**ĐẠI HỌC ĐÀ NẴNG**

**TRƯỜNG ĐẠI HỌC BÁCH KHOA**



**KHOA CÔNG NGHỆ THÔNG TIN**

**BÁO CÁO**

**TÌM KIẾM HÌNH ẢNH BẰNG CAPTION**

**GIẢNG VIÊN HƯỚNG DẪN**

**TS. HUỲNH HỮU HƯNG**

**SINH VIÊN THỰC HIỆN:**

1. Hồ Minh Hiếu, 102190310, 19.16A
2. Phạm Minh Tiến, 102190340, 19.16A
3. Nguyễn Minh Quang, 102190334, 19.16A
4. Hồ Ngọc Hoàng Minh, 102190324, 19.16A

**Đà Nẵng, 12/2022**

**MỤC LỤC**

[DANH MỤC HÌNH ẢNH 3](#_Toc119963704)

[LỜI NÓI ĐẦU 4](#_Toc119963705)

[I. QUÁ TRÌNH TRIỂN KHAI 5](#_Toc119963706)

[1. Danh sách công việc 5](#_Toc119963707)

[2. Phân công nhiệm vụ 6](#_Toc119963708)

[II. GIỚI THIỆU 7](#_Toc119963709)

[1. Mục đích 7](#_Toc119963710)

[2. Phạm vi 7](#_Toc119963711)

[3. Mô tả 7](#_Toc119963712)

[III. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG 7](#_Toc119963713)

[1. Chuẩn bị dữ liệu 7](#_Toc119963714)

[2. Mô hình huấn luyện 9](#_Toc119963715)

[3. Điều chỉnh tham số 17](#_Toc119963716)

[IV. KẾT QUẢ VÀ HƯỚNG PHÁT TRIỂN 18](#_Toc119963717)

[1. Kết quả chương trình 18](#_Toc119963718)

[2. Hướng phát triển 19](#_Toc119963719)

[TÀI LIỆU THAM KHẢO 20](#_Toc119963720)

# DANH MỤC HÌNH ẢNH

Hình 1: Output mô phỏng 7

Hình 2: Mô tả sử dụng hàm dịch ngôn ngữ trong google sheet 8

Hình 3: Kiến trúc Residual block 10

Hình 4: Các thông số chi tiết của mô hình Resnet 10

Hình 5: Kiến trúc model DenseNet 11

Hình 6: Khối các lớp tích chập với các kết quả được nối với nhau. 11

Hình 7: Các thông số chi tiết của mô hình DenseNet 11

Hình 8: Kết quả train so với các mô hình khác 12

Hình 9: Ví dụ cho việc thu phóng mô hình trên các kích thước khác nhau 13

Hình 10: So sánh việc thu phóng theo chiều rộng và theo các độ sâu, độ phân giải ở các mức độ. 13

Hình 11: Kiến trúc mạng Efficient Net 14

Hình 12: Kiến trúc model Xception 14

Hình 13: Kiến trúc gồm nhiều layers tại encoder của model BERT 16

Hình 14: Kiến trúc của CLIP model 16

Hình 15: Giao diện trang web tìm kiếm của nhóm 18

Hình 16: Giao diện trang web chức năng upload 19

# LỜI NÓI ĐẦU

Trong những thập niên gần đây, sự phát triển mạnh mẽ của khoa học công nghệ mà đi đầu là lĩnh vực công nghệ thông tin đã làm thay đổi sâu sắc trong đời sống xã hội và nhận thức của con người. Từ đó, việc ứng dụng những thành quả của khoa học công nghệ vào trong đời sống sinh hoạt để đáp ứng những nhu cầu của con người một cách nhanh chóng, hiệu quả hơn đã trở thành xu hướng tất yếu.

Cùng với sự phát triển đó, con người đã có thể kết nối, trao đổi và làm việc với nhau bất kể khoảng cách địa lý thông qua các thiết bị thông minh. Chính vì vậy, việc truy cập thông tin cũng trở nên rất phổ biến và rộng rãi hơn bao giờ hết trong giai đoạn Covid vừa qua. Do đó, để có thể hỗ trợ người dùng nâng cao khả năng tìm kiếm thông tin một cách dễ dàng và tiện lợi hơn. Chúng em đã quyết định thực hiện đề tài “**Tìm kiếm hình ảnh bằng caption**”.

Đến với đề tài nghiên cứu này, chúng em mong muốn có thể tạo ra một ứng dụng hỗ trợ việc tìm kiếm thông tin một cách nhanh chóng, tiện lợi và mang lại độ chính xác cao. Từ đó, có thể mở rộng phát triển và triển khai trên các hệ thống, dự án mang quy mô lớn.

Vì thời gian triển khai có hạn và việc tìm hiểu công nghệ mới còn gặp nhiều khó khăn, nên trong quá trình nghiên cứu cũng như làm bài báo cáo, khó tránh khỏi sai sót, rất mong thầy cô thông cảm và bỏ qua. Bên cạnh đó, do còn hạn chế về kiến thức và kinh nghiệm thực tiễn, chúng em rất mong nhận được những ý kiến đóng góp quý báu của quý thầy cô để đề tài nghiên cứu này được hoàn thiện. Cuối cùng, chúng em xin chân thành cảm ơn thầy Huỳnh Hữu Hưng của Khoa CNTT đã luôn nhiệt tình hướng dẫn để chúng em hoàn thành tốt đề tài này.

# QUÁ TRÌNH TRIỂN KHAI

## Danh sách công việc

*Bảng 1: Bảng danh sách công việc*

|  |  |
| --- | --- |
| **Ngày** | **Nội dung công việc** |
| 29/08/2022 | Báo cáo quá trình tìm hiểu đề tài, dataset |
| 05/09/2022 | Xác định hướng đi của bài toán (chuyển label sang tiếng việt).  Xem các phương pháp chuyển label từ tiếng anh sang tiếng việt |
| 12/09/2022 | Vietnamese Word Segmentation (VNCoreNLP), PhoBERT |
| 19/09/2022 | Model Xception, Densenet |
| 03/10/2022 | Model ResNet, EfficientNet |
| 10/10/2022 | Train model Xception, DenseNet, ResNet, EfficientNet Demo |
| 17/10/2022 | Code data augmentation |
| 31/10/2022 | Train, tính lại độ chính xác trên tập train |
| 07/11/2022 | Thay đổi tham số đầu vào cải thiện thuật toán |
| 14/11/2022 | Viết báo cáo |
| 21/11/2022 | Hoàn thiện báo cáo |

