

TRƯỜNG ĐẠI HỌC SÀI GÒN
KHOA CÔNG NGHỆ THÔNG TIN



PHÁT TRIỂN PHẦN MỀM MÃ NGUỒN MỞ

Phát triển ứng dụng nhận diện hình ảnh con vật trên môi trường
Linux

GVHD: TỪ LÃNG PHIÊU

SV: NGUYỄN TUẤN KIỆT - 3120410272

SV : NGÔ VĂN TÍN - 3120410534

SV : TRẦN CHÁNH QUỐC ĐỊNH - 3120411050

TP. HỒ CHÍ MINH, THÁNG 5/2024

Mục lục

1	MÔ TẢ BÀI TOÁN	2
2	CÁC CÔNG NGHỆ SỬ DỤNG	3
2.1	Nền tảng ứng dụng	3
2.2	Ngôn ngữ lập trình Python	3
2.3	CNN Network	3
2.4	Tkinter và CustomTkinter	4
3	Giao diện phần mềm	5
4	DEMO CHƯƠNG TRÌNH	6
4.1	Chức năng chèn hình ảnh	6
4.1.1	Giao diện	6
4.1.2	Xử lý	6
4.2	Chức năng Generate(Đưa ra kết quả)	7
4.2.1	Giao diện trước generate	7
4.2.2	Giao diện sau generate	8
4.2.3	Xử lý	8
4.3	Chức năng xóa hình ảnh	9
4.3.1	Giao diện	9
4.3.2	Xử lý	9
5	KẾT LUẬN	11
5.1	Tổng kết công việc đã thực hiện	11
5.2	Các mặt hạn chế	11
5.3	Hướng phát triển tiếp theo trong tương lai	11



1 MÔ TẢ BÀI TOÁN

Bài toán nhận diện hình ảnh chó và mèo bằng CNN trong Python là một nhiệm vụ phổ biến trong lĩnh vực thị giác máy tính và trí tuệ nhân tạo. Nó có ứng dụng trong nhiều lĩnh vực khác nhau như nghiên cứu khoa học, y học thú y, hoặc trong ứng dụng thực tế như hệ thống giám sát, hệ thống phân loại trong nông nghiệp hoặc công nghệ theo dõi và giám sát thú cưng.

Mục tiêu chính của bài toán là phát triển một mô hình trí tuệ nhân tạo dựa trên kiến trúc mạng CNN để phân loại hình ảnh chó và mèo. Mô hình sẽ nhận vào một bức ảnh và trả về một nhãn (label) tương ứng với loại động vật trong ảnh là chó hay mèo :

- Hiệu quả trong xử lý hình ảnh: CNN là một loại mạng nơ-ron được thiết kế đặc biệt để xử lý dữ liệu hình ảnh. Nó có khả năng tự động trích xuất các đặc trưng từ hình ảnh, giúp phân biệt giữa các loài động vật dựa trên hình dạng, màu sắc, kết cấu và các đặc điểm khác.
- Độ chính xác cao: CNN đã chứng minh được độ chính xác cao trong nhiều bài toán liên quan đến nhận diện hình ảnh.
- CNN có khả năng xử lý nhiều loại hình ảnh với các điều kiện ánh sáng, góc nhìn và bối cảnh khác nhau. Điều này rất quan trọng trong việc nhận diện chó và mèo, vì chúng có thể xuất hiện trong các điều kiện khác nhau.
- Python có nhiều thư viện hỗ trợ cho việc phát triển mạng CNN, chẳng hạn như TensorFlow, Keras, và PyTorch, giúp quá trình phát triển và triển khai mô hình trở nên dễ dàng hơn.

Mục tiêu của đồ án nhận diện hình ảnh chó và mèo bằng CNN trong Python là phát triển một mô hình trí tuệ nhân tạo có khả năng phân loại tự động hình ảnh chó và mèo từ các bức ảnh đầu vào. Mục tiêu này nhằm đáp ứng nhu cầu của nhiều lĩnh vực như nghiên cứu khoa học, y học thú y, ứng dụng thực tế trong hệ thống giám sát, phân loại trong nông nghiệp, hoặc công nghệ theo dõi và giám sát thú cưng.

