# CHƯƠNG 1 : TỔNG QUAN VỀ ĐỀ TÀI

## 1.1 Đặt vấn đề

Với xu hướng càng ngày càng phát triển hơn của xã hội trong công cuộc công nghiệp hoá và hiện đại hoá của đất nước thì nó kích thích cái nhu cầu của con người trở nên ngày càng tăng cao. Đặt biệt là cái nhu cầu về sử dụng công nghệ thông tin trong thời kì này nó thành xu hướng cả trong và ngoài nước. Thời đại 4.0 như bây giờ thì nhiều ngành nghề, dịch vụ như: ngành giáo dục, tài chính, điện tử,… đều được lồng ghép công nghệ để trở nên ngày càng tiện dụng. Các thiết bị thông minh đang dần trở nên cải tiến để phục vụ các yêu cầu thiết yếu cho con người. Qua đó, ta thấy khách hàng thì có nhu cầu tiêu dùng luôn rất đa dạng phong phú, vì vậy các lập trình viên cần có tư duy, tìm tòi học hỏi về các chuyên môn có liên quan để đáp ứng tiêu chí của người dùng. Như ở tình hình bùng dịch covid 19 giai đoạn trước thì lĩnh vực giáo dục việc học online để trở nên không bị gián đoạn, hay các nhà văn, nhân viên văn phòng, nhà thơ,… Họ có nhu cầu lưu trữ những tài liệu, thông tin đã viết ra giấy lên máy tính để lưu trữ dưới dạng văn bản word. Nhận ra đây là một nhu cầu đang được tìm kím rất cao nên nhóm em quyết định làm đề tài “ Xây dụng ứng dụng chuyển đổi văn bản viết tay sang word trên thiết bị di động”. Vừa có thể kiểm tra, cũng cố kiến thức trong những thời gian được học tập tại trường, vừa ứng dụng được Machine Learning vào các vấn đề thực tiễn đang nghiên cứu hiện nay.

## 1.2 Mục tiêu của đề tài

* Xây dụng hay thiết kế mô hình , thuât toán có độ chính xác cao.
* Thiết kế ra một ứng dụng có chức năng chuyển đổi văn bản viết tay sang phần mềm word trên thiết bị martphone.
* Đáp ứng yêu cầu sử dụng của con người , tiết kiệm được nhiều thời gian.
* Tìm hiểu các phần mềm , các ngôn ngữ lập trình , thư viện cần dùng để thực hiện.
* Phân tích ra được thiết kế hệ thống phần mềm.
* Đề xuất thêm những tính năng mới , tích hợp vào ứng dụng.
* Xây dựng chương trình , cài đặt nó, chạy thử nghiệm xong đưa ra đánh giá kết quả.
* Đề xuất thêm những tính năng mới , tích hợp vào ứng dụng.

## 1.3 Nhiệm vụ đề tài

* Sử dụng Flutter để khởi tạo giao diện cho người dùng cho ứng dụng.
* Sử dụng cơ sở dữ liệu Firebase.
* Xử lý nhận dạng chữ viết tay bằng Api của Google.

## 1.4 Đối tượng và phạm vi nghiên cứu

* Đối tượng:
* Nhu cầu thiết yếu của người dùng đối với ứng dụng đang làm nghiên cứu.
* Các phần mềm đã được bán ra trên thị trường hiện nay.
* Phạm vi nghiên cứu:
* Các phương pháp nhận diện chữ viết, trích xuất , xử lý hình ảnh.
* Giải thuật huấn luyện và nhận dạng dữ liệu.
* Hệ cơ sở dữ liệu sẵn có trong python.
* Kiến thức về lập trình python, xử lý ảnh.

- Tìm hiểu các thư viện trong Open CV , thuật toán CNN, KNN trong python.

## 1.5 Giới thiệu nhận dạng chữ viết tay

Nhận dạng chữ viết tay tiếng anh là Handwriting Recognition \_ HWR hay nhận dạng văn bản viết tay tiếng anh là Handwriten Text Recognition \_ HTR : là thiết bị máy tính có khả năng nhận diện và phân tích dữ liệu những chữ viết bằng tay từ giấy, ảnh và các thiết bị khác. Hình ảnh các văn bản viết bằng tay có thể lấy bằng cách đem một mảnh giấy có chữ viết đi quét quang học ( nhận dạng ký tự quang học) hoặc nhận dạng kí tự từ thông minh”. Ứng dụng hai công nghệ số tiểu biểu là trí tuệ nhân tạo( Artifical Intelligent) và máy học (Meachine Learning).

Nhận dạng chữ viết bằng tay thì nó vẫn là thử thách lớn cho các nhà nghiên cứu mảng này. Công trình này nó không thể nào làm trọn vẹn được vì nó phụ thuộc vào quá nhiều yếu tố mà chỉ cần thay đổi một chút có thể ra kết quả khác như là mức độ ánh sáng, sức khoẻ của người viết hay tinh thần của họ và đa dạng cách để viết.

### 1.5.1 Thời kì 1: từ năm 1905-1985

* Vào năm 1905, khi đang phát triễn ra một thiết bị giúp đỡ người mù thì ông Tyuring đã suy nghĩ đến nhận dạng chữ viết.
* Các sản phẩm được tạo ra từ năm 1949, khi máy tính lần đầu tiên trên thế giới nó giới thiệu về cái chức năng mà nhập liệu và lưu trữ dữ liệu hai chiều bằng cây viết trên thiết bị có gán cảm ứng.
* Công nghệ này vừa ra mắt giúp nhà khoa học nghiên cứu và thực nghiệm trên các mô hình toán nhận diện chữ viết bằng tay.
* Mô hình nhận diện kí tự được đưa ra từ năm 1952 do ông tên M.Sheppard được kêu là GISMO, nó là một con robot biết đọc viết.
* Năm 1955, J.Rainbow lần đầu tiên phát minh ra máy nhận dạng chữ viết mà nó để đọc các chữ viết in hoa nhưng lại rất chậm.
* Năm 1966, công ty IBM đã đem cái hệ thống nhận dạng chữ viết này lên thị trường để thương mại hoá nó.

### 1.5.2 Thời kì 2: từ năm 1985-nay

* Giai đoạn này phần chú trọng các các hệ thống dành cho nhận diện thời gian thực.
* Các kĩ thuật phương pháp luận kết hợp với nhận dạng trong lĩnh vực máy học.
* Một số ứng dụng máy hiệu quả như “mạng nơ ron, SVM tiếng anh là (Support Vector Machine) và quy trình xử lý các loại ngôn ngữ,...”

## 1.6 Ý nghĩa thực tiễn của đề tài

Vận dụng Meachine Learning (ML) trong các vấn đề thực tiễn đang nghiên cứu hiện nay. Cho phép sao chép, lưu trữ một đoạn văn bản bất kỳ trên sách, báo, giấy tờ, tạp chí… để sử dụng hoặc chỉnh sửa bằng phần mềm word trên máy tính hoặc smartphone . Tối ưu hoá về lợi ích củng như tiết kiệm thời gian trong công việc của con người. Đảm bảo khả năng vận hành ứng dụng “đáp ứng nhu cầu khách hàng trong cuộc sống”. Giúp con người tiếp cận hơn đối với các công nghệ mới ngày càng hiện đại trong tương lai. Đáp ứng khả năng nghiên cứu cơ bản giành cho học sinh, sinh viên đem lại ý nghĩa giải quyết vấn đề thực tế trong cuộc sống

# CHƯƠNG 2 CƠ SỞ LÝ THUYẾT

## 2.1 Giới thiệu Flutter

Flutter là một framework ứng dụng trong phát triển di động được triển khai bởi Google, ngôn ngữ lập trình sử dụng chính là Dart. Dart : được tạo ra từ năm 2011, hiện này được nhiều người sử dụng phổ biến, được thiết kế để kết hợp các tính năng của các ngôn ngữ hiện đại như JavaScript, C ++ và Java”.

Các thành phần của Flutter gồm có:

Widget: Widget là khối cơ bản của giao diện dành cho người dùng trong Flutter. Widget là một đối tượng Dart và được thiết kế để xây dựng các giao diện cho người dùng, các phần tử UI như nút, văn bản, hình ảnh, phông chữ, màu sắc và các thành phần khác.

Stateful và Stateless Widget: Có hai loại widget chính trong Flutter: Stateful và Stateless Widget. Stateless Widget là các widget không thay đổi sau khi được tạo ra và không có trạng thái (state) bên trong. Trong khi đó, Stateful Widget là các widget có thể biến đổi trạng thái (state) bên trong.

Flutter Architecture: Flutter cung cấp một kiến trúc gọi là Flutter Architecture, trong đó có ba cấu thành chính: “View, Model và Controller”. View dùng để hiển thị giao diện cho người dùng, Model là đại diện cho dữ liệu của phần mềm, và Controller là thành phần dùng quản lý luồng dữ liệu giữa View và Model.

Asynchronous Programming: Flutter ứng dụng hỗ trợ lập trình không đồng bộ (asynchronous programming) để đa luồng trong ứng dụng. Flutter sử dụng các khái niệm như Future, async và await để thực hiện lập trình không đồng bộ.

Dart Libraries: Dart có một vài thư viện chuẩn đi kèm với nó để hỗ trợ lập trình, bao gồm thư viện đối tượng (Object library), thư viện bất đồng bộ (Async library), thư viện dữ liệu (Collection library) và thư viện số học (Math library).

Tóm lại, Flutter nó là một framework sử dụng ngôn ngữ chính là Dart để phát triễn ra ứng dụng điện thoại di động. với các thành phần chính gồm các Widget, kiến trúc Flutter, lập trình bất đồng bộ và thư viện chuẩn.

### 2.1.1 Widget trong flutter

**Widget** là một khái niệm cơ bản trong Flutter, được sử dụng để xây dựng giao diện người dùng và tương tác với người dùng. Widget trong Flutter có hai loại: Stateless Widget và Stateful Widget.

Graphical user interface, website

Description automatically generated

*Hình 2.1.1: widgets của Flutter*

**Stateless Widget:** Là Widget không có trạng thái (state), được sử dụng khi giao diện người dùng trạng thái quá trình chạy ứng dụng. Mỗi lần Widget được tái tạo, nó sẽ trả về cùng một kết quả. Stateless Widget có phương thức build() để giao diện cơ bản cho người dùng.

**Stateful Widget**: Là Widget có trạng thái (state), được sử dụng khi giao diện người dùng cần thay đổi trong quá trình chạy ứng dụng. Mỗi lần trạng thái của Widget thay đổi, Flutter sẽ gọi phương thức build() để cập nhật giao diện mới.

Các Widget trong Flutter được chia thành hai loại: Basic Widgets và Composite Widgets.

**Basic Widgets**: Các Widget cơ bản được áp dụng để thực hiện giao diện người dùng, bao gồm:

**Text**: Hiển thị văn bản

**Image**: Hiển thị hình ảnh

**Icon**: Hiển thị biểu tượng

**Container**: Chứa các Widget con và cung cấp các thuộc tính layout như độ rộng, chiều cao và màu sắc.

**Padding**: Thêm khoảng trắng cho Widget con

**Row và Column**: Sắp xếp các Widget con theo chiều ngang hoặc dọc

**ListView**: Trình bày danh sách các Widget con

**GridView**: Hiển thị lưới các Widget con

**Stack**: Chứa các Widget con và sắp xếp chúng lên nhau

**Composite Widgets**: Các Widget được tạo bằng cách kết hợp nhiều Widget cơ bản để cấu tạo ra những thành phần giao diện phức tạp hơn, bao gồm:

**Form**: Tạo một Form để nhập liệu từ người dùng

**AppBar**: Thanh tiêu đề của ứng dụng

**TabBar**: Thanh chuyển đổi giữa các màn hình

**AlertDialog**: Hộp thoại cảnh báo

**Card**: Hiển thị nội dung trong một thẻ

Widget là một cái phần quan trọng của Flutter và được áp dụng rộng rãi để thiết kế cái giao diện cơ bản cho người dùng sài trong ứng dụng di động và web. Việc hiểu rõ về Widget sẽ giúp bạn tạo dựng ra các ứng dụng chất lượng cao và hiệu quả hơn.

