**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA CÔNG NGHỆ PHẦN MỀM**

A picture containing graphics, logo, clipart, symbol

Description automatically generated

**BÁO CÁO ĐỒ ÁN 1**

**HỎI ĐÁP THỊ GIÁC**

**Visual question answering**

**CỬ NHÂN NGÀNH KỸ THUẬT PHẦN MỀM**

**LÊ THỊ BÍCH LOAN**

**TP. HỒ CHÍ MINH, 2023**

**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA CÔNG NGHỆ PHẦN MỀM**

**LÊ THỊ BÍCH LOAN – 21521083**

**BÁO CÁO ĐỒ ÁN 1**

**HỎI ĐÁP THỊ GIÁC**

**Visual question answering**

**CỬ NHÂN NGÀNH KỸ THUẬT PHẦN MỀM**

**GIẢNG VIÊN HƯỚNG DẪN**

**TS. NGUYỄN TẤN TRẦN MINH KHANG**

**TP. HỒ CHÍ MINH, 2023**

**LỜI CẢM ƠN**

Vai trò của ngành công nghệ thông tin nói chung và trí tuệ nhân tạo nói riêng trong đời sống ngày càng trở nên quan trọng. Thực tế cho thấy, việc ứng dụng thành tựu công nghệ thông tin vào các công đoạn như sản xuất, quản lý, học tập … đã tạo nên một lợi ích to lớn, dẫn đến cuộc cách mạng số hóa bùng nổ trong thời đại hiện nay. Tuy nhiên, việc ứng dụng công nghệ không phải luôn dễ dàng, đặc biệt là với những nước đang phát triển và thiếu hụt nguồn nhân lực chất lượng trong lĩnh vực công nghệ thông tin như Việt Nam. Điều đó đặt ra yêu cầu các sản phẩm công nghệ phải dễ sử dụng, thân thiện với những người không có kiến thức công nghệ thông tin và vẫn đảm bảo tính hiệu quả, đúng đắn của nó.

Sau khi được học tập và rèn luyện dưới sự hướng dẫn của TS. Nguyễn Tấn Trần Minh, trường Đại học Công nghệ thông tin – ĐHQGHCM, em đã được trang bị những kiến thức cơ bản để có thể phát triển một phần mềm ứng dụng thỏa mãn yêu cầu trên. Vì vậy, em lựa chọn thực hiện đồ án 1 với đề tài “Hỏi đáp thị giác” để có thêm kinh nghiệm và cái nhìn sâu hơn về ứng dụng trí tuệ nhân tạo trong việc phát triển phần mềm và bên cạnh đó học hỏi thêm nhiều kiến thức mới trong việc huấn luyện và tạo ra các mô hình học máy và học sâu.

Em xin chân thành cảm ơn TS. Nguyễn Tấn Trần Minh Khang đã tận tình truyền dạy những kiến thức cần thiết và hướng dẫn em hoàn thành đồ án này.

Trong quá trình thực hiện đồ án, không thể tránh khỏi những sai sót, rất mong được nhận sự phản hồi góp ý của thầy/cô và mọi người để em rút kinh nghiệm và hoàn thiện hơn trong hành trình tiếp theo.

Trân trọng cảm ơn!

MỤC LỤC

[Chương 1. MỞ ĐẦU 11](#_Toc154997671)

[1.1. Đặt vấn đề 11](#_Toc154997672)

[1.2. Giới thiệu bài toán 12](#_Toc154997673)

[1.3. Mục tiêu đề tài 13](#_Toc154997674)

[1.4. Nội dung nghiên cứu 13](#_Toc154997675)

[1.5. Bố cục báo cáo 13](#_Toc154997676)

[Chương 2. CƠ SỞ LÝ THUYẾT 15](#_Toc154997677)

[2.1. Tổng quan về React Native 15](#_Toc154997678)

[2.1.1. Giới thiệu 15](#_Toc154997679)

[2.1.2. Ưu điểm 16](#_Toc154997680)

[2.1.3. Nhược điểm 16](#_Toc154997681)

[2.2. Tổng quan về Firebase 17](#_Toc154997682)

[2.2.1. Giới thiệu 17](#_Toc154997683)

[2.2.2. Ưu điểm 19](#_Toc154997684)

[2.2.3. Nhược điểm 20](#_Toc154997685)

[2.3. Tổng quan về Azure 21](#_Toc154997686)

[2.3.1. Giới thiệu 21](#_Toc154997687)

[2.3.2. Ưu điểm [5] 23](#_Toc154997688)

[2.3.3. Nhược điểm [6] 24](#_Toc154997689)

[2.4. Mô hình client – server 24](#_Toc154997690)

[2.4.1. Client (Khách hàng) 25](#_Toc154997691)

[2.4.2. Server (Máy chủ) 25](#_Toc154997692)

[2.4.3. Giao tiếp giữa client và server 25](#_Toc154997693)

[2.4.4. Ưu điểm 26](#_Toc154997694)

[2.4.5. Nhược điểm 27](#_Toc154997695)

[2.5. Vision-and-Language Pre-training 27](#_Toc154997696)

[2.5.1. Sự tương tác giữa 2 phương thức 28](#_Toc154997697)

[2.5.2. Phân loại mô hình Vision-and-Language 28](#_Toc154997698)

[2.6. ViLT: Vision-and-Language Transformer Without Convolution or Region Supervision 29](#_Toc154997699)

[2.6.1. Các bước quan trọng trong ViLT 31](#_Toc154997700)

[2.6.2. Kiến trúc cụ thể của ViLT 31](#_Toc154997701)

[2.6.3. Thông số mô hình 32](#_Toc154997702)

[Chương 3. THIẾT KẾ DỮ LIỆU 33](#_Toc154997703)

[3.1. Sơ đồ cơ sở dữ liệu quan hệ 33](#_Toc154997704)

[3.2. Danh sách bảng dữ liệu 33](#_Toc154997705)

[3.3. Mô tả chi tiết các bảng dữ liệu 33](#_Toc154997706)

[3.3.1. Bảng NGUOIDUNG 33](#_Toc154997707)

[3.3.2. Bảng LICHSU 34](#_Toc154997708)

[Chương 4. THIẾT KẾ GIAO DIỆN 35](#_Toc154997709)

[4.1. Danh sách các màn hình 35](#_Toc154997710)

[4.2. Chi tiết các màn hình 37](#_Toc154997711)

[4.2.1. Màn hình đăng nhập 37](#_Toc154997712)

[4.2.2. Màn hình đăng ký 37](#_Toc154997713)

[4.2.3. Màn hình đăng ký thành công 38](#_Toc154997714)

[4.2.4. Màn hình quên mật khẩu 38](#_Toc154997715)

[4.2.5. Màn hình gửi email thành công 39](#_Toc154997716)

[4.2.6. Màn hình trang chủ 39](#_Toc154997717)

[4.2.7. Màn hình lịch sử 40](#_Toc154997718)

[4.2.8. Màn hình chi tiết lịch sử câu hỏi 40](#_Toc154997719)

[4.2.9. Màn hình quản lý tài khoản 41](#_Toc154997720)

[4.2.10. Màn hình thông tin tài khoản 41](#_Toc154997721)

[4.2.11. Màn hình đổi mật khẩu 42](#_Toc154997722)

[4.2.12. Màn hình thông tin ứng dụng 42](#_Toc154997723)

[Chương 5. CÀI ĐẶT VÀ THỬ NGHIỆM 43](#_Toc154997724)

[5.1. Kết luận 43](#_Toc154997725)

[5.1.1. Ưu điểm 43](#_Toc154997726)

[5.1.2. Nhược điểm 43](#_Toc154997727)

[5.2. Hướng phát triển 44](#_Toc154997728)

DANH MỤC HÌNH

[Hình 1.1: Quá trình phát triển quá lĩnh vực trí tuệ nhân tạo 11](#_Toc154997729)

[Hình 1.2: Hình ảnh đầu vào bài toán 12](#_Toc154997730)

[Hình 2.1: Những ứng dụng lớn sử dụng framework React Native 15](#_Toc154997731)

[Hình 2.2: Một số dịch vụ của Firebase 19](#_Toc154997732)