2 CÁC CÔNG NGHỆ SỬ DỤNG

2.1 Nền tảng ứng dụng

Dự kiến phát triển ứng dụng trên nền tảng Linux OS vì đây là hệ điều hành mạnh mẽ và ổn định, đặc biệt thích hợp cho việc triển khai các ứng dụng trí tuệ nhân tạo và xử lý dữ liệu phức tạp. Ứng dụng sẽ là một phần mềm độc lập chạy trên nền tảng Linux, cho phép người dùng cài đặt trực tiếp trên máy tính

2.2 Ngôn ngữ lập trình Python

Python là một lựa chọn phù hợp cho việc phát triển ứng dụng này vì có nhiều lợi ích như:

- Hỗ trợ các thư viện mạnh mẽ: Python có nhiều thư viện và framework nổi tiếng hỗ trợ việc phát triển và huấn luyện các mô hình CNN. Các thư viện như TensorFlow, Keras, PyTorch, và scikit-learn cung cấp các công cụ và tiện ích mạnh mẽ giúp phát triển và triển khai các mô hình trí tuệ nhân tạo dễ dàng hơn.
- Python là một ngôn ngữ đa năng và mạnh mẽ, có thể được sử dụng cho nhiều mục đích khác nhau từ phát triển web, xử lý dữ liệu, đến trí tuệ nhân tạo và học máy.
- Có một cộng đồng lớn và tích cực, cung cấp nhiều thư viện và framework hữu ích giúp việc phát triển ứng dụng trở nên nhanh chóng và dễ dàng hơn.
- Miễn phí và mã nguồn mở, giúp tiết kiệm chi phí phát triển và tạo điều kiện thuận lợi cho sự mở rộng và phát triển của dự án trong tương lai.

2.3 CNN Network

Sử dụng CNN Network để phát triển ứng dụng mang lại nhiều ưu điểm quan trọng

- Sự hỗ trợ của các thư viện và công cụ mạnh mẽ: Các thư viện như TensorFlow, Keras, và PyTorch cung cấp các công cụ và tiện ích mạnh mẽ cho việc phát triển và triển khai mô hình CNN. Sự phổ biến và hỗ trợ từ cộng đồng cho các thư viện này giúp người dùng dễ dàng truy cập tài liệu và hướng dẫn chi tiết, hỗ trợ quá trình nghiên cứu và phát triển dự án.
- Tiết kiệm thời gian huấn luyện và dự đoán: Mạng CNN được tối ưu hóa để xử lý hình ảnh một cách hiệu quả, cho phép huấn luyện và dự đoán nhanh chóng. Điều này giúp giảm thời gian phát triển và triển khai ứng dụng.
- Tính ổn định và hiệu quả: CNN đã được chứng minh là một giải pháp ổn định và hiệu quả trong nhiều bài toán liên quan đến thị giác máy tính, bao gồm nhận diện chó và mèo. Điều này giúp ứng dụng đạt được hiệu suất cao và đáng tin cậy trong việc phân loại hình ảnh.
- Khả năng mở rộng: Mạng CNN có thể dễ dàng mở rộng để xử lý các loài động vật khác nhau hoặc các bài toán phân loại hình ảnh khác. Điều này cung cấp tiềm năng lớn cho việc phát triển và mở rộng ứng dụng trong tương lai.
- Khả năng tích hợp: CNN có thể được tích hợp một cách dễ dàng với các hệ thống và ứng dụng khác, giúp phát triển các giải pháp hoàn chỉnh và liên kết với các phần khác của ứng dụng một cách hiệu quả.