## Phân công nhiệm vụ

*Bảng 2: Bảng phân công nhiệm vụ*

|  |  |
| --- | --- |
| **Họ tên** | **Nhiệm vụ** |
| **Phạm Minh Tiến (L)** | * Quản lý tiến độ công việc * Xác định hướng đi bài toán và phương pháp * Tìm hiểu các phương pháp dịch câu với số lượng lớn * Tìm hiểu image encoder … * Triển khai model thành các API với Flask |
| **Hồ Minh Hiếu** | * Tìm hiểu phương pháp tiền xử lý cho text encoder bằng VNCoreNLP * Tìm hiểu các phương pháp dịch câu với số lượng lớn * Tìm hiểu image encoder … * Nghiên cứu, đánh giá và thử nghiệm việc sử dụng các loại data augmentation. * Hoàn thiện báo cáo |
| **Nguyễn Minh Quang** | * Xử lý các dữ liệu bị missing và bị sai * Tìm hiểu các phương pháp dịch câu với số lượng lớn * Tìm hiểu text encoder PhoBert * Tìm hiểu image encoder … * Cài đặt giao diện để dễ dàng sử dụng |
| **Hồ Ngọc Hoàng Minh** | * Tìm hiểu image encoder EfficientNet * Tổng hợp thông tin và cài đặt chương trình để có thể train và đánh giá model. * Tạo ra metric để đánh giá model. * Nghiên cứu điều chỉnh các bộ tham số, cải thiện độ chính xác của model. |

# GIỚI THIỆU

## Mục đích

Tài liệu mô tả các yêu cầu, phân tích, thiết kế hệ thống cũng như quá trình triển khai và kết quả chương trình của đề tài “*Tìm kiếm hình ảnh bằng caption.*”.

Với sự trợ giúp của tài liệu này sẽ giúp người đọc nắm rõ được nội dung, quy trình cũng như các kỹ thuật đã sử dụng của dự án. Nói cách khác, tài liệu này sẽ cung cấp cơ sở để xác nhận và xác minh.

## Phạm vi

Hiện tại, dự án chỉ được triển khai trên quy mô học tập và nghiên cứu, nên sản phẩm cụ thể chỉ là các kết quả của việc nghiên cứu cũng như demo đơn giản đề làm rõ kết quả đó.

Hơn nữa, dự án nghiên cứu này có thể được áp dụng rộng rãi hơn thông qua việc tích hợp vào trong các dự án lớn. Là cơ sở, tiền đề để phát triển dự án tốt nghiệp cũng như các dự án lớn khác.

## Mô tả

Input: Là một đoạn caption ngắn mô tả cho một hành động, một sự vật, một sự việc,..

Output: Kết quả trả về là những tấm ảnh có độ tương quan lớn nhất đối với caption.

*Ví dụ:*

* *Input:* Hai con chó trên biển.
* *Output:*



*Hình 1: Output mô phỏng*

# PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

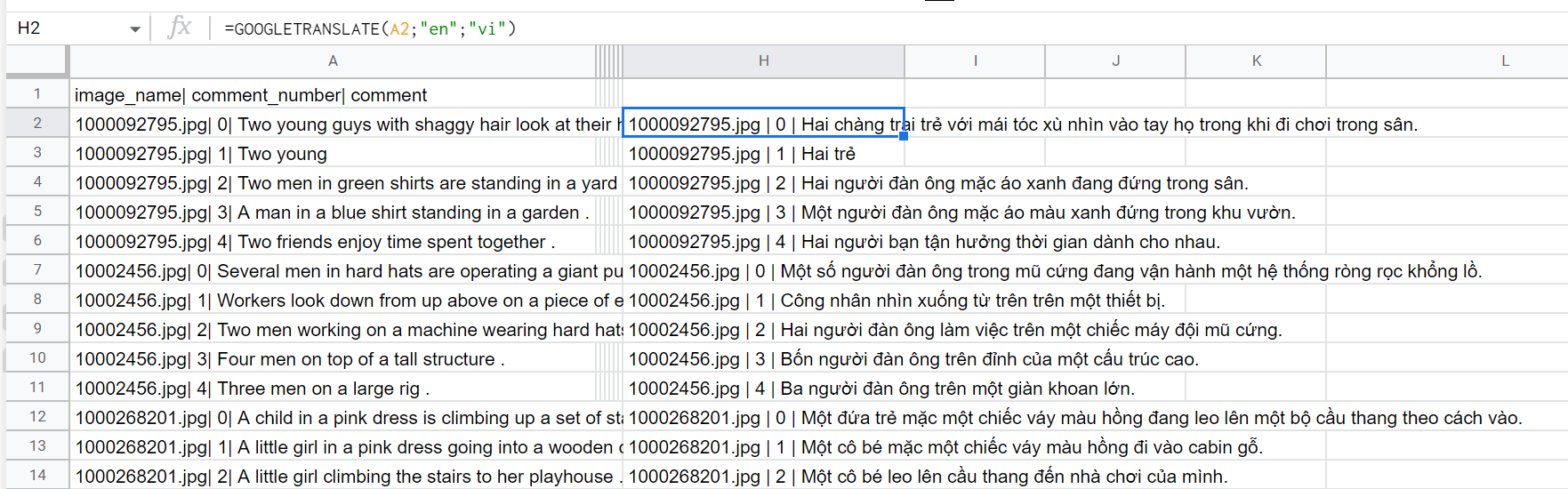
## Chuẩn bị dữ liệu

1. **Lọc dữ liệu:**

Trong dữ liệu tại dòng thứ 19999 bị thiếu, team đã fill dòng đó với caption là: “A dog runs across the grass.".

1. **Dịch câu:**

Đầu tiên ta cần chuyển các caption hiện tại từ tiếng anh sang tiếng việt, bằng cách sử dụng function có sẵn của google sheet để dịch từ tiếng anh sang tiếng việt.



*Hình 2: Mô tả sử dụng hàm dịch ngôn ngữ trong google sheet*

Ở trong hình phía trên ta đang sử dụng function GOOGLE TRANSLATE để dịch các caption từ tiếng anh sang tiếng việt. Tham số đầu tiên là tọa độ tại ô mà ta muốn dịch, tham số thứ hai chọn loại ngôn ngữ gốc và ngôn ngữ muốn phiên dịch (trong trường hợp ở trên ta đang dịch từ tiếng anh sang tiếng việt).

Sau khi đã chuyển được các caption từ tiếng anh sang tiếng việt ta bắt đầu kiểm tra độ chính xác của việc chuyển đổi.

*Bảng 3: Mô tả số lượng câu sai trong việc dịch ngôn ngữ*

|  |  |
| --- | --- |
| **Các câu kiểm tra** | **Số lượng sai** |
| 23 -> 32 | 1 (32) |
| 1653 -> 1662 | 1 (1652) |
| 2047 -> 2056 | 1 (2048) |
| 4957 -> 4966 | 1 (4958) |

Ta chọn ngẫu nhiên 4 cụm dữ liệu, mỗi cụm gồm 10 caption. Sau khi kiểm tra thì mỗi cụm dịch sai 1 câu. Các câu dịch sai này chủ yếu dịch sai về cấu trúc câu, còn các từ trong câu vẫn dịch đúng. Cụ thể như bên dưới.