[Hình 2.3: Mô hình Microsoft Azure 22](#_Toc154997733)

[Hình 2.4: Mô hình client – server 24](#_Toc154997734)

[Hình 2.5: Nguyên lý hoạt động Client-Server 25](#_Toc154997735)

[Hình 2.6: Bốn loại mô hình Vision-and-Language 28](#_Toc154997736)

[Hình 2.7 So sánh trực quan các kiến trúc VLP thông thường và ViLP được đề xuất 30](#_Toc154997737)

[Hình 2.8: Tổng quan mô hình (lấy cảm hứng từ Dosovitskiy et al) 31](#_Toc154997738)

[Hình 3.1: Sơ đồ cơ sở dữ liệu quan hệ 33](#_Toc154997739)

[Hình 4.1: Màn hình đăng nhập 37](#_Toc154997740)

[Hình 4.2: Màn hình đăng ký 37](#_Toc154997741)

[Hình 4.3: Màn hình đăng ký thành công 38](#_Toc154997742)

[Hình 4.4: Màn hình quên mật khẩu 38](#_Toc154997743)

[Hình 4.5: Màn hình đổi email thành công 39](#_Toc154997744)

[Hình 4.6: Màn hình trang chủ 39](#_Toc154997745)

[Hình 4.7: Màn hình lịch sử 40](#_Toc154997746)

[Hình 4.8: Màn hình chi tiết lịch sử câu hỏi 40](#_Toc154997747)

[Hình 4.9: Màn hình quản lý tài khoản 41](#_Toc154997748)

[Hình 4.10: Màn hình thông tin tài khoản 41](#_Toc154997749)

[Hình 4.11: Màn hình đổi mật khẩu 42](#_Toc154997750)

[Hình 4.12: Màn hình thông tin ứng dụng 42](#_Toc154997751)

DANH MỤC BẢNG

[Bảng 3.1: Danh sách các bảng dữ liệu 33](#_Toc154997752)

[Bảng 3.2: Mô tả chi tiết bảng NGUOIDUNG 34](#_Toc154997753)

[Bảng 3.3: Mô tả chi tiết bảng LICHSU 34](#_Toc154997754)

[Bảng 4.1: Danh sách các màn hình 36](#_Toc154997755)

DANH MỤC TỪ VIẾT TẮT

|  |  |  |
| --- | --- | --- |
| **STT** | **Ký hiệu chữ viết tắt** | **Chữ viết đầy đủ** |
|  | CNN | Convolutional Neural Network |
|  | VQA | Visual Question Answering |
|  | LSTM | Long Short Term Memory networks |
|  | VLP | Vision-and-Language pretraining |
|  | ViLT | Vision-and-Language Transformer |

TÓM TẮT ĐỀ TÀI

Đề tài này hướng đến việc cung cấp cho sinh viên những trải nghiệm từ việc sử dụng các model học máy, học sâu có sẵn và tích hợp vào ứng dụng di động do sinh viên xây dựng đưa trên các framework và công nghệ được hỗ trợ phổ biến hiện nay. Đề tài Hỏi đáp thị giác (Visual Question Answer) yêu cầu sinh viên nghiên cứu và ứng dụng mô hình của bài toán VQA vào ứng dụng được xây dựng để trực quan đầu vào và đầu ra của bài toán.

# MỞ ĐẦU

## Đặt vấn đề

Trí tuệ nhân tạo đang ngày càng phát triển và tác động đáng kể vào đời sống của mỗi chúng ta. Tần suất chúng ta nghe thấy các cụm từ ‘artificial intelligence’, ‘machine learning’, ‘deep learning’ cũng ngày một tăng lên. [1] Và đặc biệt là sự ra đời của Deep Learning – Học sâu (2012) đã tạo bước nhảy vọt lớn trong lĩnh vực này, đánh dấu sự phát triển mạnh mẽ của trí tuệ nhân tạo cũng như các lĩnh vực ứng dụng nó trong những năm trở lại đây.



Hình 1.1: Quá trình phát triển quá lĩnh vực trí tuệ nhân tạo

Hỏi đáp thị giác là một bài toán thuộc lĩnh vực trí tuệ nhân tạo. Đây là công việc dễ dàng đối với con người, tuy nhiên, nó lại là một vấn đề vô cùng to lớn đối với máy tính. Cũng vì lẽ đó, hỏi đáp thị giác (VQA) là một nhiệm vụ đầy thách thức và luôn nhận được sự quan tâm trong lĩnh vực thị giác máy tính và xử lý ngôn ngữ tự nhiên. [2] Hiện nay đã có rất nhiều giải pháp được thiết kế để giải quyết bài toán này. Tuy nhiên, số lượng ứng dụng di động giải quyết bài toán hỏi đáp thị giác vẫn còn hạn chế. Đây là lý do thúc đẩy tôi thực hiện đề tài này nhằm xây dựng một ứng dụng di động phục vụ nhu cầu sử dụng bài toán hỏi đáp thị giác của người Việt hiện nay.

## Giới thiệu bài toán

Bài toán VQA (Visual Question Answering) là một thể hiện của lĩnh vực trí tuệ nhân tạo. Mục tiêu của bài toán này là xây dựng các hệ thống máy tính có khả năng tự động hiểu và trả lời các câu hỏi liên quan đến hình ảnh bằng tiếng Anh với đầu vào là một hình ảnh cùng với câu hỏi liên quan đến nội dung hình ảnh đó và đầu ra bài toán là một câu trả lời bằng tiếng Anh cho câu hỏi dựa trên hình ảnh đầu vào. Đây là bài toán kết hợp giữa xử lý ngôn ngữ tự nhiên và thị giác máy tính, nơi máy tính cần phải hiểu cả nội dung hình ảnh và nội dung câu hỏi để tạo ra câu trả lời chính xác.

Ví dụ về bài toán VQA:

* Đầu vào bài toán:
  + Hình ảnh: Một bức tranh về con mèo đang đứng phía sau bức tường.
  + Câu hỏi: What animal is standing behind the wall?
* Đầu ra bài toán:
  + Câu trả lời: A cat

A cat looking at something

Description automatically generated

Hình 1.2: Hình ảnh đầu vào bài toán

Bài toán Visual Question Answering có một số thách thức lớn như sau:

* Độ Phức Tạp Cao của Nội Dung Hình Ảnh: Hình ảnh có thể chứa nhiều đối tượng, vùng phức tạp và mối liên quan giữa chúng, làm tăng độ khó trong việc hiểu nội dung hình ảnh.
* Khả Năng Nhận Diện Chưa Hoàn Hảo: Các mô hình thị giác máy có thể gặp khó khăn trong việc nhận diện chính xác các đối tượng trong hình ảnh, đặc biệt là khi đối tượng xuất hiện trong các điều kiện ánh sáng yếu hoặc góc nhìn khó khăn.
* Hiểu Biết Ngôn Ngữ Tự Nhiên: Để trả lời câu hỏi, mô hình cần hiểu sâu sắc ngôn ngữ tự nhiên, bao gồm cả ngữ pháp, ngữ nghĩa, và ngữ cảnh của câu hỏi.
* Quản Lý Kiến Thức Toàn Diện: Mô hình cần có khả năng quản lý kiến thức toàn diện về nhiều chủ đề và lĩnh vực để cung cấp câu trả lời đầy đủ và chính xác.

Bài toán VQA có nhiều ứng dụng thực tế trong việc tạo ra các hệ thống thông minh có khả năng tương tác với hình ảnh và ngôn ngữ, như hệ thống trả lời tự động cho các câu hỏi từ hình ảnh trong môi trường chatbot, hỗ trợ khách hàng tự động, gợi ý sản phẩm dựa trên hình ảnh và nhiều ứng dụng khác.

## Mục tiêu đề tài

Mục tiêu của đề tài là xây dựng một ứng dụng di động tích hợp mô hình của bài toán VQA có giao diện thân thiện với người dùng, cung cấp và hỗ trợ đầy đủ các tính năng, đảm bảo yêu cầu về hiệu suất và tính ứng dụng vào thực tế phục vụ nhu cầu hiện nay.