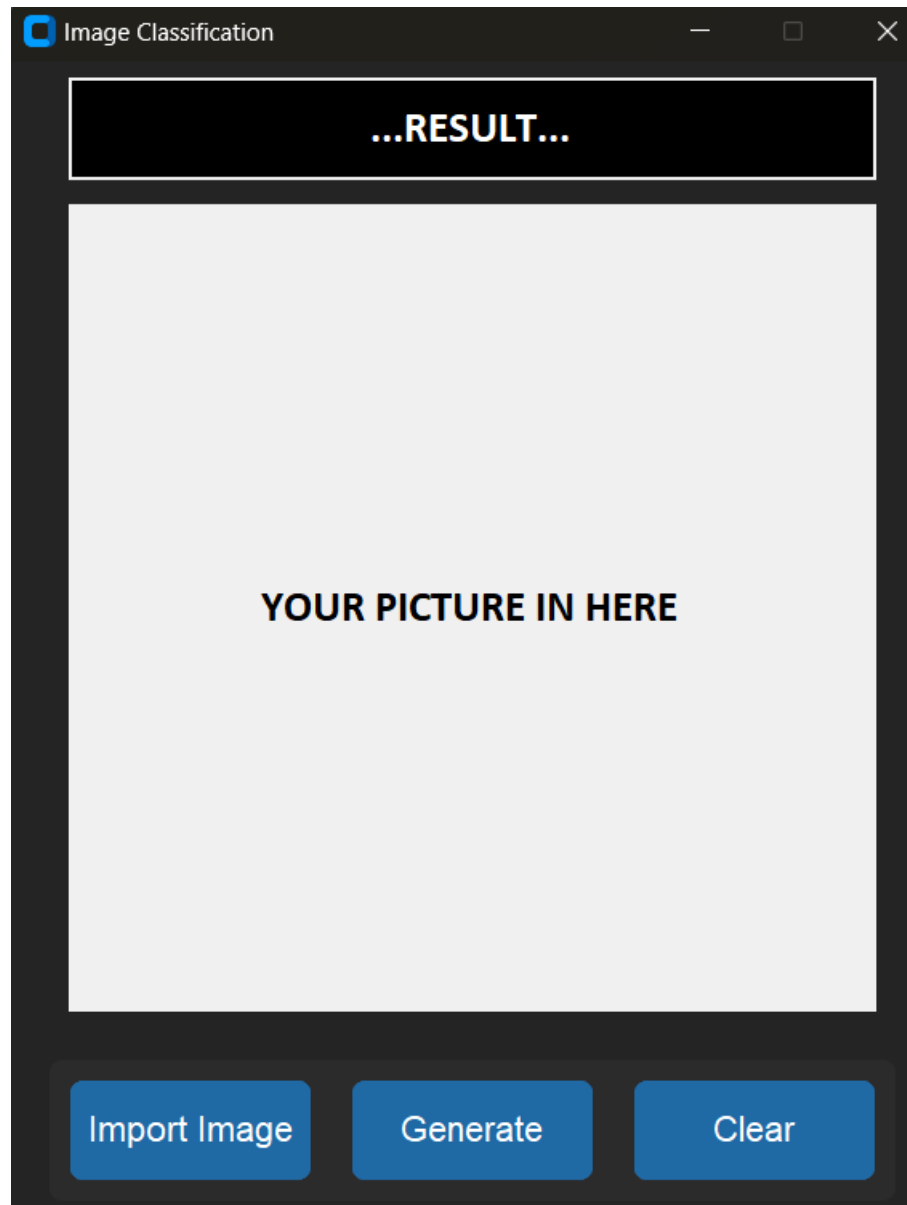
2.4 Tkinter và CustomTkinter

Tkinter là một thư viện giao diện người dùng đồ họa (GUI) chuẩn của Python, cho phép bạn phát triển các ứng dụng GUI dễ dàng và nhanh chóng. CustomTkinter là một framework mở rộng của Tkinter, cung cấp các widget và công cụ bổ sung để cải thiện giao diện và trải nghiệm người dùng. Sử dụng Tkinter và CustomTkinter cho ứng dụng GUI mang lại nhiều lợi ích:

