐỒ ÁN TỐT NGHIỆP**

**ĐỀ TÀI: XÂY DỰNG ỨNG DỤNG TRÊN THIẾT BỊ DI  
ĐỘNG HỖ TRỢ NGƯỜI MÙ MÀU**

Giảng viên hướng dẫn: ThS. TRẦN THỊ DUNG

Sinh viên thực hiện: NGUYỄN ANH TUẤN

Lớp : CÔNG NGHỆ THÔNG TIN

Khoá : 59

Tp. Hồ Chí Minh, năm 2022

**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI**  
**PHÂN HIỆU TẠI TP. HỒ CHÍ MINH**  
**BỘ MÔN CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO ĐỒ ÁN TỐT NGHIỆP**

**ĐỀ TÀI: XÂY DỰNG ỨNG DỤNG TRÊN THIẾT BỊ DI  
ĐỘNG HỖ TRỢ NGƯỜI MÙ MÀU**

Giảng viên hướng dẫn: ThS. TRẦN THỊ DUNG

Sinh viên thực hiện: NGUYỄN ANH TUẤN

Lớp : CÔNG NGHỆ THÔNG TIN

Khoá : 59

Tp. Hồ Chí Minh, năm 2022

**NHIỆM VỤ THIẾT KẾ TỐT NGHIỆP**  
**BỘ MÔN: CÔNG NGHỆ THÔNG TIN**  
-----\*\*\*-----

**Mã sinh viên:** 5951071117

**Họ tên SV:** NGUYỄN ANH TUẤN

**Khóa:** 59

**Lớp:** Công nghệ thông tin

**1. Tên đề tài**

XÂY DỰNG ỨNG DỤNG TRÊN THIẾT BỊ DI ĐỘNG HỖ TRỢ NGƯỜI MÙ MÀU

**2. Mục đích, yêu cầu**

**a. Mục đích :**

ColorBlind được tạo ra với mục đích đánh giá, kiểm tra cũng như hỗ trợ người mù màu. Thị giác là một trong năm giác quan quan trọng của con người, bệnh mù màu sẽ ảnh hưởng nghiêm trọng đến khả năng nhìn, khả năng nghiên cứu, học tập và làm việc của mỗi người. Tiêu chuẩn vàng hiện tại ColorBlind là đánh giá, kiểm tra, cũng như hỗ trợ cho mắt, nhưng đôi khi kết quả đánh giá sẽ là khá khách quan, không chỉ phụ thuộc vào dữ liệu Ishihara để đánh giá, mà còn phụ thuộc các yếu tố phụ như: ánh sáng môi trường xung quanh, cũng như chất lượng màn hình thiết bị điện thoại..., ColorBlind hướng đến sự tương tác hiệu quả, tiết kiệm thời gian của người dùng muốn kiểm tra về bệnh mù màu, mà không phải thông qua bất kỳ nền tảng khác. ColorBlind hướng đến mong muốn trở thành một trong những công cụ hỗ trợ về thị giác, chi tiết là về bệnh mù màu, cũng như không chỉ riêng biệt cho người có khả năng bị mù màu, mà chính là hỗ trợ, đánh giá, kiểm tra các căn bệnh khác về thị giác.

ColorBlind không chỉ tập trung vào mục đích hỗ trợ người bị mù màu, mà có thể những người dùng muốn học màu sớm, nhận biết màu sắc. Ví dụ như trẻ con muốn nhận biết kiểm tra thử màu sắc, cũng như cho người dùng muốn trải nghiệm nhận biết màu sắc.

**b. Yêu cầu :**

- Yêu cầu công nghệ :
  - Sử dụng ngôn ngữ lập trình Dart.
  - Sử dụng Framework Flutter.
  - Sử dụng Android Studio để lập trình.
  - Sử dụng Figma thiết kế giao diện.
- Yêu cầu chức năng :
  - Hệ thống di động cho phép quyền sử dụng Camera.
  - Hệ thống di động cho phép quyền truy cập mở File.
  - Đánh giá khách quan khả năng mù màu của người dùng.
  - Hỗ trợ xác định tối thiểu các màu sắc mù của người dùng.
- Yêu cầu phi chức năng :
  - Tốc độ truy cập nhanh, chính xác.
  - Giao diện phù hợp với đa số người dùng.
  - Thao tác đơn giản, dễ dàng sử dụng.
  - Ứng dụng sử dụng hiệu năng thấp nhất.

### **3. Nội dung và phạm vi đề tài**

#### **a. Nội dung :**

- Tổng quan về bài toán yêu cầu.
- Khảo sát bài toán.
- Phân tích và thiết kế hệ thống theo yêu cầu.
- Thiết kế giao diện phù hợp.
- Xây dựng ứng dụng ColorBlind kiểm tra, đánh giá mức độ mù màu, dưới mức độ khách quan, hỗ trợ nhận diện màu sắc cho người bị mù màu.
- Kiểm thử và ứng dụng thực nghiệm.

#### **b. Phạm vi đề tài :**

- Nghiên cứu vận dụng kiến thức đã được học và kinh nghiệm tham gia phát triển ứng dụng di động.

- Phát triển ứng dụng phù hợp với khả năng bên cạnh sử dụng nhiều công nghệ mới bên ngoài trường lớp.
- Nghiên cứu ngôn ngữ Dart, Framework Flutter, Detection..., và một số thư viện hỗ trợ liên quan.

#### **4. Công nghệ, công cụ và ngôn ngữ lập trình**

##### **a. Công nghệ :**

- Framework Flutter

##### **b. Công cụ :**

- Android Studio
- Figma

##### **c. Ngôn ngữ lập trình :**

- Dart

#### **5. Các kết quả chính dự kiến sẽ đạt được và ứng dụng**

##### **a. Kết quả dự kiến đạt được :**

- Người dùng sẽ kiểm tra mức độ mù màu của bản thân với kết quả khách quan.
- Ứng dụng hỗ trợ nhận diện những màu sắc tối thiểu của người mù màu.
- Xây dựng trò chơi luyện tập về mắt nhận diện màu sắc.
- Gợi ý các ứng dụng thực tế hỗ trợ về căn bệnh mù màu trên CHPLAY.
- Bài kiểm tra mức độ mù màu theo dạng trắc nghiệm.

##### **b. Ứng dụng :**

- Trong tương lai ColorBlind mong muốn trở thành ứng dụng, công cụ, và hơn hết là người bạn đồng hành thiết thực, mang lại hiệu quả tốt nhất cho người dùng, kết quả sẽ được tối ưu với tỷ lệ chính xác nhất.
- Tiếp tục phát triển đưa ColorBlind không chỉ tập trung hỗ trợ về căn bệnh mù màu, phát triển đưa ColorBlind hỗ trợ về những căn bệnh về thị lực...
- Triển khai nâng cấp ứng dụng, về UI, UX, cũng như đưa lên chợ CHPL, xuất ra iOS.

**6. Giáo viên và cán bộ hướng dẫn**

Họ tên: ThS. Trần Thị Dung

Đơn vị công tác: Bộ Môn CNTT

Điện thoại: 0388389579

Email: ttdung@st.utc2.edu.vn

**Ngày tháng 03 năm 2022**

**Trưởng BM Công nghệ Thông tin**

**Đã giao nhiệm vụ TKTN**

**Giảng viên hướng dẫn**

**ThS. Trần Phong Nhã**

**ThS. Trần Thị Dung**

Đã nhận nhiệm vụ TKTN

Sinh viên: Nguyễn Anh Tuấn

Email: tuanura2010@gmail.com

## LỜI CẢM ƠN

Lời đầu tiên em xin cảm ơn Cô TRẦN THỊ DUNG đã giúp đỡ và hướng dẫn nhiệt tình cho em trong suốt thời gian em làm Báo Cáo Tốt Nghiệp này. Tiếp theo em xin cảm ơn quý Thầy Cô trường Đại Học Giao Thông Vận Tải phân hiệu tại TP.HCM đã giảng dạy và truyền cảm hứng đến tất cả những sinh viên như em, để chúng em có thêm nhiều kiến thức, mở mang thêm cách tư duy, sáng tạo trong cuộc sống.

Em rất mong nhận được lời khuyên, lời góp ý chân thành từ quý Thầy Cô để em tự hoàn thành hơn nữa vốn kiến thức của mình.

Với những kiến thức còn thiếu sót, những kỹ năng còn hạn hẹp, cũng như em đã dành hết thời gian và tâm huyết của mình để hoàn thành Báo Cáo Tốt Nghiệp, em không thể tránh được những sai lầm, em mong Thầy Cô có thể thông cảm và bỏ qua cho em.

Lời cuối cùng em xin chúc quý Thầy Cô phân hiệu cũng như Bộ Môn Công Nghệ Thông Tin và đặc biệt là Cô Trần Thị Dung có thật nhiều niềm vui, sức khỏe và luôn thành công trong công việc.

Em xin chân thành cảm ơn!

## NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

*Tp. Hồ Chí Minh, ngày ..... tháng 07 năm 2022*

**Giáo viên hướng dẫn**

**Trần Thị Dung**



# MỤC LỤC

<b>NHIỆM VỤ THIẾT KẾ TỐT NGHIỆP.....</b>	<b>i</b>
<b>LỜI CẢM ƠN .....</b>	<b>v</b>
<b>NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN.....</b>	<b>vi</b>
<b>MỤC LỤC .....</b>	<b>vii</b>
<b>DANH MỤC CHỮ VIẾT TẮT .....</b>	<b>xi</b>
<b>HÌNH VẼ .....</b>	<b>xii</b>
<b>CHƯƠNG 1. MỞ ĐẦU .....</b>	<b>1</b>
<b>1.1 Tổng quan đề tài :.....</b>	<b>1</b>
<b>1.2 Mục tiêu nghiên cứu : .....</b>	<b>1</b>
<b>1.3 Phạm vi nghiên cứu :.....</b>	<b>2</b>
<b>1.4 Cấu trúc báo cáo tốt nghiệp :.....</b>	<b>2</b>
1.4.1 Chương 1: Mở đầu .....	2
1.4.2 Chương 2: Cơ sở lý thuyết .....	2
1.4.3 Chương 3: Bài toán nhận diện màu sắc, chuẩn đoán căn bệnh mù màu.....	2
1.4.4 Chương 4: Giao diện ứng dụng, phân rã chức năng .....	3
<b>CHƯƠNG 2. CƠ SỞ LÝ THUYẾT .....</b>	<b>4</b>
<b>2.1 Các dạng mù màu phổ biến .....</b>	<b>4</b>
2.1.1 Mù màu đỏ - xanh lá : .....	4
2.1.2 Mù màu xanh lam - vàng : .....	4
2.1.3 Mù màu hoàn toàn : .....	4
2.1.4 Chuẩn đoán : .....	5
2.1.5 Phòng tránh, hạn chế bệnh mù màu và điều trị : .....	5
<b>2.2 Khái quát về căn bệnh mù màu : .....</b>	<b>7</b>
2.2.1 Mù màu là gì?.....	7

2.2.2 Thị lực của người bình thường.....	8
<b>2.3 Triệu chứng mù màu, dấu hiệu bệnh mù màu .....</b>	<b>8</b>
2.3.1 Triệu chứng mù màu : .....	8
2.3.2. Dấu hiệu bệnh mù màu .....	8
<b>2.4 Nguyên nhân bệnh mù màu.....</b>	<b>9</b>
<b>2.5 Ảnh hưởng của căn bệnh mù màu đến người bệnh .....</b>	<b>9</b>
<b>2.6 Giới thiệu về Flutter:.....</b>	<b>11</b>
2.6.1 Khái quát về Flutter.....	11
2.6.2 Quá trình phát triển Flutter.....	11
2.6.3 Cấu trúc Flutter ?.....	12
2.6.4 Ưu nhược điểm Flutter .....	13
2.6.5 Các đặc tính vượt trội của Flutter: .....	14
<b>2.7 Ngôn ngữ lập trình Dart : .....</b>	<b>15</b>
2.7.1 Dart là gì? .....	15
2.7.2 Tại sao lại chọn Dart ? .....	15
2.7.3 Những ưu điểm của Dart :.....	16
2.7.4 Tìm hiểu về Null safety Dart: .....	16
<b>2.8 GetX State Manager : .....</b>	<b>16</b>
<b>CHƯƠNG 3 . BÀI TOÁN NHẬN DIỆN MÀU SẮC, CHUẨN ĐOÁN CĂN BỆNH MÙ MÀU.....</b>	<b>18</b>
<b>3.1 Bài toán nhận diện màu sắc : .....</b>	<b>18</b>
3.1.1 Giới thiệu xử lý ảnh .....	18
3.1.2 Qui trình xử lý ảnh .....	18
3.1.2 Một số khái niệm trong xử lý ảnh .....	19
3.1.3 Đặc trưng Haar-like.....	20
3.1.4 Thuật toán Adaboost .....	23
3.1.5. Mạng nơ-ron tích chập và phương pháp Transfer Learning .....	28

a. Mạng nơ-ron tích chập .....	28
b) Phương pháp Transfer Learning .....	29
<b>3.2 Quy trình bài toán nhận dạng hình ảnh màu sắc.....</b>	<b>30</b>
3.2.1 Bài toán nhận dạng và huấn luyện .....	30
3.2.2 Vấn đề và giải pháp liên quan đến bài toán .....	30
3.2.3 Thư viện Tensorflow .....	31
3.2.4 Quy trình bài toán nhận diện màu sắc.....	31
3.2.5 Vấn đề và giải pháp liên quan đến bài toán .....	31
<b>3.3 Chuẩn đoán căn bệnh mù màu .....</b>	<b>32</b>
3.3.1 Dữ liệu chuẩn đoán căn bệnh mù màu .....	32
3.3.2 Cơ sở chuẩn đoán căn bệnh mù màu.....	32
<b>CHƯƠNG 4 . GIAO DIỆN ỨNG DỤNG, PHÂN RÃ CHỨC NĂNG.....</b>	<b>36</b>
<b>4.1 Giao diện ứng dụng :.....</b>	<b>36</b>
4.1.1 Giao diện Splash, giới thiệu ứng dụng.....	36
4.1.2 Giao diện trang chủ :.....	37
4.1.3 Giao diện tính năng kiểm tra mù màu :.....	38
4.1.4 Giao diện màn hình kết quả sau khi kiểm tra : .....	39
4.1.5 Giao diện tính năng đề xuất các ứng dụng hỗ trợ mắt : .....	40
4.1.6 Giao diện tính năng nhận diện màu sắc : .....	41
4.1.7 Giao diện tính năng luyện tập mắt : .....	43
<b>4.2 Phân tích thiết kế hệ thống :.....</b>	<b>45</b>
4.2.1 Sơ lược hệ thống : .....	45
4.2.2 Sơ đồ phân rã chức năng :.....	46
<b>KẾT LUẬN VÀ KIẾN NGHỊ.....</b>	<b>47</b>
<b>1. Kết quả : .....</b>	<b>47</b>
1.1 Kết quả đạt được : .....	47
1.2 Hạn chế còn thiếu sót : .....	47

2. Hướng phát triển :.....	47
PHỤ LỤC .....	48
Hướng dẫn cài đặt .....	48
TÀI LIỆU THAM KHẢO .....	49

## DANH MỤC CHỮ VIẾT TẮT

STT	Mô tả	Ý nghĩa	Ghi chú
1	UI	User Interface	
2	UX	User Experience	
3	AI	Artificial Intelligence	
4	AOT	Alignment Optical Telescope	
5	ARM	Global Positioning System	
6	SDK	Software Development Kit	
7	CHPLAY	Google Play	

## HÌNH VẼ

Hình 2.1 Mù màu có nhiều thể với nhau .....	5
Hình 2.2 Màu sắc của người mù màu đỏ.....	9
Hình 2.3 Ảnh hưởng của bệnh mù màu đến người bệnh .....	10
Hình 2.4 Hình Logo Flutter.....	11
Hình 2.5 Hình cấu trúc Flutter.....	12
Hình 2.6 Hình tính năng vượt trội Flutter. ....	14
Hình 2.7 Hình Logo Dart.....	15
Hình 3.1 Các bước cơ bản trong một hệ thống xử lý ảnh. ....	18
Hình 3.2 Bốn đặc trưng cơ bản của Haar-like.....	20
Hình 3.3 Đặc trưng cạnh. ....	21
Hình 3.4 Đặc trưng đường.....	21
Hình 3.5 Đặc trưng xung quanh tâm. ....	21
Hình 3.6 Integral Image.....	22
Hình 3.7 Tính toán Integral Image. ....	22
Hình 3.8 Hình màu sắc chuẩn. ....	33
Hình 3.9 Hình nhận diện mù màu đỏ. ....	33
Hình 3.10 Hình nhận diện mù màu đỏ - xanh lục. ....	33
Hình 3.11 Hình nhận diện mù màu đỏ - xanh lam. ....	34
Hình 3.12 Hình nhận diện mù xanh lá cây. ....	34
Hình 3.13 Hình nhận diện mù màu. ....	35
Hình 4.1 Màn hình Splash.....	36
Hình 4.2 Màn hình trang chủ ứng dụng. ....	38
Hình 4.3 Giao diện tính năng kiểm tra mù màu .....	39
Hình 4.4 Màn hình trả kết quả, gợi ý ứng dụng. ....	40
Hình 4.5 Màn hình giao diện gợi ý các ứng dụng hỗ trợ khác.....	41

Hình 4.6 Giao diện tính năng nhận diện màu sắc.....	42
Hình 4.7 Màn hình kết quả tính năng nhận diện màu sắc. ....	43
Hình 4.8 Màn hình giao diện tính năng luyện tập mắt. ....	44
Hình 4.9 Màn hình kết quả tính năng luyện mắt. ....	45
Hình 4.10 Sơ đồ phân rã chức năng .....	46

# CHƯƠNG 1. MỞ ĐẦU

## 1.1 Tổng quan đề tài :

Con người sở hữu một đôi mắt được xếp vào là một trong những thứ tốt nhất trong tự nhiên. Tuy nhiên, không phải ai cũng may mắn có được đôi mắt có thể nhìn được thế giới muôn màu, có những người chỉ thấy mọi vật xung quanh với màu sắc ảm đạm hơn rất nhiều – đó là do chứng bệnh “mù màu” gây nên.