## Nội dung nghiên cứu

Trong quá trình thực hiện đề tài này, tôi tập trung nghiên cứu về bài toán VQA thông qua những nghiên cứu được công bố trước đó về bài toán này, tìm hiểu phương pháp, bộ dữ liệu, ... Từ đó có thể tái sử dụng mô hình dựa trên bộ dữ liệu và mã nguồn có sẵn. Bên cạnh đó, chúng tôi cũng nghiên cứu việc tích hợp mô hình đã huấn luyện vào ứng dụng di động theo mục tiêu đề ra ban đầu.

## Bố cục báo cáo

Trong báo cáo này, chúng tôi tập trung giới thiệu ứng dụng được xây dựng từ framework React Native tích hợp mô hình hỏi đáp thị giác được huấn luyện trước đó theo các phần như sau. Phần 2 là “Cơ sở lý thuyết”, nơi chúng tôi trình bày các nghiên cứu liên quan được sử dụng trong quá trình xây dưng ứng dụng này. Tiếp theo, phần 3 trình bày các thành phần của cơ sở dữ liệu và sơ đồ cơ sở dữ liệu quan hệ của ứng dụng. Phần 4 bao gồm danh sách, sơ đồ liên kết và mô tả các màn hình. Cuối cùng, phần 5 chứa kết luận và hướng phát triển đề xuất trong tương lai.

# CƠ SỞ LÝ THUYẾT

## Tổng quan về React Native

### Giới thiệu

React Native được Facebook phát hành lần đầu tiên vào năm 2015 dưới dạng một dự án open source. Chỉ trong vài năm, React Native đã trở thành một trong những giải pháp hàng đầu được sử dụng để phát triển mobile. [3] Hiện nay, React Native đã trở thành một trong những công cụ phổ biến để phát triển ứng dụng di động đa nền tảng. Nó giúp cho việc xây dựng ứng dụng di động trở nên nhanh chóng, tiết kiệm thời gian và mang lại trải nghiệm người dùng tốt trên cả hai nền tảng iOS và Android.

* Xây dựng ứng dụng di động đa nền tảng sử dụng JavaScript.
* Sử dụng React để xây dựng giao diện người dùng.
* Sử dụng các thành phần giao diện nguyên bản của hệ điều hành.
* Chia sẻ mã nguồn giữa các nền tảng iOS và Android.
* Truy cập các tính năng phần cứng của thiết bị thông qua API mạnh mẽ.

A blue background with icons and a person in a space suit

Description automatically generated

Hình 2.1: Những ứng dụng lớn sử dụng framework React Native

### Ưu điểm

1. Phát triển đa nền tảng: React Native cho phép bạn phát triển ứng dụng di động cho cả iOS và Android từ cùng một mã nguồn. Điều này tiết kiệm thời gian và công sức so với việc phát triển riêng lẻ trên hai nền tảng này.
2. Sử dụng JavaScript: React Native sử dụng JavaScript để xây dựng ứng dụng di động. Đây là một ngôn ngữ phổ biến và có nhiều nguồn lực và cộng đồng hỗ trợ lớn. Điều này giúp giảm thời gian học và tăng tốc độ phát triển.
3. Giao diện nguyên bản: Ứng dụng React Native sử dụng các thành phần giao diện nguyên bản của hệ điều hành, chứ không phải giao diện web. Điều này giúp ứng dụng chạy mượt mà và tự nhiên trên thiết bị di động, mang lại trải nghiệm người dùng tốt hơn.
4. Hiệu suất cao: Với React Native, ứng dụng được biên dịch thành mã nguồn nền tảng cụ thể. Điều này giúp tăng hiệu suất so với các ứng dụng sử dụng giao diện web. React Native cũng cung cấp các cơ chế tối ưu hóa để đảm bảo hiệu suất tốt trên thiết bị di động.
5. Chia sẻ mã nguồn: Một trong những ưu điểm lớn của React Native là khả năng chia sẻ mã nguồn giữa các nền tảng. Bạn có thể sử dụng lại nhiều thành phần và logic giữa iOS và Android, giảm bớt công việc lặp lại và tăng tốc độ phát triển.
6. Cộng đồng và hỗ trợ: React Native có một cộng đồng sôi động với rất nhiều tài liệu, ví dụ, ví dụ mã, thư viện bổ sung và công cụ phát triển. Bạn có thể tìm kiếm sự giúp đỡ và hỗ trợ từ cộng đồng này khi gặp phải vấn đề hoặc cần tư vấn.

### Nhược điểm

1. Hiệu năng: So với ứng dụng được viết bằng ngôn ngữ natvie (Objective-C/Swift cho iOS, Java/Kotlin cho Android), React Native có thể có hiệu năng không tốt hơn. Điều này do việc sử dụng giao diện nguyên bản và cơ chế trung gian để giao tiếp giữa mã JavaScript và nền tảng di động.
2. Khả năng truy cập các tính năng phức tạp: Mặc dù React Native cung cấp một API để truy cập các tính năng phần cứng của thiết bị, nhưng đôi khi nó không hỗ trợ tất cả các tính năng phức tạp như việc điều khiển đèn flash, quét vân tay, NFC, hoặc các tính năng đặc biệt của các thiết bị di động cụ thể.
3. Thư viện bên thứ ba: Mặc dù có sẵn một số thư viện bên thứ ba cho React Native, nhưng số lượng và chất lượng của chúng vẫn ít hơn so với các nền tảng native. Điều này có thể khiến việc tìm kiếm và sử dụng các thư viện phụ trợ khó khăn hơn.
4. Cấu hình và triển khai phức tạp: Để triển khai một ứng dụng React Native trên các cửa hàng ứng dụng (App Store và Google Play), bạn cần đảm bảo rằng cấu hình và quy trình triển khai được thực hiện chính xác cho cả hai nền tảng. Quy trình này có thể phức tạp hơn so với việc triển khai ứng dụng native đơn giản.
5. Sự phụ thuộc vào phiên bản React Native: React Native là một framework phát triển nhanh và liên tục, và nó thường có các bản cập nhật và thay đổi mới. Điều này có thể tạo ra sự phụ thuộc vào phiên bản cụ thể và khả năng tương thích ngược với các phiên bản cũ hơn hoặc các thư viện và công cụ bên thứ ba.
6. Kỹ năng và tài nguyên nhân lực: Mặc dù JavaScript là một ngôn ngữ phổ biến, việc phát triển ứng dụng React Native yêu cầu kiến thức về React và các khái niệm riêng của nó. Điều này có nghĩa là bạn cần có nhân lực có kỹ năng phù hợp hoặc sẵn sàng đào tạo để làm việc với React Native.

## Tổng quan về Firebase

### Giới thiệu

Firebase là một nền tảng phát triển ứng dụng di động và web của Google. Nó cung cấp các công cụ và dịch vụ để xử lý nhiều khía cạnh khác nhau của ứng dụng, từ cơ sở dữ liệu và xác thực người dùng đến phân tích và thông báo đẩy. [4] Firebase cung cấp một số dịch vụ chính như:

1. Firebase Realtime Database: Đây là một cơ sở dữ liệu thời gian thực lưu trữ dưới dạng JSON. Nó cho phép đồng bộ dữ liệu trực tiếp giữa các thiết bị và các nguồn dữ liệu, cho phép tạo ứng dụng thời gian thực như các ứng dụng trò chơi, ứng dụng chat và nhiều hơn nữa.
2. Firebase Authentication: Dịch vụ xác thực người dùng giúp quản lý việc đăng nhập, đăng ký và xác thực người dùng bằng các phương thức như email/mật khẩu, Google, Facebook, Twitter và nhiều phương thức xác thực khác.
3. Firebase Cloud Firestore: Đây là một cơ sở dữ liệu linh hoạt, phân tán và tài liệu hướng cung cấp khả năng truy vấn linh hoạt và đồng bộ dữ liệu nhanh chóng cho các ứng dụng di động và web.
4. Firebase Storage: Dịch vụ lưu trữ dùng để lưu trữ và quản lý các tệp như hình ảnh, video, âm thanh trong ứng dụng của bạn. Nó cung cấp tích hợp dễ dàng với các dịch vụ khác trong Firebase.
5. Firebase Cloud Functions: Đây là dịch vụ tích hợp để viết và triển khai các chức năng (functions) máy chủ không cần máy chủ riêng biệt. Bạn có thể sử dụng Firebase Cloud Functions để xử lý các sự kiện như việc tạo người dùng, gửi thông báo đẩy và thực hiện các tác vụ máy chủ.
6. Firebase Analytics: Dịch vụ phân tích giúp bạn hiểu rõ hơn về hành vi người dùng, hiệu suất ứng dụng và hiệu quả của chiến dịch tiếp thị. Nó cung cấp các báo cáo thống kê chi tiết và tích hợp với các dịch vụ tiếp thị khác.
7. Ngoài ra, Firebase còn cung cấp các dịch vụ khác như Firebase Cloud Messaging (thông báo đẩy), Firebase Hosting (lưu trữ và triển khai ứng dụng web), Firebase Test Lab (kiểm thử tự động trên nhiều thiết bị di động), và nhiều hơn nữa.