### 2.1.2 State trong flutter

Trong Flutter, State là một khái niệm khá quan trọng được áp dụng để mà quản lý trạng thái của các Widget và cập nhật giao diện người dùng sài tương ứng. Mỗi Widget trong Flutter đều có một State object tương ứng để lưu trữ thông tin các thay đổi của Widget đó.

A picture containing diagram

Description automatically generated

*Hình 2.1.2: Giới thiệu về State trong Flutter*

Flutter sử dụng mô hình quản lý trạng thái của Widget rất hiệu quả. Khi trạng thái của một Widget thay đổi, Flutter sẽ xây dựng lại Widget đó và các Widget liên quan đến nó để cập nhật giao diện người dùng.

Có hai loại State trong Flutter:

**Stateless Widget**: Là Widget không thể biến đổi hình thái của chính nó. Trạng thái của Stateless Widget xác định dựa trên các thuộc tính được cung cấp trong lúc khởi tạo.

**Stateful Widget**: Là Widget có thể thay đổi hình thái của chính nó. Stateful Widget bao gồm hai phần: State object và Widget object. Widget object sẽ không thay đổi, nhưng State object có thể thay đổi trong quá trình chạy. Khi State object thay đổi, Flutter sẽ tái tạo lại Widget object và cập nhật giao diện người dùng.

Graphical user interface, diagram

Description automatically generated

*Hình 2.1.3: Giới thiệu stateful widget*

Các bước để tạo Stateful Widget trong Flutter:

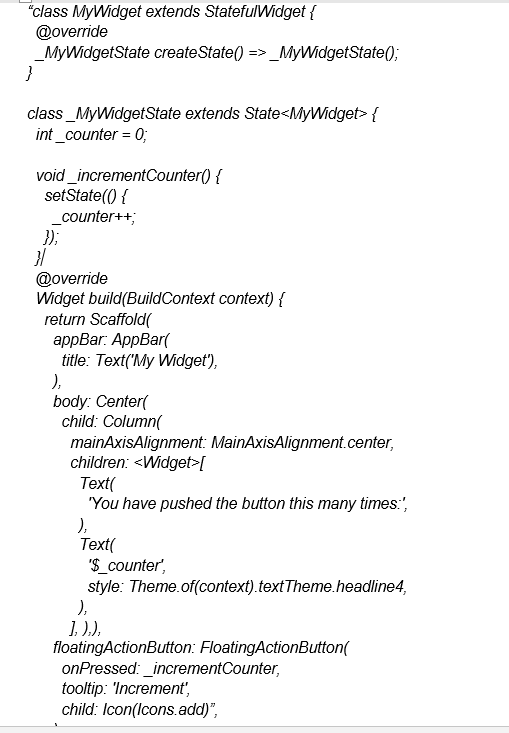
Tạo một class cho Stateful Widget và kế thừa từ StatefulWidget.

Tạo một class cho State object và kế thừa từ State<StatefulWidget>.

Override phương thức createState() của StatefulWidget để tạo ra một instance của State object.

Trong State object, sử dụng phương thức setState() để cập nhật trạng thái của Widget và giao diện người dùng.

Ví dụ về việc sử dụng State trong Flutter:



Trong ví dụ trên, \_MyWidgetState là State object cho MyWidget. Biến \_counter được sử dụng để lưu trữ trạng thái của Widget và được cập nhật bằng phương thức `\_increment

### 2.1.3 Event và Handler trong flutter

Trong Flutter, Event và Handler được áp dụng trong tương tác với người dùng và giải quyết các sự kiện. Event là một tiến trình và được thực hiện bởi người dùng trên thiết bị, chẳng hạn như nhấn một nút, cuộn chuột hoặc chạm vào màn hình. Handler là một cách thức được gọi để xử lý Event đó.

Graphical user interface, text, application

Description automatically generated

*Hình 2.1.4: Giới thiệu về event và handers*

Flutter cung cấp một số Widget có sẵn để xử lý Event, bao gồm:

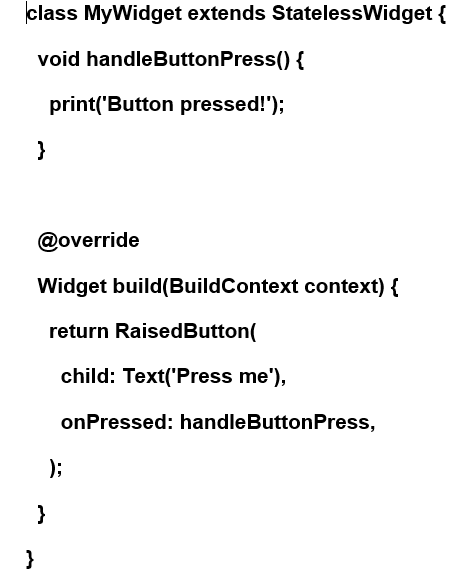
**GestureDetector**: Widget này cho phép xác định các cái sự kiện thao tác của người dùng như bấm, vuốt, kéo, chạm, v.v. bằng cách sử dụng các thuộc tính như onTap, onDoubleTap, onLongPress, v.v.

**InkWell**: Widget này là một cái phiên bản khá đơn giản của GestureDetector, có chức năng giống nhau nhưng được áp dụng để tạo những cái hiệu ứng hình thức khi người dùng chạm vào Widget.

**RaisedButton, FlatButton, IconButton**: Đây là các Widget button được sử dụng để giải quyết sự kiện nhấn nút. Các tiến trình được xử lý thông qua thuộc tính onPressed.

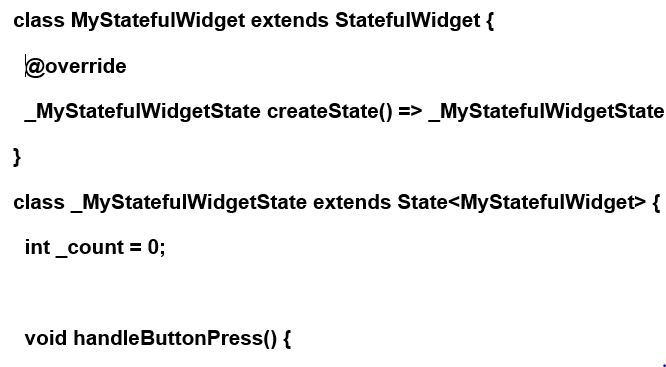
**Slider**: Widget này người sử dụng được phép chọn giá trị từ một phạm vi bằng cách kéo thanh trượt. Sự kiện thay đổi giá trị được xử lý thông qua thuộc tính onChanged.

Các Handler có thể được áp dụng để xử lý các quá trình tương tác. Chúng ta có thể nói như là một phương thức để giải quyết sự kiện và gọi nó thông qua thuộc tính mà nó tương ứng của Widget. Ví dụ:



Trong ví dụ trên, khi người dùng nhấn nút, phương thức handleButtonPress() sẽ được gọi và in ra màn hình 'Button pressed!'.

Các Handler cũng có thể được định nghĩa bên trong một class State để xử lý các sự kiện cho Widget Stateful. Chúng ta có thể sử dụng phương thức setState() để cập nhật trạng thái của Widget và tái tạo giao diện người dùng tương ứng với trạng thái mới. Ví dụ:



### 2.1.4 Threading trong flutter

Trong Flutter, Threading được áp dụng để mà đảm bảo rằng các tác vụ mà nó phức tạp có thể được thực hiện mà nó không làm đóng băng giao diện người dùng. Threading trong Flutter có thể được triển khai bằng cách áp dụng các khái niệm như Isolate, Future, Stream và Event Loop.

Graphical user interface

Description automatically generated

*Hình 2.1.5: Giới thiệu về threading trong Flutter*

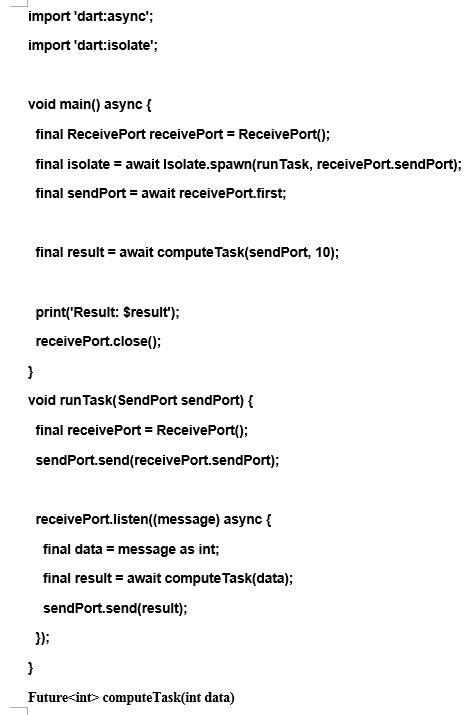
**Isolate**: Là một đơn vị xử lý độc lập trong ứng dụng Flutter. Isolate có thể được sử dụng để thực hiện các tác vụ phức tạp như tính toán, I/O và mạng mà không làm đóng băng giao diện người dùng. Isolate làm việc như các tiến trình độc lập, có thể chạy trên một luồng riêng biệt và truyền thông qua các tin nhắn.

**Future**: Là một cái đối tượng trong Flutter để giải quyết các tác vụ bất đồng bộ. Future đại diện cho một giá trị sẽ được tính toán trong tương lai. Khi giá trị được tính toán, Future sẽ gửi một thông báo về kết quả hoặc lỗi. Future được áp dụng để mà thực hiện các tác vụ bất đồng bộ trên Isolate khác và tránh làm đóng băng giao diện người dùng.

**Stream**: Là một luồng dữ liệu trong Flutter, cho phép gửi và nhận dữ liệu liên tục. Stream được sử dụng để thực hiện các tác vụ bất đồng bộ như tải trang web hoặc tải các tệp dữ liệu lớn.

**Event Loop**: Là một quy trình xử lý sự kiện trong Flutter, cho phép xử lý các sự kiện người dùng và các sự kiện hệ thống khác. Event Loop được áp dụng để bảo đảm rằng giao diện người dùng sài luôn phản hồi và tránh các trường hợp đóng băng.

Ví dụ, đoạn mã sau sử dụng Isolate và Future để thực hiện một tác vụ tính toán phức tạp:



### 2.1.5 Routing trong flutter

Routing trong Flutter là cách để điều hướng giữa các màn hình và các thành phần trong ứng dụng. Routing được sử dụng để quản lý các thành phần của ứng dụng, bao gồm các trang, các nút điều hướng, các mục menu và các hộp thoại.

Trong Flutter, một màn hình được gọi là một Route và được định nghĩa bởi một Widget. Một Navigator là một widget cha của tất cả các Route, quản lý quá trình điều hướng giữa các Route. Khi bạn điều hướng đến một Route mới, Navigator sẽ tạo một hiệu ứng chuyển động giữa các Route.

Graphical user interface

Description automatically generated with low confidence

*Hình 2.1.6: Navigation trong Flutter*

Có hai cách chính để điều hướng giữa các Route trong Flutter:

Sử dụng Navigator.push(): Phương thức này cho phép bạn chuyển từ màn hình hiện tại đến màn hình mới bằng cách thêm một Route vào ngăn xếp của Navigator. Để sử dụng phương thức này, bạn cần cung cấp một BuildContext và một Route.

Navigator.push(context, MaterialPageRoute(builder: (context) => NewScreen())); Navigator.push(context, MaterialPageRoute(builder: (context) => NewScreen()));

Sử dụng Navigator.pop(): Phương thức này cho phép bạn quay lại màn hình trước đó bằng cách loại bỏ Route hiện tại khỏi ngăn xếp của Navigator.

Navigator.pop(context);

Route, bạn có thể sử dụng các tham số của Route. Để nhận tham số từ Route mới, bạn có thể sử dụng ModalRoute.of(context).settings.arguments.

Ví dụ:



Ở DetailScreen, bạn có thể lấy item bằng cách sử dụng: **final Item item = ModalRoute.of(context).settings.arguments;**

Có thể hình thành các Route bằng cách kế thừa từ lớp PageRoute. PageRoute cung cấp các phương thức để tạo hiệu ứng chuyển động tùy chỉnh giữa các Route. Bạn có thể kế thừa từ PageRoute và ghi đè phương thức buildTransitions để thêm hiệu ứng chuyển động tùy chỉnh.