*Bảng 4: So sánh chi tiết lỗi của các câu dịch sai*

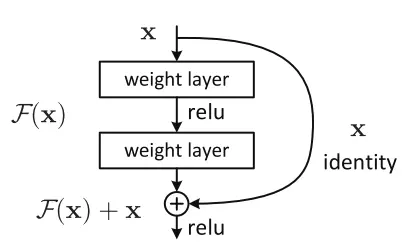
|  |  |  |
| --- | --- | --- |
| Câu | Hình ảnh | Chuyển đổi |
| 32 |  | Câu gốc: A man sits in a chair while holding a large stuffed animal of a lion.  Câu dịch: Một người đàn ông ngồi trên ghế trong khi cầm một con vật nhồi bông lớn của một con sư tử. |
| 1652 |  | Câu gốc: A black dog playing fetch with a ball of rope.  Câu dịch: Một con chó đen chơi lấy một quả bóng của dây thừng. |
| 2048 |  | Câu gốc: The little boy is wearing a black pinstripe shirt and walking on a deck.  Câu dịch: Cậu bé đang mặc một chiếc áo phông đen đen và đi trên một bộ bài. |
| 4958 |  | Câu gốc: Sled dogs trekking through snow near a forest.  Câu dịch: Chó trượt tuyết trekking qua tuyết gần một khu rừng. |

## Mô hình huấn luyện

* 1. **Image encoder:**

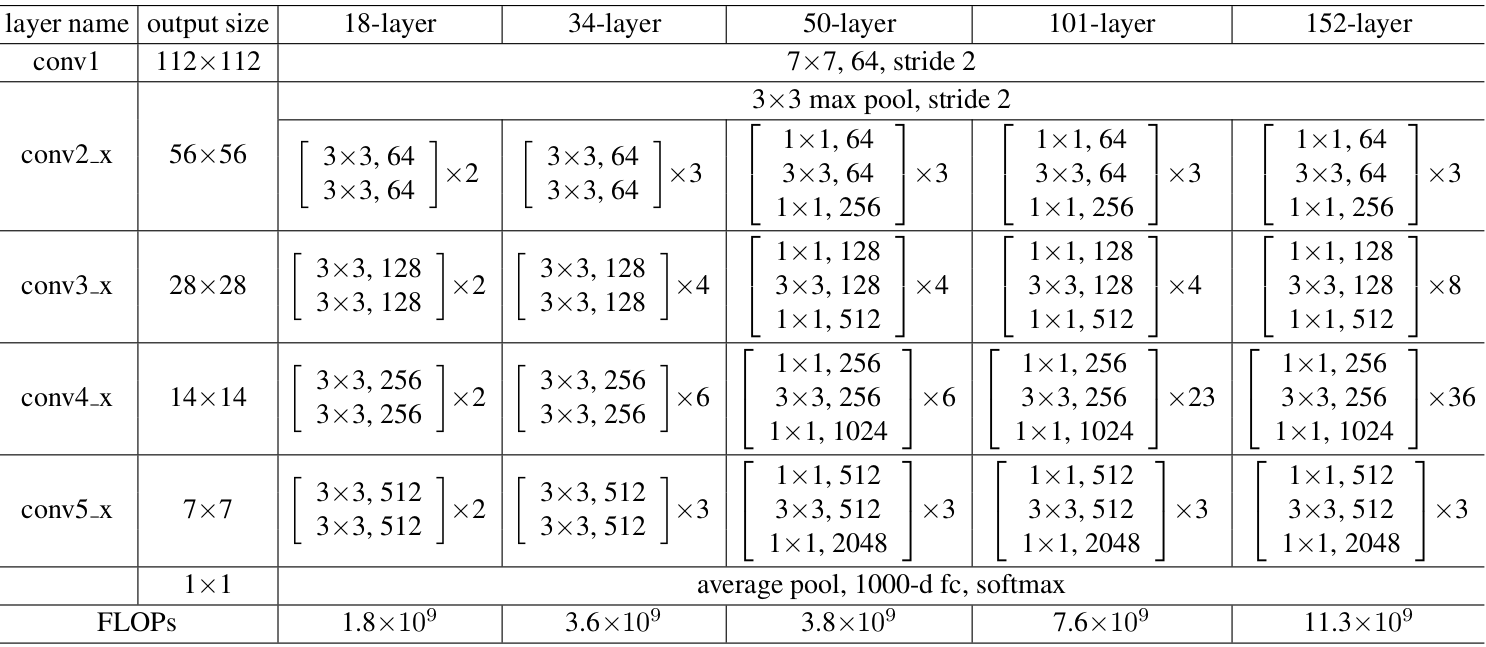
**ResNet50**

Mạng ResNet (R) là một mạng CNN được thiết kế để làm việc với hàng trăm lớp. Một vấn đề xảy ra khi xây dựng mạng CNN với nhiều lớp chập sẽ xảy ra hiện tượng *Vanishing Gradient* dẫn tới quá trình học tập không tốt.



*Hình 3: Kiến trúc Residual block*

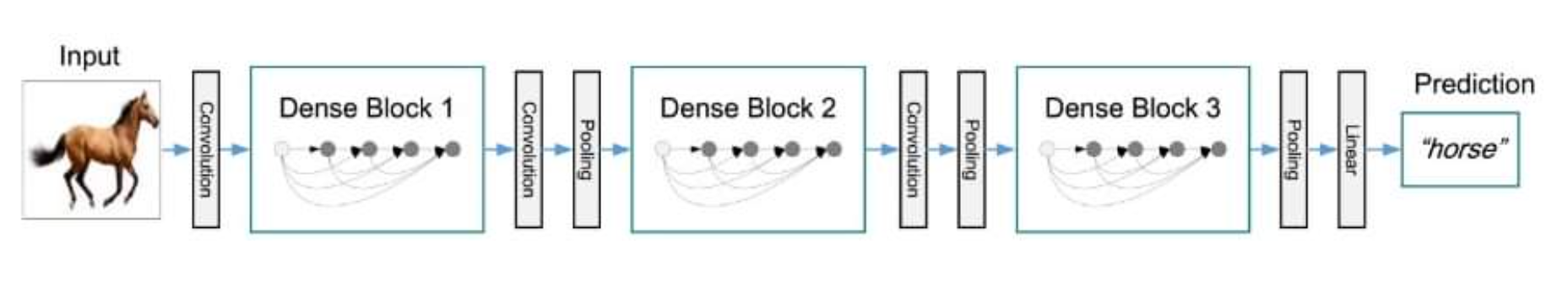
ResNet gần như tương tự với các mạng gồm có convolution, pooling, activation và fully-connected layer. Ảnh bên trên hiển thị khối dư được sử dụng trong mạng. Xuất hiện một mũi tên cong xuất phát từ đầu và kết thúc tại cuối khối dư. Hay nói cách khác là sẽ bổ sung Input X vào đầu ra của layer, hay chính là phép cộng mà ta thấy trong hình minh họa, việc này sẽ chống lại việc đạo hàm bằng 0, do vẫn còn cộng thêm X. Với H(x) là giá trị dự đoán, F(x) là giá trị thật (nhãn), chúng ta muốn H(x) bằng hoặc xấp xỉ F(x). [6]