- Dễ sử dụng và tùy biến: Tkinter có cú pháp đơn giản và dễ hiểu, giúp phát triển giao diện người dùng một cách nhanh chóng. CustomTkinter cung cấp thêm các widget nâng cao, giúp bạn tùy biến giao diện theo nhu cầu của ứng dụng.
- Tích hợp sẵn với Python: Tkinter là thư viện chuẩn của Python, do đó bạn có thể dễ dàng tích hợp GUI vào ứng dụng hiện có mà không cần cài đặt thư viện bên ngoài. CustomTkinter có thể được cài đặt nhanh chóng để mở rộng khả năng của Tkinter.
- Khả năng tương thích đa nền tảng: Ứng dụng GUI được phát triển bằng Tkinter và CustomTkinter có thể chạy trên nhiều nền tảng khác nhau như Windows, macOS và Linux mà không cần sửa đổi mã nguồn.
- Tính mở rộng và hỗ trợ cộng đồng: Cả Tkinter và CustomTkinter đều có cộng đồng hỗ trợ tích cực, cung cấp tài liệu, hướng dẫn và các giải pháp cho các vấn đề phát sinh trong quá trình phát triển.
- Thân thiện với người dùng: Các widget và công cụ của Tkinter và CustomTkinter giúp tạo giao diện người dùng trực quan, thân thiện và dễ sử dụng, cải thiện trải nghiệm của người dùng.
- Tính linh hoạt: CustomTkinter cho phép bạn dễ dàng tùy chỉnh giao diện và chức năng của ứng dụng để phù hợp với yêu cầu của dự án.