Lý do nêu trên là một trong những nguồn động lực, cũng như mục tiêu của ColorBlind ra đời nhằm phục vụ cho việc đánh giá, kiểm tra, hỗ trợ ở mức tối thiểu nhất có thể cho người dùng có khả năng mắc căn bệnh “mù màu”, cũng như những người dùng muốn kiểm tra khả năng đôi mắt nhận diện màu sắc như thế nào với tính năng luyện tập mắt, hoặc có thể những học sinh nhỏ tuổi muốn được nhận biết sớm về màu sắc, thể hiện sự nhạy bén về nhận biết màu sắc, sự nhanh tay khi so sánh nhận biết 2 màu sắc. ColorBlind hướng đến là nơi tin cậy, tin tưởng, hỗ trợ nhanh chóng nhận diện chính xác cho người dùng kiểm tra, đánh giá về mắt, với kết quả khách quan về mức độ đánh. ColorBlind sẽ không bỏ qua những chi tiết dù nhỏ nhất đó là hướng đến sự tương tác hiệu quả giữa người dùng với ứng dụng, tiết kiệm thời gian cho người dùng khi muốn trải nghiệm kiểm tra đánh giá, cũng như được hỗ trợ vấn đề về “mù màu”, mong muốn trở thành một công cụ hỗ trợ đến người dùng với góc cảm nhận như một người bạn tin cậy.

## 1.2 Mục tiêu nghiên cứu :

ColorBlind hiện tại đã hoàn thành khá tốt so với mục tiêu đề ra ban đầu nghiên cứu đồ án, mặc dù vậy ColorBlind luôn luôn mong muốn phát triển hơn và tham vọng trở thành ứng dụng, công cụ, và hơn hết là người bạn đồng hành thiết thực, mang lại hiệu quả thiết thực cho người dùng.

Qua những mục tiêu nghiên cứu ban đầu như trên đã nói, thì điều tiên quyết đó là cải về cách tổ chức mã chương trình từ UI, UX, cũng như về mặt y học, đi sâu nghiên cứu về thị giác con người, để trả về kết quả không chỉ còn là khách quan, mà là kết quả thực sự chuẩn xác hơn.

Tổng hợp các ý trên thì đã nói rõ về mục tiêu nghiên cứu của đề tài nghiên cứu tiếp theo, nói rõ hơn là về ColorBlind, còn hiện tại ColorBlind đã và đang thực hiện các mục tiêu đề ra là: Hỗ trợ, kiểm tra, đánh giá ở mức độ mức độ mù màu với kết quả khách quan, hỗ trợ người dùng không bị mù, mà hỗ trợ về nhận diện màu sắc sớm ở trẻ con.



### **1.3 Phạm vi nghiên cứu :**

Nghiên cứu vận dụng kiến thức đã được học và kinh nghiệm tham gia phát triển ứng dụng di động qua môn học Java Android, và tự nghiên cứu học hỏi qua công nghệ Flutter, ngôn ngữ lập trình Dart, cũng như các thư viện hỗ trợ liên quan đến đề tài.

Nghiên cứu ngôn ngữ sâu hướng đối tượng, Dart, mặt y học về thị giác con người.

### **1.4 Cấu trúc báo cáo tốt nghiệp :**

#### **1.4.1 Chương 1: Mở đầu**

Với chương 1 Mở đầu, em sẽ trình bày chi tiết về Mục tiêu nghiên cứu của đề tài tốt nghiệp, cũng như phạm vi nghiên cứu, chi tiết về chương 1 như sau:

- Tổng quan đề tài
- Mục tiêu nghiên cứu
- Phạm vi nghiên cứu
- Cấu trúc báo cáo tốt nghiệp

#### **1.4.2 Chương 2: Cơ sở lý thuyết**

Với chương tiếp theo là chương 2 cơ sở lý thuyết, sẽ chỉ rõ về căn bệnh mù màu, đi rõ về các dạng mù màu, nguyên nhân..., cũng như chi tiết về công nghệ và ngôn ngữ lập trình sử dụng cho đề tài nghiên cứu, chi tiết hơn về cơ sở lý thuyết chương 2 như sau:

- Các dạng mù màu phổ biến
- Khái quát về căn bệnh mù màu
- Triệu chứng mù màu, dấu hiệu bệnh mù màu
- Nguyên nhân bệnh mù màu
- Ảnh hưởng căn bệnh mù màu đến người bệnh
- Giới thiệu về Flutter
- Ngôn ngữ lập trình Dart

#### **1.4.3 Chương 3: Bài toán nhận diện màu sắc, chuẩn đoán căn bệnh mù màu**

Chương 3, sẽ đi rõ về bài toán nhận diện màu sắc, cũng như bài toán chuẩn đoán căn bệnh mù màu. Với chương 3, em sẽ trình bày rõ về bài toán nhận diện màu sắc, quy trình nhận dạng, cũng như cơ sở để chuẩn đoán căn bệnh mù màu. Chương 3 sẽ được chia theo bố cục như sau:

- Bài toán nhận diện màu sắc
- Quy trình bài toán nhận dạng hình ảnh màu sắc
- Chuẩn đoán căn bệnh mù màu

#### **1.4.4 Chương 4: Giao diện ứng dụng, phân rã chức năng**

Chương cuối đó là chính là chương 4: Giao diện ứng dụng, phân rã chức năng. Sẽ tập trình bày về giao diện của ứng dụng ColorBlind, nêu lên khái quát về cách hoạt động, tính năng chính và phụ của ứng dụng. Cũng như không thể thiếu đó là phân tích thiết kế hệ thống của đề tài.

- Giao diện ứng dụng
- Phân tích thiết kế hệ thống

#### **Kết luận và kiến nghị**

Phần kết luận và kiến nghị sẽ tập trung đi sâu về kết quả đạt được, cũng như các hạn chế khi triển khai đề tài. Và hướng đi phát triển của ứng dụng ColorBlind trong tương lai.

- **Kết quả**
- **Hướng phát triển**

#### **Phụ lục**

## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

### 2.1 Các dạng mù màu phổ biến

#### 2.1.1 Mù màu đỏ - xanh lá :

Ở vùng thị lực của những bệnh nhân này thì thường tế bào nón màu đỏ - xanh lá hoạt động sai hoặc không hoạt động. Biểu hình là:

Mù màu xanh lá thể nhẹ: Lúc này người bệnh có xu hướng nhìn màu xanh lá và vàng sang đỏ và có khả năng nhận biết từ màu xanh trong màu tím. Thể này thường tỉ lệ nam giới mắc là 5%, ít xảy ra ở nữ giới.

Mù màu đỏ thể nhẹ: Tỉ lệ nam giới mắc thể này là 1% còn nữ giới thường không có. Màu đỏ - vàng thường nhìn sang xanh, kém tươi do tế bào hình nón màu đỏ không hoạt động.

Mù màu đỏ: Tương tự trên, tỉ lệ này ở nam giới là 1%, nữ giới không có. Màu đỏ có xu hướng nhìn sang màu xám đen còn cam, xanh lá nhìn sang vàng.

Mù màu xanh lá: Xanh lá có thể chuyển sang màu be còn đỏ nhìn sang màu vàng nâu. Tỉ lệ này cũng là 1% [6]

#### 2.1.2 Mù màu xanh lam - vàng :

Loại mù màu này có thể xảy ra ở cả hai giới nữ và nam.

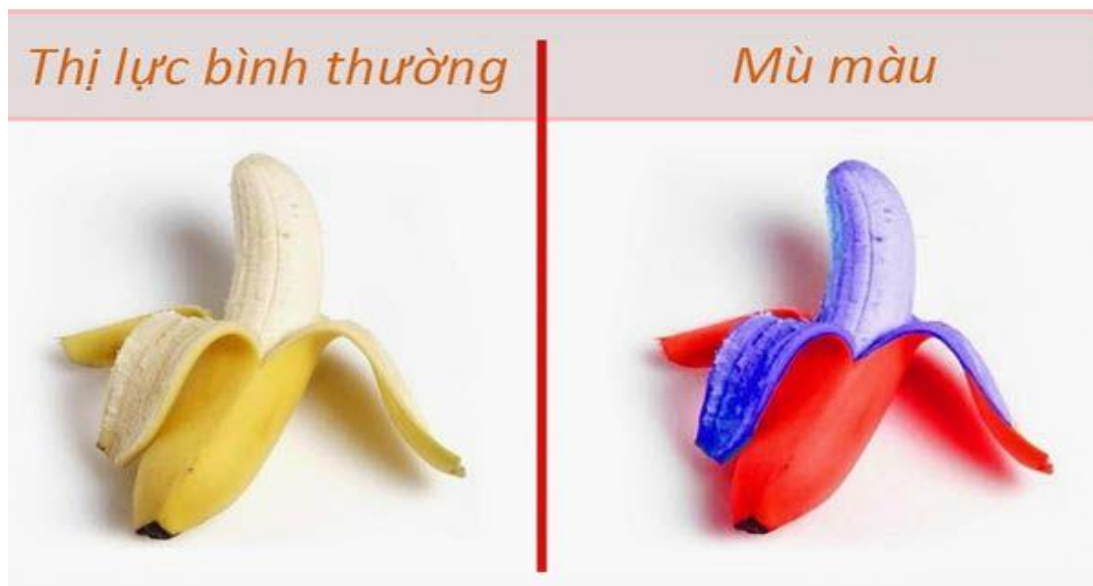
Mù màu xanh lam - vàng thể nhẹ: Tuy hiếm nhưng vẫn có một số người nhìn xanh lam xu hướng sang xanh lá còn đỏ và vàng thì khó quan sát được màu hồng trong đó.

Mù màu xanh lam - vàng: Xanh lam thành xanh lá còn vàng xu hướng chuyển sang tím hoặc xám. Tỉ lệ này cũng hiếm gặp.

#### 2.1.3 Mù màu hoàn toàn :

Hình nón: Khi các tế bào nón màu đỏ, xanh lá, xanh lam chỉ hoạt động hai trong ba hoặc hiệu quả kém. Điều này dẫn đến là bạn khó có thể phân biệt màu từ một màu sắc khác loại. Và nếu trong trường hợp là màu xanh lam thì tầm nhìn của bạn cũng hoạt động yếu hơn, hình ảnh nhìn cũng kém sắc nét. Tình trạng có thể xảy ra là bạn bị cận, không kiểm soát mắt nhìn cố định được hay còn có tên gọi khác là rung giập nhãn cầu.

Hình que: Đây là tình trạng bệnh nghiêm trọng nhất với các tế bào hình nón không hoạt động dẫn đến là màu sắc bạn nhận biết chỉ là xám, trắng, đen. Nếu trong điều kiện ánh sáng có những sắc tố rực rỡ có thể gây ảnh hưởng xấu đến mắt.



**Hình 2.1 Mù màu có nhiều thể với nhau.**

#### **2.1.4 Chuẩn đoán :**

Bệnh nhân bị mù màu khi đi khám sẽ thường được bác sĩ chỉ định làm bài kiểm tra để đánh giá khả năng nhận biết màu sắc. Thường sẽ là một bảng màu có nhiều chấm và việc của bạn là tìm ra hình trong bảng đó.

Bằng cách này bác sĩ sẽ nhận định được là bạn đang bị mù màu gì. Ngoài ra bạn có thể làm thêm bài kiểm tra sắp xếp thứ tự các màu để nhận định rõ hơn. Bệnh có thể được phát hiện ở độ tuổi trẻ con vì thế trẻ nên được khám khi 3 - 5 tuổi để sớm hỗ trợ cũng như điều hướng trẻ theo các ngành nghề phù hợp.

#### **2.1.5 Phòng tránh, hạn chế bệnh mù màu và điều trị :**

a. Phòng tránh bệnh mù màu :

Để phòng tránh bệnh mù màu xảy ra, cần chú ý một số vấn đề sau:

- Hàng năm bạn nên đi kiểm tra sức khỏe toàn diện để phát hiện sớm cũng như yếu tố di truyền ảnh hưởng đến thể hệ sau. Trẻ con cũng nên đi khám thị giác từ khi 3 - 5 tuổi.
- Không tự ý sử dụng thuốc điều trị khi chưa có sự chỉ định từ bác sĩ.
- Thị giác tổn thương có thể dẫn đến mù màu nên tránh tối đa các hoạt động động chạm gây hại đến vùng mắt.
- Khi làm các công việc tiếp xúc với hóa chất thì cần trang bị đồ bảo hộ khu vực mắt đầy đủ.
- Nếu thị giác có vấn đề thì nên đi khám sớm.

Trên đây là những thông tin về bệnh mù màu. Người bệnh không cần có chế độ ăn uống đặc biệt mà việc quan trọng là tập thích nghi với bệnh cũng như nên có các công cụ hỗ trợ việc nhận biết màu xung quanh.

#### b. Hạn chế bệnh mù màu :

Mặc dù không có cách chữa dứt điểm bệnh mù màu ở người bệnh bị mù màu nhưng nếu biết cách áp dụng một số thói quen trong sinh hoạt và cuộc sống sẽ giúp hạn chế diễn tiến của bệnh mù màu đáng kể. Người bệnh cần:

- Học cách ghi nhớ thứ tự của các màu sắc (khi tham gia giao thông sẽ dễ dàng hơn khi gặp đèn xanh đèn đỏ).
- Nhờ người thân có khả năng phân biệt màu sắc bình thường để dán tên màu lên những đồ dùng quen thuộc hoặc quần áo.
- Thăm khám sức khỏe định kỳ và sử dụng thuốc điều trị (nếu muốn) theo chỉ dẫn của bác sĩ.
- Xây dựng chế độ làm việc, nghỉ ngơi hợp lý để không gia tăng thêm các bệnh lý về mắt.
- Có chế độ ăn uống đảm bảo khoa học với các chất dinh dưỡng giúp mắt sáng khỏe.

#### c. Điều trị

Bệnh mù màu có chữa được không? Như đã nói ở trên nếu do biến chứng của bệnh hay sử dụng thuốc thì triệu chứng của bệnh có thể chữa khỏi nhưng nếu do yếu tố di truyền thì hiện nay chưa có phương thức điều trị.

Tuy nhiên người bệnh có thể cải thiện, phân biệt được màu rõ hơn thông qua sáng chế kính lọc màu sắc để tăng độ tương phản của màu đỏ, xanh biển, xanh lá.

Bên cạnh đó, bạn có thể học cách nhớ các thứ tự màu sắc của những vật dụng xung quanh môi trường sống. Ví dụ như đèn đường giao thông hoặc đánh dấu màu sắc của quần áo để dễ dàng kết hợp chúng với nhau hơn.

## **2.2 Khái quát về căn bệnh mù màu :**

### **2.2.1 Mù màu là gì?**

#### **a. Khái niệm mù màu:**

Mù màu hay còn được gọi là rối loạn sắc giác, là tình trạng khả năng phân biệt màu sắc bị giảm. Khi mắc bệnh này, người bệnh sẽ gặp khó khăn trong việc phân biệt giữa các màu đỏ, xanh lá cây, màu xanh biển hoặc các loại màu sắc được pha trộn từ những màu này. Ngay cả đối với những người mù màu thì rất hiếm khi xuất hiện trường hợp không thể nhìn thấy màu nào cả.

Khi gặp phải khó khăn trong việc phân biệt các màu sắc khác nhau chính là biểu hiện của bệnh mù màu hay còn gọi là rối loạn sắc giác. Người bệnh có thể nhìn rõ vật nhưng không phân biệt được màu sắc, đặc biệt là màu xanh với màu đỏ hoặc màu xanh dương với màu vàng.

#### **b. Mù màu là do đột biến gì và lịch sử thú vị của nó ?**

Mù màu là một bệnh di truyền do đột biến có liên quan đến cặp nhiễm sắc thể giới tính (ở nữ giới là XX và ở nam là XY). Người mắc bệnh mù màu do đột biến hoặc thiếu một gen trên nhiễm sắc thể X, gây ra sự rối loạn tế bào cảm thụ ánh sáng ở mắt cần để phân biệt màu sắc (thông thường gen này là gen lặn). Những thai nhi là bé trai mà nhận được ở mẹ loại gen này thì sẽ gặp khó khăn trong việc phân biệt màu sắc vì nhiễm sắc thể Y là nhiễm sắc thể sẽ không có gen màu sắc trội để lấn át gen mù màu.

Bệnh mù màu là một tật bẩm sinh. Tuy bệnh đã có từ cổ xưa, nhưng do hạn chế về kiến thức và người bệnh vẫn có khả năng nhìn nhận sự vật bình thường (chỉ không phân biệt được một số màu) nên không một bệnh nhân nào biết khuyết tật của mình. Người ta cho rằng người đầu tiên phát hiện ra bệnh mù màu là John Dalton (1766-1844), nhà vật lý học nổi tiếng sống ở cuối thế kỷ 18 đầu thế kỷ 19, người đặt nền móng cho lý thuyết nguyên tử.

Nhân ngày lễ Noel, John Dalton mua biếu mẹ một đôi tất màu gụ. Không ngờ bà mẹ xem xong rồi cười, bà hỏi Dalton: “Tuổi tác như mẹ làm sao có thể đi tất màu đỏ được?”. Thì ra Dalton tưởng rằng tất có màu gụ, nhưng thực ra là màu đỏ. John Dalton đi hỏi nhiều

người, họ đều xác nhận đó là màu đỏ, cuối cùng phát hiện ra chính ông là người mù màu đỏ và màu xanh. Cho đến nay vẫn có nhiều người gọi bệnh mù màu là bệnh Dalton.

Người phương Đông ít bị mù màu hơn người phương Tây. Theo thống kê của nước ngoài, chỉ có 4-5% đàn ông phương Đông bị mù màu, còn người phương Tây thì lên tới 8-9%

### **2.2.2 Thị lực của người bình thường**

Thị lực có khả năng phân biệt màu do tổng hợp ba cảm giác màu ứng với một bước sóng ánh sáng khác nhau: cam – tím (có bước sóng 450 nm), vàng – lục (520 nm) và đỏ (630 nm).