A diagram of a website

Description automatically generated

Hình 2.2: Một số dịch vụ của Firebase

### Ưu điểm

1. Dễ sử dụng: Firebase được thiết kế để đơn giản hóa việc phát triển ứng dụng. Nó cung cấp các giao diện lập trình ứng dụng (API) dễ hiểu và tài liệu phong phú, giúp bạn bắt đầu nhanh chóng và dễ dàng tích hợp các dịch vụ vào ứng dụng của mình.
2. Tích hợp linh hoạt: Firebase cung cấp một loạt các dịch vụ đám mây như cơ sở dữ liệu, xác thực người dùng, lưu trữ, phân tích, thông báo đẩy và nhiều hơn nữa. Tất cả các dịch vụ này được tích hợp chặt chẽ với nhau, cho phép bạn dễ dàng sử dụng và quản lý các tính năng khác nhau trong ứng dụng của mình.
3. Dịch vụ thời gian thực: Firebase cung cấp các cơ sở dữ liệu thời gian thực như Realtime Database và Cloud Firestore. Điều này cho phép bạn đồng bộ dữ liệu ngay lập tức giữa các thiết bị và cung cấp trải nghiệm thời gian thực cho người dùng.
4. Xác thực và quản lý người dùng: Firebase cung cấp một hệ thống xác thực mạnh mẽ và linh hoạt, cho phép bạn xác thực người dùng bằng cách sử dụng email/mật khẩu, các nhà cung cấp xã hội (Google, Facebook, Twitter), và các phương thức xác thực khác. Nó cũng cung cấp các công cụ quản lý người dùng để quản lý thông tin người dùng và cung cấp tính năng quên mật khẩu, xác minh email và nhiều hơn nữa.
5. Tích hợp thông qua SDK: Firebase cung cấp các SDK cho nhiều ngôn ngữ lập trình phổ biến như JavaScript, Swift, Kotlin, và nhiều ngôn ngữ khác. Điều này giúp tích hợp Firebase vào ứng dụng của bạn trở nên dễ dàng và linh hoạt.
6. Hỗ trợ quản lý hạ tầng: Firebase được quản lý và triển khai trên hạ tầng đám mây của Google, giúp bạn loại bỏ nhu cầu quản lý cơ sở hạ tầng và tập trung vào việc phát triển ứng dụng.
7. Hỗ trợ đa nền tảng: Firebase không chỉ hỗ trợ phát triển ứng dụng di động (iOS và Android), mà còn hỗ trợ phát triển ứng dụng web và cả các nền tảng khác như Unity và C++. Điều này giúp bạn phát triển ứng dụng cho nhiều nền tảng khác nhau trong cùng một môi trường phát triển.

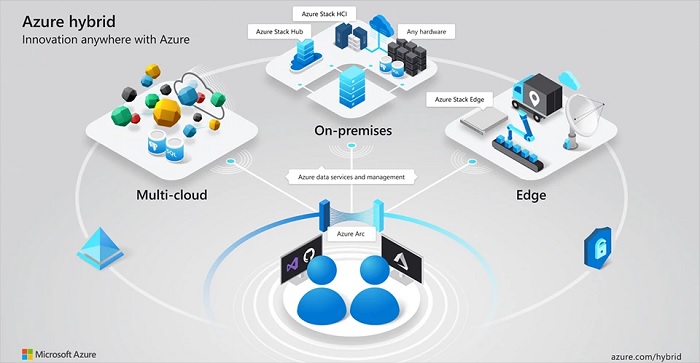
### Nhược điểm

1. Giới hạn miễn phí: Firebase cung cấp một phiên bản miễn phí với giới hạn sử dụng cho mỗi dịch vụ. Khi ứng dụng của bạn phát triển và sử dụng quy mô lớn hơn, bạn có thể phải nâng cấp lên các gói trả phí để đáp ứng nhu cầu sử dụng của mình. Việc chuyển từ miễn phí sang gói trả phí có thể tạo ra chi phí tài chính.
2. Phụ thuộc vào hạ tầng của bên thứ ba: Firebase hoạt động trên hạ tầng đám mây của Google. Mặc dù điều này có nghĩa là bạn không cần quản lý cơ sở hạ tầng của mình, nhưng nó cũng đồng nghĩa với việc phụ thuộc vào sự ổn định và khả năng mở rộng của hạ tầng đám mây của Google.
3. Khả năng tùy chỉnh giới hạn: Firebase cung cấp các dịch vụ tiện ích có sẵn, nhưng có thể gặp khó khăn khi bạn cần một tính năng tùy chỉnh hoặc điều chỉnh phức tạp. Trong một số trường hợp, bạn có thể phải tìm giải pháp bên ngoài hoặc sử dụng các công cụ khác để đáp ứng các yêu cầu cụ thể.
4. Độ phức tạp của cấu hình: Mặc dù Firebase cung cấp giao diện dễ sử dụng và tài liệu phong phú, nhưng cấu hình và triển khai các dịch vụ Firebase có thể trở nên phức tạp đối với người mới bắt đầu hoặc các ứng dụng phức tạp. Điều này đặc biệt đúng khi kết hợp nhiều dịch vụ và xử lý các tác vụ phức tạp hơn.
5. Tùy chỉnh giao diện người dùng: Firebase cung cấp các dịch vụ cho phần backend của ứng dụng, nhưng không cung cấp một khung frontend hoàn chỉnh. Điều này có nghĩa là bạn cần phải xây dựng giao diện người dùng riêng của mình bằng sử dụng các công nghệ frontend như React hoặc Angular.
6. Phụ thuộc vào mạng internet: Firebase yêu cầu kết nối mạng internet để truy cập và sử dụng dịch vụ. Điều này có nghĩa là ứng dụng của bạn phải luôn có kết nối internet để hoạt động và có thể gặp khó khăn khi sử dụng ở các môi trường không có kết nối mạng ổn định.

## Tổng quan về Azure

### Giới thiệu

Microsoft Azure là nền tảng điện toán đám mây công cộng bao gồm: PaaS – nền tảng dưới, IaaS – cơ sở hạ tầng và SaaS – phần mềm được cung cấp bởi Microsoft. Tất cả đều ở dạng dịch vụ. Microsoft Azure được ứng dụng cho những dịch vụ như: lưu trữ, phân tích, mạng, ảo hóa máy tính,… Ngoài ra, Azure còn có thể được dùng hoặc là bổ sung thêm cho các máy chủ ngay tại chỗ. [5]



Hình 2.3: Mô hình Microsoft Azure

Một số dịch vụ chính của Microsoft Azure bao gồm:

1. Azure Virtual Machines: Cung cấp máy ảo có thể tùy chỉnh cho các mục đích như phát triển ứng dụng, chạy dịch vụ và quản lý hệ thống.
2. Azure App Service: Cho phép triển khai và quản lý ứng dụng web, mobile và API một cách dễ dàng.
3. Azure SQL Database: Dịch vụ cơ sở dữ liệu quan hệ trong đám mây, giúp lưu trữ và quản lý dữ liệu hiệu quả.
4. Azure Storage: Dịch vụ lưu trữ đám mây, bao gồm lưu trữ tệp, bảng và hàng loạt dữ liệu.
5. Azure Cognitive Services: Bộ các dịch vụ AI và Machine Learning giúp tích hợp tính năng thông minh vào ứng dụng như nhận dạng hình ảnh, xử lý ngôn ngữ tự nhiên, v.v.
6. Azure DevOps: Dịch vụ hỗ trợ quản lý quy trình phát triển và triển khai ứng dụng.
7. Azure Functions: Cho phép viết và triển khai mã chức năng mà không cần quản lý cơ sở hạ tầng.
8. Azure Machine Learning: Dịch vụ hỗ trợ xây dựng, đào tạo và triển khai mô hình Machine Learning.
9. Azure Kubernetes Service: Hỗ trợ triển khai và quản lý các ứng dụng dựa trên Kubernetes.