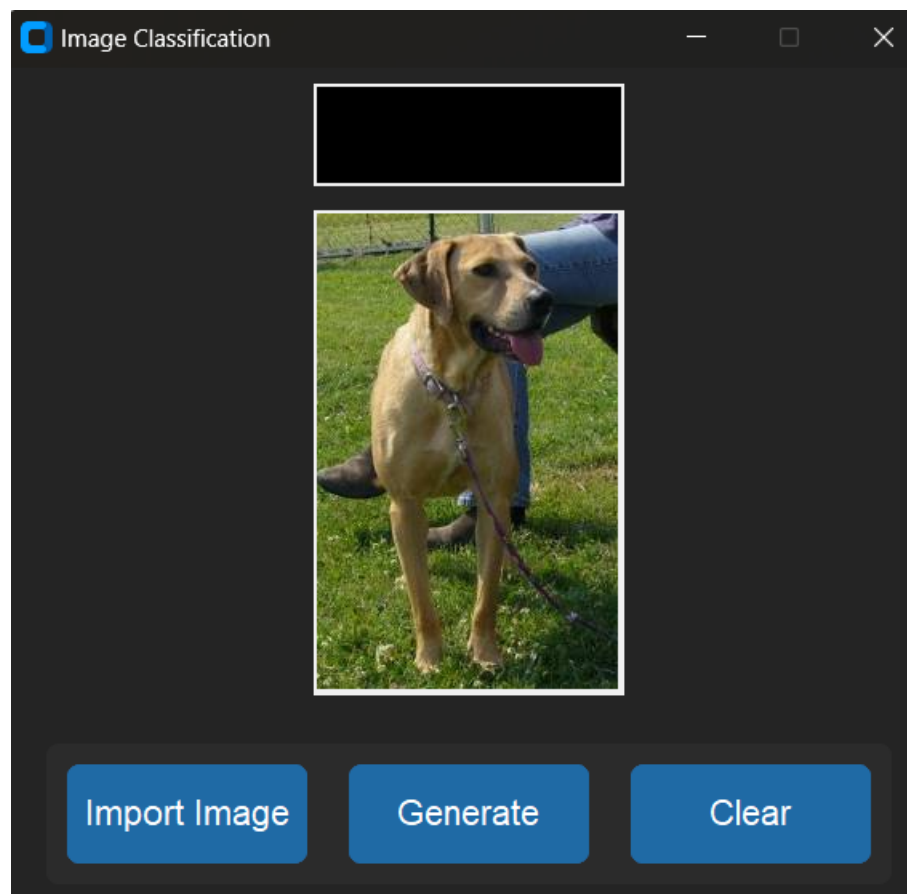
3 Giao diện phần mềm



4 DEMO CHƯƠNG TRÌNH

4.1 Chức năng chèn hình ảnh

4.1.1 Giao diện



4.1.2 Xử lý

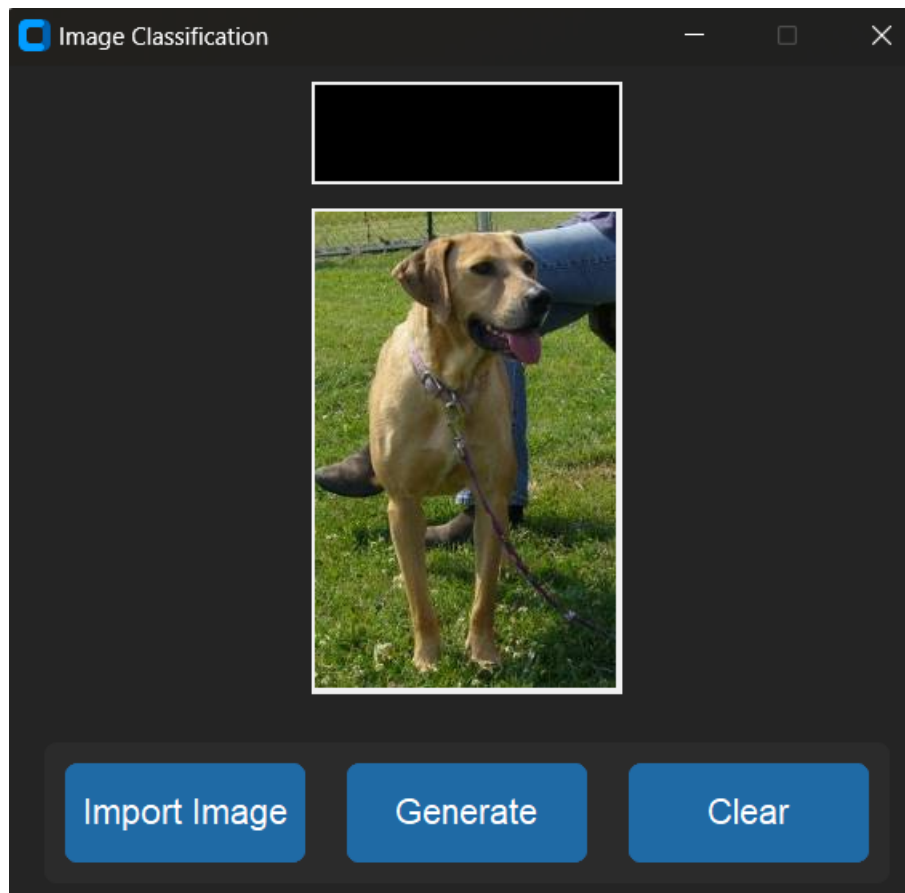
```
1 def importImage(self):
2     self.canvas_result.delete('all')
3     self.file_path = filedialog.askopenfilename()
4     if(self.file_path):
5         image = Image.open(self.file_path)
6         self.img_width = image.width
7         image = image.resize((image.width, image.height), Image.BILINEAR)
8         self.canvas.configure(width=image.width, height=image.height)
9         self.canvas_result.configure(width=image.width)
10        image = ImageTk.PhotoImage(image)
11        self.canvas.image = image
12        self.canvas.create_image(0, 0, image=image, anchor="nw")
13    else:
```