## 2.2 Giới thiệu phần mềm lập trình Flutter (Visual Studio)

## 

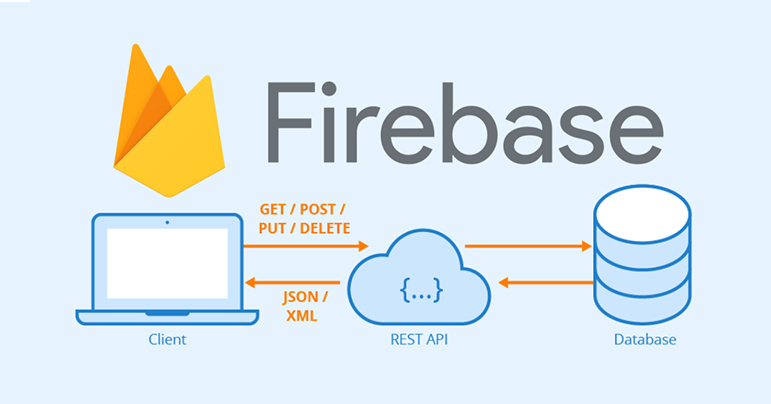
*Hình 2.2: Giao diện visual studio*

Visual Studio là “phần mềm được sử dụng rộng rãi trong phát triễn phần mềm, gồm nhiều ngôn ngữ có thể viết được như: C, Flutter, Dart, HTML,CSS, Java,…” Nó cung cấp nhiều tiện ích và các công cụ hỗ trợ tốt cho việc viết, sửa lỗi, quản lý mã nguồn, đồng thời cho phép người dùng triễn khai trực tiếp từ IDE. Nó cũng hổ trợ cho nhiều nền tảng khác nhau của các phiên bản Flutter.

## 2.3 Giới thiệu Firebase

### 2.3.1 Firebase là gì?

Firebase nó là một dịch vụ hoạt động chính trên nền tảng đám mây hay còn được gọi là (cloud) và nó cung cấp cơ sở dữ liệu cho người sử dụng. Kèm theo nó còn được coi là hệ thống máy chủ vô cùng mạnh mẽ của Google. Hệ thống có những chức năng chính sẽ giúp cho người sử dụng thông qua những cách đơn giản hoá các cái thao tác trên những thành phần cơ sở dữ liệu để mà lập trình ra ứng dụng.



*Hình 2.2.1: Hoạt động của Firebase?*

Quá trình hình thành & phát triển các cái nền tảng của Firebase:

Vào năm 2010, Firebase được cho ra đời trên thị trường với tên gọi cũ là Envolve bởi James Tamplin và Andrew Lee. Nó chuyên về cung cấp những cái API cần thiết để tích hợp thêm cái tính năng chat vào bên trong các trang web và được coi là một cái nền tảng khá là đơn giản.

Tháng 4/2021, Firebase với chức năng là Realtime được ra đời với nền móng là một công ty cách biệt hoàn toàn là Backend-as-a-Service.

Firebase được ông lớn Google mua lại và phát triển để nó thành một cái ứng dụng đa năng được tích hợp trên những nền tảng androi và các cái nền tảng web bây giờ vào năm 2014.

Chức năng của firebase hoạt động được như thế nào?

Chức năng hoạt động của công cụ Firebase này là gì?

Bắt đầu từ khi khi ông lớn Google mua lại chính thức firebase và phát triển nó thì đến hiện tại firebase có một vài hoạt động khá nổi bật như sau:

* Firebase\_Authentication

Hoạt động được xem là nổi bật nhất của firebase chắc có lẽ là xây dựng thành công ứng dụng xác nhận thông tin người dùng thông qua các nền tảng mạng xã hội như: facebook, zalo, email,Twitter,... Ngoài ra thì Firebase\_Authentication còn có thể hôx trợ trực tiếp các ứng dụng xác định những thông tin nặc danh. Hoạt động xác thực này của Firebase mang lại cho người sử dụng cách bảo mật thông tin cá nhân trở nên an toàn hơn. Điều này cũng bảo đảm cho thông tin mỗi người dùng ứng dụng và tài khoản của họ không bị đánh cắp”.



*Hình 2.2.2: Firebase Hosting*

* Firebase\_Hosting
* Phương thức hoạt động thứ hai là Firebase\_Hosting. Nó là một phương thức mà được hệ thống mạng CDN phân phối dựa trên các cái tiêu chuẩn của những cái công nghệ bảo mật SSI .
* CDN được gọi tắt là (Content Delivery Network) : thể hiện là một cái mạng lưới các máy chủ có khả năng lưu trữ lại những cái bản sao từ các nội dung tĩnh. Những cái nội dung tĩnh này được phân phối một cách trực tiếp đến các máy chủ PoP khác và nằm ở phía trong website. Hệ thống mạng lưới của các máy chủ từ CDN thì được lắp đặt ở tất cả khu vực trên thế giới. Nguồn dữ liệu mà được gửi cho người dùng cuối sẽ xuất phát từ các máy chủ Pop Points of Presence.
* Firebase\_Realtime\_Database

Firebase\_Realtime\_Database là “một JSON được đồng bộ thời gian đến với tất cả các client kết nối”. Các lập trình viên cần phải “đăng ký tài khoản ở trên Firebase” để có thể sử dụng nó. Dữ liệu bên trong database sẽ “cập nhật một cách liên tục và tự động khi phát triển ứng dụng”. Những dữ liệu này “sẽ được truyền tải đi thông qua những kết nối SSl có 2048 bit” sau khi nó đã được cập nhật.

Ưu nhược điểm của Firebase



*Hình 2.2.3: Firebase có thể làm gì?*

### 2.3.2 Ưu điểm

Không tự nhiên mà nền tảng của firebase lại được áp dụng ở khắp mọi nơi trên thế giới. Theo đó nó sở hữu cho mình những điều riêng biệt bởi các ưu điểm nổi bật không phải hệ thống nào cũng có. Điển hình:

Người dùng có thể sử dụng nó dễ dàng chỉ cần thông qua việc đăng kí một tài khoản bằng một số liên kết mạng. Thay vào đó trong quá trình triễn khai cái ứng dụng cở bản ta có thể sử dụng nền tảng này tuỳ ý trong quá trình phát triển .

Tốc độ phát triển firebase khá nhanh: Nó hỗ trợ người lập trình trong việc phát triển ứng dụng trở nên nhanh hơn. Qua đó thì các người lập trình viên sẽ phần nào giảm được các áp lực về thời gian trong lúc phát triển mô hình hay quảng cáo nó trên thị trường cho ứng dụng.

Nền tảng được phát triển bởi google:  Như đã được biết thì firebase chính thức được Google mua lại và đem nó phát triển thành một phần dịch vụ. Vì vậy, khi nó xuất hiện giúp chúng ta khai thác một cách triệt để toàn bộ các năng lực tiềm ẩn hay các dịch vụ mới mà Google hiện đang có hoạt động.

Chú tâm hơn trong phần giao diện người dùng: Firebase là hệ thống mà có một kho dữ liệu backend mẫu phải nói là vô cùng đa dạng. Từ đó các lập trình viên có thể sử dụng tuỳ ý để phát triển giao diện người dùng dễ dàng thông qua các mẫu sẵn có.

Không có máy chủ nhất định giành cho firebase app: Điều này sẽ làm cho firebase nó tối ưu hơn trong phần hiệu năng cũng như hiệu suất làm việc được đẩy mạnh dựa vào các cụm database được mở rộng ra.

Học máy: Như đã biết firebase hỗ trợ trong việc tạo ứng dụng thì học máy sẽ giúp các người lập trình phát triển ứng dụng cơ bản tốt hơn bình thường.

Phần lưu lượng truy cập lớn: Firebase\_App cho chúng ta đơn giản hoá việc thành lập các chỉ mục. Và các ứng dụng được tích hợp firebase app sẽ giúp nó tăng cao thứ hạng trên bảng xếp hạng ứng dụng của Google. Nhờ thế mà lượng truy cập của nó sẽ được đẩy mạnh lên một cách trông thấy.



*Hình 2.2.4: Chức năng bảo mật*

Kiểm soát lỗi: Thì là một công cụ giúp bạn khắc phục các thành phần lỗi của ứng dụng một cách hoàn hảo. Vậy nên khi mà áp dụng nó ta không sợ mình bỏ sót các lỗi cơ bản khi lập trình.

Khả năng sao lưu tốt: Nguồn dung lượng luôn cung cấp sẵn có để có thể sao lưu dữ liệu liên tục không sợ trễ. Điều này cũng giúp cho việc bảo vệ thông tin bảo mật được an toàn hơn bởi tính nhanh chóng.

### 2.3.3 Nhược điểm:

Firebase nó không phải là một ứng dụng mã nguồn mở vì vậy khi phát triển ứng dụng sẽ không tối ưu hoá cho các nhà phát triển lựa chọn. Chúng ta cũng không thể nào mà tự ý thay thế hay là sửa đổi được mã nguồn của firebase.

Việc truy cập mã nguồn khó khăn giữa các người dùng: Với các ứng dụng lớn đang phát triển mạnh mẽ thì việc chuyển đổi nhà cung cấp nó không phải là một việc dễ dàng. Để mà chuyển đổi như thế buộc toàn bộ dữ liệuBackend phải được tái thiết lập từ đầu.

Hạn chế hoạt động trên một vài quốc gia: Firebase có thể coi là một Subdomain trên Google. Phải nói rằng Website chính thức của nó còn bị hạn chế trên nhiều lãnh thổ trên thế giới trong đó còn có cả Trung Quốc.

Hoạt động của firebase hoàn toàn trên các CSDL và nói không với SQL: Người dùng không thể giải quyết các data một cách dễ dàng và nhanh chóng. Các JSON được áp dụng hầu hết trên tất cả hệ thống và hầu như không có SQL. Chính vì vậy để có thể chuyển dời cơ sở dữ liệu sẽ vô cùng rắc rối.

Google Cloud nơi mà firebase hoạt động chính: ở đây chính là nói firebase là một phần của google bao gồm các cơ sở hạ tầng của nó đều hoạt động phụ thuộc vào ứng dụng của Google Cloud. Và ứng dụng không thể nào khởi động trên bất cứ một nơi cung cấp đám mây nào khác .



*Hình 2.2.5: Các tiện ích, hướng dẫn*

Firebase và nhược điểm của nó:

Tham vấn còn khá chậm: và đúng vậy trong tuỳ trường hợp sẽ sử dụng cơ sở dữ liệu khác nhau nên nó sẽ không đúng trong tuỳ hoàn cảnhvà không có ngoại lệ nào bao gồm cả cloud firestore. Nên là việc này sẽ gây mất thời gian hơn khi chúng ta muốn sử dụng ứng dụng.

Vì là mã nguồn đóng nên các dịch vụ sẽ không hoàn toàn miễn phí. Và người dùng không thể sử dụng các dịch vụ trên các gói Spark mà chỉ có thể sài những chức năng có sẵn trên gói Blaze của đám mây.

Giá thành về dịch vụ của nó khá cao: Việc giá thành cao sẽ dẫn đến việc cung cấp thêm nhiều chức năng vô cùng hữu ích cho người dùng tuy vậy thì nó vẫn quá cao so với nhà cung cấp Iass thuần tuý trước đó.

Chưa có hợp đồng cho doanh nghiệp: Như đã nói nó không có các hợp đồng giành cho doanh nghiệp hay các cái lựa chọn về Dedicated Servers. Và chỉ có thể sử dụng cấu trúc Severless trong phát triển ứng dụng nhưng nó lại không linh hoạt lắm.

API GraphQL chưa được cung cấp:  API GraphQL thì chưa được cung cấp và như là quá trình để mà thiết lập các tiêu chuẩn . Tuy nhiên sẽ có các biện pháp khác để thay thế cho nó nhưng đa phần thì Rest được sử dụng nhiều nhất cho lựa chọn mặc định.

Firebase và các ứng dụng phổ biến của nó:

Một vấn đề phải nói là khá được chú ý đến gần đây của các người dùng về firebase. Rằng hiện tại thì nó được đem đi phát triển các ứng dụng nổi trội như thế nào?