Azure cung cấp linh hoạt cho việc triển khai ứng dụng và dịch vụ trên đám mây, từ doanh nghiệp nhỏ đến các tổ chức lớn. Nó giúp tiết kiệm thời gian, tài nguyên và nâng cao khả năng mở rộng của các ứng dụng và dịch vụ của bạn.

### Ưu điểm [5]

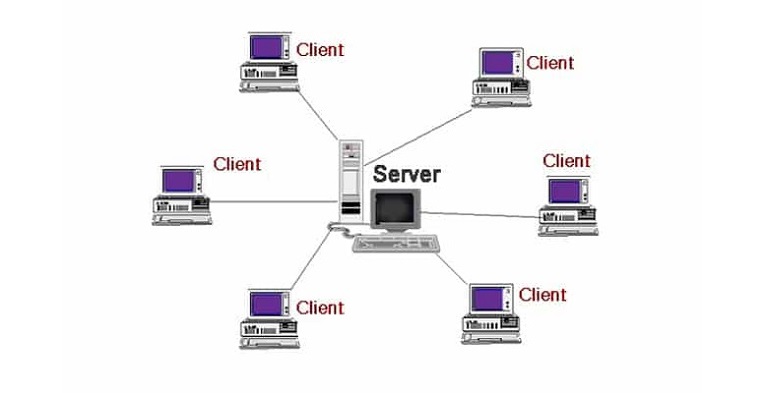
1. Khôi phục dữ liệu và chống thất thoát dữ liệu: Dịch vụ được tích hợp tính năng phục hồi nâng cao và tính linh hoạt, nhờ vậy người dùng có thể sao chép lại dữ liệu bằng hầu hết các loại ngôn ngữ, ở bất kỳ vị trí nào và trên tất cả các hệ điều hành. Chưa hết, dịch vụ Azure còn cho phép người dùng tùy chỉnh quá trình sao chép tự động theo ý của mình. Ngoài ra, trên Microsoft Azure còn có tính năng sao chép ngoại vi với các bản sao được lưu trữ thành 3 bản sao khác nhau tại 3 vị trí của trung tâm dữ liệu và 3 bản sao tại trung tâm dữ liệu Azure.
2. Lưu trữ, phát triển các ứng dụng cho web và di động: Nó sẽ giúp các ứng dụng tự chủ, tự thích ứng với tính năng quản lý vá tự động. Trong đó AutoScale và tích hợp sẽ thực hiện cho các ứng dụng ngay tại chỗ. Với Azure, các bạn cũng có thể liên kết các ứng dụng web với nhau 1 cách liền mạch. Việc truy cập tại các địa điểm khác nhau của nhân viên hay đối tác luôn an toàn với hệ thống tường lửa bên trong.
3. Phân phối và bổ sung Active Directory: Active Directory là một dịch vụ về thư mục được dùng trong Windows Server và bạn hoàn toàn có thể tích hợp nó vào Azure nhằm bổ sung thêm khả năng truy cập và danh tính. Nhờ vậy sẽ giúp cho DNS được phổ rộng toàn cầu, bảo mật tốt hơn và quản lý tập trung. Hiện tại không có nhà cung cấp dịch vụ đám mây nào có khả năng tiếp cận Active Directory giống như Azure.
4. Đổi mới các giải pháp IoT công nghiệp: Với tính linh hoạt, bảo mật cao và khả năng mở rộng, Microsoft Azure là nguồn lực hoàn hảo nhất cho các doanh nghiệp đang hướng tới giải pháp IoT. Người dùng có thể kết nối với những thiết bị của mình qua đám mây thông qua sử dụng giải pháp được tích hợp cùng cơ sở hạ tầng có sẵn. Sau đó thu thập dữ liệu mới về doanh nghiệp.

### Nhược điểm [6]

1. Tiềm ẩn nguy cơ mất quyền kiểm soát tài khoản cho Microsoft
2. Tăng sự cạnh tranh giữa các nguồn lực phát triển
3. Chạy theo kinh doanh số lượng
4. Sự suy giảm trong lợi nhuận trước mắt và doanh thu
5. Tăng tốc toàn cầu khóa và cạnh tranh thị trường
6. Tăng chi phí lưu trữ và tích hợp

## Mô hình client – server

Mô hình client-server là một kiến trúc phần mềm phổ biến trong lập trình và phát triển ứng dụng mạng. Trong mô hình này, hệ thống được chia thành hai thành phần chính: client (khách hàng) và server (máy chủ). Hai thành phần này tương tác với nhau để cung cấp dịch vụ và tài nguyên cho người sử dụng hoặc ứng dụng khác. [7]



Hình 2.4: Mô hình client – server

### Client (Khách hàng)

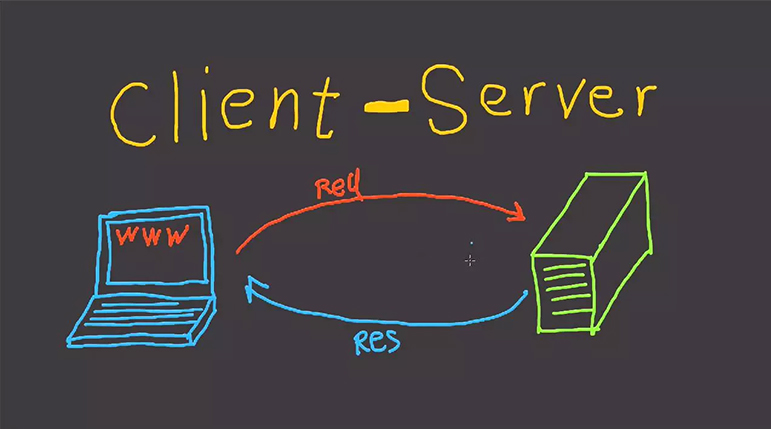
* Chức năng: Là phần giao diện của người dùng hoặc ứng dụng cuối. Người dùng tương tác với client để truy cập và sử dụng dịch vụ hoặc tài nguyên.
* Đặc điểm: Thường có giao diện người dùng đồ họa (GUI) hoặc dòng lệnh (CLI). Có thể chạy trên nhiều loại thiết bị như máy tính, điện thoại di động, máy tính bảng, v.v.

### Server (Máy chủ)

* Chức năng: Là nơi lưu trữ và quản lý dữ liệu, cung cấp dịch vụ, xử lý yêu cầu từ client và trả về kết quả.
* Đặc điểm: Thường có cấu hình phần cứng mạnh mẽ hơn so với client để xử lý nhiều yêu cầu cùng một lúc. Chạy các ứng dụng và dịch vụ có thể được chia sẻ với nhiều client.

### Giao tiếp giữa client và server

* Giao thức: Để truyền thông giữa client và server, thường sử dụng các giao thức như HTTP, TCP/IP, và giao thức tùy chỉnh phát triển.
* Yêu cầu-Phản hồi: Client gửi yêu cầu đến server, và server xử lý yêu cầu đó rồi trả lại kết quả cho client.