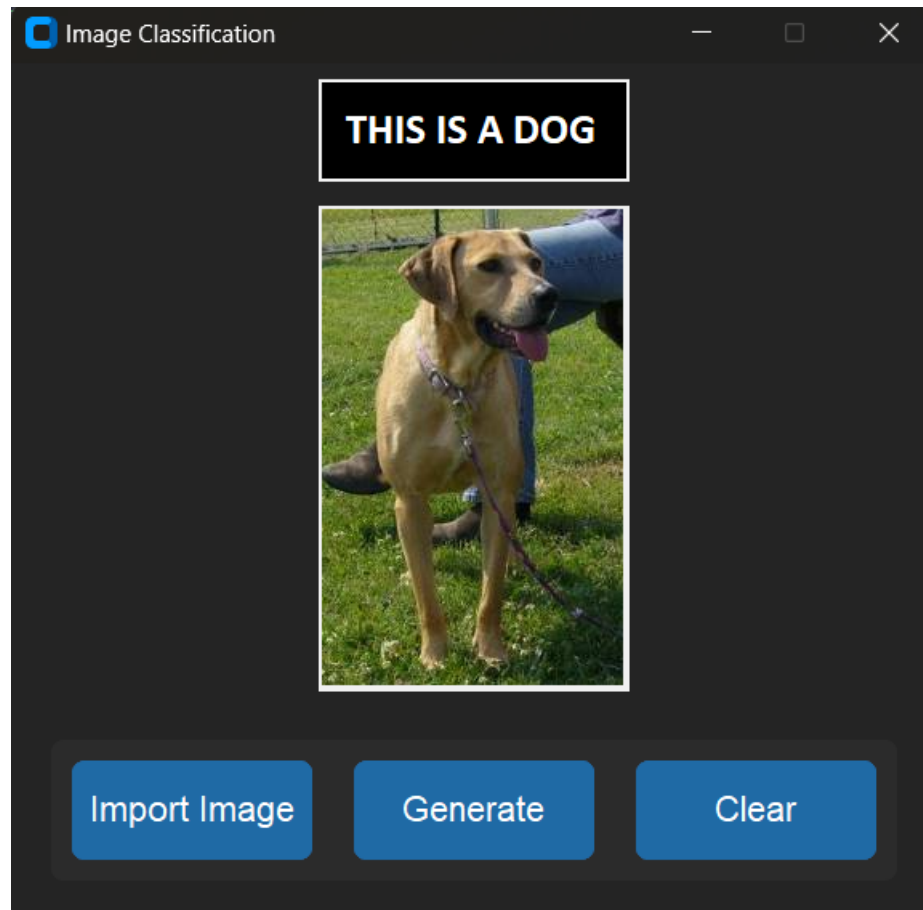
```
14 self.canvas_result.create_text(original_width/2, 30, text="PLEASE PICK YOUR  
    PICTURE !", fill="red", font=('Calibri 20 bold'))
```