Ở người, võng mạc có ba loại tế bào nón, mỗi loại mẫn cảm nhất với một bước sóng nhất định là 430 nm (ứng với tím), 540 nm (ứng với xanh lục) và 575 nm (ứng với đỏ). Có lẽ ở các loại đó có các opsin khác nhau cho mỗi loại. Khi hòa trộn các màu cơ bản đó với nhau theo một tỷ lệ nhất định có thể có muôn màu sắc khác nhau..

## **2.3 Triệu chứng mù màu, dấu hiệu bệnh mù màu**

### **2.3.1 Triệu chứng mù màu :**

Rối loạn sắc giác có thể chia làm hai mức độ:

- Khuyết sắc (không phân biệt được giữa màu lục và màu đỏ và loại không phân biệt được giữa màu xanh da trời và màu vàng.)
- Mù màu (hoàn toàn không phân biệt được giữa các màu).

### **2.3.2. Dấu hiệu bệnh mù màu :**

Dấu hiệu đầu tiên và dễ dàng nhận biết nhất là bạn khó có khả năng phân biệt được các màu sắc như đỏ, xanh lá, xanh biển đặc biệt trong môi trường ánh sáng yếu còn các màu khác như vàng, tím, cam,... vẫn có thể biết được.

- Mắt trở nên nhạy cảm hơn nếu trong điều kiện quá sáng.
- Một số trường hợp hiếm bệnh nhân còn chỉ có thể nhận biết được các màu âm tính như đen, xám, trắng.
- Với người bình thường có thể nhìn thấy hàng nghìn sắc tố màu khác nhau nhưng bệnh nhân bị bệnh mù màu chỉ nhận biết được một số sắc thái.
- Có những vấn đề về thị lực nhưng thậm chí người bệnh còn không nhận biết được.

## 2.4 Nguyên nhân bệnh mù màu

Tuy yếu tố di truyền phổ biến nhất đối với căn bệnh này nhưng không loại trừ các yếu tố khác là:

**Biến chứng của bệnh:** Những bệnh nhân đang mắc phải các căn bệnh như Alzheimer, Parkinson, thoái hóa điểm vàng, cao huyết áp, bệnh bạch cầu,... đều có khả năng bị mù màu ở 1 bên mắt hoặc cả hai bên. Tuy nhiên nếu bệnh giảm thì tình trạng mù màu cũng có thể được cải thiện hoặc thậm chí là khỏi hẳn.

**Sử dụng thuốc:** Một số thuốc được bác sĩ chỉ định để điều trị bệnh cũng có khả năng dẫn đến mù màu như thuốc chữa cao huyết áp, điều trị tim mạch, rối loạn thần kinh hay chức năng cương dương, nhiễm trùng.

**Lão hóa:** Khi tuổi càng cao thì nguy cơ bị nhiễm mù màu cũng cao hơn so với hồi bạn còn trẻ



**Hình 2.2 Màu sắc của người mù màu đỏ.**

## 2.5 Ảnh hưởng của căn bệnh mù màu đến người bệnh

Về mặt cấu tạo, tất cả cả tế bào và đường dẫn truyền thần kinh ở trong mắt và não bộ của con người đều hiện diện từ khi mới sinh ra, phần phía sau của mắt, trong võng mạc có 2 loại tế bào: Tế bào hình que và tế bào hình nón.



Tế bào hình que: Là những tế bào có sự nhạy cảm với ánh sáng nhưng không thể nhìn thấy được các màu sắc khác nhau, tế bào hình que có nhiệm vụ giúp con người nhìn được mọi thứ vào ban đêm nhưng chỉ ở sắc thái xám, đen hoặc trắng.

Tế bào hình nón: Là những tế bào có sự phản ứng với ánh sáng nhiều hơn nên có thể giúp con người nhìn chi tiết hơn các vật thể và màu sắc của vật thể. Tế bào hình nón có 3 loại và mỗi loại sẽ tiếp nhận các màu riêng biệt bởi các tia sáng đỏ, xanh dương và xanh lá cây. Nhờ sự kết hợp thông tin của từng cặp tế bào hình nón mà con người có được dải quang phổ rộng thường thấy. Những người bị mù màu sẽ bị khiếm khuyết 1 vài tế bào nón này.

Người bị mù màu sẽ có những ảnh hưởng nhất định đến công việc và cuộc sống mặc dù vẫn có thể nhìn thấy mọi đồ vật xung quanh. Những khó khăn khi nhận biết màu sắc và phân biệt đồ vật có màu sắc khác nhau sẽ làm ảnh hưởng đến công việc hàng ngày, đặc biệt là những việc liên quan đến thiết kế, đồ họa, lái xe...Ngoài những người bị mù màu do di truyền thì cũng có những người mù màu tạm thời do gặp phải chấn thương làm tổn thương não hoặc võng mạc hoặc các bệnh lý ở mắt....Đối với những người bị mù màu xanh lá và đỏ mà làm công việc lái xe thì họ có thể sẽ được lấy bằng lái xe nhưng không được cấp bằng lái thương mại và không được phép lái xe vào ban đêm, đây có thể là ảnh hưởng khá lớn đến công việc mà người bệnh sẽ phải chấp nhận.



**Hình 2.3 Ảnh hưởng của bệnh mù màu đến người bệnh.**

Đối với những người bị mù màu tạm thời, họ có thể không nhận ra cho đến khi có người phản ánh và điều này sẽ tác động liên tâm lý của người bệnh, có người sẽ bị sốc và khó chấp nhận hiện tượng này.

Trong cuộc sống, những người bị mù màu sẽ gặp phải rất nhiều khó khăn trong việc phân biệt đồ vật có màu sắc khác nhau hoặc đơn giản là lựa chọn để phối đồ mặc hàng ngày và như vậy lại phải cần đến sự giúp đỡ của người thân.

Cho đến thời điểm hiện tại thì y học vẫn chưa tìm ra cách chữa bệnh mù màu, tuy nhiên, người bệnh có thể cải thiện khả năng phân biệt màu sắc của mình bằng cách sử dụng các loại kính dành riêng cho người mù màu (kính lọc và kính tiếp xúc nhuộm màu)...Thay vì tìm cách chữa trị thì người bị mù màu cần học cách chấp nhận và sống chung với căn bệnh này, mặc dù có những ảnh hưởng nhất định đến cuộc sống nhưng về lâu dài sẽ dần quen với việc này.

## **2.6 Giới thiệu về Flutter:**

### **2.6.1 Khái quát về Flutter**

Flutter là bộ công cụ phát triển phần mềm UI framework mã nguồn mở được phát triển bởi Google và phát hành vào tháng 5 năm 2017. Flutter hiện nay với Flutter 3 cho phép người dùng tạo ra các giao diện chất lượng cao trên OS, Android, Linux, Mac, Windows, Web,.. trong khoảng thời gian ngắn. [1]



**Hình 2.4 Hình Logo Flutter.**

### **2.6.2 Quá trình phát triển Flutter**

Vào năm 2015, Google đã công bố Flutter, một SDK mới dựa trên ngôn ngữ Dart, làm nền tảng tiếp theo để phát triển Android và vào năm 2017, phiên bản alpha của nó (0.0.6) đã được phát hành cho công chúng lần đầu tiên.

Tại I/O 2017, Google đã trình diễn việc sử dụng Flutter và các khả năng đa nền tảng của nó, đồng thời tiếp tục quảng bá nó tại I/O 2018. Kể từ đó, Google đã đầu tư vào Flutter và khuyến nghị đây là cách mà mọi người nên phát triển ứng dụng di động.

Vào ngày 4 tháng 12 năm 2018, Flutter 1.0 đã được phát hành tại sự kiện Flutter Live và có sẵn để các nhà phát triển có thể bắt đầu sử dụng SDK để tạo ứng dụng dễ dàng hơn. Đây được đánh dấu là phiên bản “stable” đầu tiên.

Tại Google I/O 2019, việc hỗ trợ Flutter cho desktop và web platforms đã được công bố công khai, các công cụ để phát triển ứng dụng Flutter cho Windows, macOS, Linux và web đã được phát hành. Cùng với đó là sự ra đời của Flutter 1.12.

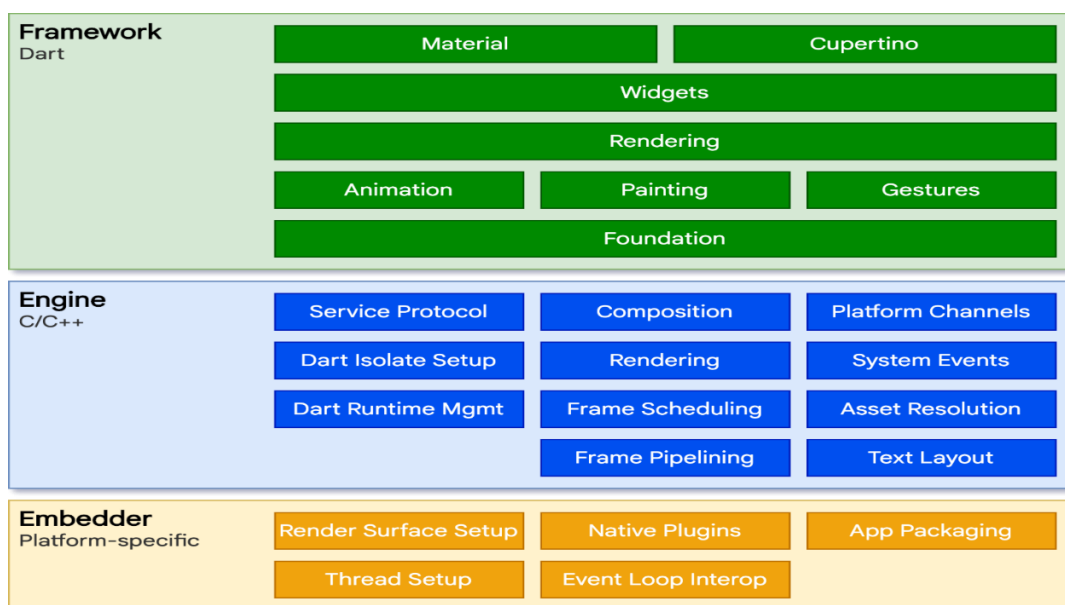
Ngày 30 tháng 4 năm 2021, 2.0.6 chính là phiên bản cập nhật mới của Flutter.

Tính đến tháng 5 năm 2022, 3.0 là phiên bản mới nhất được cập nhật của Flutter...[5]

### 2.6.3 Cấu trúc Flutter ?

Cấu trúc Flutter chủ yếu bao gồm bốn thành phần:

- Flutter Engine: Phương tiện của Flutter.
- Foundation Library: Thư viện nền tảng.
- Widgets: Các widget.
- Design Specific Widgets: Thiết kế những widget riêng.



Hình 2.5 Hình cấu trúc Flutter.

## 2.6.4 Ưu nhược điểm Flutter

Dưới đây sẽ nêu chi tiết về ưu nhược điểm của Flutter như sau :

### ❖ Flutter : Ưu điểm:

- Mạnh về hiệu ứng, hiệu suất ứng dụng rất cao.
- Giao tiếp gần như trực tiếp với hệ thống
- Ngôn ngữ kiểu tĩnh nhưng với cú pháp hiện đại (tương tự JS, Python, Java), compiler linh động khi dùng AOT (cho sản phẩm cuối) và JIT (cho quá trình phát triển với hot reload)
- Có thể chạy được giả lập mobile ngay trên web, tiện cho việc phát triển. Các bộ đo lường chỉ số hiệu suất được hỗ trợ sẵn giúp lập trình viên kiểm soát tốt hiệu suất của ứng dụng.
- Có thể dùng để xây dựng các nền tảng gắn vào ứng dụng native để tăng hiệu suất.

### ❖ Flutter : Nhược điểm:

- Bộ render UI được nhóm phát triển gần như viết lại, không liên quan tới UI có sẵn của Framework native, dẫn đến memory sử dụng khá nhiều. Hơn nữa, các UI không đi chung với iOS, mà được phát triển riêng, nghĩa là cùng 1 phiên bản Flutter khi tạo ra ứng dụng cho iOS thì iOS 8.x -> 12.x đều y chang nhau, tương tự như với Android. Nhưng UI của Android thì tất nhiên khác với iOS.
- Phải học thêm ngôn ngữ Dart: lập trình viên biết về Dart không nhiều, cũng có rủi ro là học xong Dart sẽ dính liền luôn với Dart ở mảng phát triển ứng dụng mobile. Chứ không uyển chuyển như JS hay Python có thể nhảy qua lại giữa front, back hay AI...
- Là con cưng của Google, tuy nhiên hãng dính nhiều phốt với thói quen “quăng con giữa chợ” nên cũng cần cân nhắc. Tuy nhiên, mình nhận thấy Flutter rất tốt, tốt hơn nhiều so với những cái mà Google từng làm ra như Angular.

### 2.6.5 Các đặc tính vượt trội của Flutter:

❖ Native đúng nghĩa :

- Bộ engine để render UI hoàn toàn thuộc Flutter, vì vậy bạn có thể tạo ra bất kỳ hiệu ứng và đồ họa nào cho cả 2 nền tảng iOS và Android. Bạn có thể tưởng tượng Flutter như 1 engine game, có thể điều chỉnh đến từng pixel và phủ trên đó là hệ thống Widget đa dạng, phong phú.

❖ Hot Reload :

- Khi làm việc với những thứ liên quan đến UI thì Hot Reload thật khiến cho lập trình viên dễ chịu, càng tuyệt hơn khi Flutter cung cấp khả năng ghi nhớ state của ứng dụng. Hot Reload cung cấp cho bạn trải nghiệm tuyệt vời khi lập trình Flutter, so với React Native thì điểm này Flutter hơn.
- Hãy tưởng tượng khi gặp lỗi khi chạy chương trình. Trong Flutter, có thể sửa nó ngay lập tức, tiếp tục nơi đã dừng mà không cần khởi động lại toàn bộ.

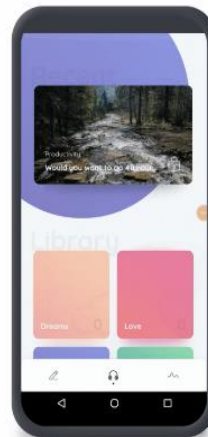
❖ Expressive và Flexible UI



#### Expressive, beautiful UIs

Delight your users with Flutter's built-in beautiful Material Design and Cupertino (iOS-flavor) widgets, rich motion APIs, smooth natural scrolling, and platform awareness.

[Browse the widget catalog](#)



Reflectly

Download: [iOS](#), [Android](#)  
[Learn more](#)

**Hình 2.6 Hình tính năng vượt trội Flutter.**

- Đây là lý do thúc đẩy Flutter ra đời. Trước đó việc xây dựng các ứng dụng UI đẹp, animation mượt mà rất phức tạp và gian nan với hầu hết các nhà phát triển ứng dụng di động.
- Trong Flutter, giao diện người dùng được xây dựng với các Widget, các khối xây dựng UI nhỏ được lắp ráp bằng một kỹ thuật gọi là Composition. Bản

thân Flutter cũng có sẵn rất nhiều Widget dựng sẵn, các Developer có thể tận dụng dễ dàng hơn thay vì xây dựng chúng từ ban đầu.

❖ Dart DevTools :

- DevTools tuy mới ở giai đoạn beta nhưng độ hoàn thiện cũng rất cao, so với DevTools của React/React Native thì DevTools của Flutter tốt hơn.

## 2.7 Ngôn ngữ lập trình Dart :

### 2.7.1 Dart là gì?

Để lập trình với Flutter, bạn sẽ sử dụng một ngôn ngữ lập trình gọi là Dart. Ngôn ngữ này được tạo bởi Google vào năm 2011, nó đã được cải thiện rất nhiều trong những năm qua. [2]

Dart tập trung vào phát triển front end, và bạn có thể sử dụng nó để tạo ứng dụng di động, ứng dụng web...



**Hình 2.7 Hình Logo Dart.**

### 2.7.2 Tại sao lại chọn Dart ?

Các nhà phát triển tại Google và các nơi khác sử dụng Dart để tạo các ứng dụng chất lượng cao, quan trọng cho iOS, Android và web. Với các tính năng nhằm đến sự phát triển phía khách hàng, Dart rất phù hợp cho cả ứng dụng di động và web. Dart giúp bạn tạo ra những trải nghiệm đẹp, chất lượng cao trên tất cả các màn hình với:

- Một ngôn ngữ được tối ưu hóa cho client
- Framework mạnh mẽ
- Công cụ linh hoạt

### 2.7.3 Những ưu điểm của Dart :

Năng suất Cú pháp Dart rõ ràng và súc tích, công cụ của nó đơn giản nhưng mạnh mẽ. Type-safe giúp bạn xác định sớm các lỗi tĩnh. Dart có các thư viện cốt lõi và một hệ sinh thái gồm hàng ngàn package.

- Sở hữu cú pháp rõ ràng và súc tích, cùng bộ công cụ đơn giản nhưng mạnh mẽ.
- Có type-safe hỗ trợ hiệu quả cho việc xác định lỗi

Nhanh Dart cung cấp tối ưu hóa việc biên dịch trước thời hạn để có được dự đoán hiệu suất cao và khởi động nhanh trên các thiết bị di động và web.

Di động Dart biên dịch thành mã ARM và x86, để các ứng dụng di động của Dart có thể chạy tự nhiên trên iOS, Android, Window, Linux, Web và hơn thế nữa. Đối với các ứng dụng web, chuyển mã từ Dart sang JavaScript.

Dễ gần Dart quen thuộc với nhiều nhà phát triển hiện có, nhờ vào cú pháp và định hướng đối tượng không gây ngạc nhiên của nó. Nếu bạn đã biết C ++, C # hoặc Java, bạn có thể làm việc hiệu quả với Dart chỉ sau vài ngày.

### 2.7.4 Tìm hiểu về Null safety Dart:

Vào ngày 3 tháng 3 năm 2021, team Flutter đã công bố Flutter 2 và Dart 2.12. Trong đó, có một số thay đổi rất quan trọng là Sound Null Safety.