*Hình 2.2.6: Ứng dụng firebase phổ biến*

* Một vài cái ứng dụng của Firebase đang hot:

- Alibaba

- eBay Motors

- Todoist

- La figaro

- Thời tiết New York

Các dịch vụ cơ bản mà Firebase cung cấp

Những cái dịch vụ nổi bật mà được mang đến cho chúng ta thì nó xuất phát từ các cái sự hữu ích mà Google Firebase đem lại. Đa phần các dịch vụ này đều hoạt động dựa trên 2 nhóm công cụ sau:

* Đầu tiên là Firebase Develop & Test Your App

Thì công cụ này là công cụ nhà phát triển dùng để kiểm thử những cái ứng dụng được đã và đang thiết kế. Trong đó thì có thể nói đến một vài cái như sau:



*Hình 2.2.7: Nhóm công cụ firebase*

Realtime Database: Một dịch vụ giành cho việc lưu trữ hay đồng bộ các dữ liệu của người sử dụng theo thời gian thực. Qua đó ta có thể dễ dàng lưu trữ hay lấy dữ liệu về từ các máy chủ của hệ thống một cách đơn giản.

Crashlytics: Một dịch vụ nổi bật dùng để lưu trữ cũng như theo dỗi các thành phần lỗi của ứng dụng. Khi thu thập các thông tin này một cách triệt để thì nó sẽ được sắp xếp lại một cách hợp lý.

Cloud Firestore: giúp người dùng lưu trữ hay đồng bộ dữ liệu các cái thiết bị máy khác nhau với quy mô toàn cầu.

Authentication: dùng để quản lý người dùng khá an toàn và tương đối đơn giản”. Đồng thời nó còn cung cấp cho ta các phương pháp để xác thực người dùng thông qua các dịch vụ của Google hay tài khoản mạng xã hội.

Cloud Functions: không cần đến các máy chủ riêng biệt mà vẫn có thể mở rộng các cái tuỳ chọn cho ứng dụng một cách hoàn hảo thông qua các cái mã phù trợ tuỳ chỉnh.

Cloud Storage: được xem là nơi dùng để lưu trữ và chia sẽ các nội dung mà chúng ta sáng tạo ra với một bộ nhớ khủng và tiết kiệm chi phí.

Hosting: đối với các trang web hiện đại thì nó sẽ giúp đơn giản đi các việc lưu trữ web theo từng yêu cầu sử dụng bằng các công cụ thực hiện và có tính nâng cao web.

Test Lab for Android: giúp khởi chạy thử các cái ứng dụng cũng như tuỳ chỉnh theo nhu cầu trên các máy ảo hay thiết bị mô phỏng vật lý được cung cấp bởi google .

Performance Monitoring: Có khả năng kiểm tra các vấn đề đối với hiệu suất của ứng dụng. Nó thì vô cùng có ích đối với những người lập trình vì giúp quá trình triễn khai xây dựng ứng dụng trở nên đơn giản và tối ưu hơn bao giờ hết.

Tiếp theo là Firebase Grow & Engage Your Audience:



*Hình 2.2.8: Google Analytics for Firebase*

Google Analytics: Cho ra những phân tích về các cái thuộc tính và hành vi của người sử dụng trong các bảng điều khiển. Nó tiếp nhận các thông tin về thời gian thực trong khi sử dụng thông qua những báo cáo cho đến khi dữ liệu thô được gửi đi đến cho Google BigQuery.

Cloud Messaging Dùng để đưa các thông báo và các tin đến cho các người dùng trên những nền tảng như: iOS, Android và Web. Nó có thể gửi đi hàng tỉ tin cũng như thông báo trên các ứng dụng mỗi ngày.

Predictions: dự đoán hành vi của người dùng qua đó tạo nhóm người dùng năng động. Từ đó ta nhắm tới họ như là đối tượng trong các sản phẩm khác như là các cấu hình từ xa hay thông báo.

Firebase Dynamic Links:  Sử dụng liên kết động để có thể cung cấp các trãi nghiệm cho người sử dụng. Và nó cũng hộ trợ trên những nền tảng web của điện thoại di động để đẩy mạnh cơ cấu chuyển đổi dữ liệu ứng dụng gốc,…

Remote Config: thiết lập được các tuỳ chỉnh của ứng dụng khi nó hiển thị. Nó được dùng để nhằm thay đổi cho giao diện của ứng dụng hay là cơ cấu thêm các tính năng mới cho ứng dụng,...

Invites: Người dùng có thể chia sẽ các đoạn mã giới thiệu hay nội dung các bảng tin yêu thích của mình trên các nền tảng mạng thông qua giao thức sms hay email.

App Indexing: được tích hợp trên google dùng để làm công cụ tìm kiếm để mà thu hút những người đã từng sử dụng dễ dàng tiếp cận các ứng dụng trước đây.

AdMob: Là các cái trang quảng cáo nhằm chia sẽ đến cho người dùng để giúp thu hút người dùng đến các ứng dụng và được các người lập trình viên dùng nó để kiếm thêm lợi nhuận thông qua việc chạy quảng cáo trên các nền tảng mạng.

AdWords giúp chúng ta có thêm có thêm thông tin qua việc chạy quảng cáo tự động để thu hút được người dùng và hiển thị tự động trên mục tìm kiếm. Chúng ta có thể dùng nó cho việc tối ưu hiệu sức chiến lược và mục tiêu cần đạt được của quảng cáo.

Giá thành từng dịch vụ trong firebase



*Hình 2.2.9: Dịch vụ firebase*

Khả năng lưu trữ mà được cung cấp miễn phí cho người dùng trong gói Spark hiện nay là 10GB. Nhưng đối với gói Blaze khi người dùng muốn sử dụng được nó phải trả phí để mua nó với giá cả linh hoạt theo từng nhu cầu dùng. Thì nó sẽ bao gồm các tính năng của gói Spark và bổ sung thêm các tính năng mới vào để mà cải tiến ứng dụng. Gói Blaze hiện tại thì giá thành nó bán khoản 0,027USD/GB khoảng trống này có thể dùng để lưu trữ và tương đương như từng GB dung lượng bình thường.

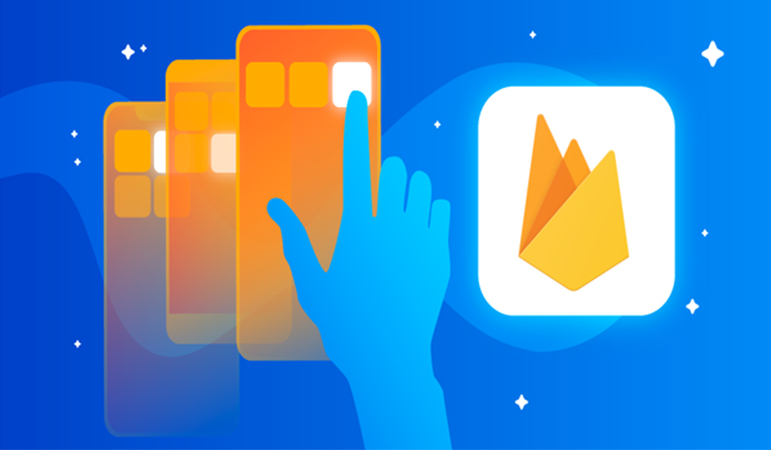
Các giải pháp thay thế khác cho firenbase

Thì firebase là một nền tảng vô cùng tuyệt vời dành cho phát triển ứng dụng mà không thể phũ nhận. Để sử dụng nó hiệu quả thì chúng ta có thể tìm đọc các hướng dẫn sử dụng hay các hướng dẫn dùng firebase cho hợp lý. Bên cạnh đó, chúng ta cũng còn một vài giải pháp khác thay thế để lựa chọn như:

Back4App: được xem là một nền tảng mã nguồn mở đem sử dụng khá là rộng rãi. Nó hoạt động y như ứng dụng Low-Code Backend chủ yếu là đẩy nhanh tốc độ phát triển khi triển khai ứng dụng.

Backendless: được xem là các công cụ dùng để quản lý và hệt như các công cụ phát triển ứng dụng vô cùng mạnh mẽ. Nó đem lại cho người dùng phong phú các tuỳ chọn về các máy chủ của hệ thống đám mây.

AWS Amplify: Nó đem lại cho ta nhiều tính năng vượt trội đa phần dùng để phát triển ứng dụng về AWS Full-stack. Ứng dụng này thì được ưu tiên trong các dự án giành riêng cho nền tảng Androi và cả Font-end.



*Hình 2.2.10: Giải pháp thay thế firebase*

## 2.4 Thuật toán MLKIT của Google

Google ML Kit là SDK dành cho thiết bị di động cung cấp khả năng máy học (ML) trên thiết bị dễ sử dụng cho Android và iOS. ML Kit đi kèm với các mô hình được xây dựng sẵn cho các trường hợp sử dụng khác nhau, bao gồm ghi nhãn hình ảnh, phát hiện ra các dạng khuôn mặt, xác nhận ra các từ trong văn bản, quét mã vạch,...

Dưới đây là một số thuật toán được tạo ra và sử dụng trực tiếp trong Google ML Kit:

Ghi nhãn hình ảnh: API gắn nhãn hình ảnh của ML Kit sử dụng mạng nơ-ron sâu để phân tích hình ảnh và nhận dạng các đối tượng bên trong chúng. Thuật toán sử dụng các dữ liệu có sẳn sau đó thêm vào mô hình để nhận ra các hình ảnh và cung cấp cho nó.

Phát hiện khuôn mặt: API phát hiện khuôn mặt sử dụng mô hình máy học để xác định khuôn mặt người trong luồng hình ảnh và video. Việc phân tích các pixel trong hình ảnh để xác định các đặc điểm khuôn mặt và khớp chúng với một tập hợp các mẫu đã biết để nhận dạng ra được khuôn mặt.

Cách nhận dạng văn bản: API nhận dạng văn bản sử dụng công nghệ nhận dạng ký tự quang học (OCR) để phát hiện và nhận dạng văn bản trong hình ảnh. Thuật toán nhận định được các ký tự và hình thức riêng lẻ trong hình ảnh và sử dụng học máy để khớp chúng với các ký tự và từ đã biết.

Quét mã vạch: API quét mã vạch sử dụng thị giác máy tính và máy học để phát hiện và đọc mã vạch trong thời gian thực. Thuật toán phân tích hình ảnh để xác định mã vạch riêng ra và từ đó giải mã phần tử chứa trong đó.

Đây chỉ là một vài ví dụ về các thuật toán được sử dụng trong Google ML Kit. Nền tảng này cung cấp một loạt các mô hình được xây dựng sẵn cho các trường hợp sử dụng khác nhau, cũng như các công cụ để các nhà phát triển xây dựng và đào tạo các mô hình tùy chỉnh của riêng họ.

### 2.4.1 Dữ liệu train của Google MLKIT

Google ML Kit cung cấp cả mô hình dựng sẵn cho các trường hợp sử dụng phổ biến cũng như các công cụ để nhà phát triển xây dựng và đào tạo các mô hình tùy chỉnh của riêng họ. Để đào tạo mô hình ML Kit tùy chỉnh, bạn sẽ cần thu thập và chuẩn bị một cách tốt nhất.

Tập dữ liệu được gắn nhãn được sử dụng để đào tạo mô hình học máy để nó có thể nhận dạng và phân loại chính xác các đầu vào tương tự trong tương lai. Tập dữ liệu phải bao gồm một tập hợp các đầu vào (ví dụ: hình ảnh, âm thanh, văn bản) cùng với các nhãn tương ứng của chúng (ví dụ: mèo, chó, chim). Bạn càng có nhiều ví dụ được gắn nhãn, mô hình của bạn sẽ càng chính xác.

