**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ ĐÔNG Á**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÀI TẬP LỚN**

**HỌC PHẦN: XỬ LÝ ẢNH VÀ THỊ GIÁC MÁY TÍNH**

**ĐỀ TÀI SỐ 09 : XÂY DỰNG HỆ THỐNG PHÂN LOẠI CẢM XÚC QUA KHUÔN MẶT CỦA CON NGƯỜI**

**Giảng viên hướng dẫn: Lương Thị Hồng Lan**

|  |  |  |  |
| --- | --- | --- | --- |
| **TT** | **Mã sinh viên** | **Sinh viên thực hiện** | **Lớp hành chính** |
| **1** | **20210140** | **Nuyễn Xuân Thái** | **DCCNTT 12.10.1** |
| **2** | **20210467** | **Ngô Quang Triệu** | **DCCNTT 12.10.1** |
| **3** | **20210142** | **Nguyễn Thế Tuấn** | **DCCNTT 12.10.1** |
| **4** | **20210123** | **Phan Ngọc Phạm Bằng** | **DCCNTT 12.10.1** |

**Bắc Ninh, năm 2024**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ ĐÔNG Á**

**KHOA CÔNG NGHỆ THÔNG TIN**

**BÀI TẬP LỚN**

**HỌC PHẦN: XỬ LÝ ẢNH VÀ THỊ GIÁC MÁY TÍNH**

**ĐỀ TÀI SỐ 09 : XÂY DỰNG HỆ THỐNG PHÂN LOẠI CẢM XÚC QUA KHUÔN MẶT CỦA CON NGƯỜI**

**Giảng viên hướng dẫn: Lương Thị Hồng Lan**

|  |  |  |  |
| --- | --- | --- | --- |
| **TT** | **Mã sinh viên** | **Sinh viên thực hiện** | **Lớp hành chính** |
| **1** | **20210140** | **Nuyễn Xuân Thái** | **DCCNTT 12.10.1** |
| **2** | **20210467** | **Ngô Quang Triệu** | **DCCNTT 12.10.1** |
| **3** | **20210142** | **Nguyễn Thế Tuấn** | **DCCNTT 12.10.1** |
| **4** | **20210123** | **Phan Ngọc Phạm Bằng** | **DCCNTT 12.10.1** |

|  |  |
| --- | --- |
|  |  |

**Bắc Ninh, năm 2024**

|  |  |
| --- | --- |
| TRƯỜNG ĐẠI HỌC CÔNG NGHỆ ĐÔNG Á  **KHOA CÔNG NGHỆ THÔNG TIN** | **KỲ THI KẾT THÚC HỌC PHẦN**  **HỌC KỲ 1, NĂM HỌC 2024** – **2025** |

|  |  |
| --- | --- |
| **PHIẾU CHẤM THI BÀI TẬP LỚN KẾT THÚC HỌC PHẦN**  **Mã đề thi: 09**  **Tên học phần: XỬ LÝ ẢNH VÀ THỊ GIÁC MÁY TÍNH**  **Lớp Tín chỉ:**  **XATGMT.03.K12.01.LH.C04.1\_LT** | |
| **Cán bộ chấm thi 1**  *(Ký và ghi rõ họ tên)*  Lương Thị Hồng Lan | **Cán bộ chấm thi 2**  *(Ký và ghi rõ họ tên)* |

| **TT** | **TIÊU CHÍ** | **THANG ĐIỂM** | **Nguyễn Xuân Thái** | **Ngô Quang Triệu** | **Nguyễn Thế Tuấn** | **Phan Ngọc Phạm Bằng** |
| --- | --- | --- | --- | --- | --- | --- |
| 20210140 | 20210467 | 20210142 | 20210123 |
| **1** | **Nội dung báo cáo trên Word đầy đủ** | **3.5** |  |  |  |  |
| 1.1 | Có bố cục rõ ràng (mục lục, phần mở đầu, nội dung chính, kết luận). | 0,5 |  |  |  |  |
| 1.2 | Nội dung phân tích rõ ràng, logic. | 0,5 |  |  |  |  |
| 1.3 | Có dẫn chứng, số liệu minh họa đầy đủ. | 0,5 |  |  |  |  |
| 1.4 | Ngôn ngữ và trình bày chuẩn, không lỗi chính tả. | 0,5 |  |  |  |  |
| 1.5 | Có trích dẫn tài liệu tham khảo đúng quy cách. | 0,5 |  |  |  |  |
| 1.6 | Được trình bày chuyên nghiệp (canh lề, font chữ, khoảng cách dòng hợp lý). | 0,5 |  |  |  |  |
| 1.7 | Tài liệu đầy đủ, bám sát yêu cầu của đề bài. | 0,5 |  |  |  |  |
| **2** | **Nội dung thuyết trình đầy đủ** | **1.0** |  |  |  |  |
| 2.1 | Trình bày tự tin, phát âm rõ ràng, mạch lạc. | 0,5 |  |  |  |  |
| 2.2 | Nội dung thuyết trình đúng trọng tâm, không lan man. | 0,5 |  |  |  |  |
| **3** | **Slides báo cáo đầy đủ nội dung + Hỏi đáp** | **3.0** |  |  |  |  |
| 3.1 | Slides có bố cục rõ ràng (mở đầu, nội dung, kết luận). | 0,5 |  |  |  |  |
| 3.2 | Thiết kế slides đẹp, chuyên nghiệp (màu sắc, hình ảnh minh họa). | 0,5 |  |  |  |  |
| 3.3 | Nội dung trên slides ngắn gọn, dễ hiểu, súc tích. | 0,5 |  |  |  |  |
| 3.4 | Nội dung slides phù hợp với nội dung báo cáo. | 0,5 |  |  |  |  |
| 3.5 | Trả lời câu hỏi đầy đủ, chính xác. | 0,5 |  |  |  |  |
| 3.6 | Trả lời câu hỏi tự tin, thuyết phục. | 0,5 |  |  |  |  |
| **4** | **Code đầy đủ** | **2.5** |  |  |  |  |
| 1.1 | Code được trình bày rõ ràng, có chú thích đầy đủ. | 0,5 |  |  |  |  |
| 1.2 | Code chạy đúng, không lỗi. | 0,5 |  |  |  |  |
| 1.3 | Code tối ưu, không dư thừa. | 0,5 |  |  |  |  |
| 1.4 | Đáp ứng đầy đủ các yêu cầu chức năng theo đề bài. | 0,5 |  |  |  |  |
| 1.5 | Có tính sáng tạo hoặc cải thiện so với yêu cầu. | 0,5 |  |  |  |  |
| **TỔNG ĐIỂM BẰNG SỐ:** | | **10** |  |  |  |  |
| **TỔNG ĐIỂM BẰNG CHỮ:** | | *Mười tròn* |  |  |  |  |

MỤC LỤC

[LỜI MỞ ĐẦU 8](#_Toc184658631)

[CHƯƠNG 1 : CỞ SỞ LÝ THUYẾT 9](#_Toc184658632)

[1.1 Tổng quan về xử lý ảnh 9](#_Toc184658633)

[1.2 Các quá trình xử lý ảnh 9](#_Toc184658634)

[1.3 Ảnh và biểu diễn ảnh 11](#_Toc184658635)

[1.3.1 Phạm vi ứng dụng của xử lý ảnh 13](#_Toc184658636)

[1.3.2 Các loại tệp cơ bản trong xử lý ảnh 13](#_Toc184658637)

[1.4 Học máy 29](#_Toc184658638)

[1.4.2 Học không giám sát 29](#_Toc184658640)

[1.4.3 Học bán giám sát 30](#_Toc184658641)

[1.4.1Học có giám sát 29](#_Toc184658639)

[CHƯƠNG 2: XÂY DỰNG HỆ THỐNG NHẬN DIỆN CẢM XÚC QUA KHUÔN MẶT 31](#_Toc184658642)

[2.1 Thư viện 31](#_Toc184658643)

[2.1.1 Opencv 31](#_Toc184658644)

[2.1.2 Tensorflow 31](#_Toc184658645)

[2.1.3 một số thư viện khác và công cụ sử dụng 32](#_Toc184658646)

[2.2 Xây Dựng Hệ Thống Nhận Diện Cảm Xúc Qua Khuôn Mặt 34](#_Toc184658647)

[2.2.1 Kiến Trúc Tổng Quan Của Hệ Thống 34](#_Toc184658648)

[2.3 Phân tích code 35](#_Toc184658649)

[2.3.1 Final.py 35](#_Toc184658650)

[2.3.2 Train.py 36](#_Toc184658651)

[2.3.3 haarcascade\_frontalface\_default 37](#_Toc184658652)

[2.3.4 Train.ipynb 37](#_Toc184658653)

[CHƯƠNG 3 : THỰC NGHIỆM CHƯƠNG TRÌNH 40](#_Toc184658654)

[3.1 Trang chủ 40](#_Toc184658655)

[3.2 Nhận diện cảm xúc từ ảnh 40](#_Toc184658656)

[3.3 Nhận diện cảm xúc qua video 41](#_Toc184658657)

[KẾT LUẬN 42](#_Toc184658658)

[TÀI LIỆU THAM KHẢO 43](#_Toc184658659)

# LỜI MỞ ĐẦU

Môn Xử lý ảnh và Thị giác máy tính là một môn học đã cung cấp cho em cái nhìn toàn diện về các cạnh cơ bản của số hình ảnh xử lý, đưa ra cách biểu diễn và chuyển đổi hình ảnh dữ liệu, cùng với các ứng dụng của thuật toán trong phân tích image. Không chỉ dừng lại ở lý thuyết, môn học mà còn giúp em khám phá các khái niệm nâng cao như phân đoạn ảnh, nhận dạng đối tượng và đặc biệt là học sâu – một lĩnh vực đang dẫn đầu xu hướng công nghệ. Các chủ đề đều được trình bày logic và dễ hiểu, giúp em từng bước nắm chắc những nguyên lý cốt lõi và cách áp dụng chúng trong thực tế.