Null Safety, còn được biết đến với tên gọi nnbd (Non Nullable By Default), là sự đảm bảo của ngôn ngữ rằng trừ khi lập trình viên chỉ định, không biến nào là có thể null. Nếu lập trình viên đặc biệt tạo một biến “nullable”, trình phân tích và biên dịch đảm bảo rằng lập trình viên sẽ xử lý các trường hợp, mà giá trị của biến có thể là null.

Sound Null safety giúp chúng ta tránh được các lỗi xảy ra mà không lường trước được, trường hợp điển hình và dễ thấy nhất đó là giá trị của một biến bị null nhưng thực tế thì chúng không nên null trong lúc chương trình đang chạy. [4]

### 2.8GetX State Manager :

Câu nói mà nhắc đến GetX sẽ nhớ đầu tiên đó chính là cực kỳ nhẹ nhàng, mạnh mẽ cho Flutter. Qua câu nói trên cũng đã nói lên giường như hết tất cả về toàn bộ GetX, và dưới đây sẽ đi sơ lược qua những ý chính cần biết về GetX lý thuyết. [3]

GetX có 3 nguyên tắc cơ bản. Điều này có nghĩa là đây là ưu tiên cho tất cả các tài nguyên trong thư viện: năng suất, hiệu suất và tổ chức.

- Hiệu suất: GETX tập trung vào hiệu suất và tiêu thụ tài nguyên tối thiểu. GetX không sử dụng luồng hoặc `changeNotifier`.

- Năng suất: GetX sử dụng cú pháp dễ dàng và dễ chịu. Bất kể bạn muốn làm gì, luôn có một cách dễ dàng hơn với `getx`. Nó sẽ tiết kiệm giờ phát triển và sẽ cung cấp hiệu suất tối đa mà ứng dụng của bạn có thể cung cấp. Nói chung, nhà phát triển nên quan tâm đến việc loại bỏ bộ điều khiển khỏi bộ nhớ.

- Tổ chức: GetX cho phép tổng số quá trình phân tách của chế độ xem, logic trình bày, logic kinh doanh, tiêu phụ thuộc và điều hướng. Bạn không cần bối cảnh để điều hướng giữa các tuyến đường, vì vậy bạn không phụ thuộc vào cây widget (trực quan hóa) cho việc này. Bạn không cần bối cảnh để truy cập các bộ điều khiển/khối của mình thông qua một kế thừa, vì vậy bạn hoàn toàn tách rời logic trình bày và logic kinh doanh từ lớp trực quan hóa của bạn. Bạn không cần phải đưa các lớp điều khiển/mô hình/khối của bạn vào cây widget của bạn thông qua `Multiproviders`. Đối với điều này, GETX sử dụng tính năng tiêu phụ thuộc của riêng mình, tách hoàn toàn DI khỏi quan điểm của nó.

Getx thật sự thú vị. Nó có vô số tính năng cho phép bạn bắt đầu lập trình mà không phải lo lắng về bất cứ điều gì, nhưng mỗi tính năng này đều nằm trong các thùng chứa riêng biệt và chỉ được bắt đầu sau khi sử dụng. Nếu bạn chỉ sử dụng quản lý nhà nước, chỉ có quản lý nhà nước sẽ được biên soạn. Nếu bạn chỉ sử dụng các tuyến đường, không có gì từ quản lý nhà nước sẽ được biên soạn. GetX có một hệ sinh thái khổng lồ, một cộng đồng lớn, một số lượng lớn các cộng tác viên và sẽ được duy trì miễn là sự rung động tồn tại. GetX cũng có khả năng chạy với cùng một mã trên Android, iOS, Web, Mac, Linux, Windows và trên máy chủ của bạn. Có thể sử dụng lại hoàn toàn mã của bạn được thực hiện trên mặt trận trên phần phụ trợ của bạn với `Get Server`.



## CHƯƠNG 3 . BÀI TOÁN NHẬN DIỆN MÀU SẮC, CHUẨN ĐOÁN CĂN BỆNH MÙ MÀU

### 3.1 Bài toán nhận diện màu sắc :

#### 3.1.1 Giới thiệu xử lý ảnh

Con người thu nhận thông tin qua các giác quan, trong đó thị giác đóng vai trò quan trọng nhất. Những năm trở lại đây với sự phát triển của phần cứng máy tính, xử lý ảnh và đồ họa đã phát triển một cách mạnh mẽ và có nhiều ứng dụng trong cuộc sống. Xử lý ảnh và đồ họa đóng một vai trò quan trọng trong tương tác người máy.

Quá trình xử lý ảnh được xem như là quá trình thao tác ảnh đầu vào nhằm cho ra kết quả mong muốn. Kết quả đầu ra của một quá trình xử lý ảnh có thể là một ảnh “tốt hơn” hoặc một kết luận.

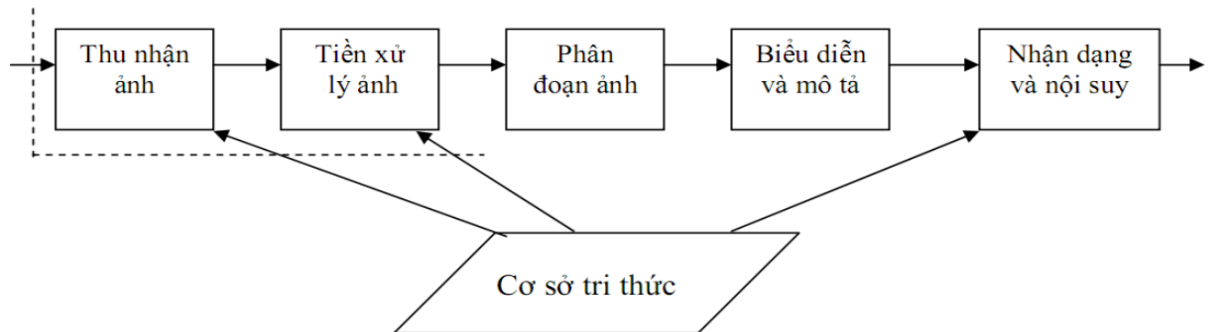
Có thể hiểu một cách khác, xử lý ảnh hay cao cấp hơn nữa là thị giác máy tính (Computer Vision) bao gồm tất cả các lý thuyết và kỹ thuật liên quan, cho phép tạo lập một hệ thống có khả năng tiếp nhận thông tin từ các hình ảnh thu được, lưu trữ và xử lý theo nhu cầu.

Mục đích của xử lý ảnh gồm:

- Biến đổi ảnh làm tăng chất lượng hình ảnh.
- Tự động nhận dạng ảnh, đoán nhận ảnh, đánh giá các nội dung của ảnh

#### 3.1.2 Qui trình xử lý ảnh

Quá trình xử lý ảnh là một quá trình gồm một chuỗi các thao tác với mục đích cuối cùng là nhận dạng được hình ảnh đó. Các bước trong quá trình xử lý ảnh bao gồm: thu nhận ảnh, tiền xử lý ảnh, phân loại ảnh, biểu diễn và mô tả, nhận dạng và nội suy.



**Hình 3.1** Các bước cơ bản trong một hệ thống xử lý ảnh.

- **Thu nhận ảnh:** Quá trình tiếp nhận thông tin từ vật thể thông qua camera màu hoặc trắng đen, ảnh thu nhận được có thể là ảnh tương tự hoặc ảnh đã số hóa.

- **Tiền xử lý ảnh:** Sau bộ thu nhận, ảnh có thể nhiễu, độ tương phản thấp nên cần đưa vào bộ tiền xử lý để nâng cao chất lượng. Chức năng chính của bộ tiền xử lý là lọc nhiễu, nâng độ tương phản để làm ảnh rõ hơn, nét hơn.

- **Phân đoạn ảnh:** Là tách một ảnh đầu vào thành các vùng thành phần để biểu diễn phân tích, nhận dạng ảnh. Ví dụ: để nhận dạng chữ (hoặc mã vạch) trên phong bì thư cho mục đích phân loại bưu phẩm, cần chia các câu, chữ về địa chỉ hoặc tên người thành các từ, các chữ, các số (hoặc các vạch) riêng biệt để nhận dạng. Đây là phần phức tạp khó khăn nhất trong xử lý ảnh và cũng dễ gây lỗi, làm mất độ chính xác của ảnh. Kết quả nhận dạng ảnh phụ thuộc rất nhiều vào công đoạn này.

- **Biểu diễn ảnh:** Đầu ra ảnh sau phân đoạn chứa các điểm ảnh của vùng ảnh (ảnh đã phân đoạn) cộng với mã liên kết với các vùng lân cận. Việc biến đổi các số liệu này thành dạng thích hợp là cần thiết cho xử lý tiếp theo bằng máy tính. Việc chọn các tính chất để thể hiện ảnh gọi là trích chọn đặc trưng (Feature Selection) gắn với việc tách các đặc tính của ảnh dưới dạng các thông tin định lượng hoặc làm cơ sở để phân biệt lớp đối tượng này với đối tượng khác trong phạm vi ảnh nhận được. Ví dụ: trong nhận dạng ký tự trên phong bì thư, chúng ta miêu tả các đặc trưng của từng ký tự giúp phân biệt ký tự này với ký tự khác.

- **Nhận dạng và nội suy ảnh:** Nhận dạng ảnh là quá trình xác định ảnh. Quá trình này thường thu được bằng cách so sánh với mẫu chuẩn đã được học (hoặc lưu) từ trước. Nội suy là phán đoán theo ý nghĩa trên cơ sở nhận dạng. Ví dụ: một loạt chữ số và nét gạch ngang trên phong bì thư có thể được nội suy thành mã điện thoại.

### 3.1.2 Một số khái niệm trong xử lý ảnh

**Ảnh và điểm ảnh:** Góc của ảnh (ảnh tự nhiên) là ảnh liên tục về không gian và độ sáng. Để xử lý bằng máy tính, ảnh cần phải được số hoá. Số hoá ảnh là sự biến đổi gần đúng một ảnh liên tục thành một tập điểm phù hợp với ảnh thật về vị trí (không gian) và độ sáng (mức xám). Khoảng cách giữa các điểm ảnh đó được thiết lập sao cho mắt người không phân biệt được ranh giới giữa chúng. Mỗi một điểm như vậy gọi là điểm ảnh và ảnh được xem như là 1 tập hợp các điểm ảnh.

**Độ phân giải của ảnh:** Độ phân giải (Resolution) của ảnh là mật độ điểm ảnh được ấn định trên một ảnh số được hiển thị. Mức xám của ảnh: Mức xám của điểm ảnh là cường

độ sáng của nó được gán bằng giá trị số tại điểm đó. Giá trị mức xám thông thường: 16, 32, 64, 128, 256.

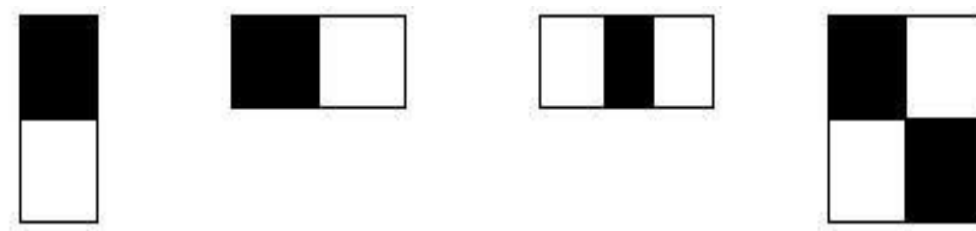
**Ảnh đen trắng:** là ảnh có hai màu đen, trắng (không chứa màu khác) với mức xám ở các điểm ảnh có thể khác nhau.

**Ảnh nhị phân:** là ảnh chỉ có 2 mức đen trắng phân biệt tức dùng 1 bit mô tả 2 mức khác nhau. Nói cách khác: mỗi điểm ảnh của ảnh nhị phân chỉ có thể là 0 hoặc 1.

### 3.1.3 Đặc trưng Haar-like

Hình ảnh màu sắc là đặc trưng bởi tập hợp các pixel trong vùng biến báo mà các pixel này tạo lên những điểm khác biệt so với các vùng pixel khác. Tuy nhiên với một ảnh đầu vào, việc sử dụng các pixel riêng lẻ lại không hiệu quả. Vì vậy những nhà nghiên cứu đã đưa ra tư tưởng kết hợp các vùng pixel lại cùng với nhau để tạo ra những đặc trưng có khả năng để phân loại các biến báo với nhau. Các đặc trưng Haar-like (còn gọi là mặt nạ Haar-like) được công bố bởi Viola và Jones là các hình chữ nhật có kích thước xác định với các vùng sáng tối đan xen lẫn nhau. Giá trị đặc trưng của Haar-like là chênh lệch tổng mức xám của các điểm ảnh trong các vùng tối so với các vùng sáng, giá trị này sẽ được sử dụng để huấn luyện phân loại mạnh theo thuật toán Adaboost cho phép nhận diện biến báo giao thông và các đối tượng trên biến báo giao thông như màu sắc, hình dạng,... Phương pháp này là một phương pháp đơn giản và kết quả phát hiện là tương đối cao, lên tới 98%, Các hãng sản xuất máy ảnh nổi tiếng như Canon, Nikon, Samsung,... cũng đã tích hợp nó trong sản phẩm của mình.

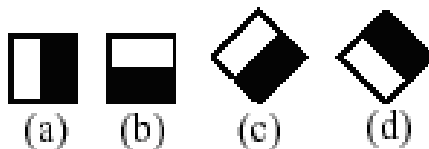
Đặc trưng Haar-like được tạo thành bằng việc kết hợp các hình chữ nhật màu trắng và các hình chữ nhật màu đen với nhau theo một trật tự nhất định, một kích thước nào đó. Haar-like có 4 đặc trưng cơ bản, mỗi đặc trưng của Haar-like chứa 2, 3 hoặc 4 miền hình học có màu “đen” hoặc “trắng” được trình bày như sau :



**Hình 3.2 Bốn đặc trưng cơ bản của Haar-like.**

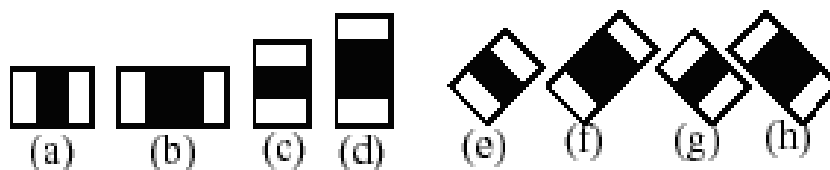
Hình vẽ trên là 4 loại đặc trưng cơ bản của Haar-like bao gồm 2, 3 hoặc 4 hình chữ nhật tạo thành. 4 đặc trưng Haar-like cơ bản được mở rộng và chia làm 3 tập đặc trưng như sau:

- Đặc trưng cạnh (edge features):



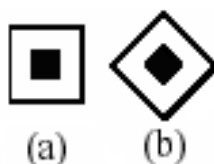
**Hình 3.3 Đặc trưng cạnh.**

- Đặc trưng đường (line features):



**Hình 3.4 Đặc trưng đường.**

- Đặc trưng xung quanh tâm (center-surround features):



**Hình 3.5 Đặc trưng xung quanh tâm.**

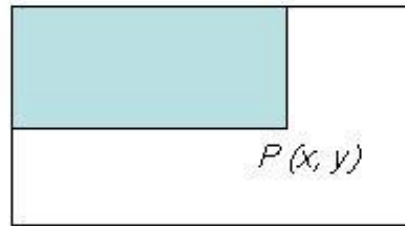
Giá trị của một đặc trưng Haar-like chỉ đơn giản là sự khác biệt giữa tổng các giá trị xám của các pixel trong vùng “đen” với tổng giá trị xám của các pixel trong vùng “trắng” theo công thức sau:

$$f(x) = \text{Tổng}_{\text{vùng đen}}(\text{pixel}) - \text{Tổng}_{\text{vùng trắng}}(\text{pixel})$$

Vậy khi đặc lên một vùng ảnh, đặc trưng Haar-like sẽ tính toán và đưa ra giá trị đặc trưng  $h(x)$  của vùng ảnh đó.

Nhận thấy ý nghĩa của đặc trưng Haar-like là nó thể hiện được tri thức về các đối tượng ảnh (thông qua việc thể hiện mối quan hệ giữa các bộ phận của đối tượng). Đặc trưng Haar-like được tính toán thông qua khái niệm “Integral Image”, được đề xuất bởi Viola và Jones. Ở đây, Integral Image tại vị trí  $(x, y)$  (Hình 2.6) có giá trị bằng tổng các giá trị pixel nằm trong hình chữ nhật được định bởi góc trái là  $(0,0)$  và góc phải là  $(x, y)$ :

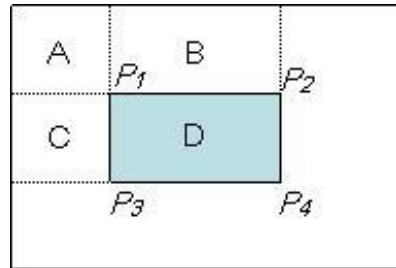
$$P(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y')$$



**Hình 3.6 Integral Image.**

Từ đây thì ta có thể tính được tổng các giá trị pixel của bất kỳ một hình chữ nhật nào đó thông qua Integral Image tại 4 đỉnh của nó.

Ví dụ: Tính Integral Image của vùng ảnh D.



**Hình 3.7 Tính toán Integral Image.**

Việc tính tổng điểm ảnh của một vùng bất kì nào đó trên ảnh được thực hiện như sau:

- A, B, C, D lần lượt là tổng các giá trị pixel của hình chữ nhật tương ứng.
- P1, P2, P3, P4 là Integral Image tại 4 đỉnh của hình chữ nhật D.

Ta có:  $P_1 = A$ ,  $P_2 = A + B$ ,  $P_3 = A + C$ ,  $P_4 = A + B + C + D$

Suy ra:

$$P_1 + P_4 - P_2 - P_3 = A + (A + B + C + D) - (A + B) - (A + C) = D$$

Vậy ta có:  $D = P_1 + P_4 - P_2 - P_3$ .