*Hình 4: Các thông số chi tiết của mô hình Resnet*

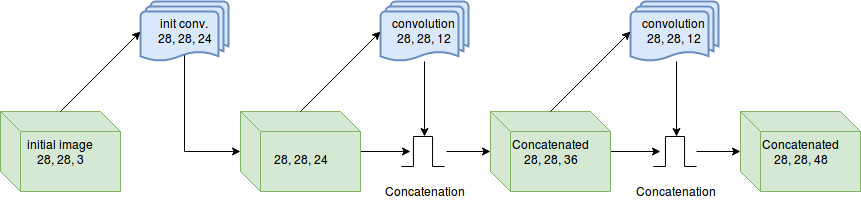
**DenseNet**

Densenet (Dense connected convolutional network) là một trong những network mới nhất cho visual object recognition. DenseNet là một trong những biến thể mở rộng của Resnet nên cũng gần giống Resnet nhưng có một vài điểm khác biệt. Densenet có cấu trúc gồm các dense block và các transition layers. Được stack dense block- transition layers-dense block- transition layers như hình vẽ. Với CNN truyền thống nếu chúng ta có L layer thì sẽ có L connection, còn trong densenet sẽ có L(L+1)/2 connection. [5]

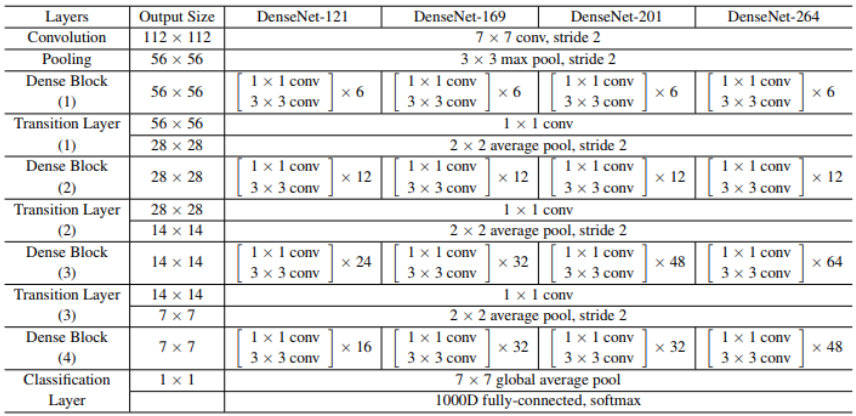


*Hình 5: Kiến trúc model DenseNet*

Hãy tưởng tượng ban đầu ta có 1 image size (28,28,3). Đầu tiên ta khởi tạo feature layer bằng Conv tạo ra 1 layer size (28,28,24). Sau mỗi layer tiếp theo (Trong dense block ) nó sẽ tạo thêm K= 12 feature giữa nguyên width và height. Khi đó output tiếp theo sẽ là (28,28,24 +12),(28,28,24 +12+12). Ở mỗi dense block sẽ có normalization, nonlinearity và dropout. Để giảm size và depth của feature thì transition layer được đặt giữa các dense block, nó gồm Conv kernel size =1, average pooling (2x2) với stride = 2 nó sẽ giảm output thành (14,14,48). [5]



*Hình 6: Khối các lớp tích chập với các kết quả được nối với nhau.*



*Hình 7: Các thông số chi tiết của mô hình DenseNet*

**Một số ưu điểm của DenseNet:**

* Densenet training tham số ít hơn 1 nửa so với Resnet nhưng có cùng độ chính xác trên tập dữ liệu Imagenet.



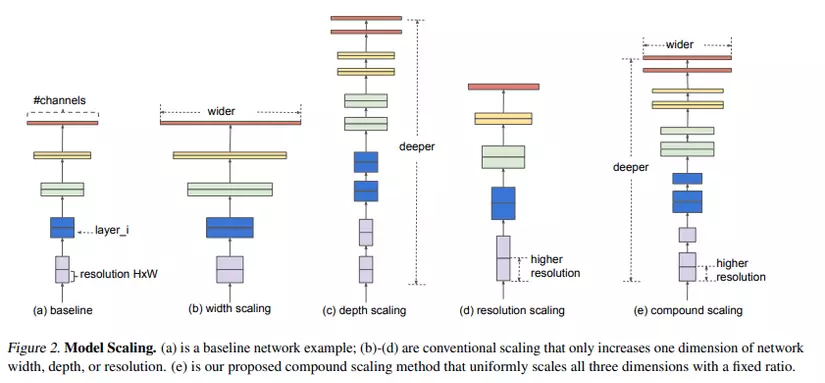
*Hình 8: Kết quả train so với các mô hình khác.*

* DenseNet chống overfitting rất hiệu quả.
* Sử dụng lại feature hiệu quả hơn.
* Giảm được vanishing gradient.

**EfficientNet**

Như ta đã biết, có ba kích thước tỷ lệ của CNN (Depth, Width, Resolution):

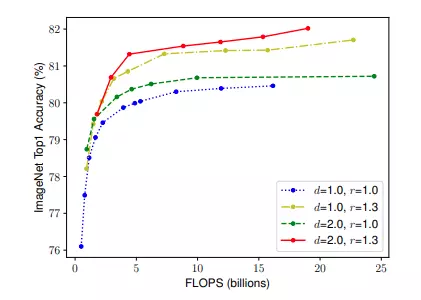
* Depth là độ sâu của mạng tương đương với số lớp trong đó.
* Width là độ rộng của mạng. Ví dụ: một thước đo chiều rộng là số kênh trong lớp Conv.
* Resolution là độ phân giải hình ảnh được chuyển đến CNN.



*Hình 9: Ví dụ cho việc thu phóng mô hình trên các kích thước khác nhau*

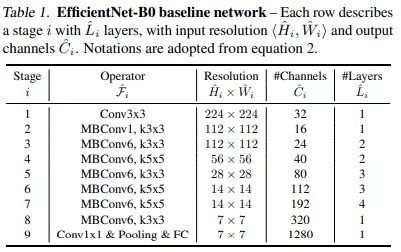
Theo nhóm tác giả quan sát các chiều của các mô hình thường không thu nhỏ/thu phóng đọc lập với nhau. Bởi vậy nên, theo trực giác của bản thân, nhóm tác giả cho rằng chúng ta cần phối hợp và cân bằng các kích thước tỷ lệ khác nhau hơn là chia tỷ lệ một chiều thông thường. [7]

Để xác thực trực giác của mình, nhóm tác giả so sánh việc thu phóng theo chiều rộng và theo các độ sâu cũng như độ phân giải mạng ở các mức độ khác nhau.