4.2 Chức năng Generate(Đưa ra kết quả)

4.2.1 Giao diện trước generate



4.2.2 Giao diện sau generate



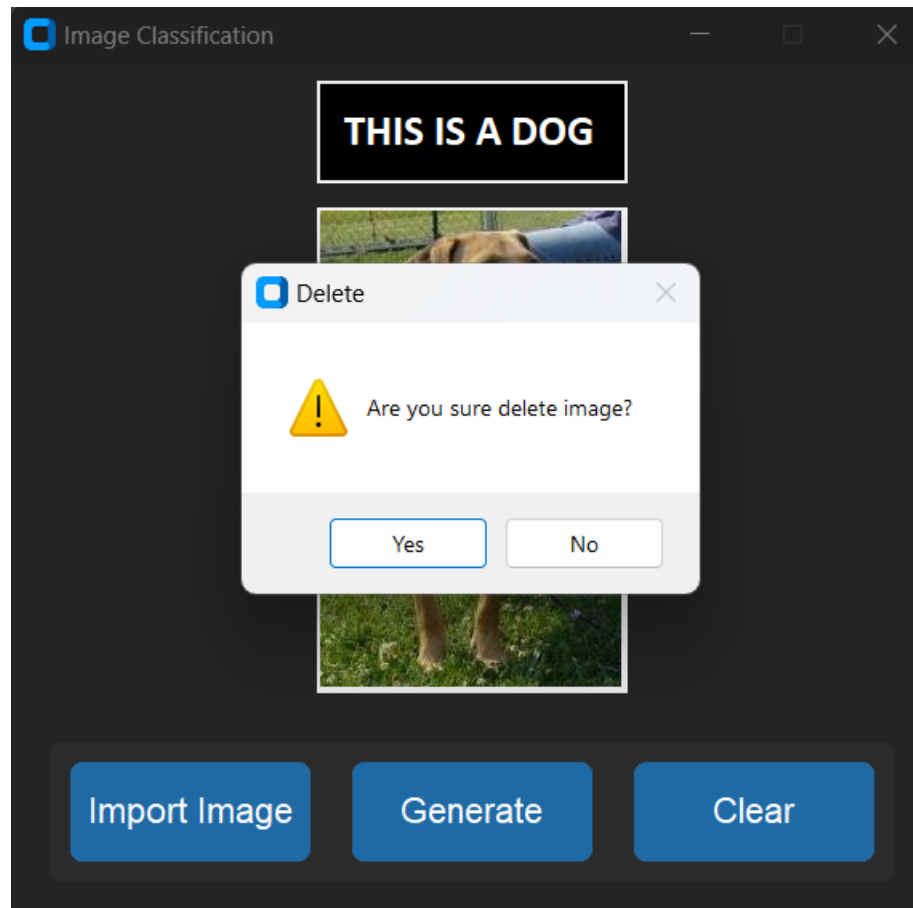
4.2.3 Xử lý

```
1 def predictImage(self):
2     if(self.file_path==""):
3         self.canvas_result.delete('all')
4         self.canvas_result.create_text(original_width/2, 30, text="PLEASE PICK YOUR
5             PICTURE !", fill="red", font=('Calibri 20 bold'))
6     else:
7         self.canvas_result.delete('all')
8         image = img.imread(self.file_path)
9         image = cv.resize(image,(128,128))
10        image = tf.expand_dims(image, axis=0)
11        pred = model.predict(image)
12        print(np.argmax(pred))
13        if(np.argmax(pred)==0):
14            self.canvas_result.create_text(self.img_width/2, 30, text="THIS IS A
15                CAT", fill="white", font=('Calibri 20 bold'))
16        else:
```

```
15 self.canvas_result.create_text(self.img_width/2, 30, text="THIS IS A  
DOG", fill="white", font=('Calibri 20 bold'))
```

4.3 Chức năng xóa hình ảnh

4.3.1 Giao diện



4.3.2 Xử lý

```
1 def clearAll(self):  
2     ans = messagebox.askquestion('Delete', 'Are you sure delete image?',  
3     icon='warning')  
4     if ans == 'yes':  
5         self.canvas.delete("all")  
6         self.canvas_result.delete('all')  
7         self.file_path=""  
8  
9         self.canvas.create_text(original_width/2, original_height/2, text="YOUR  
10         PICTURE IN HERE", fill="black", font=('Calibri 20 bold'))
```



```
9         self.canvas_result.create_text(original_width/2, 30, text="...RESULT...",  
10             fill="white", font=('Calibri 20 bold'))  
11  
12         self.canvas.configure(width=original_width, height=original_height)  
13         self.canvas_result.configure(width=original_width, height=60)
```



5 KẾT LUẬN

5.1 Tổng kết công việc đã thực hiện

- Nắm bắt được nhu cầu cũng như là khoa học kĩ thuật ngày càng đổi mới nên nhóm đã phát triển ứng dụng để ứng dụng vào thực tế và hướng con người đến gần hơn với công nghệ.
- Phát triển các chức năng cơ bản của một ứng dụng nhận diện đối tượng qua hình ảnh.

5.2 Các mặt hạn chế

- Thời gian phát triển đồ án có phần hạn chế dẫn đến quá trình hoàn thành ứng dụng bị trễ so với deadline ban đầu đề ra.
- Cần nhiều thời gian để train model dẫn tới thông tin đến CNN chưa được phong phú.

5.3 Hướng phát triển tiếp theo trong tương lai

- Tích hợp thêm các API về AI khác như nhập liệu bằng giọng nói, dịch thuật,...
- Phát triển thêm nhiều chức năng khác như scan hình ảnh sang văn bản, dịch văn bản,... để phát triển thành một ứng dụng riêng biệt và có thể đưa ra thị trường.