Khi áp dụng vào tính toán các giá trị đặc trưng ta có thể thấy:

- Đặc trưng cạnh được tính thông qua 6 giá trị điểm ảnh tích hợp.
- Đặc trưng đường và đặc trưng xung quanh tâm được tính thông qua 8 giá trị điểm ảnh tích hợp.
- Đặc trưng chéo được tính thông qua 9 giá trị điểm ảnh tích hợp.

### 3.1.4 Thuật toán Adaboost

#### a. Giới thiệu về thuật toán

Kỹ thuật Boosting: nguyên tắc cơ bản của Boosting là kết hợp các bộ phân lớp yếu (hay các bộ phân lớp cơ sở) để tạo nên một bộ phân lớp mạnh. Các bộ phân lớp yếu này thậm chí chỉ cần nhỉnh hơn phương pháp random một chút. Bằng cách này, ta nói bộ phân lớp đã được “boost.”

Adaboost (Adaptive Boosting) là một thuật toán sử dụng kỹ thuật Boosting được đưa ra bởi Schapire và Freund vào năm 1996. Thuật toán này có thể được sử dụng cùng với nhiều thuật toán khác để cải thiện hiệu suất cũng như là độ chính xác trong bài toán nhận diện. Thuật toán Adaboost được sử dụng phổ biến trong các bài toán xử lý ảnh phát hiện đối tượng nhất là trong nhận diện khuôn mặt.

Điểm cải tiến của thuật toán Adaboost là ta sẽ gán mỗi mẫu một trọng số. Ý nghĩa của việc gán trọng số này như sau: Ở mỗi vòng lặp của quá trình huấn luyện, khi một bộ phân lớp yếu  $y_i$  đã được xây dựng, ta sẽ tiến hành cập nhật trọng số cho các mẫu. Việc cập nhật này được tiến hành như sau: ta sẽ tăng trọng số của các mẫu bị phân lớp sai bởi bộ phân lớp yếu  $y_i$  và giảm trọng số của các mẫu được phân lớp đúng bởi  $y_i$ . Bằng cách này, ở vòng lặp kế, ta sẽ xây dựng bộ phân lớp yếu  $y_{i+1}$  theo hướng: tập trung vào các mẫu bị phân loại sai bởi bộ phân loại yếu trước đó.

Sau cùng, để có được bộ phân loại mạnh là ta phải kết hợp tuyến tính các bộ phân loại yếu đã tìm được lại với nhau. Mỗi bộ phân loại yếu sẽ được đánh một trọng số tương ứng với độ tốt của bộ phân loại yếu đó.

Như vậy chúng ta có thể hiểu ý nghĩa của từ “Adaptive” trong tên gọi của thuật toán Adaboost là: các bộ phân loại yếu tạo thành một chuỗi, trong đó bộ phân loại yếu sau sẽ tập trung giải quyết các mẫu bị phân loại sai bởi các bộ phân loại yếu trước đó.

#### b) Giới thiệu về thuật toán

Cho tập ảnh huấn luyện  $(x_1, t_1), \dots, (x_n, t_n)$  với  $t_i \in \{-1, 1\}$

1. Khởi tạo hệ số trọng cho mỗi mẫu huấn luyện:  $w_n^{(1)} = 1/N$  với  $n = 1, 2, \dots, N$ .

2. Với  $m = 1, \dots, M$ :

a. Xây dựng bộ phân loại yếu  $h_m$ :

+ Với mỗi đặc trưng  $j$ , xây dựng một bộ phân lớp  $h_j$  với độ lỗi:

$$E_j = \sum_{n=1}^N w_n^{(m)} I(h_m(x_n) \neq t_n) \quad (1.0)$$

với  $I(h_m(x_n) \neq t_n) = 1$  nếu  $h_m(x_n) \neq t_n$  và  $= 0$  nếu ngược lại.

+ Chọn bộ phân lớp  $h_j$  với độ lỗi nhỏ nhất ta được  $h_m$ .

b. Cập nhật trọng số:

+ Tính:

$$\epsilon_m = \frac{\sum_{n=1}^N w_n^{(m)} I(h_m(x_n) \neq t_n)}{\sum_{n=1}^N w_n^{(m)}} \quad (1.1)$$

và:

$$\alpha_m = \ln \frac{1 - \epsilon_m}{\epsilon_m} \quad (1.2)$$

+ Cập nhật trọng số:

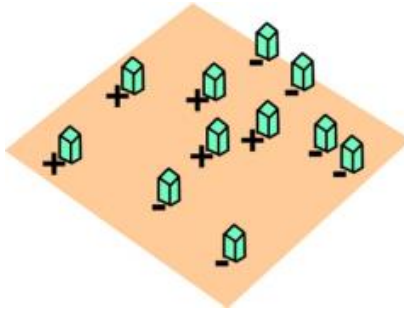
$$w_n^{(m+1)} = w_n^{(m)} \exp \{ \alpha_m I(h_m(x_n) \neq t_n) \} \quad (1.3)$$

3. Bộ phân lớp mạnh cuối cùng:

$$H_M(x) = \text{sign}[\sum_{m=1}^M \alpha_m h_m(x)] \quad (1.4)$$

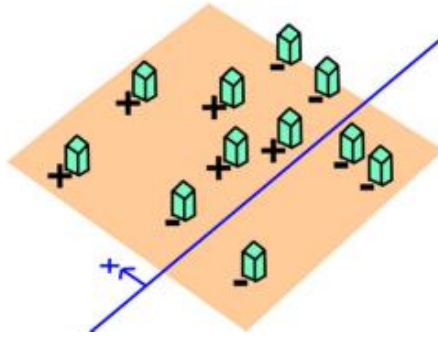
b) Giải thích thuật toán

Thuật toán bắt đầu bằng việc khởi tạo trọng số cho các mẫu huấn luyện. Các trọng số này được khởi tạo bằng nhau. Các trọng số này cho thuật toán biết được độ tương quan trọng số của mẫu.



Ở mỗi vòng lặp, ta làm 2 việc như sau:

- Thứ 1: Tìm bộ phân lớp yếu  $h_m$  dựa vào độ lỗi nhỏ nhất.



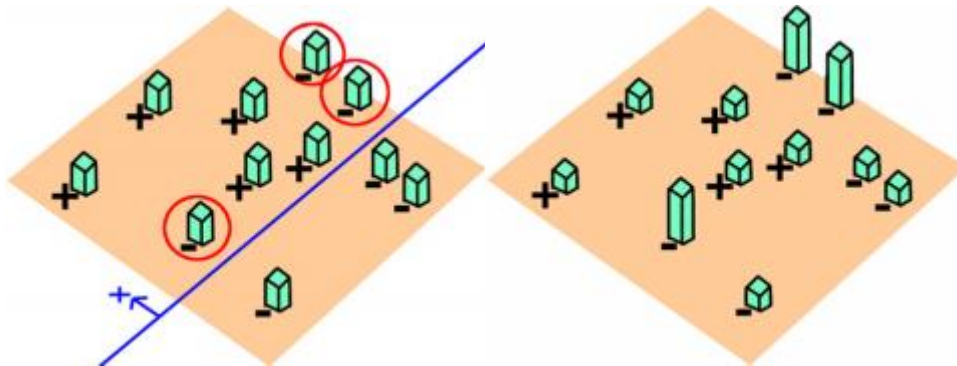
Công thức tính độ lỗi:

$$E_j = \sum_{n=1}^N w_n^{(m)} I(h_m(x_n) \neq t_n)$$

với  $I(h_m(x_n) \neq t_n) = 1$  nếu  $h_m(x_n) \neq t_n$  và  $= 0$  nếu ngược lại.

Công thức này đơn giản chỉ là tính tổng các trọng số của các mẫu bị phân loại sai.

▪ Thứ 2: cập nhật trọng số theo nguyên tắc: ta sẽ tăng trọng số cho các mẫu hiện đang bị phân lớp sai và giảm trọng số cho các mẫu hiện đang được phân lớp đúng. Bằng cách này, ở vòng lặp kế ta sẽ xây dựng bộ phân lớp yếu theo hướng tập trung giải quyết các mẫu bị phân lớp sai bởi bộ phân lớp yếu trước đó.



Công thức:

$$w_n^{(m+1)} = w_n^{(m)} \exp \{ \alpha_m I(h_m(x_n) \neq t_n) \}$$

Trong đó:

$$\alpha_m = \ln \frac{1 - \epsilon_m}{\epsilon_m}$$

với:



$$\epsilon_m = \frac{\sum_{n=1}^N w_n^{(m)} I(h_m(x_n) \neq t_n)}{\sum_{n=1}^N w_n^{(m)}}$$

Ta nhận thấy nếu mẫu được phân lớp đúng thì trọng số không thay đổi; nếu mẫu bị phân lớp sai thì:

$$w_n^{(m+1)} = w_n^{(m)} \exp(\alpha_m) = w_n^{(m)} \frac{1 - \epsilon_m}{\epsilon_m}$$

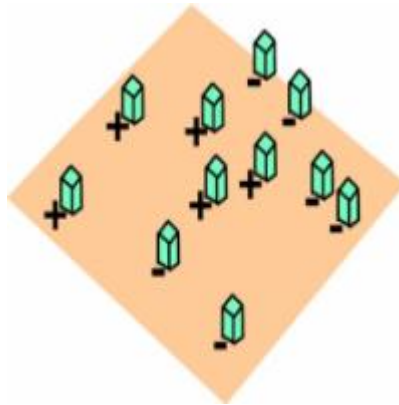
Thấy ngay  $\frac{1 - \epsilon_m}{\epsilon_m} > 1$ .

Giả sử  $\frac{1 - \epsilon_m}{\epsilon_m} < 1$  nghĩa là tỉ lệ phần sai lớn hơn phần đúng. Điều này mâu thuẫn với điều kiện bộ phân lớp yếu là tối ưu (có độ lỗi nhỏ nhất) vì rằng chỉ cần đổi chiều của mặt phẳng phân lớp lại là ta có ngay một bộ phân lớp yếu tốt hơn.

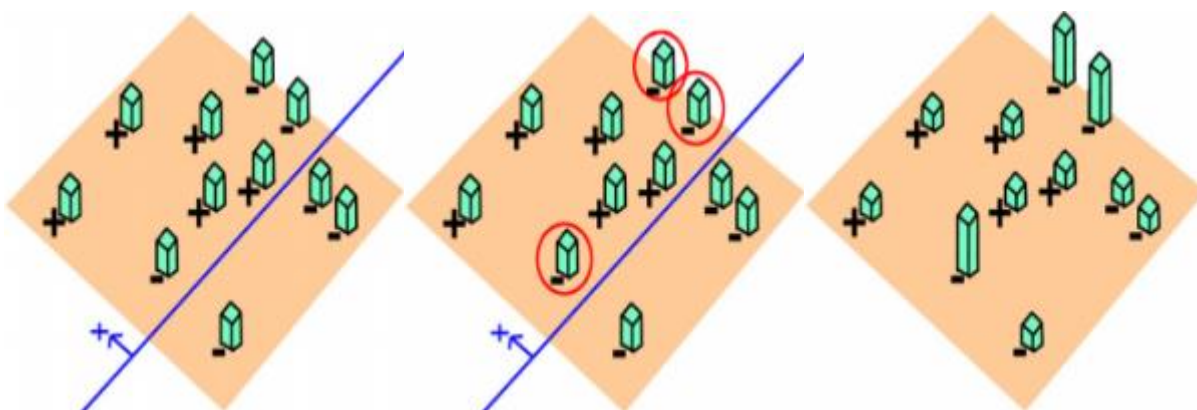
Và vì  $\frac{1 - \epsilon_m}{\epsilon_m} > 1$  nên  $w_n^{(m+1)} > w_n^{(m)}$ : trọng số của các mẫu bị phân loại sai được tăng lên.

Để trực quan, ta hãy quan sát thuật toán thông qua chuỗi các hình vẽ dưới đây:

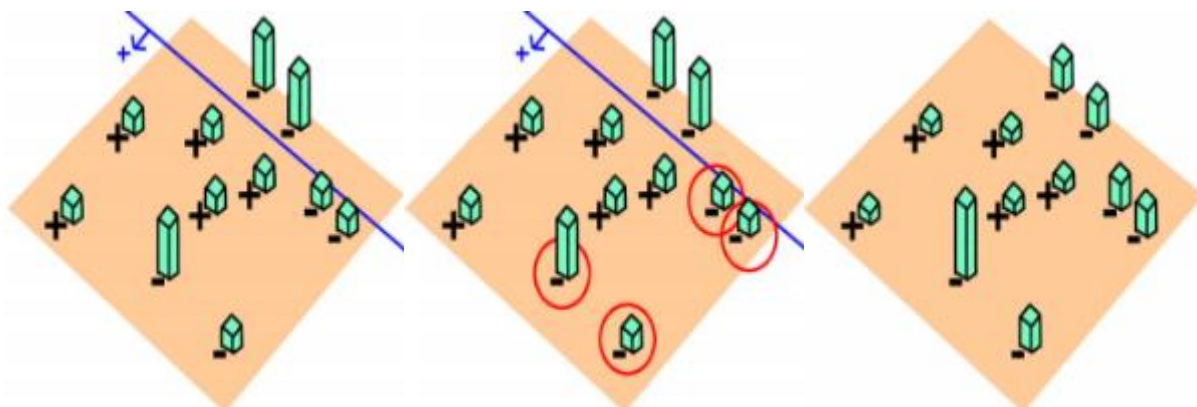
- Khởi tạo trọng số cho các mẫu:



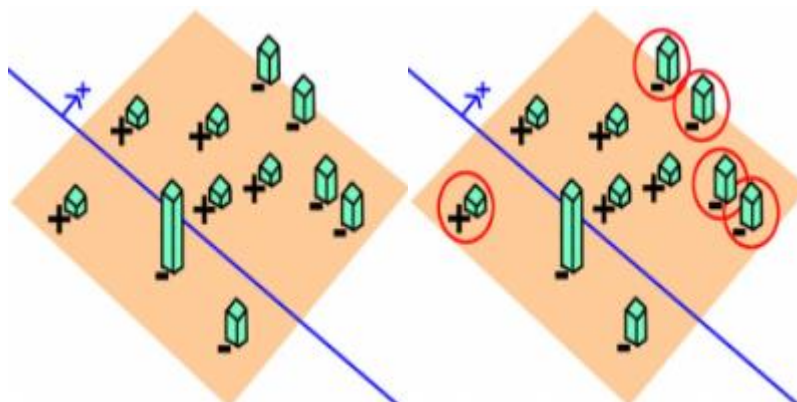
- Vòng lặp thứ nhất:



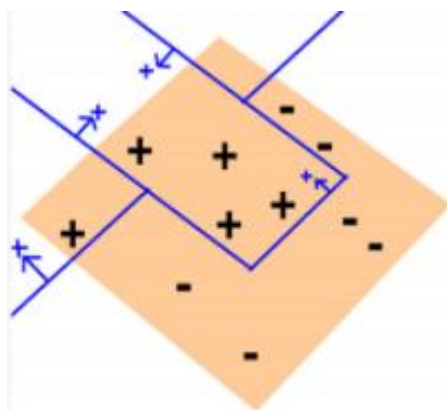
- Vòng lặp thứ hai:



- Vòng lặp thứ 3:



- Kết hợp các bộ phân loại yếu lại với nhau:



Cuối cùng, kết hợp tuyến tính các bộ phân lớp yếu lại ta được một bộ phân lớp mạnh:

$$H_M(x) = \text{sign}[\sum_{m=1}^M \alpha_m h_m(x)]$$

Nhận thấy mức độ đóng góp của mỗi bộ phân lớp yếu vào bộ phân lớp mạnh được quyết định bởi  $\alpha_m$ :  $\alpha_m = \ln \frac{1-\epsilon_m}{\epsilon_m}$

Dễ thấy  $\alpha_m$  tỉ lệ nghịch với  $\epsilon_m$  mà  $\epsilon_m$  thể hiện độ lỗi ứng với bộ phân lớp  $h_m$ . Như vậy, độ lỗi càng thấp, tức bộ phân lớp càng tốt thì mức độ đóng góp của nó vào bộ phân lớp mạnh cuối cùng sẽ càng nhiều.

Nói thêm về điều kiện dừng: trong thực tế cài đặt, người ta sử dụng một giá trị ngưỡng là tỉ lệ nhận dạng sai tối đa (max false positive) để làm điều kiện dừng. Qua các vòng lặp, tỉ lệ nhận dạng sai của bộ phân lớp mạnh sẽ giảm dần. Đến một lúc nào đó, tỉ lệ này nhỏ hơn tỉ lệ nhận dạng sai tối đa và ta sẽ dừng thuật toán.

### 3.1.5. Mạng nơ-ron tích chập và phương pháp Transfer Learning

#### a. Mạng nơ-ron tích chập

Mạng nơ-ron tích chập (Convolution neural network được viết tắt là CNN) là mô hình được Yann LeCun đề xuất, để giải quyết cho việc huấn luyện với dữ liệu là hình ảnh một cách hiệu quả nhất.

CNN là phiên bản chính quy của các tri giác đa lớp. Các tri giác đa lớp thường đề cập đến các mạng được kết nối đầy đủ, nghĩa là mỗi nơ-ron trong lớp được kết nối với tất cả các nơ-ron ở các lớp tiếp theo. Tính kết nối đầy đủ của các mạng này dễ khiến chúng bị quá tải dữ liệu. Các cách thông thường hóa bao gồm thêm một số hình thức đo cường độ của trọng số vào hàm mất. Tuy nhiên, CNN có một cách tiếp cận khác đối với việc chính quy

hóa: họ tận dụng mô hình phân cấp trong dữ liệu và lắp ráp các mẫu phức tạp hơn bằng cách sử dụng các mẫu nhỏ hơn và đơn giản hơn.