Hình 2.5: Nguyên lý hoạt động Client-Server

### Ưu điểm

* Tập trung: Ưu điểm đầu tiên của mô hình Client Server kiểu mạng khách chủ đó chính là khả năng kiểm soát tập trung (Centralization) đã được tích hợp sẵn. Theo như mô hình này thì tất cả mọi thông tin cần thiết đều sẽ được đặt ở một vị trí duy nhất. Đây là một ưu điểm vô cùng hữu ích được những người quản trị viên mạng yêu thích bởi vì họ có thể toàn quyền quản lý cũng như điều hành mọi việc. Tính năng này giúp cho mọi sự cố trong mạng đều sẽ được giải quyết ở cùng một nơi thống nhất. Đồng thời, việc cập nhật cơ sở tài nguyên, dữ liệu cũng sẽ dễ dàng hơn rất nhiều.
* Bảo mật: Trong mạng Client Server, tất cả các dữ liệu đều sẽ được bảo vệ một cách tối đa nhờ vào hệ thống kiến trúc tập trung của mạng. Thông qua đó, nó sẽ giúp người dùng kiểm soát truy cập để chỉ có những ai được cấp quyền truy cập thì mới được thực hiện các thao tác cần thiết. Muốn làm như vậy, chúng ta cần phải áp đặt thông tin đăng nhập cũng như Username hay Password. Bên cạnh đó, nếu dữ liệu của chúng ta bị mất thì các file sẽ được khôi phục một cách vô cùng dễ dàng chỉ từ một bản sao lưu duy nhất mà thôi.
* Khả năng mở rộng: Mô hình mạng kết nối Client Server có khả năng mở rộng vô cùng tốt. Chỉ cần người dùng cần sử dụng bất cứ lúc nào thì họ cũng có thể tăng được số lượng tài nguyên của mình. Ví dụ như số Client hoặc Server. Nhờ đó mà chúng ta có thể tăng kích thước của Server một cách dễ dàng mà không bị gián đoạn nhiều.
* Khả năng truy cập: Hoàn toàn không hề có sự phân biệt giữa các vị trí hay nền tảng với nhau. Tất cả mọi Client đều có khả năng đăng nhập được vào hệ thống mạng máy tính. Điều này sẽ giúp cho tất cả các nhân viên đều có thể truy cập thông tin của công ty một cách dễ dàng mà không cần phải dùng một terminal mode hoặc một bộ xử lý nào khác.

### Nhược điểm

* Tắc nghẽn lưu lượng: Nói về nhược điểm lớn nhất của mô hình mạng Client Server đó chính là tắc nghẽn lưu lượng. Trong trường hợp có quá nhiều Client tạo request từ cùng một Server thì nó có thể sẽ làm cho kết nối chậm hơn. Trong trường hợp xấu nhất còn có thể xuất hiện hiện tượng crash. Khi một server bị quá tải thì sẽ tạo ra nhiều vấn đề khi truy cập thông tin.
* Độ bền: Client Server là mạng tập trung chính vì thế, khi Server chính xảy ra sự cố hoặc bị nhiễu thì cũng đồng nghĩa với việc toàn bộ hệ thống mạng sẽ bị gián đoạn. Như vậy, bạn cần chú ý đó là mạng thiếu tính ổn định và độ bền. Bạn cần chú ý khi thực hiện.
* Chi phí: Chi phí được sử dụng để thiết lập và bảo trì Server trong Client Server thường sẽ khá cao. Lý do là vì các hệ thống mạng có sức mạnh rất lớn cũng đồng nghĩa với việc giá để chi cho việc này là rất đắt. Chính vì vậy, không phải ai cũng có khả năng chỉ trả và sử dụng.
* Bảo trì: Khi các Server thực hiện triển khai để làm việc thì nó cũng sẽ hoạt động một cách không ngừng nghỉ. Điều này đồng nghĩa với việc chúng ta cần phải quan tâm đến việc bảo trì hệ thống đúng mức. Khi xảy ra bất cứ vấn đề gì cũng cần phải giải quyết ngay lập tức. Vậy nên, cần phải có một nhà quản lý mạng chuyên biệt để tiến hành duy trì hoạt động của Server khi chúng được đưa vào và sử dụng.
* Tài nguyên: Một điều mà chúng ta rất cần phải lưu ý đó chính là không phải tất cả tài nguyên hiện có trên Server đều sử dụng được. Ví dụ một cách đơn giản đó chính là chúng ta không thể in trực tiếp được tài liệu từ trên web cũng như tiến hành chỉnh sửa bất kỳ một thông tin nào trên ổ cứng của Client cả.

## Vision-and-Language Pre-training

Các mô hình Vision-and-Language pretraining (VLP) tỏ ra hiệu quả trong việc cải thiện các downstream task liên quan đến sự kết hợp thông tin cả ngôn ngữ và hình ảnh. Để đưa vào mô hình VLP, các pixel ảnh cần được embed cùng với các language token. Cho đến nay, hầu hết các nghiên cứu VLP đều tập trung vào việc cải thiện hiệu suất bằng cách cải thiện bước embed hình ảnh. Tuy nhiên bước embed này có những hạn chế trong thế giới thực do sử dụng mô hình thường rất nặng, dẫn đến tốc độ chậm khi thực hiện truy vấn trong thế giới thực. [8]

### Sự tương tác giữa 2 phương thức

Cốt lõi của các mô hình VLP hiện đại là Transformer. Mô hình nhận đầu vào là các embedding của hình ảnh, văn bản, mô hình hóa giữa các phương thức và tương tác giữa các phương thức qua các lớp, sau đó xuất ra một chuỗi feature được ngữ cảnh hóa.

### Phân loại mô hình Vision-and-Language

Trong nghiên cứu “ViLT: Vision-and-Language Transformer Without Convolution or Region Supervision”, nhóm tác giả đề xuất phân loại các mô hình Vision-and-Language dựa trên hai thông tin sau:

* So sánh hai phương thức về mặt tham số và tài nguyên tính toán
* Hai phương thức có tương tác trong một mạng học sâu hay không

Dựa vào 2 thông tin trên ta có 4 trường hợp như hình dưới. Độ cao của hình chữ nhật biểu thị cho tài nguyên tính toán tương đối của module đó. VE, TE và MI lần lượt là viết tắt của trình nhúng trực quan, trình nhúng văn bản và tương tác phương thức. [8]

A diagram of a software development process

Description automatically generated

Hình 2.6: Bốn loại mô hình Vision-and-Language

* Với hình (a) ta có model VSE++ và SCAN. 2 model này sử dụng các embedder riêng biệt cho ảnh và văn bản (với ảnh thì nặng hơn). Sau đó mô hình biểu diễn sự tương đồng giữa các embedded feature từ 2 mô hình với phép nhân vô hướng đơn giản hoặc các lớp shallow attention.
* Với hình (b) ta có model CLIP sử dụng transformer embedder cho 2 phương thức riêng biệt với độ phức tạp tương đồng. Tương tác giữa pooled image vector và text vector vẫn là "nông" (sử dụng tích vô hướng). Tuy nhiên, việc sử dụng các embedder mạnh cho từng phương thức không đảm bảo rằng mô hình sẽ học hiệu quả các task vision-and-language phức tạp. Điển hình như nếu ta fine-tuning MLP head trên NLVR2 thay cho dot product của CLIP thì độ chính xác chỉ khoảng 50.99 vì vậy các biểu diễn không có khả năng học task này. Vì vậy điều này thúc đẩy việc cần xây dựng một cấu trúc tương tác giữa các phương thức chặt chẽ hơn
* Với hình (c), các mô hình VLP gần đây sử dụng một deep transformer để mô hình hóa tương tác các feature ảnh và văn bản. CNN vẫn sử dụng để trích xuất và embed image feature. Điều này làm cho mô hình có kích thước lớn, yêu cầu nhiều hơn và tài nguyên tính toán.
* ViLT chính là mô hình kinh điển đầu tiên cho hình (d). Trong đó lớp embedding nông, nhẹ. Do đó, kiến trúc này tập trung hầu hết các tính toán vào việc mô hình hóa các tương tác giữa các phương thức.