Dưới đây là các bước chung để đào tạo mô hình tùy chỉnh bằng Google ML Kit:

1. Thu thập và gắn nhãn tập dữ liệu: Bạn sẽ cần thu thập một tập hợp dữ liệu và gắn nhãn thủ công cho từng ví dụ. Để nhận dạng hình ảnh, điều này có thể liên quan đến việc gắn nhãn các đối tượng trong hình ảnh. Để nhận dạng văn bản liên quan đến việc gắn nhãn các từ hoặc cụm từ riêng lẻ.
2. Chuẩn bị tập dữ liệu: Khi bạn đã gắn nhãn dữ liệu của mình, bạn sẽ cần xử lý trước dữ liệu đó để đảm bảo rằng nó ở định dạng mà ML Kit có thể sử dụng. Điều này chuyển đổi hình ảnh sang một định dạng cụ thể hoặc chuẩn hóa dữ liệu văn bản.
3. Hình thành ra mô hình: Bạn sẽ sử dụng tập dữ liệu được chọn để đào tạo mô hình học máy của mình. Điều này liên quan đến việc đưa dữ liệu được gắn nhãn vào thuật toán ML, thuật toán này sẽ học cách trong dữ liệu và tạo ra một mô hình có thể phân loại chính xác các ví dụ mới.
4. Đánh giá mô hình: Sau khi mô hình được đào tạo, bạn sẽ cần đánh giá độ chính xác của nó trên một tập dữ liệu riêng biệt gồm các ví dụ được gắn nhãn. Điều này sẽ giúp bạn xác định xem mô hình của bạn có đủ chính xác để hữu ích hay không.
5. Tích hợp mô hình: Cuối cùng, bạn sẽ cần tích hợp mô hình đã đào tạo vào ứng dụng của mình bằng cách sử dụng API ML Kit do Google cung cấp. Điều này sẽ cho phép bạn sử dụng mô hình cho các tác vụ như ghi nhãn hình ảnh hoặc nhận dạng văn bản.

### 2.4.2 Code train của Google MLKIT

Đào tạo một mô hình tùy chỉnh bằng Google ML Kit yêu cầu một năng lượng chuẩn bị và công việc đáng kể và mã chính xác được yêu cầu sẽ phụ thuộc vào trường hợp sử dụng cụ thể và dữ liệu đang được sử dụng. Tuy nhiên, đây là tổng quan về các bước cơ bản liên quan đến việc đào tạo mô hình tùy chỉnh với ML Kit, cùng với một số mã mẫu để gắn nhãn hình ảnh:

Tải dữ liệu được gắn nhãn của bạn từ tệp hoặc cơ sở dữ liệu

Danh sách<LabelledImage> trainingData = loadTrainingData();

Tạo tập dữ liệu ghi nhãn hình ảnh Firebase ML Kit

Tùy chọn FirebaseVisionImageLabelerOptions =

FirebaseVisionImageLabelerOptions.Builder() mới

.setConfidenceThreshold(0,7f)

.build();

FirebaseVisionImageLabeler labeler =

FirebaseVision.getInstance().getOnDeviceImageLabeler(tùy chọn);

for (LabelledImage image : trainingData) {

Tải hình ảnh từ tệp hoặc URL

FirebaseVisionImage firebaseImage = FirebaseVisionImage.fromFilePath(context, image.getPath());

Gắn nhãn hình ảnh và thêm nhãn vào dữ liệu đào tạo

Tác vụ<Danh sách<FirebaseVisionImageLabel>> tác vụ = labeler.processImage(firebaseImage);

task.addOnSuccessListener(labels -> {

image.setLabels (nhãn);

});

}

1. Chuẩn bị tập dữ liệu:

Chuyển đổi hình ảnh sang định dạng yêu cầu

for (LabelledImage image : trainingData) {

Bitmap bitmap = BitmapFactory.decodeFile (image.getPath ());

FirebaseVisionImage firebaseImage = FirebaseVisionImage.fromBitmap (bitmap);

image.setFirebaseImage (firebaseImage);

}

Chuẩn hóa nhãn

for (LabelledImage image : trainingData) {

List<String> labels = new ArrayList<>();

for (nhãn FirebaseVisionImageLabel: image.getLabels()) {

Chuỗi normalizedLabel = normalizeLabel(label.getText());

labels.add(normalizedLabel);

}

image.setLabels (nhãn);

}

1. Đào tạo mô hình:

Tạo một mô hình “TensorFlow và đào tạo nó bằng cách sử dụng dữ liệu” được gắn nhãn.

Mô hình TensorFlowModel = TensorFlowModel mới ();

model.train (trainingData);

1. Cách mô hình được đánh giá:

Danh sách<LabelledImage> testData = loadTestData();

for (LabelledImage image: testData) {

List<String> predictedLabels = model.predict(image.getFirebaseImage());

int numCorrect = 0;

for (String predictedLabel: predictedLabels) {

if (image.getLabels().contains(predictedLabel)) {

numCorrect++;

}

}

độ chính xác gấp đôi = (đôi) numCorrect / (đôi) predictedLabels.size();

Log.d(TAG, "Hình ảnh: " + image.getPath() + ", Độ chính xác: " + độ chính xác);

}

1. Tích hợp mô hình:

Sử dụng mô hình đã đào tạo để gắn nhãn hình ảnh trong ứng dụng của bạn

FirebaseVisionImageLabeler labeler =

FirebaseVision.getInstance().getOnDeviceImageLabeler(tùy chọn);

Bitmap bitmap = BitmapFactory.decodeFile (imagePath);

Hình ảnh FirebaseVisionImage = FirebaseVisionImage.fromBitmap (bitmap);

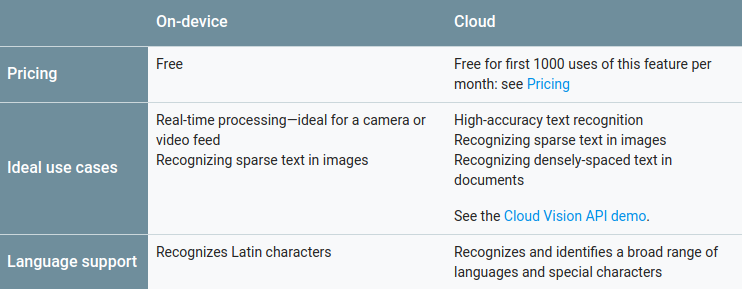
Tác vụ<Danh sách<FirebaseVisionImageLabel>> tác vụ = labeler.processImage(hình ảnh);

task.addOnSuccessListener(labels -> {

Làm điều gì đó với nhãn

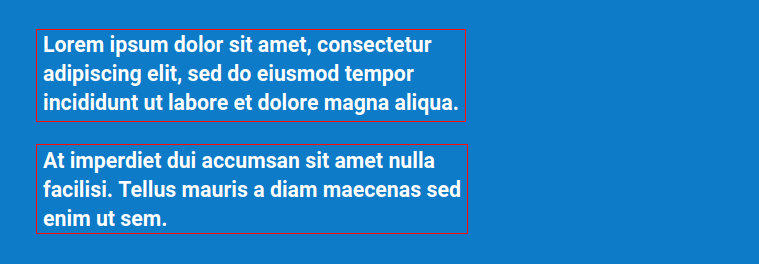
});

Lưu ý rằng mã này chỉ nhằm mục đích tổng quan và mã bổ sung sẽ được yêu cầu để xử lý những thứ như xử lý lỗi, phân luồng và tối ưu hóa hiệu suất. Ngoài ra, mã này giả định rằng bạn có một tập dữ liệu được gắn nhãn để làm việc, nhưng bạn sẽ cần tạo tập dữ liệu được gắn nhãn của riêng mình cho trường hợp sử dụng cụ thể của mình.

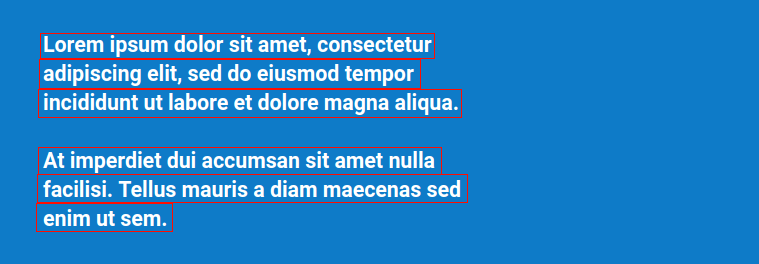


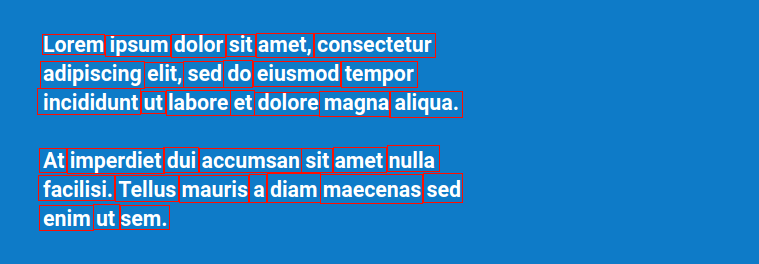
Quá trình nhận dạng chữ viết của ML kit sẽ phân đoạn văn ra làm các khối, các dòng

Các khối là một đoạn văn bản xuyên suốt.



Dòng là một dòng văn bản liên tục



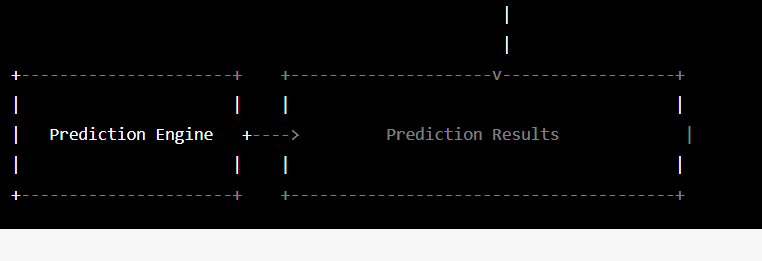
Chữ là một phần tử chữ được chọn

### 2.4.3 Sơ đồ thuật toán Google MLKIT

Dưới đây là sơ đồ đơn giản hóa về kiến trúc chung của thuật toán học máy bằng Google ML Kit:

Diagram

Description automatically generated with medium confidence



Sơ đồ cho thấy luồng dữ liệu cơ bản thông qua thuật toán học máy sử dụng Google ML Kit:

1. Labeled Training Data: Một tập dữ liệu gồm các ví dụ máy học. Mỗi ví dụ là một điểm dữ liệu (ví dụ: hình ảnh, văn bản, âm thanh) với nhãn tương ứng (ví dụ: mèo, chó, chim).

2.Training Algorithm: các dữ liệu được đưa vào để phát hiện ra các dữ liệu gồm nhiều thuật toán mới. Tạo ra mô hình được đào tạo.

3. Trained Machine Learning Model: Mô hình học máy được đào tạo là một tập hợp các tham số có thể được sử dụng để phân loại các điểm dữ liệu mới.

4. Unlabeled Data: Các điểm dữ liệu mới cần được phân loại được đưa vào công cụ dự đoán.

5. Prediction Engine: Công cụ dự đoán sử dụng mô hình học máy được đào tạo để phân loại các điểm dữ liệu mới và tạo kết quả dự đoán.

6. Prediction Results: Kết quả dự đoán là đầu ra, được sử dụng cho nhiều tác vụ khác nhau như gắn nhãn hình ảnh, phát hiện đối tượng hoặc nhận dạng văn bản.

Một sơ đồ đơn giản hóa và các thuật toán học máy thực tế có thể phức tạp hơn nhiều, liên quan đến nhiều bước và lớp xử lý hơn.

### 2.4.4 Traning và Testing

Có 2 yếu tố quan trọng ảnh hưởng trực tiếp đến máy học là ovefiting và underfiting. Các mô hình sử dụng underfiting luôn cho ra các kết quả tệ hại vì các dữ liệu đã quan sát được sử dụng. Ngược lại việc sử dụng overfỉing lại ít được sử dụng trong thực tế dù cho nó có kết quả tiêu chuẩn cao trong quá trình thực hiện. 2 yếu tố này có tính thực tế và độ chính xác không cao, vì nó quá đơn giản hoặc đôi khi lại qusa phức tạp.

Trong máy học cũng vậy ta phải kiểm thử trước rồi mới hoạt động trên mô hình thực tế để dự đoán kết quả. Có nhiều cách kiểm thử và đánh giá.