Em tin rằng những kiến ​​thức và kỹ năng mà mình tiếp thu được từ môn học này sẽ trở thành thành hành trang vô giá, giúp em không chỉ trong quá trình học tập mà còn trên con đường sự nghiệp sau này. Môn **Xử lý ảnh và Thị giác máy tính** thực sự đã mang đến cho em không chỉ những kiến ​​thức hữu ích mà còn là một góc nhìn mới, sâu sắc và đầy cảm hứng về vai trò của công nghệ trong thời đại hiện nay.

# CHƯƠNG 1 : CỞ SỞ LÝ THUYẾT

## 1.1 Tổng quan về xử lý ảnh

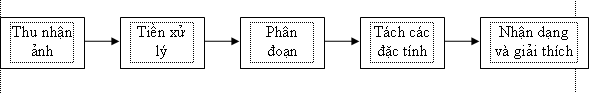
Xử lý ảnh (XLA) là đối tượng nghiên cứu của lĩnh vực thị giác máy, là quá trình biến đổi từ một ảnh ban đầu sang một ảnh mới với các đặc tính và tuân theo ý muốn của người sử dụng. Xử lý ảnh có thể gồm quá trình phân tích, phân lớp các đối tượng, làm tăng chất lượng, phân đoạn và tách cạnh, gán nhãn cho vùng hay quá trình biên dịch các thông tin hình ảnh của ảnh.

Cũng như xử lý dữ liệu bằng đồ hoạ, xử lý ảnh số là một lĩnh vực của tin học ứng dụng. Xử lý dữ liệu bằng đồ họa đề cập đến những ảnh nhân tạo, các ảnh này được xem xét như là một cấu trúc dữ liệu và được tạo bởi các chương trình. Xử lý ảnh số bao gồm các phương pháp và kỹ thuật biến đổi, để truyền tải hoặc mã hoá các ảnh tự nhiên. Mục đích của xử lý ảnh gồm:

* Biến đổi ảnh làm tăng chất lượng ảnh.
* Tự động nhận dạng ảnh, đoán nhận ảnh, đánh giá các nội dung của ảnh.

Nhận biết và đánh giá các nội dung của ảnh là sự phân tích một hình ảnh thành những phần có ý nghĩa để phân biệt đối tượng này với đối tượng khác, dựa vào đó ta có thể mô tả cấu trúc của hình ảnh ban đầu. Có thể liệt kê một số phương pháp nhận dạng cơ bản như nhận dạng ảnh của các đối tượng trên ảnh, tách cạnh, phân đoạn hình ảnh,… Kỹ thuật này được dùng nhiều trong y học (xử lý tế bào, nhiễm sắc thể), nhận dạng chữ trong văn bản.

## Các quá trình xử lý ảnh



Hinh 1.1 Các giai đoạn chính trong xử lý ảnh

Thu nhận ảnh: Đây là công đoạn đầu tiên mang tính quyết định đối với quá trình XLA. Ảnh đầu vào sẽ được thu nhận qua các thiết bị như camera, sensor, máy scanner,v.v… và sau đó các tín hiệu này sẽ được số hóa. Việc lựa chọn các thiết bị thu nhận ảnh sẽ phụ thuộc vào đặc tính của các đối tượng cần xử lý. Các thông số quan trọng ở bước này là độ phân giải, chất lượng màu, dung lượng bộ nhớ và tốc độ thu nhận ảnh của các thiết bị.

Tiền xử lý: Ở bước này, ảnh sẽ được cải thiện về độ tương phản, khử nhiễu, khử bóng, khử độ lệch,v.v… với mục đích làm cho chất lượng ảnh trở lên tốt hơn nữa, chuẩn bị cho các bước xử lý phức tạp hơn về sau trong quá trình XLA. Quá trình này thường được thực hiện bởi các bộ lọc.

Phân đoạn ảnh: phân đoạn ảnh là bước then chốt trong XLA. Giai đoạn này phân tích ảnh thành những thành phần có cùng tính chất nào đó dựa theo biên hay các vùng liên thông. Tiêu chuẩn để xác định các vùng liên thông có thể là cùng màu, cùng mức xám v.v… Mục đích của phân đoạn ảnh là để có một miêu tả tổng hợp về nhiều phần tử khác nhau cấu tạo lên ảnh thô. Vì lượng thông tin chứa trong ảnh rất lớn, trong khi đa số các ứng dụng chúng ta chỉ cần trích một vài đặc trưng nào đó, do vậy cần có một quá trình để giảm lượng thông tin khổng lồ đó. Quá trình này bao gồm phân vùng ảnh và trích chọn đặc tính chủ yếu.

Tách các đặc tính: Kết quả của bước phân đoạn ảnh thường được cho dưới dạng dữ liệu điểm ảnh thô, trong đó hàm chứa biên của một vùng ảnh, hoặc tập hợp tất cả các điểm ảnh thuộc về chính vùng ảnh đó. Trong cả hai trường hợp, sự chuyển đổi dữ liệu thô này thành một dạng thích hợp hơn cho việc xử lý trong máy tính là rất cần thiết. Để chuyển đổi chúng, câu hỏi đầu tiên cần phải trả lời là nên biểu diễn một vùng ảnh dưới dạng biên hay dưới dạng một vùng hoàn chỉnh gồm tất cả những điểm ảnh thuộc về nó. Biểu diễn dạng biên cho một vùng phù hợp với những ứng dụng chỉ quan tâm chủ yếu đến các đặc trưng hình dạng bên ngoài của đối tượng, ví dụ như các góc cạnh và điểm uốn trên biên chẳng hạn. Biểu diễn dạng vùng lại thích hợp cho những ứng dụng khai thác các tính chất bên trong của đối tượng, ví dụ như vân ảnh hoặc cấu trúc xương của nó. Sự chọn lựa cách biểu diễn thích hợp cho một vùng ảnh chỉ mới là một phần trong việc chuyển đổi dữ liệu ảnh thô sang một dạng thích hợp hơn cho các xử lý về sau. Chúng ta còn phải đưa ra một phương pháp mô tả dữ liệu đã được chuyển đổi đó sao cho những tính chất cần quan tâm đến sẽ được làm nổi bật lên, thuận tiện cho việc xử lý chúng.

Nhận dạng và giải thích: Đây là bước cuối cùng trong quá trình XLA. Nhận dạng ảnh có thể được nhìn nhận một cách đơn giản là việc gán nhãn cho các đối tượng trong ảnh. Ví dụ đối với nhận dạng chữ viết, các đối tượng trong ảnh cần nhận dạng là các mẫu chữ, ta cần tách riêng các mẫu chữ đó ra và tìm cách gán đúng các ký tự của bảng chữ cái tương ứng cho các mẫu chữ thu được trong ảnh. Giải thích là công đoạn gán nghĩa cho một tập các đối tượng đã được nhận biết.

Chúng ta cũng có thể thấy rằng, không phải bất kỳ một ứng dụng XLA nào cũng bắt buộc phải tuân theo tất cả các bước xử lý đã nêu ở trên, ví dụ như các ứng dụng chỉnh sửa ảnh nghệ thuật chỉ dừng lại ở bước tiền xử lý. Một cách tổng quát thì những chức năng xử lý bao gồm cả nhận dạng và giải thích thường chỉ có mặt trong hệ thống phân tích ảnh tự động hoặc bán tự động, được dùng để rút trích ra những thông tin quan trọng từ ảnh, ví dụ như các ứng dụng nhận dạng ký tự quang học, nhận dạng chữ viết tay v.v…

## Ảnh và biểu diễn ảnh

Ảnh trong thực tế là một ảnh liên tục cả về không gian và giá trị độ sáng. Để có thể xử lý ảnh bằng máy tính thì cần thiết phải tiến hành số hóa ảnh. Quá trình số hóa biến đổi các tín hiệu liên tục sang tín hiệu rời rạc thông qua quá trình lấy mẫu (rời rạc hóa về không gian) và lượng tử hóa các thành phần giá trị mà về nguyên tắc bằng mắt thường không thể phân biệt được hai điểm liền kề nhau. Các điểm như vậy được gọi là các pixel (Picture Element) hay các phần tử ảnh hoặc điểm ảnh. Ở đây cần phân biệt khái niệm pixel hay đề cập đến trong các hệ thống đồ họa máy tính. Để tránh nhầm lẫn ta gọi khái niệm pixel này là pixel thiết bị. Khái niệm pixel thiết bị có thể xém xét như sau: khi ta quan sát màn hình (trong chế độ đồ họa), màn hình không liên tục mà gồm các điểm nhỏ, gọi là pixel. Mỗi pixel gồm một tập tọa độ (x, y) và màu.