*Hình 10: So sánh việc thu phóng theo chiều rộng và theo các độ sâu, độ phân giải ở các mức độ.*

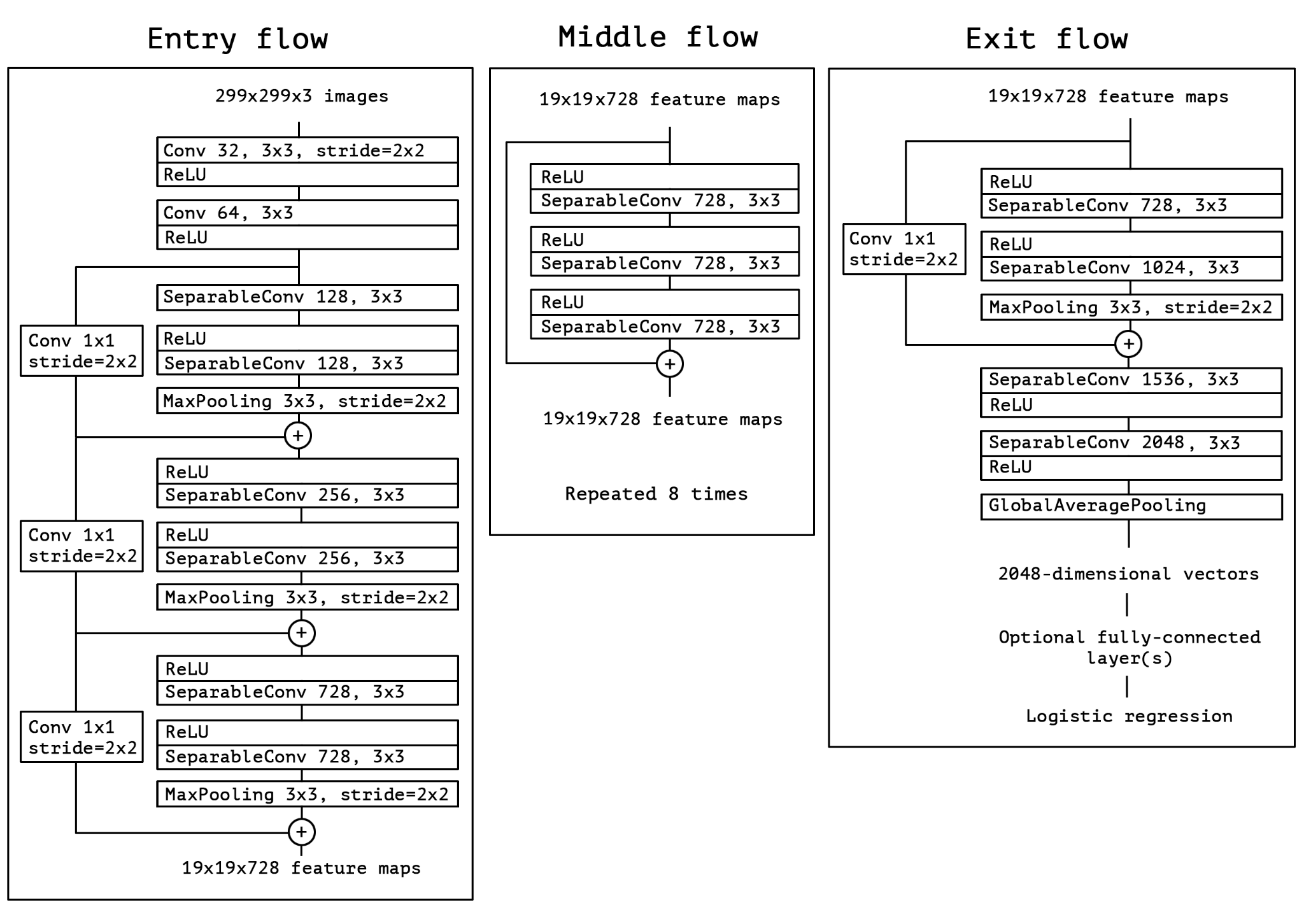
Từ kết quả được thể hiện trong hình trên, nhóm tác giả kết luận rằng để đạt được độ chính xác và hiệu quả tốt hơn, điều quan trọng là phải cân bằng tất cả các kích thước của chiều rộng, chiều sâu và độ phân giải mạng trong quá trình thu phóng quy mô ConvNet.



*Hình 11: Kiến trúc mạng Efficient Net*

**Xception Net**

Mạng Xception net là mạng dựa trên kiến trúc của Inception V3. Model đạt kết quả tốt trên tập dữ liệu Imagenet (với 350 triệu ảnh và 17 nghìn lớp). Được sáng tạo dựa trên mạng Inception và đã có nhiều cải tiến nên kết quả đạt được hiệu quả hơn mạng Inception trên tập Imagenet. Đạt vị trí thứ nhất trong cuộc thi ILSVRC 2015 trên cả 2 tập dữ liệu ImageNet ILSVRC và JFT. [4]



*Hình 12: Kiến trúc model Xception*

Phía trên là hình ảnh model của Xception net gồm có 3 khối, khối đầu tiên là entry flow với đầu vào là một tấm ảnh có kích thước 299x299x3 (ảnh màu) sau đó sẽ đi qua các 3 block như trong hình và output sẽ là một vector có kích thước 19x19x728, tiếp đến là middle flow tại đây đầu vào sẽ là vector 19x19x728 vừa thu được tại entry flow sau đó sẽ đi qua 3 block giống nhau và độ dài output vector đầu ra sẽ không đổi (19x19x728), cuối cùng sẽ đến exit flow tại đây đầu vào là output của khối middle flow sẽ đi qua thêm một block và đến cuối cùng là qua các lớp convolution, fully connected để output đầu ra của model là logistic regression. [4]

*Bảng 5: So sánh kết quả của 2 model Inception V3 và Xception trên tập Imagenet*

|  |  |
| --- | --- |
|  | **FastEval14k MAP@100** |
| **Inception V3 – no FC layers** | 6.36 |
| **Xception – no FC layers** | 6.70 |
| **Inception V3 with FC layers** | 6.50 |
| **Xception with FC layers** | **6.78** |

Từ kết quả trên ta thấy Xception cho kết quả tốt hơn Inception V3 ở cả 2 phương diện có FC layers hay không có FC layers.