## ViLT: Vision-and-Language Transformer Without Convolution or Region Supervision

Vision-and-Language Pre-training (VLP) đã được cải thiện hiệu suất trong các nhiệm vụ chung về ngôn ngữ và thị giác khác nhau. Cách tiếp cận VLP hiện tại phụ thuộc rất nhiều vào các quy trình của việc trích xuất đặc trưng hình ảnh. Hầu hết trong số đó liên quan đến giám sát khu vực (ví dụ: phát hiện đối tượng) và kiến ​​trúc tích chập (ví dụ: ResNet). Hai hạn chế lớn của các phương pháp tiếp cận hiện tại là: (1) hiệu quả và tốc độ bởi việc trích xuất đầu vào đơn giản đòi hỏi nhiều tính toán hơn các bước tương tác đa phương thức và (2) là sức mạnh biểu đạt bởi nó cao hơn giới hạn bởi sức mạnh biểu cảm của hình ảnh bộ nhúng và từ vựng trực quan được xác định trước của nó. [9] Trong nghiên cứu “ViLT: Vision-and-Language Transformer Without Convolution or Region Supervision”, nhóm nghiên cứu đã giới thiệu một mô hình VLP tối giản - Vision-and-Language Transformer (ViLT) - có tính đồng nhất trong việc xử lý đầu vào thị giác, nơi quá trình đơn giản hóa đến mức chỉ sử dụng cùng một cách tiếp cận không sử dụng tích chập như chúng ta xử lý đầu vào văn bản. Họ đã chỉ ra rằng ViLT nhanh gấp đôi, thậm chí là nhiều lần so với các mô hình VLP trước đó nhưng vẫn có hiệu suất tác vụ hạ nguồn tốt và cạnh tranh hơn.

A diagram of a software development process

Description automatically generated

Hình 2.7 So sánh trực quan các kiến trúc VLP thông thường và ViLP được đề xuất

ViLT có kiến trúc ngắn gọn như một mô hình VLP với một đường ống nhúng hình ảnh tối thiểu và tuân theo phương pháp dòng đơn. [8] Trong mô hình ViLP được đề xuất, nhóm tác giả đã loại bỏ hoàn toàn mạng lưới tích chập thần kinh từ đường dẫn VLP mà không ảnh hưởng đến hiệu suất của các tác vụ xuôi dòng. ViLT là mô hình VLP đầu tiên trong đó các thành phần dành riêng cho phương thức yêu cầu tính toán ít hơn so với thành phần máy biến áp cho các tương tác đa phương thức.

A diagram of a machine

Description automatically generated

Hình 2.8: Tổng quan mô hình (lấy cảm hứng từ Dosovitskiy et al)

### Các bước quan trọng trong ViLT

1. Embedding bắt đầu với biểu diễn của văn bản (t) và hình ảnh (v), cộng với ma trận nhúng vị trí (pos).
2. Tạo ra vector kết hợp z bằng cách ghép các vector nhúng với vector kiểu modal tương ứng và kết hợp chúng thành một chuỗi kết hợp z₀.
3. Vector ngữ cảnh z được cập nhật thông qua D lớp transformer để tạo ra chuỗi ngữ cảnh cuối cùng z₋D.
4. p là biểu diễn gộp của toàn bộ đầu vào đa dạng, được thu được bằng cách áp dụng chiếu tuyến tính và hàm tanh lên chỉ số đầu tiên của chuỗi z₋D.

### Kiến trúc cụ thể của ViLT

* Mỗi khối trong ViLT bao gồm một lớp tự chú ý đa đầu (MSA) và một lớp MLP.
* Mô hình sử dụng lớp chuẩn hóa (LN) sau MSA và MLP, khác với BERT (post-norm), ViT sử dụng LN trước MSA và MLP (pre-norm).
* Đầu vào văn bản t được nhúng thành t¯ với ma trận nhúng từ T và Tₚₒₛ (ma trận nhúng vị trí).
* Đầu vào hình ảnh I được chia thành các miếng và làm phẳng để tạo ra v. Sau đó, nó được nhúng thành v¯ với ma trận nhúng V và Vₚₒₛ (ma trận nhúng vị trí).
* Tất cả các vector nhúng được cộng với vector kiểu modal tương ứng và ghép nối để tạo thành chuỗi kết hợp z₀.
* Vector ngữ cảnh z được cập nhật qua D lớp transformer để tạo ra chuỗi ngữ cảnh z₋D.
* Biểu diễn gộp p của toàn bộ đầu vào được thu được bằng cách áp dụng chiếu tuyến tính và hàm tanh.

### Thông số mô hình

* Sử dụng trọng số từ ViT-B/32 tiền đào tạo trên ImageNet, do đó có tên là ViLT-B/32.
* Kích thước ẩn H là 768, độ sâu lớp D là 12, kích thước miếng P là 32, kích thước MLP là 3.072, và số lượng đầu chú ý là 12.

Tóm tắt lại, ViLT là kiến trúc đơn giản nhất cho đến nay cho một mô hình thị giác và ngôn ngữ vì nó giao cho mô-đun transformer để trích xuất và xử lý đặc trưng thị giác thay vì một bộ nhúng thị giác sâu riêng biệt. Thiết kế này tự nhiên dẫn đến hiệu suất thời gian chạy và tham số đáng kể. Lần đầu tiên đạt được hiệu suất cạnh tranh trong các nhiệm vụ thị giác và ngôn ngữ mà không sử dụng đặc trưng vùng hoặc bộ nhúng thị giác tích chập sâu nói chung. Đồng thời nhóm tác giả đã chứng minh theo cách thực nghiệm rằng việc che mặt toàn bộ từ và tăng cường hình ảnh, những điều chưa từng xuất hiện trong các kế hoạch đào tạo VLP trước đây, đồng thời giảm hiệu suất thụ động. [9] ViLT đại diện cho một sự đổi mới trong thiết kế mô hình thị giác và ngôn ngữ, tập trung vào sự đơn giản và hiệu suất cao thông qua việc tối ưu hóa tương tác giữa các chế độ thông tin.

# THIẾT KẾ DỮ LIỆU

## Sơ đồ cơ sở dữ liệu quan hệ

A screenshot of a computer

Description automatically generated

Hình 3.1: Sơ đồ cơ sở dữ liệu quan hệ

## Danh sách bảng dữ liệu

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên bảng dữ liệu** | **Mô tả** |
|  | NGUOIDUNG | Lưu trữ thông tin người dùng |
|  | LICHSU | Lưu trữ lịch sử hỏi đáp |

Bảng 3.1: Danh sách các bảng dữ liệu

## Mô tả chi tiết các bảng dữ liệu

### Bảng NGUOIDUNG

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
|  | MaND | String | Khóa chính | Mã người dùng |
|  | Email | String |  | Email người dùng |
|  | HoTen | String |  | Họ tên người dùng |
|  | SDT | String |  | Số điện thoại |
|  | NickName | String |  | Nickname |
|  | Avatar | String |  | Link avarta |

Bảng 3.2: Mô tả chi tiết bảng NGUOIDUNG

### Bảng LICHSU

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
|  | MaLS | String | Khóa chính | Mã lịch sử |
|  | HinhAnh | String |  | Hình ảnh đầu vào |
|  | CauHoi | String |  | Câu hỏi ứng với hình ảnh |
|  | TraLoi | String |  | Câu trả lời tương ứng |
|  | ThoiGian | Datetime |  | Thời gian hỏi |
|  | TinhTrang | String |  | Tình trạng câu hỏi |

Bảng 3.3: Mô tả chi tiết bảng LICHSU

# THIẾT KẾ GIAO DIỆN

## Danh sách các màn hình

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên màn hình** | **Chức năng** |
|  | Màn hình đăng nhập | Cho phép người dùng nhập thông tin đăng nhập để sử dụng ứng dụng |
|  | Màn hình đăng ký | Cho phép người dùng tạo tài khoản mới |
|  | Màn hình đăng ký thành công | Thông báo người dùng đăng ký tài khoản thành công |
|  | Màn hình quên mật khẩu | Cho phép người dùng nhập email và xác nhận. Tiếp đó hệ thống sẽ gửi email chứa đường link cập nhật mật khẩu về email hợp lệ mà người dùng đã nhập |
|  | Màn hình gửi emial thành công | Thông báo gửi email xác thực thành công |
|  | Màn hình trang chủ | Màn hình chính của ứng dụng. Cho phép người dùng thực hiện chức năng hỏi đáp tại đây |
|  | Màn hình lịch sử | Hiển thị lịch sử hỏi đáp của người dùng |
|  | Màn hình chi tiết lịch sử câu hỏi | Hiển thị chi tiết lịch sử hỏi đáp được chọn |
|  | Màn hình quản lý tài khoản | Cho phép người dùng chọn các chức năng quản lý tài khoản |
|  | Màn hình thông tin tài khoản | Cho phép người dùng xem và cập nhật thông tin tài khoản |
|  | Màn hình đổi mật khẩu | Cho phép người dùng cập nhật mật khẩu |
|  | Màn hình thông tin ứng dụng | Hiển thị thông tin ứng dụng và cách xử lý khi gặp sự cố |