Như vậy mỗi ảnh là tập hợp các điểm ảnh. Khi được số hóa nó thường được biểu diễn bởi mảng 2 chiều I(n,p): n là dòng và p là cột.

Về mặt toán học có thể xem ảnh là một hàm hai biến f(x,y) với x, y là các biến tọa độ. Giá trị số ở điểm (x,y) tương ứng với giá trị xám hoặc độ sáng của ảnh (x là các cột còn y là các hàng). Giá trị của hàm ảnh f(x,y) được hạn chế trong phạm vi của các số nguyên dương.

0 ≤ f(x,y) ≤ fmax.

Với ảnh đen trắng mức xám của ảnh có thể được biểu diễn bởi một số như sau:

Trong đó SBW( ) là đặc tính phổ của cảm biến được sử dụng và k là hệ số tỷ lệ xích. Vì sự cảm nhận độ sáng có tầm quan trọng hàng đầu đối với ảnh đen trắng nên *SBW( )* được chọn giống như là hiệu suất sáng tương đối. Vì *f* biểu diễn công suất trên đơn vị diện tích, nên nó bao giờ cũng không âm và hữu hạn.

*0≤ f ≤ fmax*

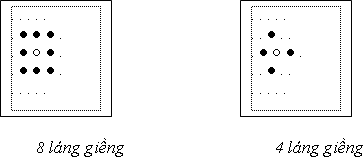
Trong đó *fmax* là giá trị lớn nhất mà *f* đạt được. Trong xử lý ảnh, *f* được chia thang sao cho nó nằm trong một phạm vi thuận lợi nào đó.

Ảnh có thể được biểu diễn theo một trong hai mô hình: mô hình Vector hoặc mô hình Raster.

Mô hình Vector: Ngoài mục đích tiết kiệm không gian lưu trữ, dễ dàng hiển thị và in ấn, các ảnh biểu diễn theo mô hình vector còn có ưu điểm cho phép dễ dàng lựa chọn, sao chép, di chuyển, tìm kiếm…Theo những yêu cầu này thì kỹ thuật biểu diễn vector tỏ ra ưu việt hơn. Trong mô hình này, người ta sử dụng hướng vector của các điểm ảnh lân cận để mã hóa và tái tạo lại hình ảnh ban đầu. Các ảnh vector được thu nhận trực tiếp từ các thiết bị số hóa như Digitalize hoặc được chuyển đổi từ các ảnh Raster thông qua các chương trình vector hóa.

Mô hình Raster: là mô hình biểu diễn ảnh thông dụng nhất hiện nay. Ảnh được biểu diễn dưới dạng ma trận các điểm ảnh. Tùy theo nhu cầu thực tế mà mỗi điểm ảnh có thể được biểu diễn bởi một hay nhiều bit. Mô hình Raster thuận lợi cho việc thu nhận, hiển thị và in ấn. Các ảnh được sử dụng trong phạm vi của đề tài này cũng là các ảnh được biểu diễn theo mô hình Raster.

Khi xử lý các ảnh Raster chúng ta có thể quan tâm đến mối quan hệ trong vùng lân cận của các điểm ảnh. Các điểm ảnh có thể xếp hàng trên một lưới (raster) hình vuông, lưới hình lục giác hoặc theo một cách hoàn toàn ngẫu nhiên với nhau.



Hình 1.2 Quan hệ trong vùng lân cận giữa các điểm ảnh.

Cách sắp xếp theo hình vuông là được quan tâm đến nhiều nhất và có hai loại: điểm 4 láng giềng (4 liền kề) hoặc 8 láng giềng (8 liền kề). Với điểm 4 láng giềng, một điểm ảnh I(i, j) sẽ có điểm kế cận theo 2 hướng i và j; trong khi đó với điểm 8 láng giềng, điểm ảnh I(i, j) sẽ có 4 điểm kế cận theo 2 hướng i, j và 4 điểm kế cận theo hướng chéo 45 o (Xem hình 1.2)

### 1.3.1 Phạm vi ứng dụng của xử lý ảnh

Xử lý ảnh đã đem lại nhiều ứng dụng trong nhiều lĩnh vực khác nhau: y học, khoa học hình hình sự, khí tượng thuỷ văn, quản lý, ...

Quản lý là là một trong những ứng dụng quan trọng của xử lý ảnh. Cùng với sự bùng nổ của kinh tế thị trường. Khối lượng quản lý càng lớn, như quản lý hồ sơ, quản lý phiếu điều tra trong công tác thống kê, các câu hỏi trắc nghiệm. Để thực hiện các công việc trên một cách chính xác, nhanh chóng và hiệu quả. Xử lý ảnh và nhận dạng đã nghiên cứu và phát triển mạnh mẽ bài toán nhập liệu tự động.

### Các loại tệp cơ bản trong xử lý ảnh

Ảnh thu đựơc sau quá trình số hoá có nhiều loại khác nhau phụ thuộc vào kỹ thuật số hoá ảnh và các ảnh thu nhận được có thể lưu trữ trên tệp để dùng cho việc xử lý các bước tiếp theo. Sau đây là một số loại tệp cơ bản và thông dụng nhất hiện nay.

1.3.2.1 File ảnh IMG

Ảnh IMG là ảnh đen trắng, phần đầu file IMG có 16 bytes chứa các thông tin cần thiết:

+ 6 bytes đầu dùng để đánh dấu nhận dạng file IMG. Giá trị của 6 bytes đầu này viết dưới dạng hexa: 0x0001 0x0008 0x0001.

+ 2 bytes chứa độ dài các mẫu tin. Đó là độ dài của một dãy các bytes lặp lại một số lần nào đó, số lần lặp này sẽ được lưu trong một file đếm. Nhiều dãy giống nhau được lưu trong một bytes. Đó chính là cách lưu trữ nén

+ 4 bytes tiếp theo mô tả kích cỡ của pixel

+ 2 bytes tiếp mô tả số pixel trên một dòng

+2 bytes cuối cho biết số dòng trong ảnh

Các dòng giống nhau được nén thành một pack. Có 4 loại pack sau:

* Loại 1: Gói các dòng giống nhau.Quy cách gói tin này 0x00 0x00 0xFF Count. 3 bytes đầu cho biết số các dãy giống nhau ,bytes cuối cho biết số các dòng giống nhau.
* Loại 2: Gói các dãy giống nhau. Quy cách gói này 0x00 Count. Bytes thứ hai cho số các dãy giống nhau được nén trong gói. Độ dài cử dãy được ghi đầu file.
* Loại 3:Dãy các pixel không giống nhau, không lặp lại và không nén được. Quy cách như sau: 0x80 Count. Bytes thứ hai cho biết độ dài dãy các pixel không giống nhau, không nén được.
* Loại 4:Dãy các pixel giống nhau. Tuỳ theo các bit cao của bytes đầu được bật hay tắt, nếu bit cao được bật(giá trị 1) thì đây là gói nén các bytes chỉ gồm bit 0, số các bytes được nén được tính bởi 7 bit thấp còn lại. Nếu bit cao tắt (giá tri 0) thì đay là gói nén các bytes toàn bit 1. Số các bytes được nén được tính bởi 7 bit thấp còn lại.

Các cảu file IMG phong phú như vậy là do ảnh IMG là ảnh đen trắng nên chỉ cần 1 bit cho một pixel thay vì 4 hoặc 8 bit như đã nói ở trên toàn bộ ảnh chỉ có điểm sáng hoặc tối tương ứng với 1 hoặc 0. Tỷ lệ nén của file này là khá cao

1.3.2.2 File ảnh PCX

Định dạng ảnh PCX là một trong những định dạng ảnh cổ điển nhất. Nó sử dụng phương pháp mã loạt dài RLE (Run-Length-Encoded) để nén dữ liệu ảnh. Quá trình nén và giải nén được thực hiện trên từng dòng ảnh. Thực tế, phương pháp giải nén PCX kém hiệu quả hơn so với kiểu IMG. Tệp PCX gồm 3 phần: đầu tệp (header), dữ liệuảnh (image data) và bảng màu mở rộng.

Header của tệp PCX có kích thước cố định gồm 128 byte và được phân bố như sau:

+ 1 byte : chỉ ra kiểu định dạng. Nếu là kiểu PCX/PCC nó luôn có giá trị là

0Ah.

+ 1 byte: chỉ ra version sử dụng để nén ảnh, có thể có các giá trị sau:

* 0: version 2.5.
* 2: version 2.8 với bảng màu.
* 3: version 2.8 hay 3.0 không có bảng màu.
* 5: version 3.0 có bảng màu.