*Bảng 6: So sánh tốc độ, tham số của 2 model Inception V3 và Xception*

|  |  |  |
| --- | --- | --- |
|  | **Parameter count** | **Steps/second** |
| **Inception V3** | 23,626,728 | 31 |
| **Xception** | 22,855,952 | 28 |

Từ kết quả trên cho ta thấy tuy tham số model Xception không chênh lệch không quá nhiều so với Inception V3 nhưng có quá trình train cho một step nhanh hơn đáng kể và kết quả thu được trên tập Imagenet cũng chính xác hơn.

* 1. **Text segmentation**

**Khái niệm:**

Word Segmentation là một phần khá cơ bản trong các kỹ thuật xử lý của NLP, nhiệm vụ chính là tách một đoạn text thành từ riêng lẻ. Đối với tiếng Anh thì việc này khá dễ dàng vì mỗi từ trong tiếng Anh đều có ý nghĩa và đa số các trường hợp thì chỉ cần sử dụng khoảng trắng là có thể tách từ được. Tuy nhiên, đối với tiếng Việt thì điều này không chính xác.

Những từ mà khi tách ra đứng một mình thì sẽ làm sai hoặc mất đi nghĩa của câu đó. (Từ phức). [2]

*Ví dụ: [Tôi, yêu, đất, nước] vs [Tôi, yêu, đất\_nước]*

*[Thuê, khách, sạn] vs [Thuê, khách\_sạn]*

Và tất nhiên nếu áp dụng trong Machine Translation, chúng ta sẽ thấy rằng tách từ đúng thì sẽ cho kết quả dịch chính xác hơn.

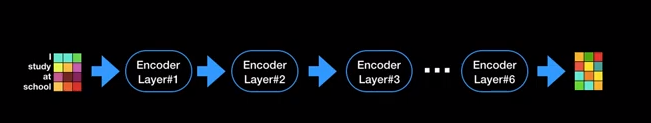
**Mục đích:**

* Sử dụng kĩ thuật text segmentation nhằm tách đoạn text thành từ riêng lẻ từ đó giữ được ý nghĩa chính xác cho câu.
* Khi áp dụng vào các việc xử lý ngôn ngữ tự nhiên thì sẽ cải thiện độ chính xác cho chương trình. Đặc biệt là đề tài nhóm em dùng sử dụng tiếng Việt, nên kĩ thuật tách từ rất cần thiết.
  1. **Text encoder**

**PhoBert:**

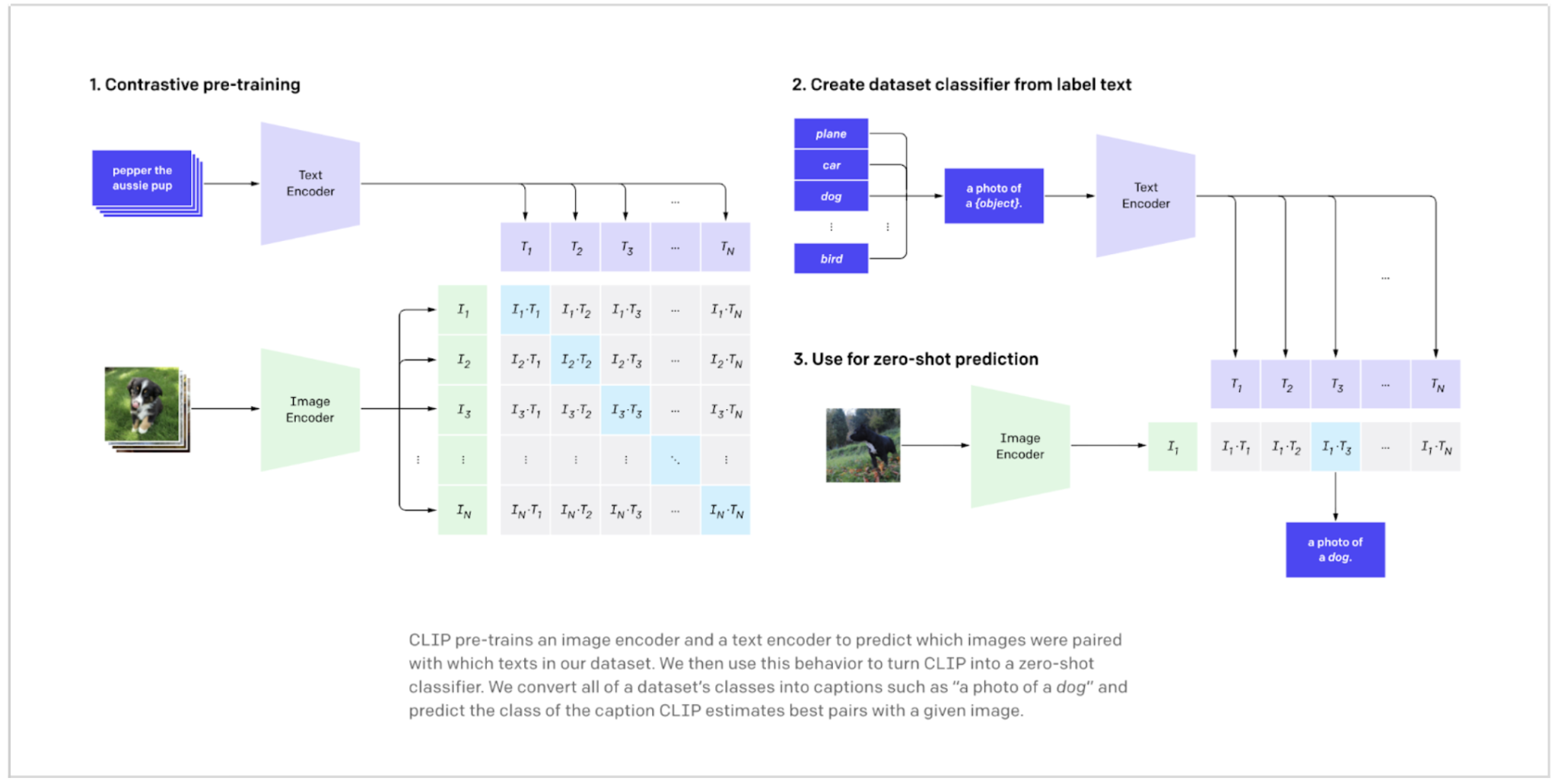
Đây là một pre-trained được huấn luyện monolingual language, tức là chỉ huấn luyện dành riêng cho tiếng Việt. BERT là viết tắt của cụm từ Bidirectional Encoder Representation from Transformer có nghĩa là mô hình biểu diễn từ theo 2 chiều ứng dụng kỹ thuật Transformer. BERT được thiết kế để huấn luyện trước các biểu diễn từ (pre-train word embedding). Điểm đặc biệt ở BERT đó là nó có thể điều hòa cân bằng bối cảnh theo cả 2 chiều trái và phải.