Cấu trúc của CNN gồm ba thành phần chính:

- Lớp tích chập (Convolution).
- Lớp giảm chiều (Pooling)
- Lớp kết nối đầy đủ (Fully Connected)

Lớp tích chập là thành phần chính giúp cho CNN xử lý dữ liệu hình ảnh một cách hiệu quả hơn so với các mạng nơ-ron (Neural Network) khác. Thay vì sử dụng các lớp của mạng nơ-ron truyền thống, thì nó sử dụng sử dụng một bộ lọc là filter để xử lý và trả về kết quả cho các lớp tiếp theo. Việc sử dụng các bộ filter như thế sẽ giúp làm giảm đi thông tin cần lưu trữ của mô hình cũng như phù hợp với tính chất tự nhiên của hình ảnh, thông tin của hình ảnh thường mang tính chất cục bộ, nên việc giữ lại một phần của hình ảnh trên phạm vi nhỏ sẽ hiệu quả hơn là việc dùng các lớp kết nối đầy đủ như các mạng nơ-ron truyền thống. Ví dụ như các pixel nằm ở hai biên của hình ảnh sẽ không nhiều thông tin như các pixel nằm gần nhau. Hơn nữa, bằng cách sử dụng bộ lọc, chúng ta có thể đạt được tính chất shift invariance bằng cách sử dụng cùng một bộ lọc cho cùng một đặc trưng, từ đó giảm được khá nhiều lượng trọng số của mô hình cần lưu trữ.

Vị trí của các đặc trưng không còn quan trọng khi đã được trích xuất, dữ liệu chúng ta cần giữ lại là vị trí tương đối của các đặc trưng với nhau. Lớp giảm chiều (Pooling) được sử dụng tính toán bằng cách lấy giá trị lớn nhất hoặc giá trị trung bình trong một cửa sổ nhỏ, nhằm giảm đi sự ảnh hưởng của vị trí các đặc trưng của ảnh, đồng thời làm giảm đi số lượng trọng số phải lưu, hạn chế học tử (overfitting). Bộ lọc dành cho lớp giảm chiều thường dùng là khoảng 2x2 để không làm mất đi quá nhiều dữ liệu.

#### b) Phương pháp Transfer Learning

Transfer Learning: giống như mạng nơ-ron, lấy cảm hứng từ việc con người có thể tổng hợp kiến thức từ các nguồn khác nhau để giải quyết vấn đề chưa được gặp. Mục tiêu của Transfer Learning là huấn luyện một mô hình (target task) mới dựa trên những đặc trưng trích xuất được từ mô hình cũ (source domain). InceptionV3 được huấn luyện trên dữ liệu ImageNet, với tổng số hình vẽ hoa là 1924 tấm, nên ta có thể xem rằng không gian đặc trưng của InceptionV3 khá tương đồng với mô hình chúng ta sẽ xây dựng, việc áp dụng Transfer Learning sẽ đạt được hiệu quả cao. Hơn nữa, Transfer Learning cũng đã đạt được những kết quả vô cùng khả quan trong thời gian gần đây.

Phương pháp áp dụng đầu tiên là thêm một lớp Softmax vào cuối cùng của InceptionV3. Lớp cuối Softmax cuối gồm 13 nút tương đương với 13 lớp cần phân loại. Cách thay đổi này cũng khá là hiển nhiên do chúng ta cần đầu ra của mô hình là một phân phối xác suất của 13 thành phần.

Lý do chúng ta chỉ huấn luyện khối Inception cuối hơn là chỉnh sửa các lớp trên hay toàn bộ mô hình là do dữ liệu ban đầu của chúng ta hạn chế. Những đặc trưng được học bởi những khối Inception đầu sẽ mang tính chất cơ bản hơn không cụ thể như những khối cuối nên việc giữ nguyên các khối đầu và thay đổi những lớp cuối là hợp lý.

## **3.2 Quy trình bài toán nhận dạng hình ảnh màu sắc**

### **3.2.1 Bài toán nhận dạng và huấn luyện**

Có nhiều phương pháp nhận dạng có một phương pháp dễ dàng thực thi và không tốn nhiều thời gian huấn luyện, không cần nhiều dữ liệu đó là Transfer learning (một phương pháp trong Deep Learning). Phương pháp này sử dụng Inception (một mô hình đã huấn luyện với 1000 lớp từ ImageNet 2012 Challenge Image DataSet, một mô hình khá nổi tiếng trên thế giới) sử dụng mô hình này để giải quyết bài toán nhận diện biển báo.

Bài toán sử dụng mô hình huấn luyện trong lập trình di động:

- Huấn luyện mô hình và sử dụng mô hình đó trên di động có một số thư viện nổi tiếng như opencv, ...
- Để phù hợp sử dụng phương pháp transfer learning ở đây Tensorflow là một cách chọn hợp lý. Tensorflow là một thư viện dành cho máy học, được phát triển bởi Google. Sử dụng Tensorflow để retrain lại một mô hình khá dễ dàng và nhanh chóng trên PC dựa vào pre-trained model của các model nổi tiếng trên thế giới để nhận dạng đối tượng mong muốn, ở tensorflow cho phép chuyển định dạng protobuffer để nó có thể sử dụng trong lập trình di động.

### **3.2.2 Vấn đề và giải pháp liên quan đến bài toán**

Thời gian thực: phát hiện đối tượng là bài toán chính vì khi phát hiện vùng chứa đối tượng để giảm chiều đặc trưng, loại bỏ các nền. Nhưng phải chọn giải pháp để xử lý với tốc độ đáp ứng.

Độ chính xác nhận dạng: độ chính xác khi huấn luyện là bài toán cần thiết cho vấn đề về các yếu tố bên ngoài, như ánh sáng, chất lượng phân giải màu sắc...

Thiết bị, môi trường huấn luyện không đáp ứng: Quá trình huấn luyện trên nhiều dữ liệu đặc biệt là dữ liệu ảnh khá lớn thiết bị cần phải có một cấu hình cao mới có thể đáp ứng được.

### **3.2.3 Thư viện Tensorflow**

TensorFlow là một thư viện phần mềm nguồn mở để tính toán số bằng các biểu đồ luồng dữ liệu. Các nút trong biểu đồ biểu thị các hoạt động toán học, trong khi các cạnh của biểu đồ biểu thị các mảng dữ liệu đa chiều (tensors) được truyền giữa chúng. Kiến trúc linh hoạt cho phép bạn triển khai tính toán cho một hoặc nhiều CPU hoặc GPU trong máy tính để bàn, máy chủ hoặc thiết bị di động với một API. TensorFlow ban đầu được phát triển bởi các nhà nghiên cứu và kỹ sư làm việc trong Nhóm Google Brain trong tổ chức nghiên cứu Máy thông minh của Google nhằm mục đích tiến hành học máy và nghiên cứu mạng nơ-ron sâu, nhưng hệ thống này đủ chung để áp dụng trong nhiều lĩnh vực khác như tốt.

### **3.2.4 Quy trình bài toán nhận diện màu sắc**

Có nhiều phương pháp nhận dạng có một phương pháp dễ dàng thực thi và không tốn nhiều thời gian huấn luyện, không cần nhiều dữ liệu đó là Transfer learning (một phương pháp trong DeepLearning). Phương pháp này sử dụng Inception (một mô hình đã huấn luyện với 1000 lớp từ ImageNet 2012 Challenge ImageDataSet, một mô hình khá nổi tiếng trên thế giới) sử dụng mô hình này để giải quyết bài toán nhận diện biển báo.

Bài toán sử dụng mô hình huấn luyện trong lập trình di động:

Huấn luyện mô hình và sử dụng mô hình đó trên di động có một số thư viện nổi tiếng như opencv, ...

Để phù hợp sử dụng phương pháp transfer learning ở đây Tensorflow là một cách chọn hợp lý. Tensorflow là một thư viện dành cho máy học, được phát triển bởi Google. Sử dụng Tensorflow để re-train lại một mô hình khá dễ dàng và nhanh chóng trên PC dựa vào pre-trained model của các model nổi tiếng trên thế giới để nhận dạng đối tượng mong muốn, ở tensorflow cho phép chuyển định dạng protobuffer để nó có thể sử dụng trong lập trình di động.

### **3.2.5 Vấn đề và giải pháp liên quan đến bài toán**

Độ chính xác nhận diện : độ chính xác khi huấn luyện là bài toán cần thiết, nhưng với tập dữ liệu, cũng như các màu sắc thật sự rất nhiều, và có những màu sắc, có sự giống nhau khó phân biệt. Phụ thuộc vào độ sáng của môi trường, chất lượng màn hình điện thoại...

Thiết bị, môi trường huấn luyện không đáp ứng: Quá trình huấn luyện trên nhiều dữ liệu đặc biệt là dữ liệu ảnh khá lớn thiết bị cần phải có một cấu hình cao mới có thể đáp ứng được. Và thời gian cho một thực thi cũng là khá lâu, làm giảm tiến độ.

### **3.3 Chuẩn đoán căn bệnh mù màu**

#### **3.3.1 Dữ liệu chuẩn đoán căn bệnh mù màu**

Ứng dụng ColorBlind được dựa trên màu Kiểm tra Ishihara. Màu sắc Kiểm tra Ishihara là một ví dụ về một thử nghiệm cảm nhận màu sắc cho thiếu màu đỏ-xanh. Nó được đặt tên sau khi nhà thiết kế của mình, Tiến sĩ Shinobu Ishihara, một giáo sư tại Đại học Tokyo, người đầu tiên được xuất bản thử nghiệm của ông trong 1917.

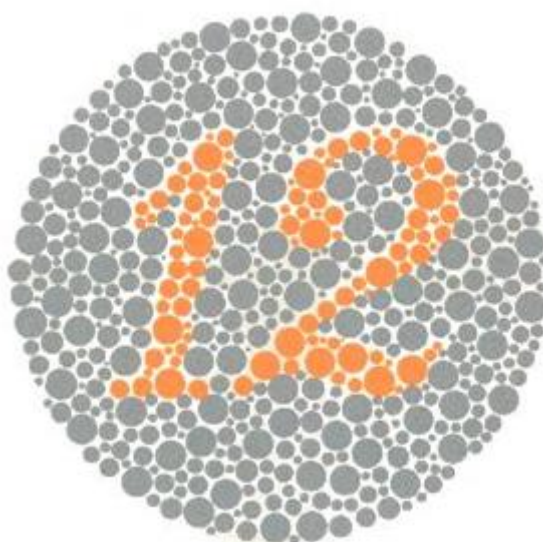
Đây là thử nghiệm được sử dụng phổ biến nhất để kiểm tra mù màu. Thật không may, thử nghiệm này chỉ có thể phát hiện mù màu đỏ-xanh. Kiểm tra mù màu Ishihara sử dụng các vòng tròn bao gồm nhiều điểm có màu sắc và kích cỡ khác nhau. Một số điểm sẽ tạo thành số nhất định.

#### **3.3.2 Cơ sở chuẩn đoán căn bệnh mù màu**

Dưới đây là dữ liệu đại diện làm cơ sở chuẩn đoán căn bệnh mù màu

a. Màu sắc chuẩn :

Màu sắc chuẩn là màu sắc tạm hiểu theo em là mỗi sắc ai cũng có thể nhìn thấy. Đây chính là cơ sở để nhận biết rằng, người dùng có cố tình trả lời sai hay là đang treo máy khi sử dụng ứng ColorBlind

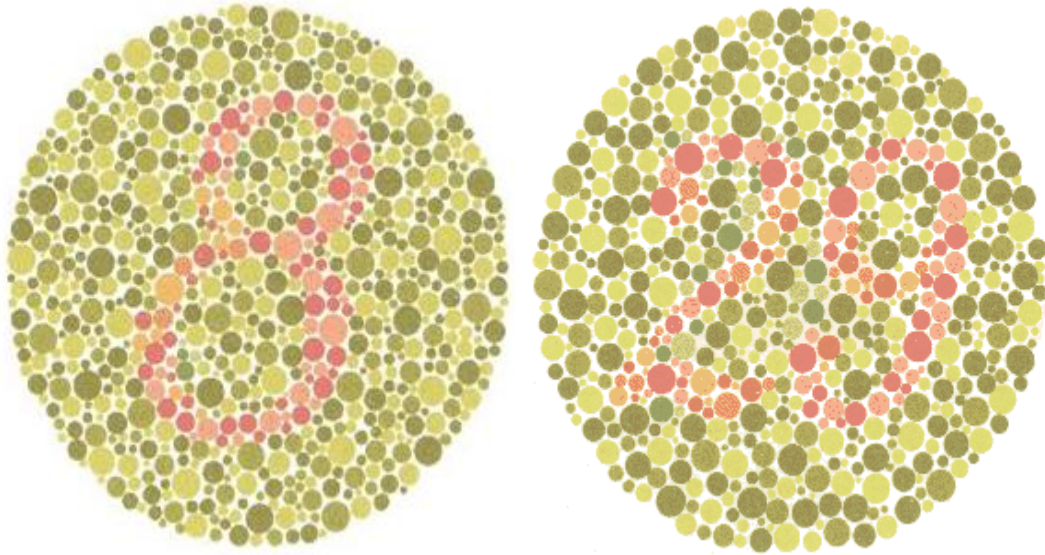




### Hình 3.8 Hình màu sắc chuẩn.

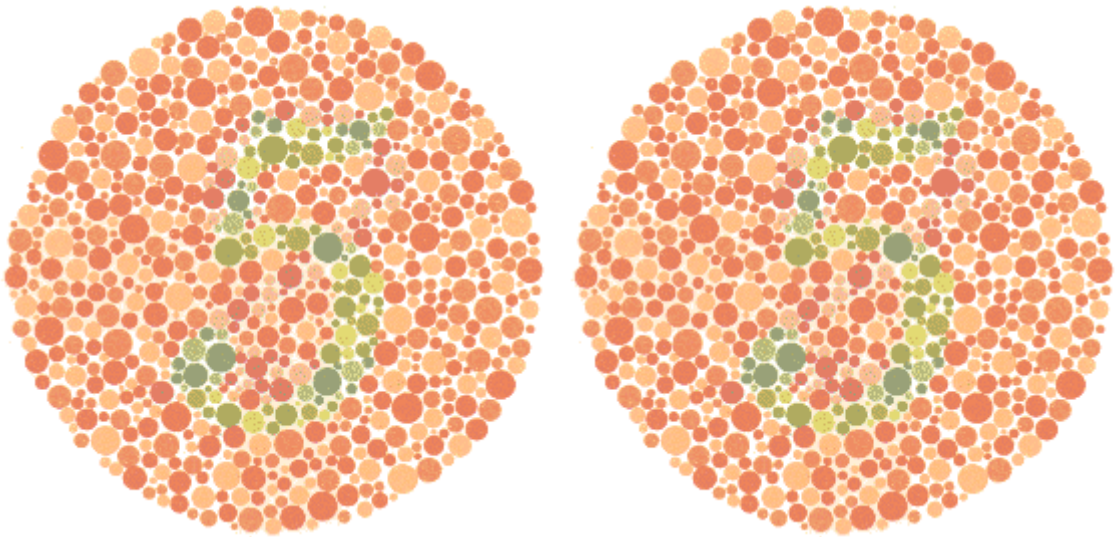
b. Màu sắc nhận diện mù màu đỏ :

Dưới đây là dữ liệu để nhận biết dấu hiệu của mù màu đỏ.



Hình 3.9 Hình nhận diện mù màu đỏ.

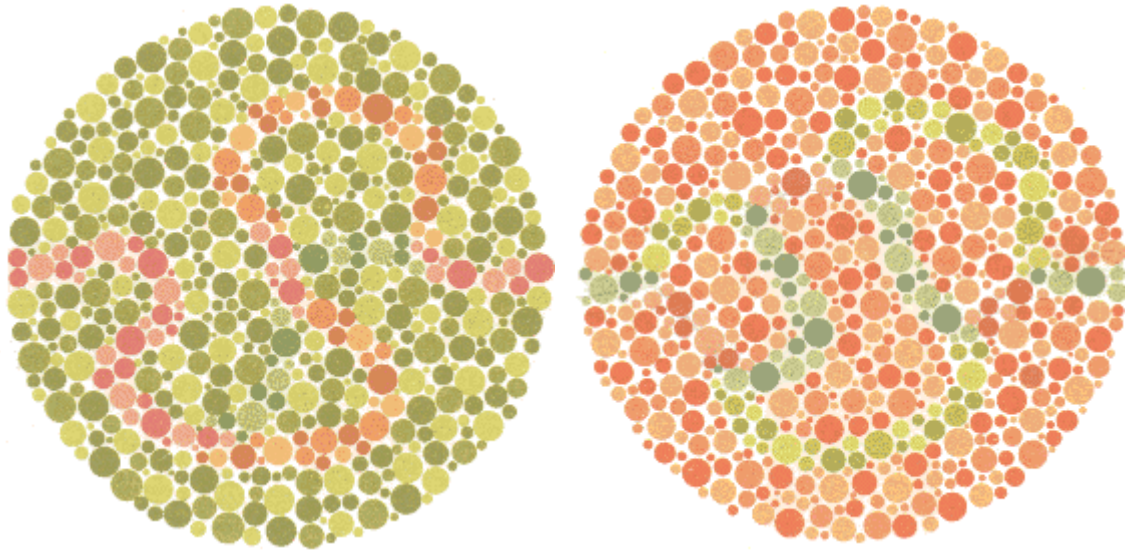
c. Màu sắc nhận diện mù màu đỏ - xanh lục :



Hình 3.10 Hình nhận diện mù màu đỏ - xanh lục.