Có nhiều cách thực hiện hơn vì nó có thể đơn giản hóa đi như dùng thời gian thực tế để sử dụng, nhiều trường hợp khó có thể đem ra thử như vậy, có kịch bản kiểm.

Cách tiếp theo để kiếm là không sử dụng một cách nào khác mà lại lấy dữ liệu thử thách không liên quan tới việc huấn luyện đa sắc. Làm tăng khả năng chính xác và hiệu quả tốt. Setting và Testing tách ra ngay các vật được cung cấp và liên quan đến nhau.

### 2.4.5 Việc huấn luyện hay còn gọi là traning set:

Các dữ liệu dùng trong việc huấn luyện và các thuật toán sẽ được sử dụng trong đây tạo ra các thứ mới gọi chung là việc huấn luyện tập. Khi học thì mọi thứ sẽ được thay đổi theo nhiều thức khác nhau ví dụ các thuật toán mới, các cách thức huấn luyện mới.

Việc mà con người hay các nhà phát triễn dùng các phương pháp cụ thể như gốc sẽ được tối ưu hóa ra nhiều lần giảm độ dốc hoặc giảm độ dốc ngẫu nhiên.

Việc huấn luyện cũng như con người khi được luyện tập hằng ngày sẽ tạo ra một thói quen cũng như kĩ năng cứng. Mỗi ngày sẽ thu nhập nhiều thứ tạo nên kĩ năng riêng mà mình có thể sử dụng.

Ví dụ khi sử dụng hồi quy tuyến tính, các mối quan hệ đường sẽ được kết hợp lại với nhau, các điểm này sẽ được tìm ra và sử dụng. Các mối uqan hệ sẽ được thiết lập. Khi vào thực tế thì không có chuyện này xãy ra. Các nhãn và đích sẽ được thông qua một cách tốt nhất.

### 2.4.6 Testing Set (kiểm thử)

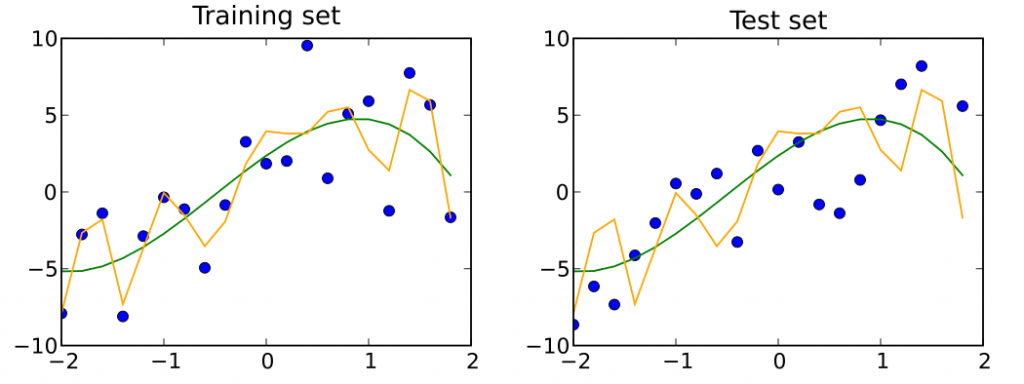
Để tạo ra các ứng dụng có khả năng hóa quát tổng các vấn đề và độ chính xác luôn cao. Thì máy học là mục tiêu quan trọng nhất. Khi ta đã tạo ra một ứng dụng hoàn chỉnh thì đây chưa phải là dấu chấm hết mà ta phải có một quá trình quan trọng khác là kiểm tra lại toàn bộ mọi thứ. Đây gọi là testing set kiểm thử mọi quy trình, mọi quy chuẩn.

Việc này là chưa đủ để hoàn thành. Việc máy học thực hiện phải được tốt nhất tất cả các vấn đề từ việc chạy được chương trình, chương trình phải trơn tru không có lỗi, các kết quả thực hiện ở mức tương đối. Sau khi thực hiện để mà biết được ứng dụng có thật sự tốt hay chưa thì ta cần một thứ gọi là tập kiểm tra. Hiệu quả được đánh giá tốt.

Tập kiểm thử này được dùng để đo và tính ra các trường hợp sai có thể xãy ra. Được đào tạo qua nhiều bài bản, nhiều trường hợp. Các trường hợp sẽ được lưu lại và so sánh với các trường hợp khác nhằm tối ưu hóa việc tìm kiếm sai sót. Bằng cách tính trung bình cộng mọi trường hợp, từ đó mà ta có khà năng biết được gần như toàn bộ các phần lỗi có phát hiện.

Việc nhận ra các lỗi này được đánh giá từ từ nhiều chỉ tiêu khác nhau như F1-Sore, MAE,… Hàm mất mát hoặc hoặc hàm mục tiêu được sử dụng phổ biến hơn hết. quá trình tạo ra ứng dụng và quá trình kiểm tra là 2 quá trình độc lập với nhau không có sự trùng lấp.

Chỉ khi quá trình thực hiện ứng dụng có vấn đề gì thì quá trình kiếm tra sẽ được thực hiện. khi sai số là không nhiều thì việc này được gọi là mẫu như hình bên dưới.



### 2.4.7 Quá trình kiểm chứng hay được gọi là validation Set

Việc sử dụng validation set sẽ giúp cho những nhà phát triễn phần mềm hoàn thành nốt giai đoạn cuối cùng và xem ứng dụng có đạt được hiệu quả tốt nhất hay không. Ngược lại việc mà cho ra kết quả tệ thì sao nhỉ? Cần phải điểu chỉnh thay đổi mô hình để triễn khai ra được tốt nhất. Việc này đang đi theo một chiều hướng nào đó không hợp lí lắm.

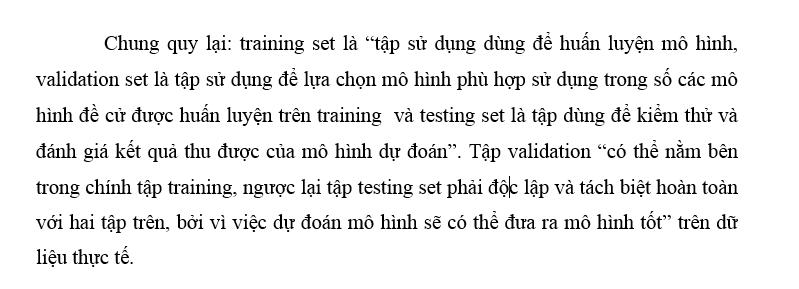
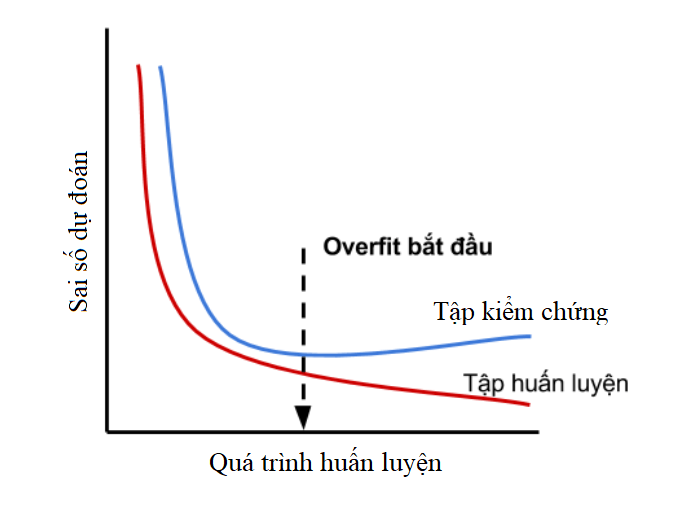
Việc mà chúng ta cố gắng đưa ra các trường hợp, các dự đoán nhằm chuẩn bị cho các trường hợp xấu nhất. Hoàn toàn không phải là đưa ra các trường hợp đã xãy ra rồi. Vì thế ta cần phải có một tập dữ liệu hoàn toàn khác làm nhiệm vụ điều chỉnh ứng dụng khi kết thúc quá trình training. Các dữ liệu này gọi là quá trình kiếm chứng validation set.

Chúng ta hoàn toàn có thể kiểm tra, test thử toàn bộ quá trình hoạt động rồi đưa ra chứng nhận cuối cùng cho bài test cuối cùng. Điều này giống như khi đi học ở bất kì đâu thì luôn luôn có những bài kiểm tra miệng hay 15p nhằm tổng kết kiểm tra lại toàn bộ những kiến thức đã học, đã được áp dụng vào thực hành. Khi mà chúng ta cần lựa chọn xem việc huấn luyện hay kiểm tra nào là phù hợp thì ta sẽ sử dụng validation set, còn việc kiểm tra cuối cùng thì ta lại sử dụng testing set.

Ở hình vẽ ở bên dưới, việc mà tạm dừng huấn luyện tại đó thì nó có sự thay đổi rất rõ rệt trong validation set. Thời gian mà diễn ra overfiting là khi đó nó hoạt động lớn nhất.

Việc kiểm thử sẽ tìm kiếm ra các thứ tốt nhất trong quá trình huấn luyện nên nó có tác dụng rất lớn. Nó được sử dụng đo các so sánh về việc sử dụng cái nào là tốt nhất như độ nhạy cảm, RMSE,…

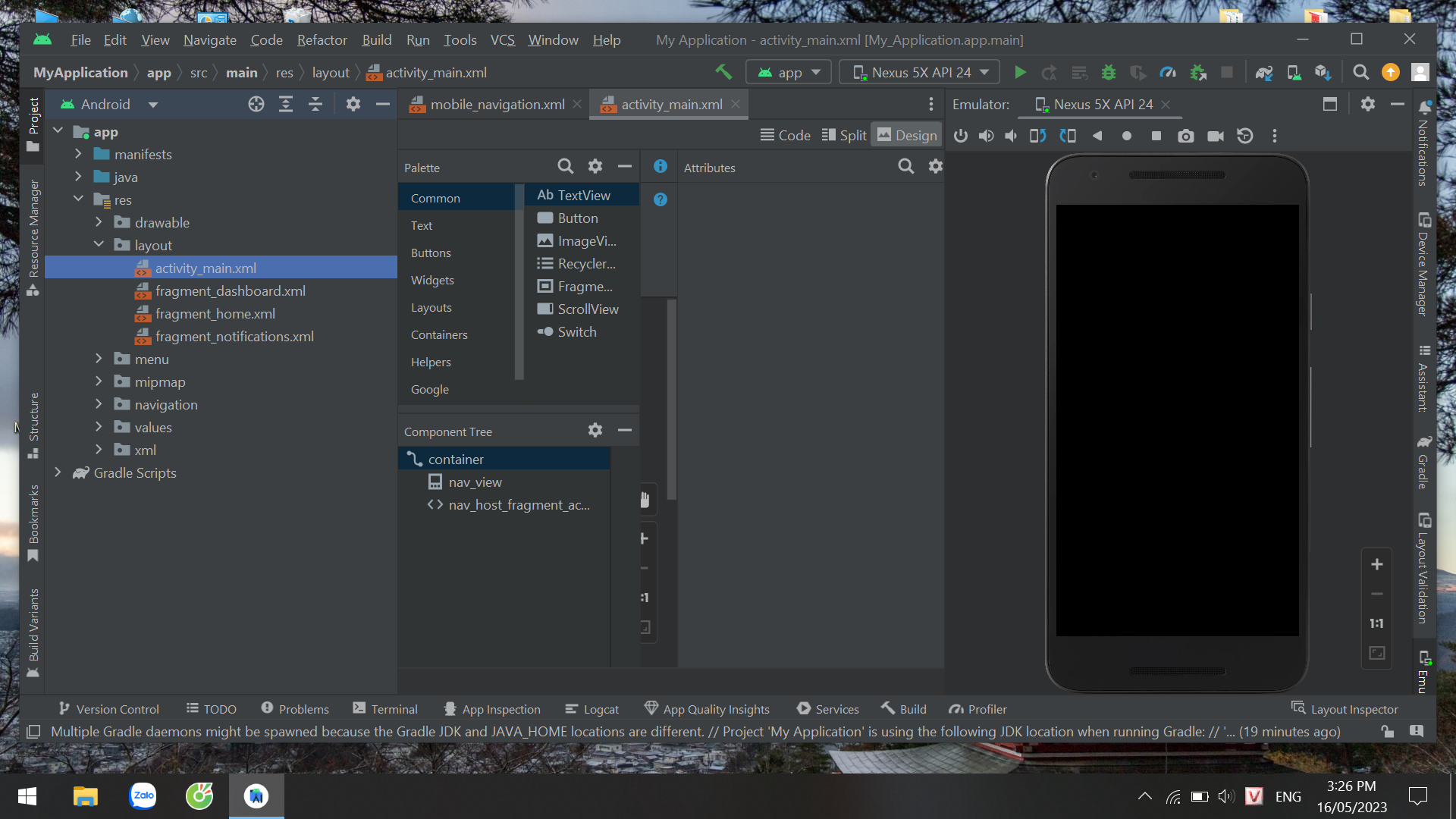
Khi mà có lỗi thì kiểm chứng cũng giúp ứng dụng dừng sớm. Khi mà lỗi đã tăng quá cao và vượt quá khả năng kiểm soát. Việc áp dụng này nhìn là tuy đơn giản mà lại có sự phức tạp trong việc điều hành. Phải xác định rõ thời gian hoạt động cũng như kết thúc.