BERT mở rộng khả năng của các phương pháp trước đây bằng cách tạo các biểu diễn theo ngữ cảnh dựa trên các từ trước và sau đó để dẫn đến một mô hình ngôn ngữ với ngữ nghĩa phong phú hơn. PhoBERT được train trên khoảng 20GB dữ liệu bao gồm khoảng 1GB Vietnamese Wikipedia corpus và 19GB còn lại lấy từ Vietnamese news corpus. Đây là một lượng dữ liệu khả ổn để train một mô hình như BERT. [2]



*Hình 13: Kiến trúc gồm nhiều layers tại encoder của model BERT. Mô hình huấn luyện từ RoBERTa cho phép ta trích xuất các đặc trưng từ những layers của encoder.*

* 1. **CLIP model**



*Hình 14: Kiến trúc của CLIP model*

Đây là model được sử dụng cuối cùng và có kiến trúc gần giống trong ảnh với text encoder là PhoBert, image encoder là ResNet. Chỉ khác phần tiền xử lý trước khi đưa vào PhoBert là có thêm VNCoreNLP chuyên dùng cho bài toán với input là tiếng Việt với chức năng là word segmentation.

## Điều chỉnh tham số

Các tham số có thể được điều chỉnh ban đầu bao gồm:

*Bảng 7: Bảng tham số có thể điều chỉnh*

|  |  |  |
| --- | --- | --- |
| **Key** | **Value** | **Ý nghĩa** |
| batch\_size | 16 | Kích thước batch |
| image\_encoder\_lr | 1e-5 | Learning rate của image encoder |
| text\_encoder\_lr | 1e-5 | Learning rate của text encoder |
| size | 224 | Kích thước ảnh input của image encoder |
| projection\_dim | 256 | Cần đưa số chiều của text embedding và image embedding về cùng một số chiều chung. |
| dropout | 0.1 | Tỉ lệ drop out. |

Với bộ tham số này, model đạt được accuracy khoảng 60%.

Ban đầu team có thử nghiệm với các model image encoder khác nhau thì ta có một nhận xét là model chính (CLIP model) hội tụ rất nhanh, khoảng 3 epochs là sẽ bắt đầu overfit. Từ đó team bắt đầu nghi ngờ lý do không nằm ở model mà nằm ở các tham số đầu vào.

Sau các thử nghiệm với các bộ tham số khác nhau thì thấy 2 tham số ảnh hưởng đến độ chính xác nhiều nhất là size và projection\_dim. Các tham số này ảnh hưởng trực tiếp tới các đặc trưng của image embedding cho nên có thể kết luận là các đặc trưng của image embedding chưa tương xứng với text embedding, cụ thể là quá ít. Thế nên model có quá ít đặc trưng để học.

Sau nhận xét trên team đã tiếp tục thử với các bộ tham số khác và tập trung chủ yếu vào 2 tham số là size và projection\_dim. Từ đó ta có kết quả tốt nhất với bộ tham số như sau:

*Bảng 8: Bảng tham số sau khi điều chỉnh*

|  |  |  |
| --- | --- | --- |
| Key | Value | Ý nghĩa |
| batch\_size | 16 | Kích thước batch |
| image\_encoder\_lr | 3e-4 | Learning rate của image encoder |
| text\_encoder\_lr | 3e-5 | Learning rate của text encoder |
| size | 336 | Kích thước ảnh input của image encoder |
| projection\_dim | 512 | Cần đưa số chiều của text embedding và image embedding về cùng một số chiều chung. |
| dropout | 0.1 | Tỉ lệ drop out. |

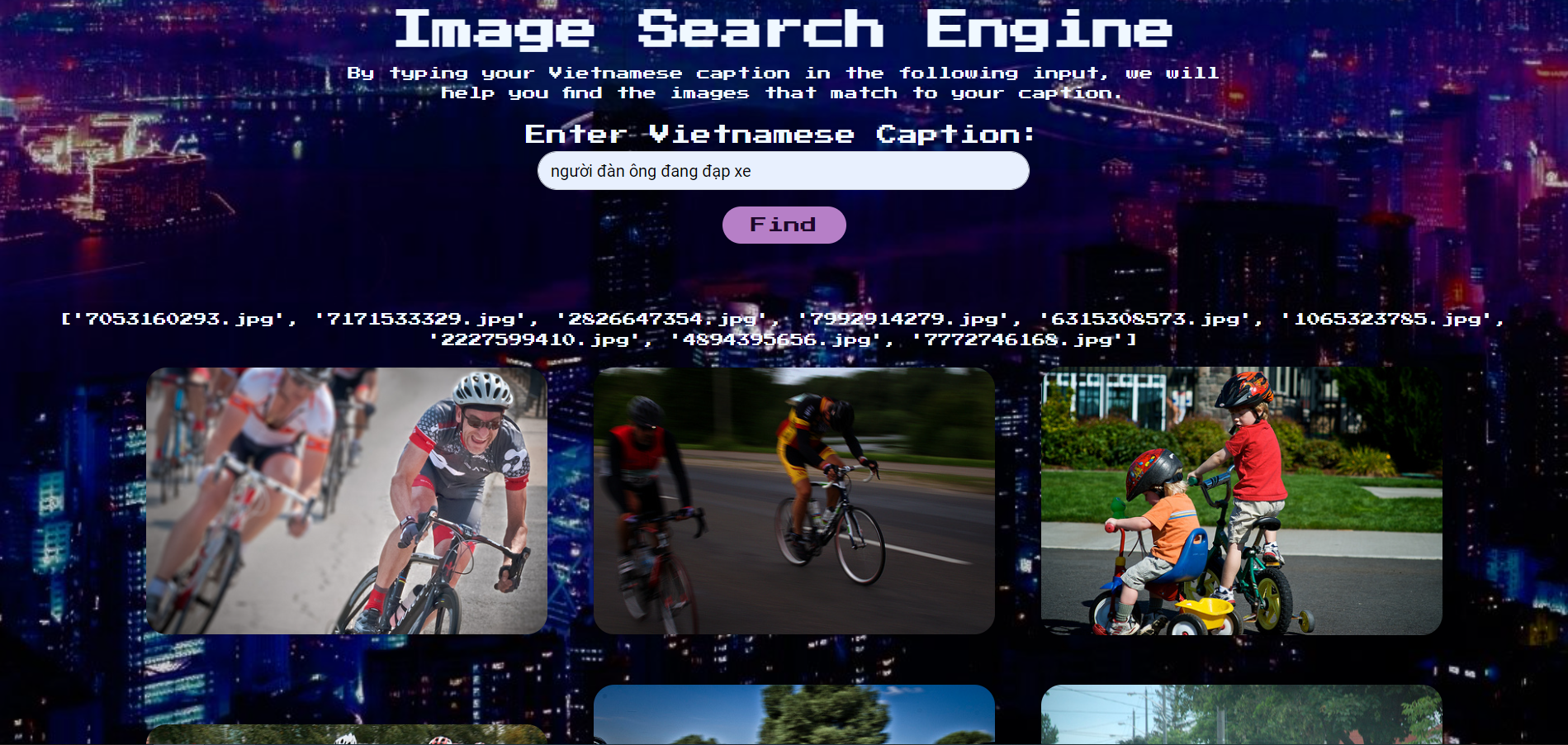
Với bộ tham số trên, team đã cải thiện độ chính xác lên 75.582%.