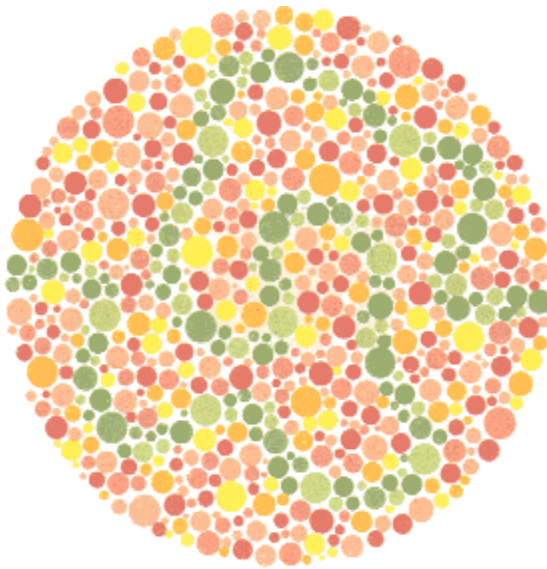
d. Màu sắc nhận diện mù đỏ - xanh lam :





**Hình 3.11 Hình nhận diện mù màu đỏ - xanh lam.**

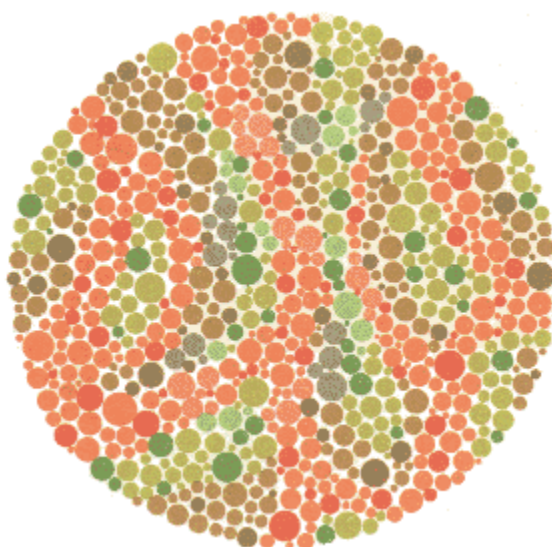
e. Màu sắc nhận diện mù màu xanh lá cây :



**Hình 3.12 Hình nhận diện mù màu xanh lá cây.**

f. Màu sắc nhận diện mù màu :

Dưới đây là đại diện hình ảnh để nhận diện sự mù màu cơ bản, khó xác định được mù màu nào rõ ràng.



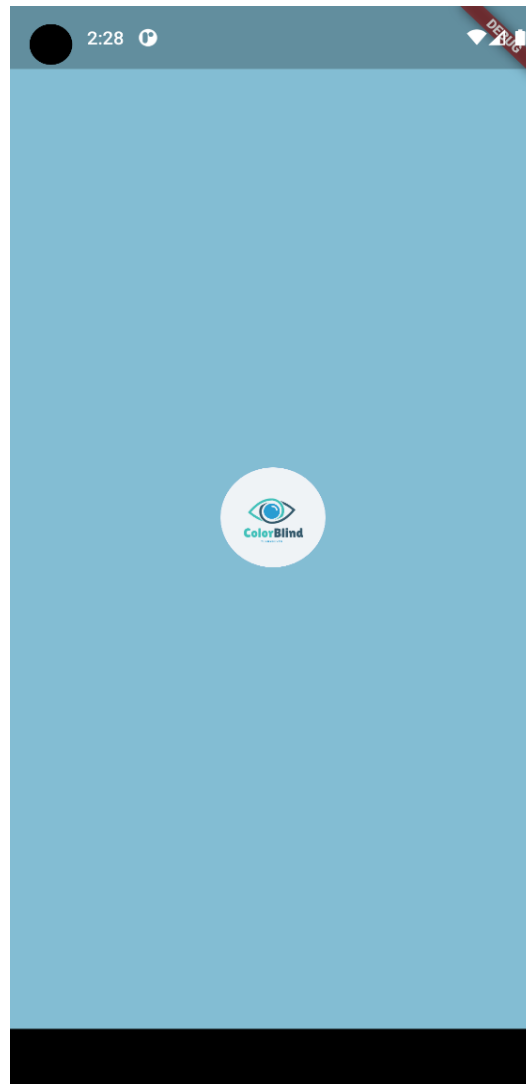
**Hình 3.13 Hình nhận diện mù màu.**

## CHƯƠNG 4 . GIAO DIỆN ỨNG DỤNG, PHÂN RÃ CHỨC NĂNG

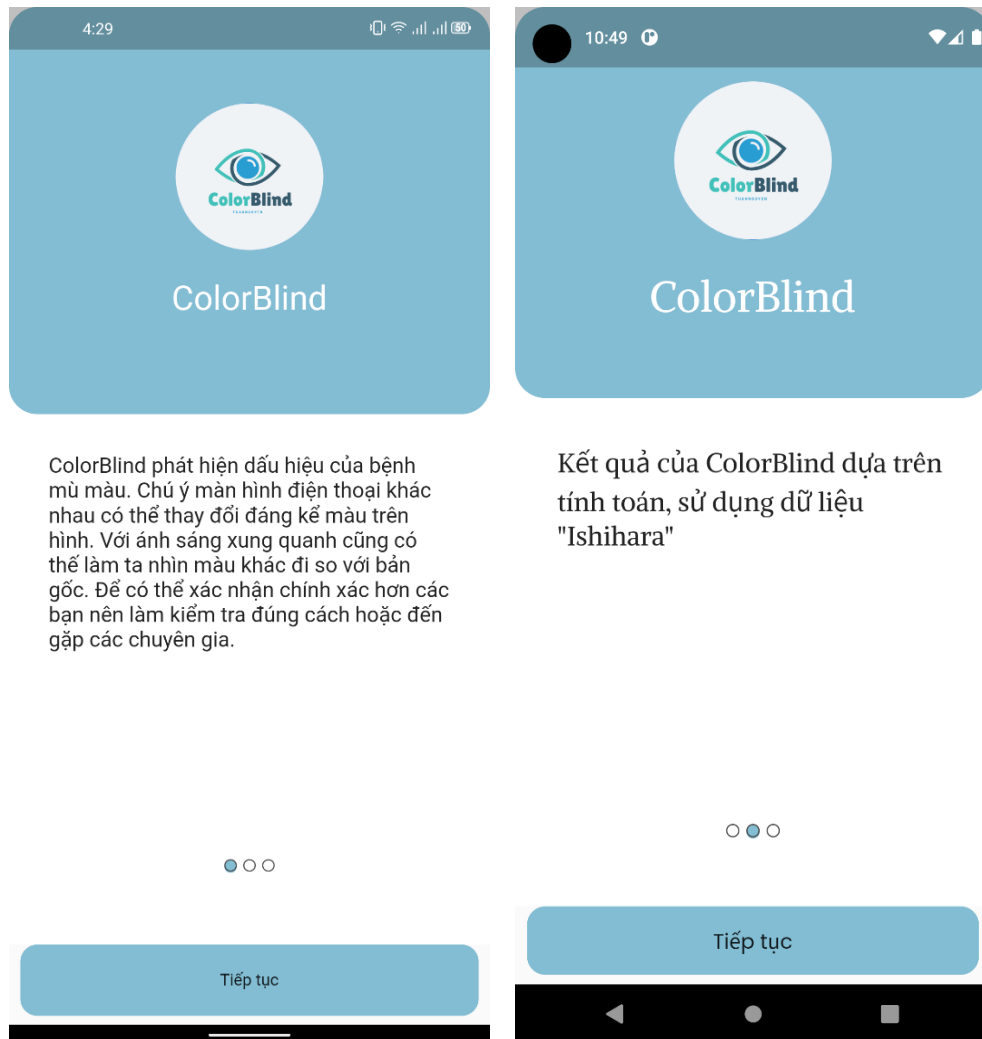
### 4.1 Giao diện ứng dụng :

#### 4.1.1 Giao diện Splash, giới thiệu ứng dụng

Sau khi cài đặt ứng dụng thành công, đây là màn hình đầu tiên khi mở ứng dụng. Người dùng đọc những thông tin giới thiệu về ứng dụng, cảnh báo về độ chính xác của ứng dụng cũng như giảm tránh trường hợp người dùng khi sử dụng tính năng nhận diện hình ảnh, để đạt được hiệu quả cao nhất, cũng như hiển thị bộ dữ liệu sử dụng là cơ sở đánh giá cho kết quả của người dùng tăng sự tin cậy, và nếu như có sai sót về kết quả hoặc xảy ra các trường hợp ngoại lệ sẽ liên hệ qua mail cá nhân.



Hình 4.1 Màn hình Splash.



**Hình 4.2** Màn hình giới thiệu ColorBlind

#### 4.1.2 Giao diện trang chủ :

Màn hình thông tin về ứng dụng sau khi người đọc xong. Các tính năng chính của ứng dụng xuất hiện ở màn hình chính.

Phía trên cùng sẽ xuất hiện những tấm hình màu, kèm chữ về màu sắc đó. Giúp hỗ trợ người dùng xác định mình đang nhìn thấy màu đó chính xác hay không?

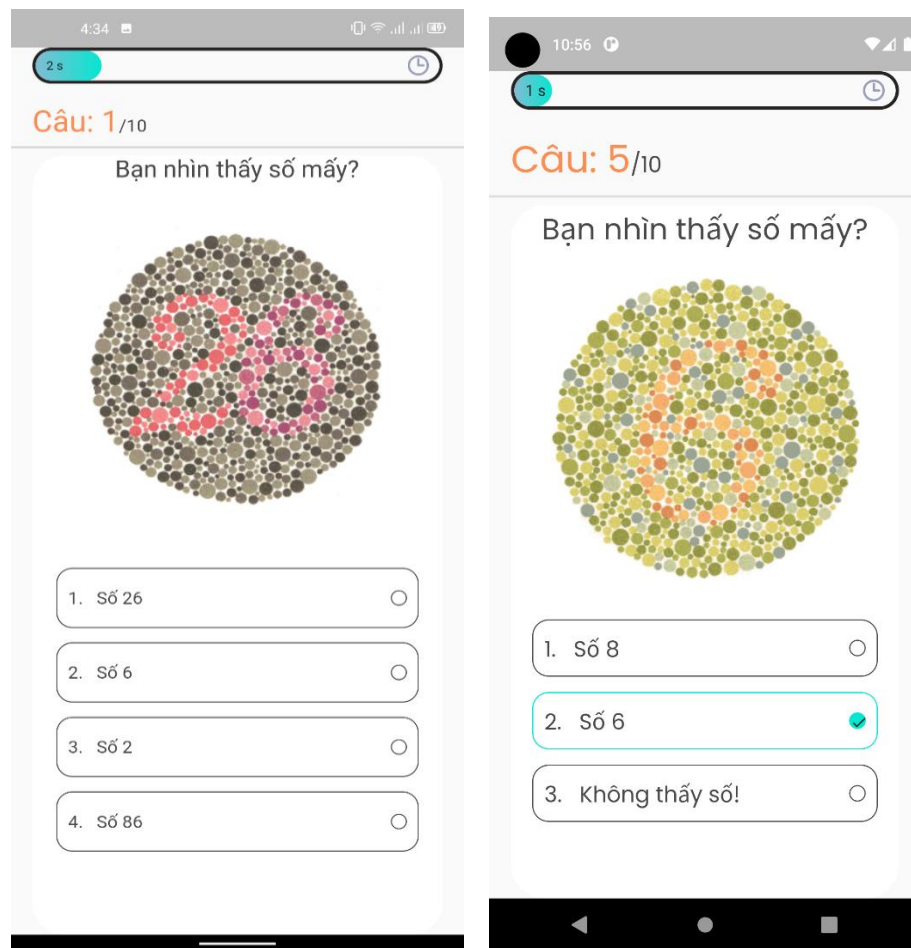


**Hình 4.2 Màn hình trang chủ ứng dụng.**

#### **4.1.3 Giao diện tính năng kiểm tra mù màu :**

Ứng dụng sẽ xuất hiện bộ câu hỏi bao gồm 10 câu hỏi. Sử dụng bộ dữ liệu Ishihara cho bộ câu hỏi trên.

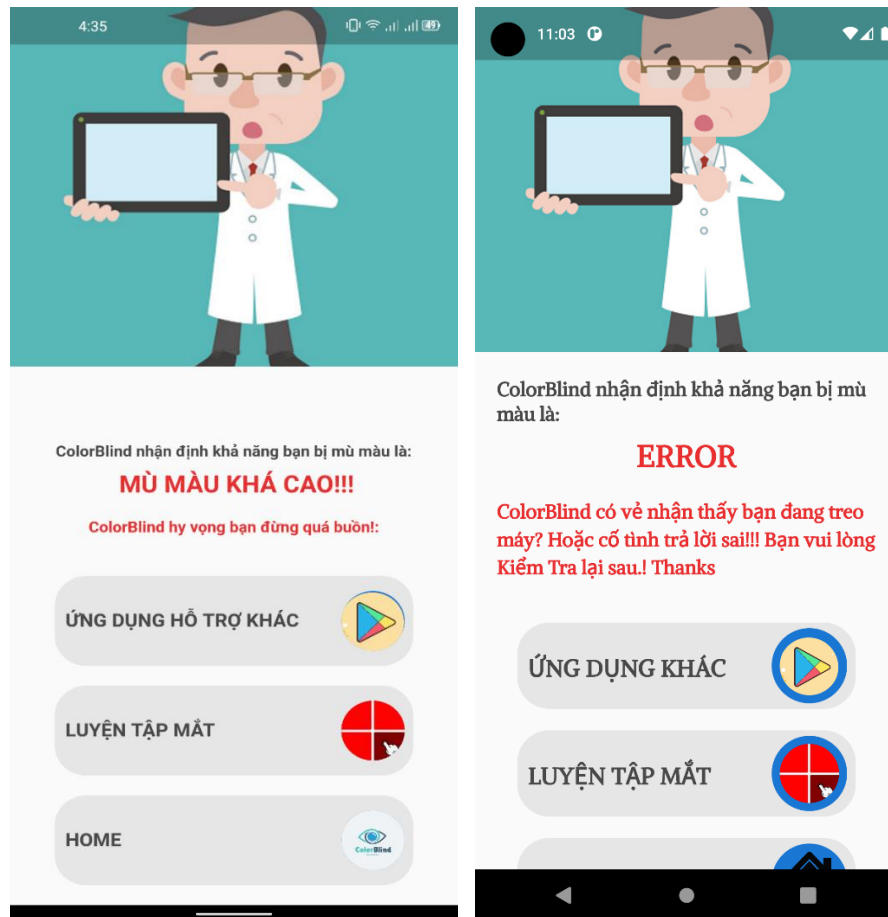
Mỗi câu hỏi sẽ 30 giây để chọn đáp án, nếu như đáp án sai sẽ trả về màu đỏ, cũng như trả về đáp án chính xác màu xanh.



**Hình 4.3 Giao diện tính năng kiểm tra mù màu.**

#### **4.1.4 Giao diện màn hình kết quả sau khi kiểm tra :**

Kết thúc bộ câu hỏi bao gồm 10 câu : bao gồm hình ảnh nhận diện về tất cả các phân loại về mù màu dựa vào đáp án người dùng trả lời đúng hay sai. Ứng dụng sẽ trả về kết quả khách quan cho người dùng kiểm thử. Ở đây em như đã nói ở phần 2 cơ sở lý thuyết, thì em sẽ sử dụng kết quả của người chọn đúng hay sai, để trả về kết quả tính năng. Với bộ câu hỏi trên, đã sử dụng thêm câu hỏi bắt điều kiện để người dùng khi đang treo máy hoặc có tình trả lời sai, đó là 1 câu hỏi bất kỳ người dùng dù là mù màu hay không mù màu đều có thể nhìn thấy được kết quả.

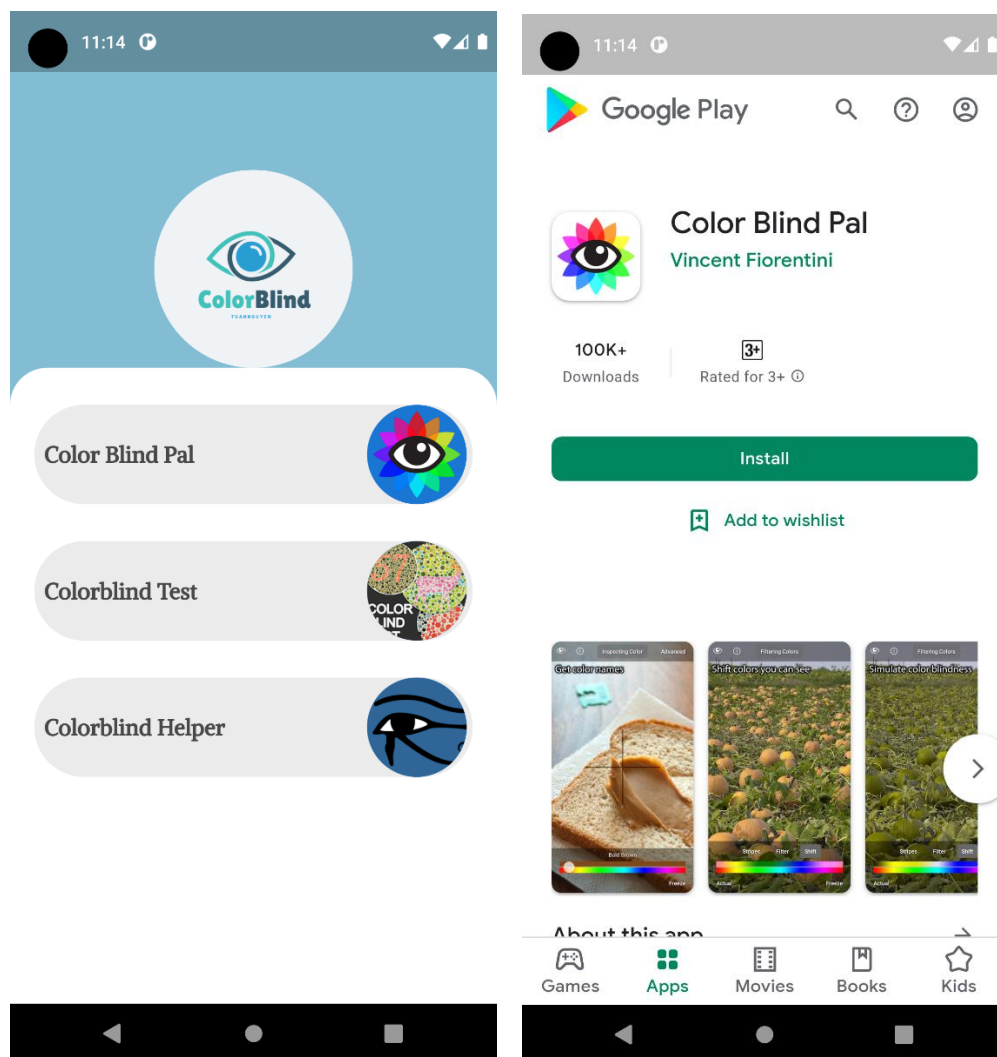


Hình 4.4 Màn hình trả kết quả.