## 2.5 Máy ảo android studio

Android studio là phần mềm nhằm để tạo ra một ứng dụng ảo điện thoại được phát triễn bởi Android. Được phát triễn để cung cấp hỗ trợ phát triễn các ứng dụng của Android như mã nguồn mở, trình biên dịch, gỡ lỗi, quản lý phiên bản, các thiết kế giao diện, các tính năng hay ho khác.

Các nhà phát triễn phần mềm dùng để tạo ra giao diện android chất lượng tốt nhất. nó hỗ trợ nhiều ngôn ngữ như flutter, java cho nhiều nền tảng, và nhiều thiết bị khác nhau.



*Hình 2.5.1: Giao diện android studio*

# CHƯƠNG 3 : PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

## 3.1 Những phân tích ban đầu

Ở cuộc sống bình thường hiện nay, công nghệ đã ngày càng phát triễn. Việc áp dụng khoa học công nghệ vào cuộc sống là vô cùng cần thiết. Việc soạn thảo văn bản, xử lý thông tin, dữ liệu nhanh là vấn đề cấp bách. Việc có những ý tưởng viết ra giấy ngay tức thì để lưu lại những ý tưởng sau đó muôn đưa nhưng ý tưởng đó lên những phần mềm soạn thảo nhanh chóng, mà việc tốn thời gian thêm cho quá trình nhập liệu là không cần thiết.

Ảnh có chứa văn bản, ảnh chụp màn hình

Mô tả được tạo tự động

*Hình 3.1: Mô phỏng cơ bản ban đầu*

Quá trình xử lý chữ viết tay, ghi chú gồm 5 bước:

Tiền xử lý (Preprocessing) : Các hình ảnh lần đầu tiên được đưa vào xử lý sẽ được làm giảm các phần nhiễu đi làm cho các lỗi trong quá trình xử lý hình ảnh được giảm đi, các dữ liệu được chuẩn hóa một cách có logic, theo một quy chuẩn nhất định, việc xử lý về sau sẽ dễ dàng hơn.

Tách chữ (Segmentation): Các đoạn văn bản không được đem ra xử lý một cách ngay lập tức mà sẽ có một giai đoạn khác là tách ra thành nhiều đoạn văn bản nhỏ. Việc sử lý các đoạn văn bản hay chữ cái đơn lẻ sẽ chính xác hơn nhiều so việc xử lý một khối văn bản lớn.

Trích chọn đặc trưng ( Representation): Để việc xử lý được trơn tru và ít phức tạp thì đây là giai đoạn vô cùng quan trọng. Tránh được sự phức tạp hóa của chữ viết. Ta sẽ biến chữ viết này thành một dạng đặc biệt khác, theo một quy định đã đặt ra trước, có các đặc điểm riêng biệt nhằm tăng khả năng chính xác của chữ viết.

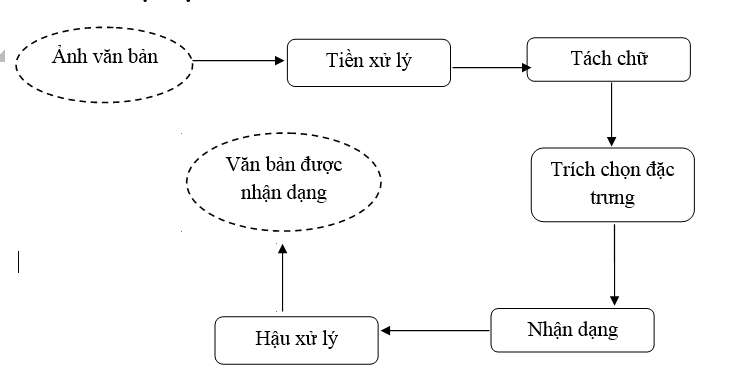
Huấn luyện và nhận dạng (Training anh regconition): đây là giai đoạn dùng để máy học thêm các loại chữ viết khác. Dùng các thuật toán do con người tạo ra nhằm học thêm nhiều mẫu chữ viết mới.

Hậu xử lý (Postprocessing): kiểm tra lại lần cuối các quá trình xử lý, làm tăng cường sự chính xác trong quá trình chuyển đổi.

Các học sinh, sinh viên có nhu cầu sử dụng rất cao, nhằm thực hiện các bài tập, ý tưởng cho cái hoạt động mình làm. Nhu cầu sử dụng của nhân viên văn phòng cũng rất cao do môi trường làm việc luôn đòi hỏi sự nhanh chóng, tiện ích. Việc đánh máy lại những ý tưởng trên giấy sẽ mất thêm một công đoạn nên sẽ mất rất nhiều thời gian. Việc tạo ra ứng dụng này sẽ giúp cho các công việc văn phòng, học tập được diễn ra nhanh chóng hơn.

## 3.2 Ý tưởng hệ thống:

Như đã đề cập thì việc tạo ra ứng dụng là vô cùng cần thiết. Vì nó sẽ tiết kiệm được nhiều thời gian và công sức cho người có nhu cầu sử dụng. Ứng dụng sẽ chuyển đổi chữ viết tay trên giấy sang word nhanh chóng.



*Hình 3.2: Sơ đồ ý tưởng*

Trong đó thì vấn đề nhân dạng văn bản tiếng việt còn một số hạn chế nhỏ : phân biệt font chữ , chữ tiếng việt có dấu , chữ gạch dưới, chữ viết tay xấu,… làm giảm khả năng nhận dạng trích xuất từ ngữ dẫn đến kết quả đầu ra có vài sai sót . Vì vậy nhóm e chọn đề tài này nằm tìm hiểu, học hỏi , phát triển để có thể nhận dạng tốt hơn nâng cao tỷ lệ trích xuất đúng các văn bản.

## 3.3 Phạm vi đề tài:

“Xây dụng ứng dụng chuyển đổi chữ viết tay sang Word ” cày đặt và chạy thử nghiệm phần mềm. Xử lý các vấn đề chuyển đổi chữ viết ghi chú sang word, tập trung phân tích, nhận ra từng ký tự tiếng việt đơn lẻ .

Từ đó tạo cơ sở xây dựng phát triển ứng dụng chuyển đổi chữ tay sang word trên ứng dụng di động, từ đó sử dụng và áp vào các việc như học tập, làm việc, xử lý hình ảnh.

Đồ án phân tích tập trung vào các hệ thống nhận dạng : tiền xử lý, phân tích đặc trưng và trích xuất dữ liệu. Sau đó tạo ra ứng dụng trên máy tính bằng flutter sau đó phát triễn và bán lên ch play, app store.

3.3 Ý tưởng tổng thể hệ thống ứng dụng:

Sử dụng các thiết bị kĩ thuật số có camera để chụp lại những đoạn văn bản, bài báo, biển bảo,… đưa lên hệ thống của ứng dụng để trích xuất các tự ngữ cần thiết trong văn bản đó rồi trả vê với dạng text. Người dùng có thể copy đưa vào word để chỉnh sữa, thay thế , xoá bớt mà không cần phải gõ lại hay viết lại chúng.

Bằng vào những hiểu biết cơ bản về open cv, cũng như xử lý ảnh nhóm em làm trích xuất các văn bản viết tay dưới dạng plandtext thông qua trình xử lý ảnh của open cv sẽ phát triển thêm các tính năng khác trong tương lai.

•Xác định ảnh đầu vào thông qua tiền xử lý quá trình này phụ thuộc vào chất lượng ảnh cụ thể hoặc tuỳ vào chức năng hệ thống.

•Nhị phân hoá ảnh chuyển ảnh về dưới dạng ảnh xám. Quá trình này gồm 2 phần : nền và chữ . Bằng cách chọn ngưỡng cố định lớn hơn ngưỡng trả về giá trị mức sách và bé hơn thì trả về giá trị tối.

•Tiến hành lọc nhiễu bỏ đi các thành phần dư thừa trong ảnh

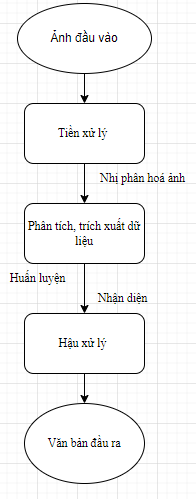
•Chuẩn hoá kích thước các thành phần trong ảnh:

Chuẩn quá kích thớc xác định trọng tâm ảnh, sau đó từ tâm ảnh xác định khoản cách các cạnh trên, dưới, trái , phải. Qua đó xác định tỉ lệ co, giãn so với ảnh góc vì vậy tính cân bằng luôn được coi trọng không bị thay đổi.

•Làm trơn chữ và trích chọn đặc trưng cơ bản.

•Quá trình được xãy ra sau cùng là hậu xử lý, thông qua các đặc trưng cơ bản trích xuất ghép các kí tự thành câu. Nhằm phát hiện ra văn bản và các lỗi trong câu, các kí tự lạ, trong các đoạn văn.

Đề tài “ xây dựng ứng dụng chuyển đổi văn bản viết tay thành văn bản word trên ứng dụng di động” thì hầu hết trên hệ thống đều có sẵn trên máy tính , nên chúng ta đi vào trực tiếp việc xử lý ảnh đầu vào của hệ thống . Ảnh đầu vào được xử lý cơ bản rồi so sánh các từ ngữ trong văn bản được trích xuất với đưa ra dữ liệu đầu ra, thống kê kết quả tỉ lệ chính xác các từ đã được trích xuất.



*Hình 3.3: Sơ đồ khối*

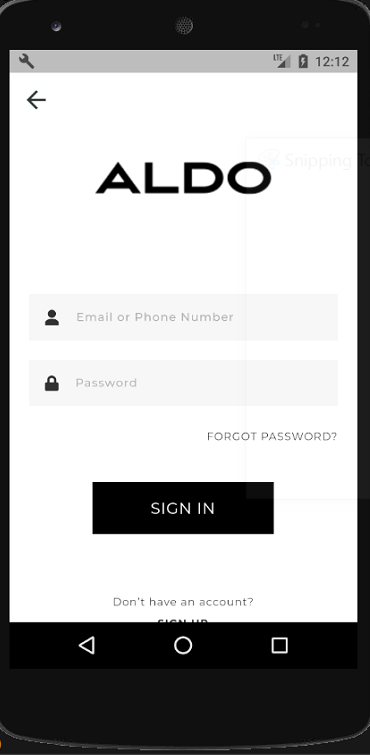
## 3.4 Giao diện ứng dụng:

## A picture containing text, multimedia, mobile phone, communication device Description automatically generatedA person sitting on a swing Description automatically generated with low confidence

## A picture containing text, mobile phone, communication device, portable communications device Description automatically generated

*Hình 3.4: Giao diện khởi tạo*

Có thể dùng giới thiệu sơ bộ về ứng dụng cũng như tạo sự thú vị cho ứng dụng thu hút người dùng.