+ 1 byte: chỉ ra phương pháp mã hoá. Nếu là 0 thì mã hoá theo phương pháp

BYTE PACKED, nếu không là phương pháp RLE.

+ 1 byte: số bit cho một điểm ảnh plane.

+ 1 word: toạ độ góc trái trên của ảnh. Với kiểu PCX nó có giá trị là (0,0); còn PCC thì khác (0,0).

+ 1 word: toạ độ góc phải dưới.

+ 1 word: kích thước bề rộng và bề cao ảnh.

+ 1 word: số điểm ảnh.

+ 1 word: độ phân giải màn hình.

+ 1 word.

+ 48 byte: chia thành 16 nhóm, mỗi nhóm 3 byte. Mỗi nhóm này chứa thông tin về một thanh ghi màu. Như vậy ta có 16 thanh ghi màu.

+ 1 byte: không dùng đến và luôn đặt là 0.

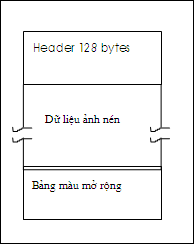
+1 byte: số bit plane mà ảnh sử dụng. Với ảnh 16 màu, giá trị này là 4, với ảnh 256 màu (1 pixel/8 bit) thì số bit plane lại là 1.

+ 1 byte: số bytes cho một dòng quét ảnh.

+ 1 word: kiểu bảng màu.

+ 58 byte: không dùng.

Tóm lại, định dạng ảnh PCX thường được dùng để lưu trữ ảnh vì thao tác đơn giản, cho phép nén và giải nén nhanh. Tuy nhiên vì cấu trúc của nó cố định, nên trong một số trường hợp nó làm tăng kích thước lưu trữ. Và cũng vì nhược điểm này mà một số ứng dụng lại sử dụng một kiểu định dạng khác mềm dẻo hơn: định dạng TIFF (Targed Image File Format) sẽ mô tả dưới đây.



**Hình 1.3 Cấu trúc tệp ảnh dạng PCX.**

1.3.2.3 Kỹ thuật nén ảnh PCX

Kiểu nén: Thông tin về giá trị điểm xám cho mỗi điểm ảnh PCX được lưu trữ theo kiểu nén, khi được lưu trữ theo kiễu nén các file phải tuân theo quy luật nhất định: là một ma trận hai chiều để lưu trữ thông tin liên quan về các giá trị mức xám. Kỹ thuật dùng để nén ảnh PCX là kỹ thuật Run Length Encode (RLE), phần tử thông tin cần nén là 1 bytes.

Tỷ số nén: Trong kỹ thuật nén ảnh người ta quan tâm nhiều đến tỷ số nén. Tỷ số nén của ảnh được tính bởi tỷ số giữa kích thước lưu trữ ảnh sau khi nén trên kích thước cần thiết để lưu trữ ảnh không nén. Giá trị của tỷ số này phụ thuộc vào mỗi file ảnh, ảnh pcx có thể là 1,4 hoặc 8 bits, nếu xét yếu tố này ảnh hưởng đến tỷ số nén ta thấy:

* Ảnh 1 bits (hay ảnh nhị phân) thì một bytes lưu trữ 8 bits khả năng xuất hiện mỗi mức xám là lớn (50% cho mỗi mức xám) làm cho tần xuất lặp bits là lớn, yếu tố này làm tăng khả năng nén. Nhưng phải ít nhất 3 bytes liên tiếp giống nhau trong một dòng quét thì mới có hiệu quả cho việc nén tức là tần xuất lặp ở đây không phải cho từng pixel mà là cả gói 8 pixel cùng lặp giống nhau, yếu tố này làm giảm khả năng nén. Vậy việc nén ảnh nhị phân chỉ có ý nghĩa đối với ảnh có nền, còn đối với một số ảnh nhị phân khác việc nén không có ý nghĩa có khi càn làm tăng thêm kích thưóc của ảnh.
* Ảnh 4 bits (hay 16 màu) tương ứng với 4 bits mã hoá một pixel, ảnh này có 2 pixel được chứa trong một bytes. Khả năng xuất hiện cho mỗi mức màu là 1/16. Yếu tố này làm giảm đi khả năng nén so với ảnh nhị phân. Cần có ít nhất 3 bytes liên tiếp giống nhau cùng trong một dòng quét thì mới có hiệu quả nén, tần số lặp pixel ở đây là lặp gói gồm hai pixel, yếu tố này làm tăng khả năng nén hơn so với ảnh nhị phân.
* Ảnh 8 bits (hay ảnh 256 màu) tương ứng với 8 bits hay 1 bytes mã hoá một pixel. Khả năng xuất hiện cho mỗi mức màu là 1/256, yếu tố này làm giảm khả năng nén so với ảnh nhị phân và ảnh 4 bits. chỉ cần ít nhất 3 bytes (hay 3 pixel) liên tiếp giống nhau mà cùng nằm trong một dòng quét thì có hiệu quả nén.

Như vậy đối với mỗi ảnh Pcx 1,4,8 bits màu thì mỗi loại đều có các yếu tăng hoặc giảm khả năng nén. nếu ảnh nào sử dụng nền hoặc chỉ dùng một số mức màu nhất định trong bảng màu thì có khả năng nén cao.

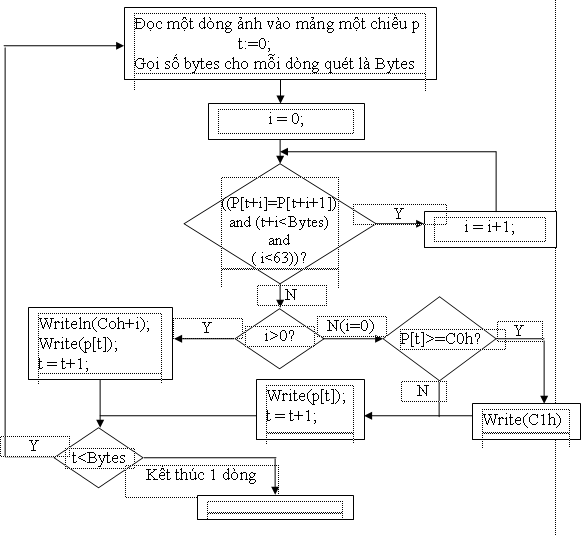
Dấu hiệu nén trong file trong ảnh PCX: Cấu trúc nén trong một dòng ảnh bao

gồm hai bytes, bytes đầu là dấu hiệu nén và số bytes được nén, bytes tiếp theo chứa

chỉ số màu của các bytes đó. Bytes dùng làm dấu hiệu nén là một bytes đặc biệt nó được chia làm hai phần như hình vẽ sau:



*Hỉnh 1.4 Cấu trúc của bytes dấu hiệu*

Phần cố định là C0h (1100 0000b), có 2 bits cao nhất là 1, số bits thấp hơn còn lại (gồm 6 bits) dùng để chỉ số bytes giống nhau liên tiếp. Như vậy mỗi cấu trúc chỉ có thể ghi được tối đa là 63 bytes giống nhau.

*Hình 1.5 Sơ đồ giải thuật nén một dòng ảnh cho file PCX*

1.3.2.4 Giải nén ảnh PCX

Quá trình nén được tiến hành theo từng dòng như sau:

+ Thứ tự đầu tiên trong file ảnh PCX là dòng đầu tiên của ảnh.

+ Việc nén file ảnh PCX phải bắt đầu từ dòng đầu tiên của ảnh.

+ Kết thúc khi tất cả các dòng đều được nén.

+ Mỗi một dòng nén phải tuân theo cùng một giải thuật nén của file PCX.

1.3.2.5 Định dạng ảnh TIFF

Kiểu định dạng TIFF được thiết kế để làm nhẹ bớt các vấn đề liên quan đến việc mở rộng tệp ảnh cố định. Về cấu trúc, nó cũng gồm 3 phần chính:

- Phần Header (IFH): có trong tất cả các tệp TIFF và gồm 8 byte:

+ 1 word: chỉ ra kiểu tạo tệp trên máy tính PC hay Macintosh. Hai loại này khác nhau rất lớn ở thứ tự các byte lưu trữ trong các số dài 2 hay 4 byte. Nếu trường này có giá trị là 4D4Dh thì đó là ảnh cho máy Macintosh; nếu là 4949h là của máy PC.

+ 1 word: version. Từ này luôn có giá trị là 42. Có thể coi đó là đặc trưng của file TIFF vì nó không thay đổi.

+ 2 word: giá trị Offset theo byte tính từ đầu file tới cấu trúc IFD(Image File Directory) là cấu trúc thứ hai của file. Thứ tự các byte ở đây phụ thuộc vào dấu hiệu trường đầu tiên.

- Phần thứ 2 (IFD): Nó không ở ngay sau cấu trúc IFH mà vị trí của nó được xác định bởi trường Offset trong đầu tệp. Có thể có một hay nhiều IFD cùng tồn tại trong file (nếu file có nhiều hơn 1 ảnh).

Một IFD gồm:

+ 2 byte: chứa các DE (Directory Entry).