Bảng 4.1: Danh sách các màn hình

## Chi tiết các màn hình

### Màn hình đăng nhập

**A screenshot of a login screen

Description automatically generated**

Hình 4.1: Màn hình đăng nhập

### Màn hình đăng ký

A screenshot of a login form

Description automatically generated

Hình 4.2: Màn hình đăng ký

### Màn hình đăng ký thành công

A cartoon of a white rabbit

Description automatically generated

Hình 4.3: Màn hình đăng ký thành công

### Màn hình quên mật khẩu

A screenshot of a login screen

Description automatically generated

Hình 4.4: Màn hình quên mật khẩu

### Màn hình gửi email thành công

A screenshot of a cell phone

Description automatically generated

Hình 4.5: Màn hình đổi email thành công

### Màn hình trang chủ

A screenshot of a chat

Description automatically generated

Hình 4.6: Màn hình trang chủ

### Màn hình lịch sử

A screenshot of a phone

Description automatically generated

Hình 4.7: Màn hình lịch sử

### Màn hình chi tiết lịch sử câu hỏi

A screenshot of a cartoon

Description automatically generated

Hình 4.8: Màn hình chi tiết lịch sử câu hỏi

### Màn hình quản lý tài khoản

A screenshot of a phone

Description automatically generated

Hình 4.9: Màn hình quản lý tài khoản

### Màn hình thông tin tài khoản

A screenshot of a phone

Description automatically generated

Hình 4.10: Màn hình thông tin tài khoản

### Màn hình đổi mật khẩu

A screenshot of a frog login

Description automatically generated

Hình 4.11: Màn hình đổi mật khẩu

### Màn hình thông tin ứng dụng

A screenshot of a phone

Description automatically generated

Hình 4.12: Màn hình thông tin ứng dụng

# CÀI ĐẶT VÀ THỬ NGHIỆM

## Kết luận

Ứng dụng hỏi đáp thị giác là một công cụ được phát triển để sử dụng trên thiết bị di động, nhằm giải quyết các thách thức trong việc hiểu và tương tác với hình ảnh thông qua ngôn ngữ tự nhiên. Với khả năng nhận diện và trả lời câu hỏi dựa trên nội dung thị giác, ứng dụng mang lại trải nghiệm người dùng tương tác và thông tin một cách hiệu quả.

### Ưu điểm

* Tương Tác Thông Minh: Ứng dụng cho phép người dùng tương tác với hình ảnh thông qua ngôn ngữ tự nhiên, tạo ra trải nghiệm tương tác tự nhiên và gần gũi.
* Trí Tuệ Nhân Tạo và Thị Giác Máy: Sử dụng trí tuệ nhân tạo và thị giác máy để nhận diện và hiểu nội dung của hình ảnh, mở ra khả năng hiểu biết và xử lý thông tin phức tạp.
* Ứng Dụng Đa Nhiệm: Có khả năng đồng thời xử lý nhiều nhiệm vụ, từ việc mô tả hình ảnh đến việc trả lời câu hỏi, mở rộng khả năng sử dụng của ứng dụng.
* Tương Tác Tự Nhiên: Giao diện sử dụng ngôn ngữ tự nhiên để tương tác giúp người dùng có trải nghiệm gần gũi và thuận tiện.
* Khả năng tích hợp: có thể sử dụng cho cả hệ điều hành android và IOS
* Trải Nghiệm Người Dùng Tích Cực: Tạo ra trải nghiệm người dùng tích cực và thuận lợi khi tương tác với thế giới xung quanh thông qua hình ảnh.

### Nhược điểm

* Độ Chính Xác: Độ chính xác của hệ thống nhận diện và xử lý hình ảnh chưa cao dẫn đến kết quả câu hỏi có thể chưa hoàn toàn chính xác và đáng tin cậy.
* Hiệu Suất: Quá trình gửi yêu cầu và nhận phản hồi đôi khi khá chậm.

## Hướng phát triển

* Độ Chính Xác: Tăng cường độ chính xác của hệ thống nhận diện và xử lý hình ảnh để đảm bảo kết quả câu hỏi là chính xác và đáng tin cậy.
* Hiệu suất: Tăng khả năng của ứng dụng trong việc hiểu và xử lý hình ảnh giúp người dùng tận dụng thông tin từ thế giới xung quanh một cách dễ dàng và nhanh chóng.
* Mở Rộng Nguồn Dữ Liệu: Đảm bảo rằng ứng dụng được đào tạo trên một loạt lớn các dữ liệu để đảm bảo tính đa dạng trong quá trình nhận diện.
* Tối Ưu Giao Diện Người Dùng: Cải thiện trải nghiệm người dùng bằng cách tối ưu hóa giao diện để làm cho việc đặt câu hỏi và hiển thị kết quả trở nên linh hoạt và trực quan
* Đa ngôn ngữ: Mở rộng hỗ trợ đa ngôn ngữ và văn bản để phục vụ nhiều nhóm đối tượng người dùng khác nhau.
* Khả năng tích hợp: Có thể tích hợp vào nhiều ứng dụng khác nhau, từ ứng dụng di động đến trang web và hệ thống tự động hóa, tăng tính linh hoạt và sử dụng rộng rãi.
* Tuân Thủ Quy Định Bảo Mật và Quyền Riêng Tư: Bảo vệ dữ liệu người dùng và tuân thủ các quy định bảo mật và quyền riêng tư

**TÀI LIỆU THAM KHẢO**

|  |  |
| --- | --- |
| [1] | Admin, "VNCoder," [Online]. Available: https://vncoder.vn/tin-tuc/chia-se/deep-learning-la-gi-lich-su-cua-deep-learning. |
| [2] | Qi Wu, Damien Teney, Peng Wang, Chunhua Shen, Anthony Dick, Anton van den Hengel, "Visual question answering: A survey of methods and datasets," *Computer Vision and Image Understanding,* 2017. |
| [3] | N. Hưng, 26 03 2022. [Online]. Available: https://vietnix.vn/react-native-la-gi/. |
| [4] | B. Stonehem, Google Android Firebase: Learning the Basics, 2016. |
| [5] | "viettelstore," [Online]. Available: https://viettelstore.vn/tin-tuc/microsoft-azure-la-gi. |
| [6] | giangpth, "BìzlyCloud," 02 05 2018. [Online]. Available: https://bizflycloud.vn/tin-tuc/12-loi-ich-va-rui-ro-cua-windows-azure-2018042616174366.htm. |
| [7] | [Online]. Available: https://fptcloud.com/client-server/. |
| [8] | "viblo," [Online]. Available: https://viblo.asia/p/paper-reading-vilt-vision-and-language-transformer-without-convolution-or-region-supervision-yZjJYxvX4OE. |
| [9] | Kim, Wonjae and Son, Bokyung and Kim, Ildoo, "ViLT: Vision-and-Language Transformer Without Convolution or Region Supervision," *Proceedings of the 38th International Conference on Machine Learning,* vol. 139, pp. 5583--5594, 2021. |
| [10] | Khanh Quoc Tran, An Trong Nguyen, An Tran-Hoai Le, Kiet Van Nguyen, "ViVQA: Vietnamese Visual Question Answering," *Proceedings of the 35th Pacific Asia Conference on Language, Information and Computation,* 2021. |
| [11] | Zhou Yu, Jun Yu, Yuhao Cui, Dacheng Tao, Qi Tian, "Deep Modular Co-Attention Networks for Visual Question Answering," *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition,* 2019. |