#### 4.1.5 Giao diện tính năng đề xuất các ứng dụng hỗ trợ mắt :

Sau khi có kết quả ứng dụng ColorBlind sẽ thêm tính năng gợi ý những ứng dụng được đánh giá cao, cũng như lượt tải lớn trên CHPLAY. Mặc dù tính năng không quá lớn, nhưng xét về hướng đi ban đầu của đề tài nghiên là phi lợi nhuận, mang đến sắc màu thú vị hơn cho người dùng bị căn bệnh mù màu, thì tính năng đã mang đến cực kỳ hiệu quả đến người dùng.

Xét về mặt “marketing” tính năng này hoàn toàn có thể phát triển ở các ứng dụng để gợi ý người dùng tải về chuỗi sản phẩm công nghệ. Đây là tính em tâm đắc về hướng đi thực tế về phi lợi nhuận, cũng như về mặt marketing lợi nhuận.



**Hình 4.5** Màn hình giao diện gợi ý các ứng dụng hỗ trợ khác.

#### **4.1.6 Giao diện tính năng nhận diện màu sắc :**

Khi người dùng chọn tính năng nhận diện màu sắc, ứng dụng sẽ cho phép người dùng, chụp hình, cũng như chọn hình ảnh trong điện thoại để nhận diện màu sắc.

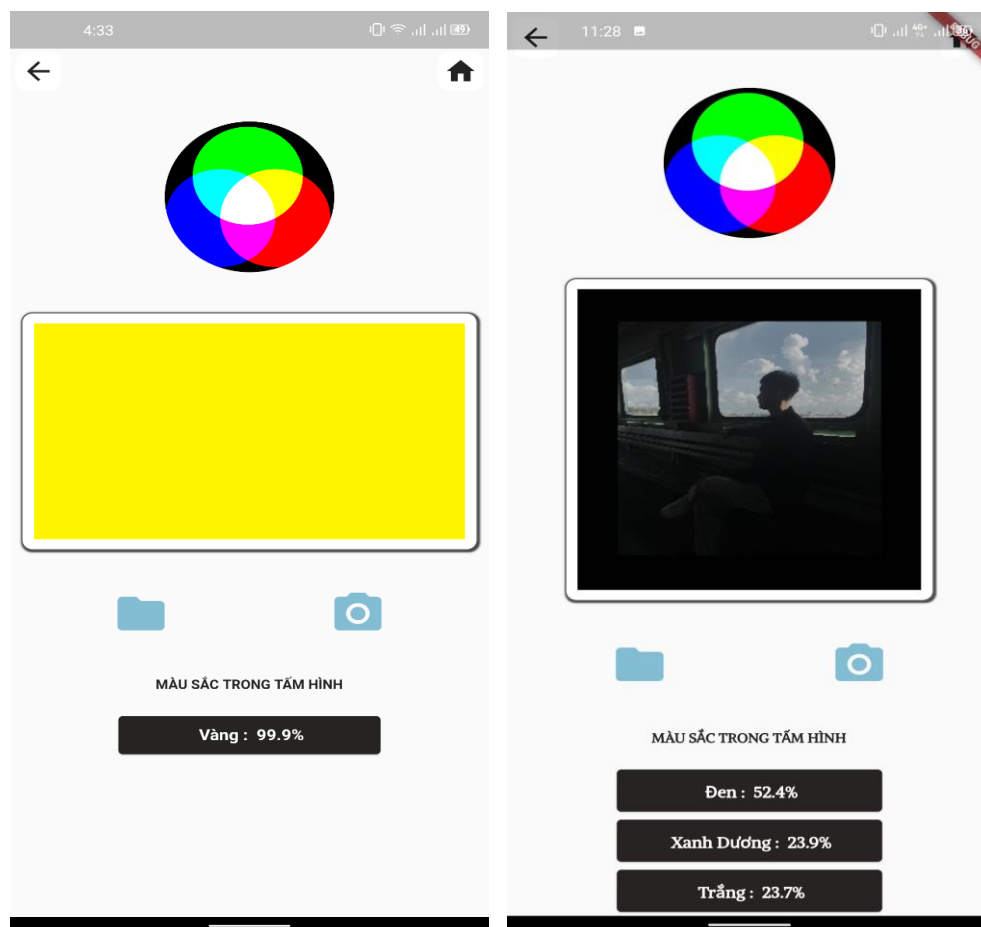
Ở đây, ứng dụng chỉ nhận diện với độ chính xác về phân tích mức độ màu sắc nào nhiều nhất theo tỷ lệ % với tỷ lệ chính xác 70-80%, số lượng màu sắc nhận diện với những màu sắc căn bản nhất, cũng như những màu sắc khả năng bị mù là 7 màu: Đen, Xanh Dương, Xanh Lá Cây, Đỏ, Trắng, Vàng, Cam.

Sau khi nhận diện xong màu sắc, có thể chọn các tấm hình khác muốn nhận diện màu sắc. Cũng như trở về giao diện màn hình chính. Với Icon “Home” góc phải màn hình.





**Hình 4.6** Giao diện tính năng nhận diện màu sắc.



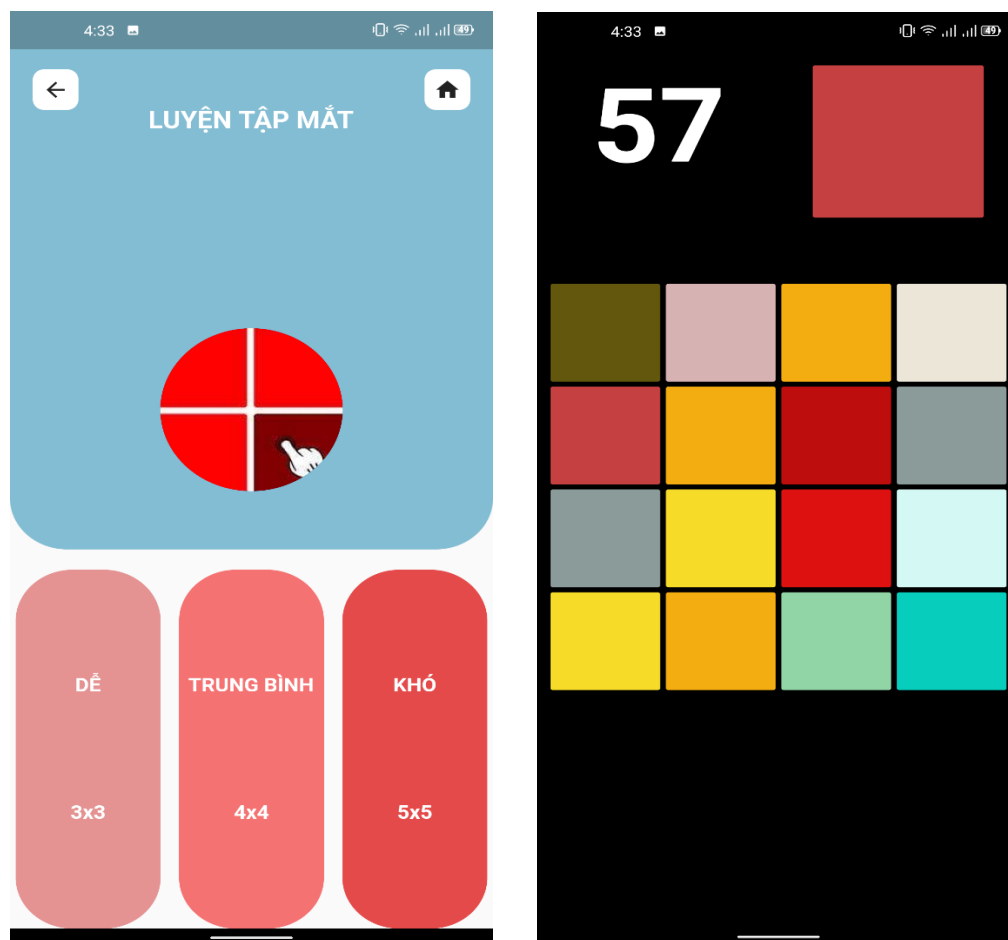
**Hình 4.7 Màn hình kết quả tính năng nhận diện màu sắc.**

#### **4.1.7 Giao diện tính năng luyện tập mắt :**

Khi người dùng chọn tính năng luyện tập mắt. Ứng dụng có 3 mức độ để người dùng trải nghiệm.

Với 60 giây, để người dùng chọn màu sắc giống với màu sắc của ứng dụng đưa ra. Sau khi kết thúc 60 giây, ứng dụng sẽ trả về kết quả những đáp án chính xác với số điểm tương ứng.

Số lượng màu sắc cho tính năng là đủ số lượng màu khác nhau.



**Hình 4.8** Màn hình giao diện tính năng luyện tập mắt.

Sau khi hoàn thành xong sau 60 giây, thì sẽ trả về kết quả của lựa chọn đúng trong 60 giây của trò chơi.



**Hình 4.9** Màn hình kết quả tính năng luyện mắt.

## **4.2 Phân tích thiết kế hệ thống :**

### **4.2.1 Sơ lược hệ thống :**

Con người sở hữu một đôi mắt được xếp vào hàng tốt nhất trong tự nhiên. Tuy nhiên, không phải ai cũng may mắn có được đôi mắt có thể nhìn được thế giới muôn màu, có những người chỉ thấy mọi vật xung quanh với màu sắc ảm đạm hơn rất nhiều – đó là do chứng bệnh “mù màu” gây nên.

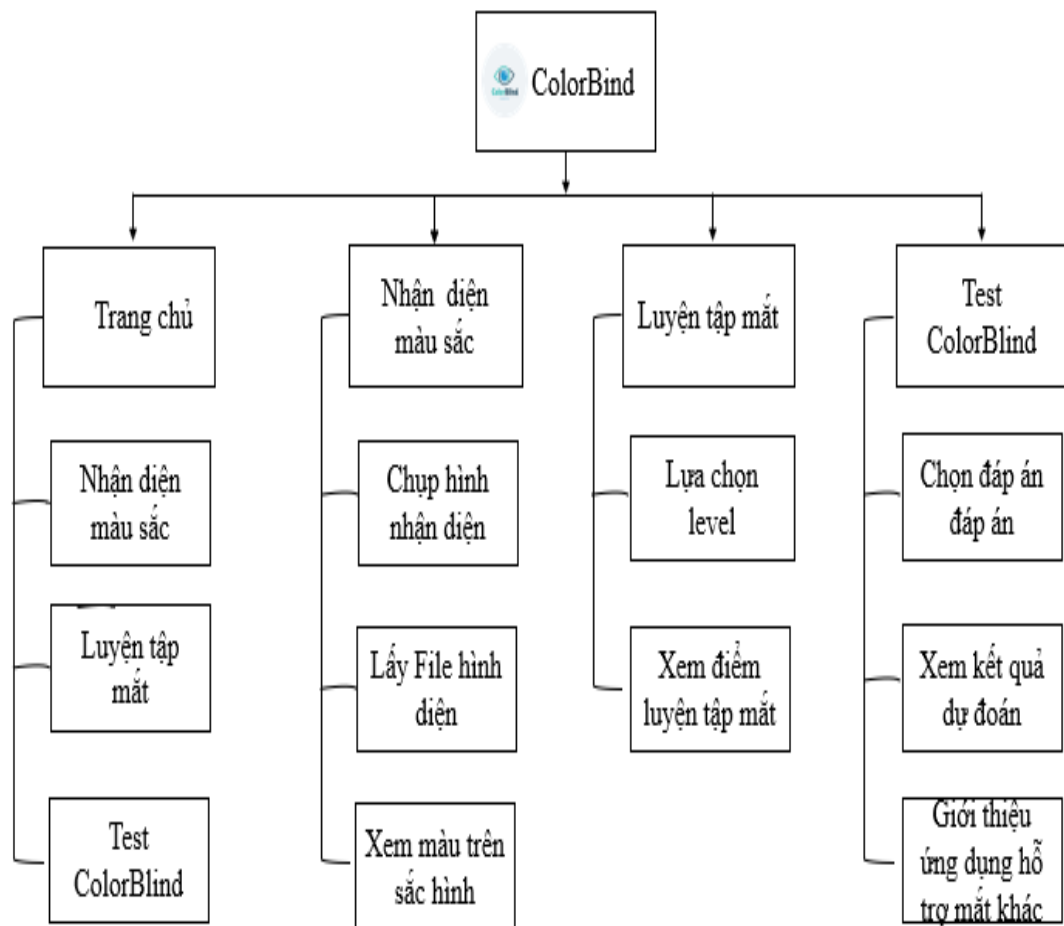
ColorBlind ra đời nhằm phục vụ cho việc đánh giá, kiểm tra, hỗ trợ ở mức tối thiểu nhất có thể cho người dùng có khả năng mắc căn bệnh “mù màu”, cũng như những người dùng muốn kiểm tra khả năng đôi mắt nhận diện màu sắc như thế nào với tính năng luyện mắt. ColorBlind hướng đến là nơi tin cậy, tin tưởng, hỗ trợ nhanh chóng nhận diện chính xác cho người dùng kiểm tra, đánh giá về mắt. ColorBlind hướng đến sự tương tác hiệu quả, tiết kiệm thời gian cho người dùng khi muốn kiểm tra đánh giá,

cũng như được hỗ trợ vấn đề về “mù màu” mong muốn trở thành một công cụ hỗ trợ cũng như là người bạn.

#### 4.2.2 Sơ đồ phân rã chức năng :

Với những yêu cầu và chức năng được chọn, ColorBlind sẽ được xây dựng và ứng dụng vào thực tế để kiểm thử và để hiểu rõ hơn ColorBlind làm gì thì chúng ta hãy thông qua vào một số chức năng.

ColorBlind có 3 mục chức năng chính được thể hiện với sơ đồ phân rã chức năng bên dưới:



**Hình 4. 10** Sơ đồ phân rã chức năng

## KẾT LUẬN VÀ KIẾN NGHỊ

### 1. Kết quả :

Trong quá trình nghiên cứu và hoàn thành đồ án tốt nghiệp với đề tài “*Xây dựng ứng dụng mobile kiểm tra, đánh giá, hỗ trợ người mù màu*”, em đã đạt được kết quả như sau:

#### 1.1 Kết quả đạt được :

- Đánh giá đạt được yêu cầu cơ bản về mức độ mù màu.
- Nhận diện những màu sắc chính, và những màu sắc dễ bị mù màu nhất.
- Tạo ra các mức độ luyện tập mắt cho người dùng trải nghiệm với 3 mức độ thích hợp.
- Hỗ trợ đề xuất các ứng dụng hỗ trợ mắt khác cho người dùng.

#### 1.2 Hạn chế còn thiếu sót :

Bên cạnh những khía cạnh đạt được, do thời gian thực hiện có hạn cùng với trình độ kiến thức còn nhiều hạn chế nên đã còn những thiếu sót một số chức năng như :

- Đánh giá chi tiết về căn bệnh mù màu nào.
- Tỷ lệ nhận diện màu còn khá thấp chỉ khoảng ~73%.
- Cài đặt cho ứng dụng trên CHPlay.
- Đánh giá ứng dụng.

### 2. Hướng phát triển :

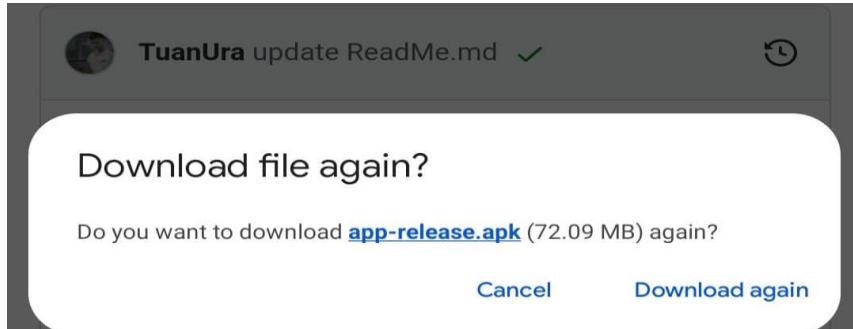
Trong tương lai, nếu có điều kiện đồ án của em sẽ được phát triển theo các hướng sau:

- Phát triển thêm tính năng khi nhận được kết quả, đề xuất hướng chữa trị.
- Nhận diện màu sắc với số lượng màu nhiều hơn.
- Cải thiện hiệu năng.
- Cải thiện giao diện người dùng đẹp hơn và thân thiện người dùng.
- Cải tiến chất lượng bộ huấn luyện phát hiện hình ảnh màu sắc.

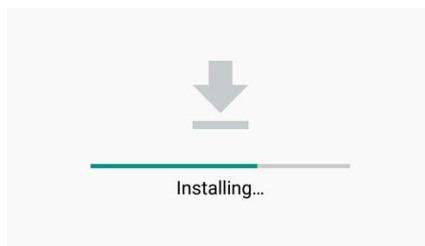
## PHỤ LỤC

### Hướng dẫn cài đặt

- Download file apk : app-release.apk trên repo Github :



- Cài đặt file apk.



## TÀI LIỆU THAM KHẢO

- [1] Trang chủ Flutter, <https://flutter.dev/>, truy cập ngày 21 tháng 04 năm 2022.
- [2] Thư viện Tensorflow, <https://www.tensorflow.org/>, truy cập ngày 23 tháng 05 năm 2022.
- [3] Thư viện quản lý trạng thái GetX, <https://pub.dev/packages/get>, truy cập ngày 23 tháng 05 năm 2022.
- [4] Jonathan Sande & Matt Galloway, “Dart Apprejtice”, 2021.
- [5] Marco L.Napoli, “Beginning Flutter: A Hands On Guide to App Development”, 2019.
- [6] National Eye Institute, “Color Blindness” <https://www.nei.nih.gov/learn-about-eye-health/eye-conditions-and-diseases/color-blindness>, truy cập ngày 13 tháng 5 năm 2022

Đường dẫn đến repository của đồ án này: [TuanUra/ColorBlind \(github.com\)](https://github.com/TuanUra/ColorBlind)