+ 12 byte là các DE xếp liên tiếp. Mỗi DE chiếm 12 byte.

+ 4 byte : chứa Offset trỏ tới IFD tiếp theo. Nếu đây là IFD cuối cùng thì trường này có giá trị là 0.

- Cấu trúc phần dữ liệu thứ 3: các DE.

Các DE có độ dài cố định gồm 12 byte và chia làm 4 phần:

+ 2 byte: Chỉ ra dấu hiệu mà tệp ảnh đã được xây dựng.

+ 2 byte: kiểu dữ liệu của tham số ảnh. Có 5 kiểu tham số cơ bản:

1: BYTE (1 byte).

2: ASCII (1 byte).

3: SHORT (2 byte).

4: LONG (4 byte).

5: RATIONAL (8 byte).

+ 4 byte: trường độ dài (bộ đếm) chứa số lượng chỉ mục của kiểu dữ liệu đã chỉ ra . Nó không phải là tổng số byte cần thiết để lưu trữ. Để có số liệu này ta cần nhân số chỉ mục với kiểu dữ liệu đã dùng.

+ 4 byte: đó là Offset tới điểm bắt đầu dữ liệu thực liên quan tới dấu hiệu, tức là dữ liệu liên quan với DE không phải lưu trữ vật lý cùng với nó nằm ở một vị trí nào đó trong file.

Dữ liệu chứa trong tệp thường được tổ chức thành các nhóm dòng (cột) quét của dữ liệu ảnh. Cách tổ chức này làm giảm bộ nhớ cần thiết cho việc đọc tệp. Việc giải nén được thực hiện theo bốn kiểu khác nhau được lưu trữ trong byte dấu hiệu nén.

Như đã nói ở trên, file ảnh TIFF là dùng để giải quyết vấn đề khó mở rộng của file PCX. Tuy nhiên, với cùng một ảnh thì việc dùng file PCX chiếm ít không gian nhớ hơn.

1.3.2.6 Định dạng ảnh GIF(Graphics Interchanger Format)

Cách lưu trữ kiểu PCX có lợi về không gian lưu trữ: với ảnh đen trắng kích thước tệp có thể nhỏ hơn bản gốc từ 5 đến7 lần. Với ảnh 16 màu, kích thướ c ảnh nhỏ hơn ảnh gốc 2-3 lần, có trường hợp có thể xấp xỉ ảnh gốc. Tuy nhiên, với ảnh 256 màu thì nó bộc lộ rõ khả năng nén rất kém. Điều này có thể lý giải như sau: khi số màu tăng lên, các loạt dài xuất hiện ít hơn và vì thế, lưu trữ theo kiểu PCX không còn lợi nữa. Hơn nữa, nếu ta muốn lưu trữ nhiều đối tượng trên một tệp ảnh như kiểu định dạng TIFF, đòi hỏi có một định dạng khác thích hợp.

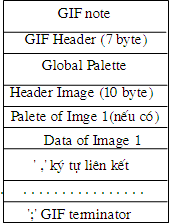
Định dạng ảnh GIF do hãng ComputServer Incorporated (Mỹ) đề xuất lần đầu tiên vào năm 1990. Với địng dạng GIF, những vướng mắc mà các định dạng khác gặp phải khi số màu trong ảnh tăng lên không còn nữa. Khi số màu càng tăng thì ưu thế của định dạng GIF càng nổi trội. Những ưu thế này có được là do GIF tiếp cận các thuật toán nén LZW(Lempel-Ziv-Welch). Bản chất của kỹ thuật nén LZW là dựa vào sự lặp lại của một nhóm điểm chứ không phải loạt dài giống nhau. Do vậy, dữ liệu càng lớn thì sự lặp lại càng nhiều. Dạng ảnh GIF cho chất lượng cao, độ phân giải đồ hoạ cũng đạt cao, cho phép hiển thị trên hầu hết các phần cứng đồ hoạ.

Định dạng tổng quát của ảnh GIF như sau:

* Chữ ký của ảnh
* Bộ mô tả hiển thị
* Bản đồ màu tổng thể
* Mô tả một đối tượng của ảnh
  + Dấu phân cách
  + Bộ mô tả ảnh
  + Bản đồ màu cục bộ
  + Dữ liệu ảnh .

Phần mô tả này lặp n lần nếu ảnh chứa n đối tượng.

* Phần đầu cuối ảnh GIF(terminator)



* + Chứ ký của ảnh GIF có giá trị là GIF87a. Nó gồm 6 ký tự, 3 kí tự đầu chỉ ra kiểu định dạng, 3 ký tự sau chỉ ra version của ảnh.
  + Bộ hình hiển thị: chứa mô tả các thông số cho toàn bộ ảnh GIF:

+ Độ rộng hình raster theo pixel: 2 byte;

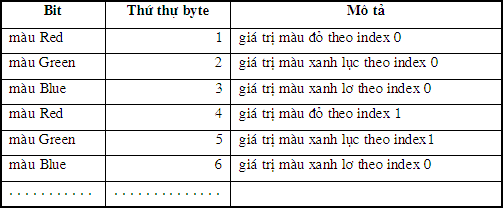
+ Độ cao hình raster theo pixel: 2 byte;

+ Các thông tin về bản đồ màu, hình hiển thị,...

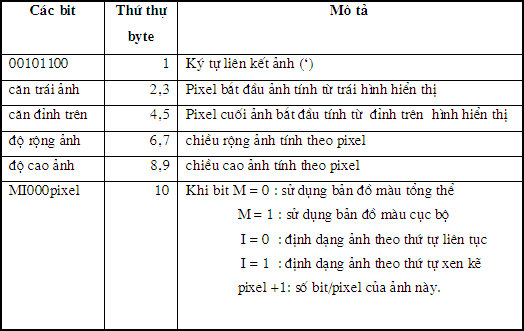
+ Thông tin màu nền: 1 byte;

+ Phần chưa dùng: 1 byte.

* + Bản đồ màu tổng thể: mô tả bộ màu tối ưu đòi hỏi khi bit M = 1. Khi bộ màu tổng thể được thể hiện, nó sẽ xác lập ngay bộ mô tả hình hiển thị. Số lượng thực thể bản đồ màu lấy theo bộ mô tả hình hiển thị ở trên và bằng 2 m, với m là lượng bit trên một pixel khi mỗi thực thể chứa đựng 3 byte (biểu diễn cường độ màu của ba màu cơ bản Red-Green-Blue). Cấu trúc của khối này như sau:



* + Bộ mô tả ảnh: định nghĩa vị trí thực tế và phần mở rộng của ảnh trong phạm vi không gian ảnh đã có trong phần mô tả hình hiển thị. Nếu ảnh biểu diễn theo ánh xạ bản đồ màu cục bộ thì cờ định nh\ghĩa phải được thiết lập. Mỗi bộ mô tả ảnh được chỉ ra bởi ký tự kết nối ảnh. Ký tự này chỉ được dùng khi định dạng GIF có từ 2 ảnh trở lên. Ký tự này có giá trị 0x2c (ký tự dấu phảy). Khi ký tự này được đọc qua, bộ mô tả ảnh sẽ được kích hoạt. Bộ mô tả ảnh gồm 10 byte và có cấu trúc như sau:



* Bản đồ màu cục bộ: bản đồ màu cục bộ chỉ được chọn khi bit M của byte thứ 10 là

1. Khi bản đồ màu được chọn, bản đồ màu sẽ chiếu theo bộ mô tả ảnh mà lấy vào cho đúng. Tại phần cuối ảnh, bản đồ màu sẽ lấy lại phần xác lập sau bộ mô tả hình hiển thị. Lưu ý là trường “pixel “ của byte thứ 10 chỉ được dùng khi bản đồ màu được chỉ định. Các tham số này không những chỉ cho biết kích thước ảnh theo pixel mà còn chỉ ra số thực thể bản đồ màu của nó.

* + Dữ liệu ảnh: chuỗi các giá trị có thứ tự của các pixel màu tạo nên ảnh. Các pixel được xếp liên tục trên một dòng ảnh, từ trái qua phải. Các dòng ảnh được viết từ trên xuống dưới.
  + Phần kết thúc ảnh: cung cấp tính đồng bộ cho đầu cuối của ảnh GIF. Cuối của ảnh sẽ xác định bởi kí tự “;” (0x3b).

Định dạng GIF có rất nhiều ưu điểm và đã được công nhận là chuẩn để lưu trữ ảnh màu thực tế (chuẩn ISO 10918-1). Nó được mọi trình duyệt Web (Web Browser) hỗ trợ với nhiều ứng dụng hiện đại. Cùng với nó có chuẩn JPEG (Joint Photograph Expert Group). GIF dùng cho các ảnh đồ hoạ (Graphic), còn JPEG dùng cho ảnh chụp (Photographic).

1.3.2.7 File ảnh BMP (BITMAP)

Khái niệm về ảnh đen trắng, ảnh màu, ảnh cấp xám.

