

**ĐẠI HỌC QUỐC GIA TP.HỒ CHÍ MINH**  
**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**



**TÓM TẮT BÁO CÁO ĐỒ ÁN CUỐI KỲ**  
**MÁY HỌC - CS114.L21**

**Đề tài:** *Nhận dạng món ăn đường phố Việt Nam*  
*(30 món)*

**Giảng viên:** *Phạm Nguyễn Trường An*  
*Lê Đình Duy*

Sinh viên thực hiện:		
STT	Họ tên	MSSV
1	Phạm Quang Vinh (leader)	19522526
2	Nguyễn Minh Trí	19522389
3	Trương Xuân Linh	19521759

**TP. HỒ CHÍ MINH – 07/2021**

## MỤC LỤC

<b>TỔNG QUAN.</b>	<b>1</b>
<b>XÂY DỰNG BỘ DỮ LIỆU.</b>	<b>2</b>
<b>I. Số lượng, độ đa dạng:</b>	<b>2</b>
<b>II. Phân chia Train/Test/Validation:</b>	<b>2</b>
<b>TRAINING VÀ ĐÁNH GIÁ.</b>	<b>3</b>
<b>I. Tóm tắt quá trình Training:</b>	<b>3</b>
<b>II. Đánh giá chung:</b>	<b>3</b>
<b>ỨNG DỤNG VÀ HƯỚNG PHÁT TRIỂN.</b>	<b>4</b>
<b>I. Xây dựng ứng dụng web:</b>	<b>4</b>
<b>II. Hướng phát triển:</b>	<b>4</b>
<b>CẬP NHẬT SO VỚI BÁO CÁO VẤN ĐÁP</b>	<b>4</b>

## TỔNG QUAN.

### ❖ Bài toán:

Với đề tài này chúng em mong muốn xây dựng một model máy học có thể thực hiện việc **nhận dạng được món ăn (food recognition)** hay giải quyết bài toán **phân lớp (classification)** cho 30 món ăn qua đầu vào hình ảnh với:

#### a. Input:

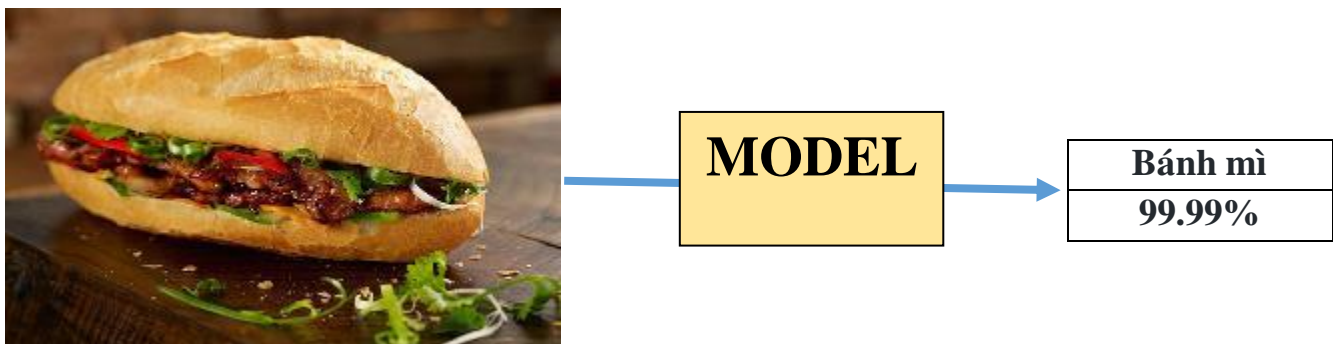
- Một bức ảnh chụp món ăn:
  - Được chụp bằng smartphone hoặc máy ảnh.
  - Góc chụp tùy ý đủ thấy toàn bộ món ăn.
  - Chỉ có một món ăn trong ảnh.

#### b. Output:

- Tên món ăn trong ảnh
- Xác suất dự đoán (probability) món đó so với các món trong danh sách.

Nếu món ăn có xác suất dự đoán cao nhất bé hơn 70% thì xuất “Can’t identify this food”

### Ví dụ:



⇒ Từ output trên, khi chúng ta nhận dạng được tên món ăn thì dễ dàng xuất ra thông tin về món ăn đó.

## XÂY DỰNG BỘ DỮ LIỆU.

### I. Số lượng, độ đa dạng:

Sau khi mất 3 tuần để thu thập từ Google Image, Bing, Instagram, Foody.com và label cho tất cả dữ liệu, kết quả thu được tổng cộng **23178** ảnh cho **30** classes món ăn, cụ thể như sau:

Class	SL	Class	SL	Class	SL
banh_beo	591	banh_xeo	1173	com_chay	815
banh_bot_loc	720	bap_xao	607	com_tam	942
banh_can	744	bot_chien	675	goi_cuon	856
banh_cuon	1140	bun_bo_hue	1530	mi_quang	884
banh_gio	641	bun_cha	510	nem_chua	542
banh_khot	835	bun_dau_mam_tom	927	pha_lau	686
banh_mi	1336	bun_thit_nuong	747	pho	807
banh_mi_nuong	423	cao_lau	618	sup_cua	687
banh_trang_nuong	795	chao_long	1073	xoai_lac	541
banh_trang_tron	494	chuoi_chien	619	xoi_gac	220

*Số lượng ảnh của từng Class*

### II. Phân chia Train/Test/Validation:

Vì chúng em sử dụng nhiều mô hình huấn luyện khác nhau nên bộ dữ liệu sẽ được chia cố định thành hai phần vào 2 folder **Train** và **Test**, với tỉ lệ:

**80% Train và 20% Test.**

Đối với các kiến trúc deep learning trước khi train chúng em sẽ chia thêm tập **Validation** từ tập Train với tỉ lệ bằng 15% tập Train, khi đó ta được:

**68% Train : 12% Validation : 20% Test**

Chia như thế để tận dụng hết dữ liệu và đủ dữ liệu cho quá trình Train

	SVM - HOG	Deep learning
<b>Train</b>	18552	15785
<b>Validation</b>		2767
<b>Test</b>	4626	4626

*Tỉ lệ Train/ Validation/ Test*

## TRAINING VÀ ĐÁNH GIÁ.

### I. Tóm tắt quá trình Training:

Ban đầu, sau khi resize các ảnh về (224,224,3), tất cả ảnh được đưa qua mô hình **HOG** để rút trích đặc trưng và train bằng **SVM**. Tuy nhiên, kết quả thu được lại rất tệ, train\_acc: 0.76, test\_acc: 0.21

Nên sau đó, chúng em chuyển sang train mô hình **VGG16** với tất cả các trọng số được train lại từ đầu và thêm một vài layer để giảm overfitting, tuy nhiên mô hình vẫn gặp tình trạng overfitting với train\_acc: 0.87, val\_acc: 0.62, test\_acc: 0.62

Tiếp theo, chúng em quyết định sử dụng **transfer learning** và **fine tuning** với mô hình **VGG16** có các trọng số được train sẵn với tập ImageNet, đóng băng một vài lớp và train lại một vài lớp, kết quả thu được tốt hơn với train\_acc: 0.99, val\_acc: 0.79, acc\_test: 0.8

Cuối cùng, chúng em sử dụng **transfer learning** với mô hình **EfficientNet** với hai chiến lược **fine tuning** khác nhau trên pretrained model có trọng số được train trên tập ImageNet. Sơ lược về các chiến lược fine tuning là thiết kế lại phần top layer và train phần top layer khi đóng băng pretrained model, rồi mở một vài layer để train đến khi có kết quả tốt nhất. Kết quả tốt nhất thu được là train\_acc: 0.96, val\_acc: 0.86, Test\_acc: 0.87

### II. Đánh giá chung:

Accuracy Model	Train	Validation	Test
<b>SVM với HOG</b>	0.759	-	0.215
<b>VGG16</b>	0.873	0.621	0.623
<b>VGG16 Transfer</b>	0.998	0.786	0.797
<b>EfficientNet B0 v1</b>	0.913	0.837	0.848
<b>EfficientNet B4 v1</b>	0.899	0.812	0.836
<b>EfficientNet B0 v2</b>	<b>0.956</b>	<b>0.862</b>	<b>0.871</b>

*Accuracy của các thuật toán*

- Qua quá trình xây dựng các model, model có kết quả tốt nhất là EfficientNet B0 với cách fine tuning thứ 2, model có kết quả kém nhất là SVM với rút trích đặc trưng HOG.

## ỨNG DỤNG VÀ HƯỚNG PHÁT TRIỂN.

### I. Xây dựng ứng dụng web:

Ứng dụng được xây dựng bằng python cho phép người dùng upload ảnh hoặc url của ảnh món ăn và ứng dụng sẽ cho biết thông tin món ăn gồm cách đọc, tên món, mô tả sơ lược, thành phần, địa điểm có thể mua và giá tiền. Ngôn ngữ hoàn toàn là tiếng Anh, deploy trên môi trường sever đám mây của Heroku.

**Link web app demo:** [Streamlit \(vietnamesefood-demo.herokuapp.com\)](https://vietnamesefood-demo.herokuapp.com)

#### Nhận xét:

- Do chạy trên nền tảng web dạng đọc với server đám mây nên ứng dụng có thể chạy được trên đa dạng các thiết bị trình duyệt cũng như hệ điều hành khác nhau.
- Thời gian truy cập còn ảnh hưởng bởi tốc độ Internet và loại trình duyệt.
- Một số ảnh ngoài danh sách bị dự đoán sai. Bị lỗi ở một vài url cá biệt.
- Còn lại nhìn chung ứng dụng hoạt động ổn định, đáp ứng đủ tính năng đặc được đặc ra, tận dụng tốt model được huấn luyện.

### II. Hướng phát triển:

Qua quá trình thực hiện đồ án với những kết quả đạt được cùng nhu cầu của thị trường em nhận thấy đề tài *Nhận dạng món ăn Việt* có khả năng phát triển cao.

Với đồ án trên chúng ta có thể tối ưu và mở rộng hơn bộ dữ liệu để có thể nhận dạng được nhiều món ăn hơn. Từ việc nhận dạng một ăn chúng ta có thể phát triển đề nhận dạng nhiều món hơn trong một bức ảnh.

Về phần ứng dụng tương đối ổn định nhưng cũng có thể phát triển thêm để ứng dụng ổn định hơn và thời gian tương tác với ứng dụng nhanh hơn. Hoặc có thể phát triển thành ứng dụng di động.

## CẬP NHẬT SO VỚI BÁO CÁO VẤN ĐÁP

- Chỉnh lại chính tả của báo cáo chi tiết.
- Thêm phần tài liệu tham khảo của phần tiền xử lý của báo cáo chi tiết.
- Tạo colab notebook để clone github repo cũng như tải dataset trên Drive của nhóm