# KẾT QUẢ VÀ HƯỚNG PHÁT TRIỂN

## Kết quả chương trình

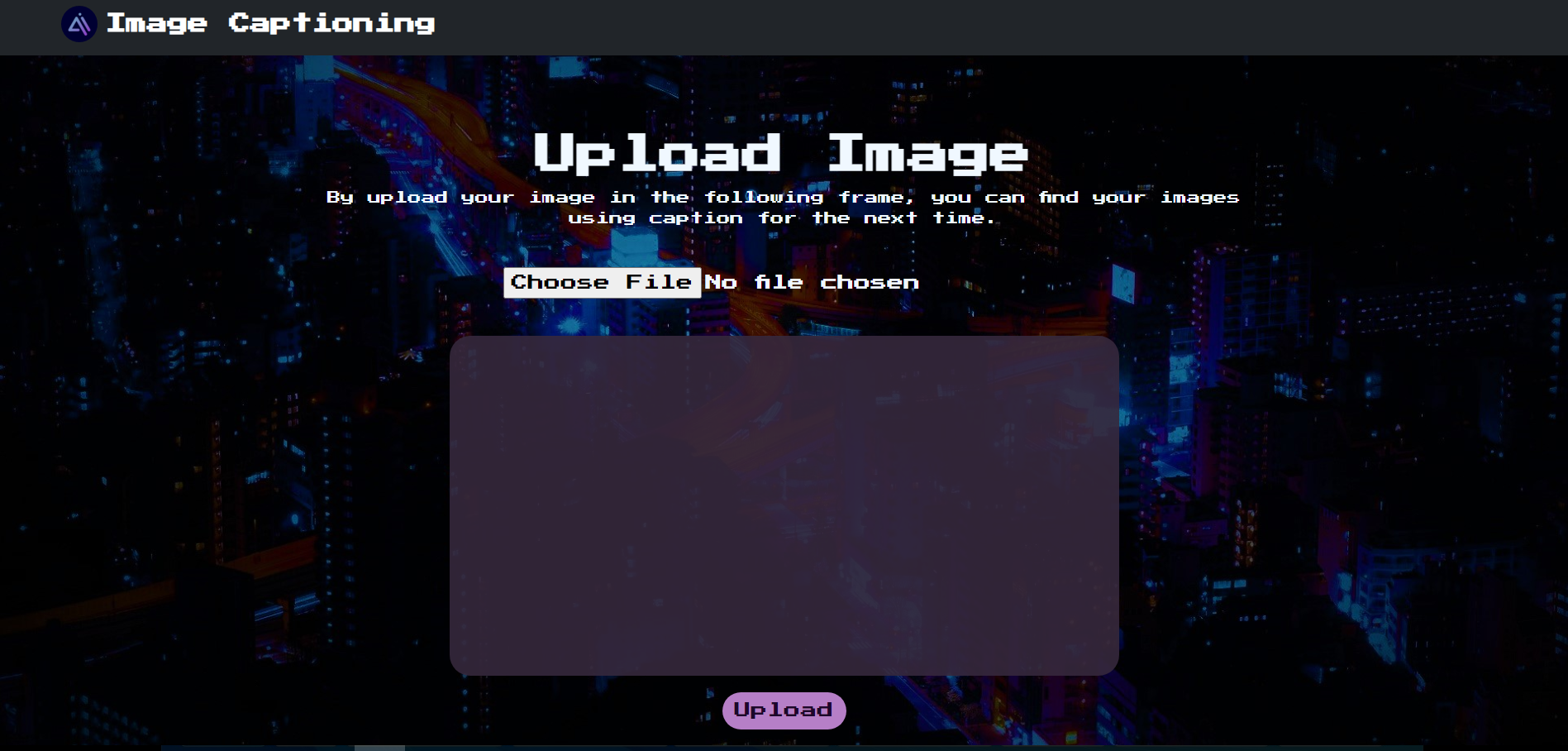
Team đã sử dụng deploy model và xây dựng giao diện để dễ dàng cho việc sử dụng. Gồm 2 chức năng:

* Tìm kiếm hình ảnh bằng caption.



*Hình 15: Giao diện trang web tìm kiếm của nhóm*

* Upload hình ảnh mà không cần caption nào.



*Hình 16: Giao diện trang web chức năng upload*

## Hướng phát triển

Việc cải thiện accuracy của model vẫn còn nhiều hướng. Đơn giản nhất là việc thay đổi bộ tham số hoặc tìm kiếm thêm dữ liệu. Khó hơn có thể nghĩ cách để thử nghiệm các kiến trúc mới hoặc thay đổi các kiến trúc có sẵn.

Cải thiện và tích hợp hệ thống vào trang web cụ thể mang quy mô lớn hơn như đồ án tốt nghiệp.

# TÀI LIỆU THAM KHẢO

**[1] Connecting Text and Image:** [*https://openai.com/blog/clip/*](https://openai.com/blog/clip/)[Truy cập ngày: 31/08/2022]

**[2] Word Segmentation:** [*https://chienuit.wordpress.com/2016/04/21/nlp-word-segmentation/*](https://chienuit.wordpress.com/2016/04/21/nlp-word-segmentation/)[Truy cập ngày: 10/09/2022]

**[3] PhoBERT:** <https://github.com/VinAIResearch/PhoBERT> [Truy cập ngày: 10/09/2022]

**[4] XceptionNet:** <https://openaccess.thecvf.com/content_cvpr_2017/html/Chollet_Xception_Deep_Learning_CVPR_2017_paper.html> [Truy cập ngày: 20/09/2022]

**[5] DenseNet:** <https://paperswithcode.com/lib/timm/densenet> [Truy cập ngày: 21/09/2022]

**[6] ResNet:** <https://rohanvarma.me/resnet/> [Truy cập ngày: 05/10/2022]

**[7] EfficientNet:** <https://arxiv.org/pdf/1905.11946.pdf>. [Truy cập ngày: 07/10/2022]

**[8] Data Augmentation:** <https://paperswithcode.com/task/data-augmentation> [Truy cập ngày: 21/10/2022]