* + - * + *Ảnh đen trắng.*

Đó là những bức ảnh mà mỗi điểm ảnh chỉ là những điểm đen hoặc trắng, được quy định bằng một bit. Nếu bit mang giá trị là 0 thì điểm ảnh là điểm đen, còn nếu mang giá trị là 1 thì điểm ảnh là điểm trắng. Do đó để biểu diễn một điểm ảnh đen trắng ta có thể dùng một ma trận nhị phân, là ma trận mà mỗi phần tử chỉ nhận một trong hai giá trị là 0 hoặc 1.

* + - * + *Ảnh màu*

Quá trình giấu tin vào ảnh màu cũng tương tự như với ảnh đen trắng nhưng trước hết ta phải chọn từ mỗi điểm ảnh ra bit có trọng số thấp nhất (LSB) để tạo thành một ảnh nhị phân gọi là ảnh thứ cấp. Sử dụng ảnh thứ cấp này như ảnh môi trường để giấu tin, sau khi biến đổi ảnh thứ cấp ta trả nó lại ảnh ban đầu để thu được ảnh kết quả.

* + - * + *Ảnh đa cấp xám*

Đối với ảnh đa cấp xám bảng màu của nó đã có sẵn, tức là những cặp màu trong bảng màu có chỉ số chênh lệch càng ít thì càng giống nhau. Vì vậy đối với ảnh đa cấp xám bit LSB của mỗi điểm ảnh là bit cuối cùng của mỗi điểm ảnh.

Quá trình tách bit LSB của ảnh đa cấp xám và thay đổi các bit này bằng thuật toán giấu tin trong ảnh đen trắng sẽ làm chỉ số của điểm màu bị thay đổi tăng hoặc giảm 1 đơn vị, do đó điểm ảnh mới sẽ có độ sáng tối của ô màu liền trước hoặc liền sau ô màu của điểm ảnh cũ. Bằng mắt thường rất khó có thể nhận thấy sự thay đổi về độ sáng tối này.

* + - * + *Ảnh nhỏ hơn hoặc bằng 8 màu*

Những ảnh thuộc loại này gồm có 16 màu (4 bit màu) và ảnh 256 màu (8 bit màu). Khác với ảnh màu, ảnh xám với số bit nhỏ hơn hoặc bằng 8 bit không phải luôn luôn được sắp xếp màu bảng màu.

Những màu ở liền kề nhau trong bảng màu có thể rất khác nhau chẳng hạn như màu đen với màu trắng vẫn có thể được xếp cạnh nhau.

Vì vậy việc xác định bit LSB của ảnh loại này rất khó. Nếu ta chỉ làm như đối với ảnh xám, tức là vẫn lấy bit cuối cùng của mỗi điểm ảnh để tạo thành ảnh thứ cấp thì mỗi thay đổi 0 -> 1 hoặc 1 ->0 trên ảnh thứ cấp có thể làm cho ảnh màu của điểm ảnh cũ và mới tương đương ứng thay đổi rất nhiều dù chỉ số màu của chúng cũng tăng hoặc giảm 1 mà thôi.

* + - * + *Ảnh hightcolor (16 bit màu)*

Ảnh 16 bit màu thực tế chỉ sử dụng 15 bit cho mỗi điểm ảnh trong đó 5 bit biểu diễn cường độ tương đối của màu đỏ, 5 bit biểu diễn cường độ tương đối của màu xanh lam, 5 bit biểu diễn cường độ tương đối của màu xanh lơ. Còn lại một bit không dùng đến là bit cao nhất của byte thứ hai trong mỗi cặp thứ hai byte biểu diễn một điểm ảnh, đó chính là bit LSB của ảnh 16 bit màu. Việc thay đổi giá trị của những bit này sẽ không hề ảnh hưởng tới màu sắc của từng điểm ảnh trong môi trường.

* + - * + *Ảnh true color (24 bit màu)*

Ảnh true color sử dụng 3 byte cho mỗi điểm ảnh, mỗi byte biểu diễn một thành phần trong cấu trúc RGB. Trong mỗi byte các bit cuối cùng của mỗi byte trong phần dữ liệu ảnh là các bit LSB của ảnh true color.

Để tăng lượng thông tin giấu được vào ảnh môi trường, từ mỗi byte của ảnh true color ra sẽ lấy nhiều hơn một bit để tạo thành ảnh thứ cấp. Thông thường cũng chỉ nên lấy nhiều nhất 4 bit cuối cùng của mỗi byte để ảnh kết quả không bị nhiễu đáng kể, khi đó lượng thông tin tối đa có thể giấu trong ảnh cũng tăng lên gấp bốn lần so với lượng thông tin tối đa giấu được trong ảnh đó nếu chỉ lấy 1 bit cuối cùng ở từng byte.

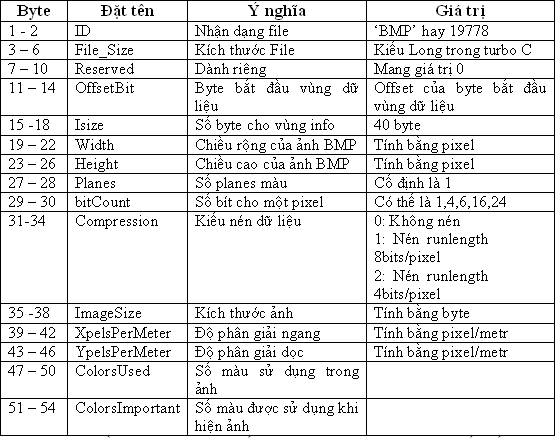
Cấu trúc ảnh BMP

Để thực hiện việc giấu tin trong ảnh, trước hết ta phải nghiên cứu cấu trúc của ảnh và có khả năng xử lý được ảnh tức là phải số hoá ảnh. Quá trình số hoá các dạng ảnh khác nhau và không như nhau. Có nhiều loại ảnh đã được chuẩn hoá như: JPEG, PCX, BMP… Sau đây là cấu trúc ảnh \*.BMP.

Mỗi file ảnh BMP gồm 3 phần:

* BitmapHeader (54 byte)
* Palette màu (bảng màu)
* BitmapData (thông tin ảnh) Cấu trúc cụ thể của ảnh:
* Palette màu (bảng màu): bảng màu của ảnh, chỉ những ảnh lớn hơn hoặc bằng 8 bit màu mới có Palette màu.
* BitmapData (thông tin ảnh): phần này nằm ngay sau phần palette màu của ảnh BMP. Đây là phần chứa giá trị màu của điểm ảnh trong ảnh BMP, các dòng ảnh được lưu từ dưới lên trên, các điểm ảnh được lưu từ trái sang phải. Giá trị của mỗi điểm ảnh là một chỉ số trỏ tới phần tử màu tương ứng của palette màu.

BitmapHeader (54 byte)



* Thành phần BitCount của cấu trúc BitmapHeader cho biết số bit dành cho mỗi điểm ảnh và số lượng màu lớn nhất của ảnh. BitCount có thể nhận các giá trị sau:

1: Bitmap là ảnh đen trắng, mỗi bit biểu diễn 1 điểm ảnh. Nếu bit mang giá trị 0 thì điểm ảnh là đen, bit mang giá trị 1 điểm ảnh là điểm trắng.

4: Bitmap là ảnh 16 màu, mỗi điểm ảnh được biểu diễn bởi 4 bit. 8: Bitmap là ảnh 256 màu, mỗi điểm ảnh biểu diễn bởi 1 byte.

16: Bitmap là ảnh highcolor, mỗi dãy 2 byte liên tiếp trong bitmap biểu diễn cường độ tương đối của màu đỏ, xanh lá cây, xanh lơ của một điểm ảnh.

24: Bitmap là ảnh true color (224 màu), mỗi dãy 3 byte liên tiếp trong bitmap biểu diễn cường độ tương đối của màu đỏ, xanh lá cây, xanh lơ (RGB) của một điểm ảnh.

* Thành phần ColorUsed của cấu trúc BitmapHeader xác định số lượng màu của palette màu thực sự được sử dụng để hiển thị bitmap. Nếu thành phần này được đặt là 0, bitmap sử dụng số màu lớn nhất tương ứng với giá trị của BitCount.
* Cấu trúc ảnh PNG

Là một dạng hình ảnh sử dụng phương pháp nén dữ liệu mới – không làm mất đi dữ liệu gốc. PNG được tạo ra nhằm cải thiện và thay thế định dạng ảnh GIF với một định dạng hình ảnh không đòi hỏi phải có giấy phép sáng chế sử dụng. PNG được hỗ trợ bởi thư viện tham chiếu libpng, một thư viện nền độc lập bao gồm các hàm của C để quản lý các hình ảnh PNG.

Những tập tin PNG thường có phần mở rộng là PNG và đã được gán kiểu chuẩn MIME là image/png.

Một tập tin PNG bao gồm 8 – byte kí hiệu (89 50 4E 47 0D 0A 1A) được viết trong hệ thống có cơ số 16, chứa các chữ “PNG” và 2 dấu xuống dòng, ở giữa là xếp theo số lượng của các thành phần, mỗi thành phần đều chứa thông tin về hình ảnh. Cấu trúc dựa trên các thành phần được thiết kế cho phép định dạng PNG có thể tương thích với các phiên bản cũ khi sử dụng. Các “thành phần” trong tập tin.