*Hình 3.5: Giao diện đăng nhập*

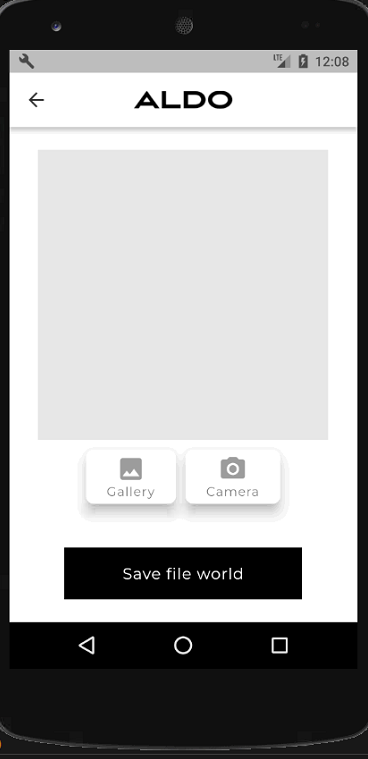
Màn hình đăng nhập ứng dụng. Ta cần sử dụng tài khoản gồm 10 kí tự. Mật khẩu gồm 1 chữ cái in hoa, có số và 8 kí tự



*Hình 3.6: Giao diện homepage*

Giao diện homepage gồm các thành phần:

* Button chuyển vào trang chuyển đổi văn bản.
* Các chức năng khác.



*Hình 3.7:Giao diện trang chuyển đổi*

Trang chuyển đổi gồm các thành phần:

* 2 button gồm: 1 button thêm ảnh từ máy, 1 button chụp ảnh hiện tại
* 1 button chuyển đổi văn bản đã chuyển đổi sang word.
* 1 màn hình hiện hình ảnh văn bản cần chuyển đổi.
* 1 khu vực hiện văn bản đã chuyển đổi.

## 3.5 Kết quả thực nghiệm

## 

A screenshot of a phone

Description automatically generated with medium confidence

Hình ảnh ứng dụng chuyển đổi chữ viết tay cơ bản, chung ta có thể copy để gửi tin hoặc save trực tiếp qua word để lưu trữ lại sau quá trình chuyển đổi.



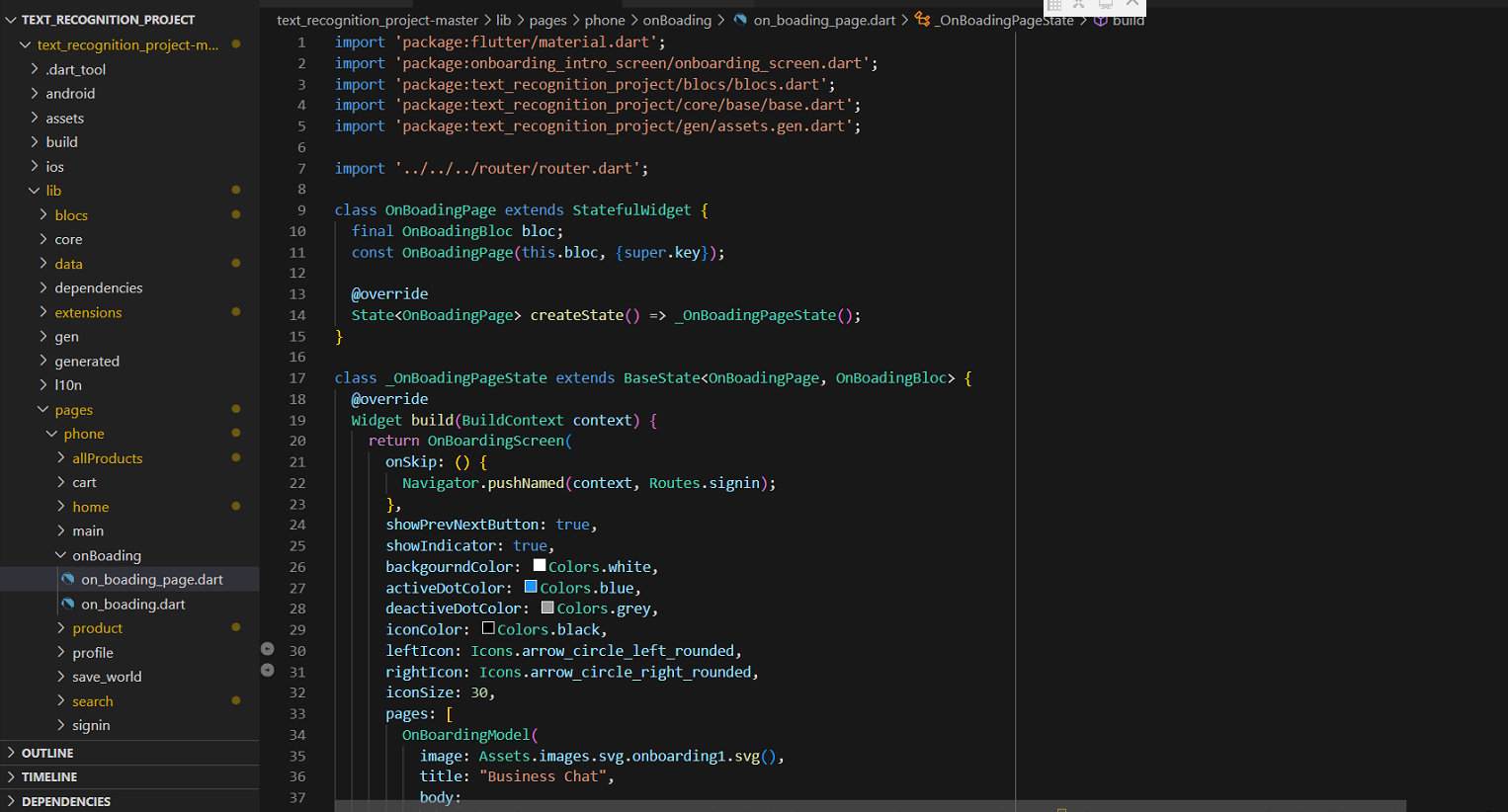
*Hình 3.8: giao diện export file*

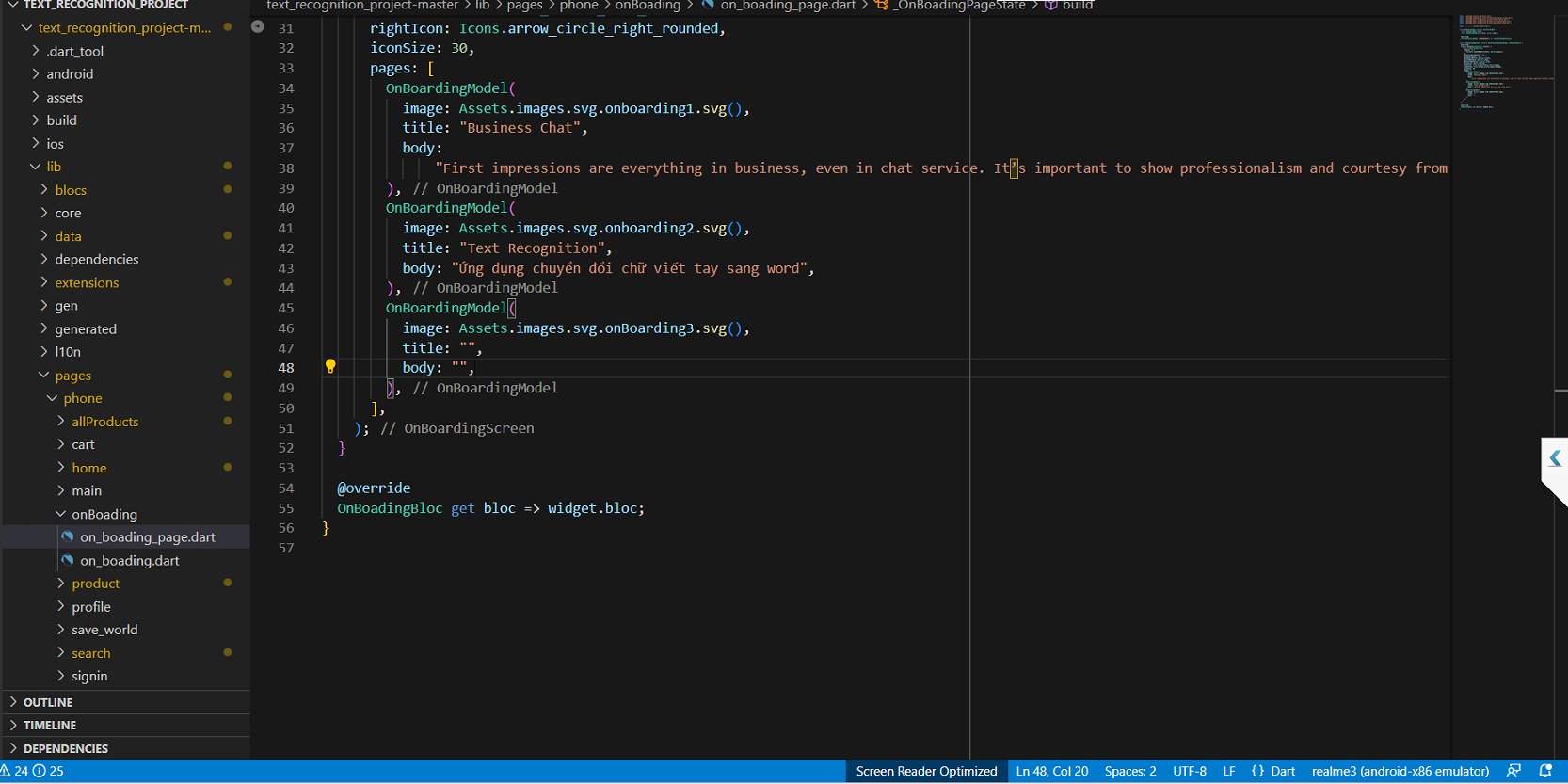
Để chuyển đổi sang word cần đặt tên cho file chuyển đổi tránh sự nhầm lẫn file trong quá trình lưu trữ và export file để chuyển sang word.



Dữ liệu được export ra word có thể tuỳ ý chỉnh sửa : thay đổi, thêm , xoá ,... Khi hoàn tất chúng ta có thể sao lưu lại và thực hiện các hoạt động khác.

* Code giao diện mở đầu:





# Chương 4 : THỰC NGHIỆM VÀ ĐÁNH GIÁ KẾT QUẢ

4.1 Kết luận

So với các việc đã thực hiện trong quá trình thực nghiệm kiểm tra khả năng chuyển đổi văn bản được viết tay sang word ở mức cơ bản. Ứng dụng đã chạy tốt và khá ổn định tuy nhiên còn chứa nhiều nhược điểm chưa khai thác hết . Hệ thống nhận diện được kí tự viết tay dưới dạng kí tự plantext và tốt hơn khi ở dạng các kí tự la tinh. Hệ thống không bao gồm chuyển đổi văn bản ở nhiều dạng style khác nhau . Trong quá trình chuyển đổi độ chính xác sẽ xác định dựa trên nhiều yếu tố khách quan như : độ sáng đèn, kiểu chữ mỗi người khác nhau, vật liệu được viết như thế nào,... Qúa trình thực nghiệm được test trên mỗi các đoạn văn ngắn hay các kí tự khoản 10-20 lần để có thể đưa ra kết luận hệ thống làm việc có ổn định hay không và độ chính xác như thế nào.

Với đề tài này phù hợp cho sinh viên phát triển khả năng tự học , tự làm việc mà còn áp dụng được những kiến thức đã học vào trong thực tiễn. Việc triễn khai mô hình hệ thống này giúp cũng cố kiến thức đã học , cũng như tạo điều kiện thuận lợi cho sinh viên tham gia vào một số dự án , phát triễn kĩ năng , làm việc có kế hoạch trước khi ra trường tiến vào môi trường doanh nghiệp.

4.2 Hướng phát triễn

Triễn khai thêm nhiều tính năng mới cho hệ thống.

Mở rộng thêm phần chuyển đổi đa ngôn ngữ.

Phát triễn thêm phần dịch trực tiếp khi viết lên các textbox trên những ứng dụng martphone.

Đưa hệ thống phát triển ra với thị trường.

Thêm lượng lớn các dữ liệu huyến luyện để tăng khả năng chính xác của mô hình .

Sử dụng ứng dụng trên đa nền tảng.