PNG là cấu trúc như một chuỗi các thành phần, mỗi thành phần chứa kích thước, kiểu, dữ liệu, và mã sửa lỗi CRC ngay trong nó.

Chuỗi được gán tên bằng 4 chữ cái phân biệt chữ hoa chữ thường. Sự phân biệt này giúp bộ giải mã phát hiện bản chất của chuỗi khi nó không nhận dạng được.

Với chữ cái đầu, viết hoa thể hiện chuỗi này là thiết yếu, nếu không thì ít cần thiết hơn ancillary. Chuỗi thiết yếu chứa thông tin cần thiết để đọc được tệp và nếu bộ

giải mã không nhận dạng được chuỗi thiết yếu,việc đọc tệp phải được hủy.

Về cơ bản, định dạng PNG đem lại cho ta những ưu thế vượt trội hơn so với các định dạng phổ thông khác hiện nay như JPG, GIF, BMP…Những ưu thế tỏ rõ sức mạnh hơn khi được sử dụng trong môi trường đồ họa web.

Giảm thiểu dung lượng: Trong tất cả các định dạng ảnh phổ thông hiện nay thì hình ảnh PNG có thể coi là dung lượng nhỏ nhất. Điều này rất quan trọng khi sử dụng PNG trong môi trường web.

Độ sâu của màu: Ảnh PNG hỗ trợ đến true color 48bit màu. Trong khi đó ảnh gif chỉ ở mức 256 màu. Để loại bỏ những khó khăn này thì việc dịch chuyển ảnh đã scan cho chuẩn với ảnh mẫu là rất cần thiết. Nó giúp tăng độ chuẩn xác khi chấm các bài thi chắc nghiệm hoặc trong các phiếu điều tra.

## 1.4 Học máy

Tổng quan về học máy là một phần quan trọng trong nghiên cứu trí tuệ nhân tạo, giúp máy tính học hỏi từ dữ liệu và đưa ra quyết định hoặc dự đoán. Trong phần này, chúng ta sẽ tìm hiểu các phương pháp học máy chính, bao gồm:

### **1.4.1Học có giám sát**

Học có giám sát (Supervised Learning) là phương pháp học từ một tập dữ liệu đã được gắn nhãn trước. Mỗi dữ liệu trong tập này bao gồm đầu vào và đầu ra mong muốn. Mục tiêu của học có giám sát là xây dựng một mô hình có khả năng dự đoán chính xác đầu ra cho dữ liệu mới dựa trên các mẫu đã học.

* **Ứng dụng:**
  + Phân loại (Classification): Chẩn đoán bệnh dựa trên triệu chứng, nhận diện chữ viết tay.
  + Hồi quy (Regression): Dự đoán giá nhà, dự đoán thời tiết.
* **Thuật toán phổ biến:**
  + Hồi quy tuyến tính (Linear Regression)
  + Cây quyết định (Decision Trees)
  + Máy vector hỗ trợ (Support Vector Machines)
  + Mạng nơ-ron nhân tạo (Artificial Neural Networks)

### **1.4.2 Học không giám sát**

Học không giám sát (Unsupervised Learning) không yêu cầu dữ liệu có gắn nhãn trước. Phương pháp này tập trung vào việc tìm kiếm cấu trúc tiềm ẩn trong dữ liệu, như nhóm hoặc mẫu.

* **Ứng dụng:**
  + Phân cụm (Clustering): Phân nhóm khách hàng, phát hiện hành vi gian lận.
  + Giảm số chiều (Dimensionality Reduction): Tăng hiệu quả trong xử lý dữ liệu lớn, hình dung dữ liệu.
* **Thuật toán phổ biến:**
  + K-means Clustering
  + Phân tích thành phần chính (PCA - Principal Component Analysis)
  + Hệ thống đề xuất (Recommendation Systems)

### **1.4.3 Học bán giám sát**

Học bán giám sát (Semi-supervised Learning) kết hợp cả hai phương pháp học có giám sát và không giám sát. Phương pháp này sử dụng một lượng nhỏ dữ liệu có gắn nhãn cùng với một lượng lớn dữ liệu chưa gắn nhãn để cải thiện hiệu suất học tập.

* **Ưu điểm:**
  + Hiệu quả cao trong các trường hợp dữ liệu có gắn nhãn khó thu thập hoặc tốn kém.
  + Tận dụng được cả dữ liệu có gắn nhãn và chưa gắn nhãn.
* **Ứng dụng:**
  + Phân loại văn bản: Email spam, phân loại tài liệu.
  + Phát hiện bất thường: Phát hiện lỗi kỹ thuật hoặc gian lận trong dữ liệu.
* **Thuật toán phổ biến:**
* Graph-based Methods
* Self-training
* Co-training

# CHƯƠNG 2: XÂY DỰNG HỆ THỐNG NHẬN DIỆN CẢM XÚC QUA KHUÔN MẶT

## 2.1 Thư viện

### 2.1.1 Opencv

OpenCV là tên viết tắt của open source computer vision library – hoàn toàn có thể được hiểu là một thư viện nguồn mở cho máy tính. Cụ thể hơn OpenCV là kho tàng trữ những mã nguồn mở được dùng để giải quyết và xử lý hình ảnh, tăng trưởng những ứng dụng đồ họa trong thời hạn thực. OpenCV được cho phép cải tổ vận tốc của CPU khi triển khai những hoạt động giải trí real time. Nó còn phân phối một số lượng lớn những mã giải quyết và xử lý Giao hàng cho quá trình của thị giác máy tính hay những learning machine khác.

Thư viện OpenCV được phát hành với giấy phép BDS. Do đó các dịch vụ nó cung cấp là hoàn toàn miễn phí và được hạn chế tối đa các rào cản thông thường. Cụ thể, bạn được phép sử dụng phần mềm này cho cả hoạt động thương mại lẫn phi thương mại. OpenCV sở hữu giao diện thiên thiện với mọi loại ngôn ngữ lập trình, ví dụ như C++, C, Python hay Java… Ngoài ra, nó cũng dễ dàng tương thích với các hệ điều hành khác nhau, bao gồm từ Windows, Linux, Mac OS, iOS cho đến cả Android.

Theo tính năng và ứng dụng của OpenCV, có thể chia thư viện này thánh các nhóm tính năng và module tương ứng như sau:

* Xử lý và hiển thị Hình ảnh/ Video/ I/O (core, imgproc, highgui)
* Phát hiện các vật thể (objdetect, features2d, nonfree)
* Geometry-based monocular hoặc stereo computer vision (calib3d, stitching, videostab)
* Computational photography (photo, video, superres)
* Machine learning & clustering (ml, flann)

CUDA acceleration (gpu)

### 2.1.2 Tensorflow

Tensorflow là một thư viện có mã nguồn mở, được dùng để tính toán machine learning với quy mô lớn. TensorFlow kết hợp một loạt các mô hình và thuật toán machine learning cùng deep learning, từ đó làm cho chúng trở nên hữu ích bằng những phép toán. TensorFlow sử dụng Python để cung cấp một API front-end thuận tiện cho việc xây dựng các ứng dụng với framework, đồng thời thực thi các ứng dụng đó bằng ngôn ngữ C++ để đạt hiệu suất cao hơn.

Kiến trúc TensorFlow hoạt động được chia thành 3 phần:

* Tiền xử lý dữ liệu
* Dựng model
* Train và ước tính model

Cách thức hoạt động:

TensorFlow cho phép bạn xây dựng biểu đồ và cấu trúc luồng dữ liệu để mô tả cách dữ liệu di chuyển qua biểu đồ hoặc di chuyển qua một seri mà các node đang xử lý. Mỗi một node trong đồ thị đại diện cho một operation toán học, có thể gọi đây là mảng dữ liệu đa chiều hay tensor.

TensorFlow sẽ cung cấp tất cả thông tin cho lập trình viên bằng ngôn ngữ lập trình Python. Python có nhiệm vụ điều phối các luồng công việc và kết nối chúng lại với nhau. Các node và tensor có trong TensorFlow cũng là những đối tượng của Python.

### 2.1.3 một số thư viện khác và công cụ sử dụng

- Cuda

CUDA là từ viết tắt của Compute Unified Device Architecture - Kiến trúc hợp nhất tính toán của các thiết bị điện tử được phát triển độc quyền bởi hãng công nghệ NVIDIA. Còn về CUDA core hay nhân CUDA thì chúng ta có thể hiểu đây là một nhân xử lý trong GPU của card đồ họa - đơn vị chịu trách nhiệm tính toán các thông tin, dữ liệu đồ họa cần được kết xuất. Nhân CUDA tích hợp trong GPU của card đồ họa rời, càng nhiều nhân CUDA thì khả năng tính toán đồng thời nhiều thông tin càng nhanh và chính xác

* Tkinter

Tkinter là một gói trong Python có chứa module Tk hỗ trợ cho việc lập trình GUI. Tk ban đầu được viết cho ngôn ngữ Tcl. Sau đó Tkinter được viết ra để sử dụng Tk bằng trình thông dịch Tcl trên nền Python. Ngoài Tkinter ra còn có một số công cụ khác giúp tạo một ứng dụng GUI viết bằng Python như wxPython, PyQt, và PyGTK

* Keras

Keras là một open source cho Neural Network được viết bởi ngôn ngữ Python. Nó là một library được phát triển vào năm 205 bởi Francois Chollet, là một kỹ sư nghiên cứu Deep Learning. Keras có thể sử dụng chung với các thư viện nổi tiếng như Tensorflow, CNTK, Theano.

* Qt Designer

Qt Designer là một công cụ để nhanh chóng xây dựng giao diện người dùng đồ họa với các widget từ khung Qt GUI . Nó cung cấp cho bạn một giao diện kéo và thả đơn giản để bố trí các thành phần như nút, trường văn bản, hộp tổ hợp và hơn thế nữa. Qt Designer tạo ra .ui các tệp. Đây là một định dạng dựa trên XML đặc biệt để lưu trữ các widget của bạn dưới dạng cây. Bạn có thể tải các tệp này trong thời gian chạy hoặc dịch chúng sang ngôn ngữ lập trình như C ++ hoặc  Python.

* Qt Designer và Python
* Nhiều người thích sử dụng Qt Designer cùng với Python vì nó là một ngôn ngữ động có lợi cho việc tạo mẫu nhanh.
* Cách dễ nhất để kết hợp Qt Designer và Python là thông qua liên kết PyQt . Để cài đặt PyQt, chỉ cần nhập nội dung sau vào dòng lệnh:
* python3 -m venv venv
* source venv/bin/activate # or "call venv\Scripts\activate.bat" on Windows

python3 -m pip install PyQt5

* PyQt5

Qt là một Application framework đa nền tảng viết trên ngôn ngữ C++ , được dùng để phát triển các ứng dụng trên desktop, hệ thống nhúng và mobile. Hỗ trợ cho các platform bao gồm : Linux, OS X, Windows, VxWorks, QNX, Android, iOS, BlackBerry, Sailfish OS và một số platform khác. PyQt là  Python interface của Qt, kết hợp của ngôn ngữ lập trình  Python và thư viện Qt, là một thư viện bao gồm các thành phần giao diện điều khiển (widgets , graphical control elements).

## 2.2 Xây Dựng Hệ Thống Nhận Diện Cảm Xúc Qua Khuôn Mặt

Đề tài này, nhóm sẽ áp dụng máy học, openCV và Python để tạo ra một ứng dụng tự động nhận diện cảm xúc và chọn ra emoji tương ứng từ hai nguồn đầu vào là live video và ảnh chọn từ máy tính.

Bài toán nhận diện có thể chia thành 2 giai đoạn chính, bao gồm:

* Vẽ bounding box và sử dụng webcam để đưa khuôn mặt và đúng khu vực cần nhận diện
* Sau khi đưa khuôn mặt vào đúng khu vực cần nhận diện thì thực hiện nhận diện cảm xúc dựa vào các features bắt được.Cuối cùng, hiển thị tên của cảm xúc nhận diện lên bounding box

### 2.2.1 Kiến Trúc Tổng Quan Của Hệ Thống

- Sơ đồ khối

Hệ thống được chia thành thành 2 giai đoạn chính: Huấn luyện và nhận dạng

Huấn luyện gồm các quá trình: Lấy ảnh đầu vào, phân tích và xử lý ảnh đầu vào, tạo cơ sở dữ liệu.



Hình 3.2.1.a.1: Huấn luyện

Nhận diện gồm các quá trình: Lấy ảnh đầu vào, xử lý đầu vào, so sánh với cơ sở dữ liệu đã tạo ra trong giai đoạn huấn luyện, đưa ra kết quả đặc trưng một cách tường minh, nó sẽ thực hiện đi kèm với phương pháp phân loại.



Hình 3.2.1.a.2: Nhận dạng

## 2.3 Phân tích code

### 2.3.1 Final.py

* + - * **Các thư viện và gói cần thiết:**
* **OpenCV (**cv2**)**: Dùng để xử lý ảnh và video.
* **NumPy (**np**)**: Dùng để thao tác với mảng, đặc biệt là thay đổi kích thước và chuẩn bị dữ liệu đầu vào cho mô hình.
* **Keras**: Cung cấp mô hình học sâu (Deep Learning) cho nhận diện cảm xúc, sử dụng mạng nơ-ron tích chập (CNN).
* **PyQt5**: Dùng để xây dựng giao diện đồ họa (GUI), cho phép người dùng tương tác với ứng dụng, tải ảnh hoặc sử dụng camera.
* haarcascade\_frontalface\_default.xml: Là mô hình Haar Cascade được huấn luyện trước dùng để phát hiện khuôn mặt.
  + - * **Kiến trúc mô hình (**emotion\_model**):**

Mô hình là một mạng nơ-ron tích chập (CNN) với các lớp để trích xuất đặc trưng không gian từ ảnh khuôn mặt.

* + **Các lớp Conv2D**: Dùng để học các đặc trưng của khuôn mặt (ví dụ: kết cấu, hình dáng).
  + **Các lớp MaxPooling2D**: Dùng để giảm độ phân giải của các đặc trưng.
  + **Các lớp Dropout**: Dùng để tránh hiện tượng overfitting trong quá trình huấn luyện.
  + **Các lớp Flatten và Dense**: Các lớp này chuyển các đặc trưng 2D thành một vector 1D và cho phép mô hình dự đoán cảm xúc.
  + Mô hình đưa ra 7 loại cảm xúc, được biểu diễn qua hàm kích hoạt softmax ở lớp cuối.
    - * **Từ điển cảm xúc (**emotion\_dict **và** emoji\_dist**):**
* emotion\_dict: Chuyển đổi đầu ra của mô hình (từ 0 đến 6) thành tên cảm xúc tương ứng.
* emoji\_dist: Chuyển đổi đầu ra của mô hình thành đường dẫn đến hình ảnh emoji tương ứng.
  + - * **Thiết kế giao diện người dùng (GUI) với PyQt5:**
* Giao diện có nhiều widget như labels (QLabel), buttons (QPushButton), và graphics views (QGraphicsView).
* fromVid: Nút bấm để bắt đầu nhận diện cảm xúc từ video trực tiếp.
* fromImg: Nút bấm để mở tệp ảnh và nhận diện cảm xúc.
* **Labels**: Hiển thị cảm xúc được nhận diện và emoji tương ứng.
  + - * **Nhận diện cảm xúc từ video trực tiếp (**fVid**):**
* Hàm này dùng để lấy video trực tiếp từ webcam (cv2.VideoCapture(0)).
* Sử dụng mô hình Haar Cascade để phát hiện khuôn mặt trong từng khung hình của video.
* Sau khi phát hiện khuôn mặt, ảnh khuôn mặt sẽ được cắt và thay đổi kích thước thành 48x48 pixel, đây là kích thước đầu vào cho mô hình nhận diện cảm xúc.
* Mô hình sẽ dự đoán cảm xúc từ ảnh khuôn mặt đã cắt, và emoji tương ứng sẽ được hiển thị trên giao diện.
  + - * **Nhận diện cảm xúc từ ảnh (**fIMG**):**
* Hàm này cho phép người dùng chọn một tệp ảnh từ máy tính qua hộp thoại tệp (QFileDialog.getOpenFileName).
* Ảnh được xử lý tương tự như video trực tiếp, phát hiện khuôn mặt và đưa qua mô hình nhận diện cảm xúc.
* Cảm xúc và emoji tương ứng sẽ được hiển thị trên giao diện.

### 2.3.2 Train.py

* + - * **Thư viện Import**
* numpy: Thư viện để xử lý các mảng số học.
* cv2: Thư viện OpenCV để xử lý ảnh và video.
* Các thư viện Keras (bình luận) đã được sử dụng cho việc xây dựng và huấn luyện mô hình học sâu (CNN) nhận diện cảm xúc. Tuy nhiên, hiện tại chúng bị bỏ qua vì chỉ có đoạn mã nhận diện khuôn mặt và xử lý video đang hoạt động.
  + - * **Mô hình Nhận diện Cảm xúc (Bị chú thích)**
* Một mô hình CNN (Convolutional Neural Network) dùng để nhận diện cảm xúc của con người từ khuôn mặt, bao gồm:
  + **Conv2D**: Các lớp tích chập để trích xuất đặc trưng.
  + **MaxPooling2D**: Lớp pooling giảm kích thước đặc trưng.
  + **Dropout**: Giảm overfitting bằng cách ngắt kết nối một số nơ-ron.
  + **Flatten**: Biến đổi đầu ra 2D thành 1D.
  + **Dense**: Các lớp fully connected với kích thước lớn.
  + **Softmax**: Dùng cho phân loại đa lớp (7 cảm xúc).
  + Mô hình này được huấn luyện để phân loại 7 cảm xúc: "Angry", "Disgusted", "Fearful", "Happy", "Neutral", "Sad", "Surprised".
    - * **Video Feed và Khuôn mặt**
* Mở camera từ webcam bằng cv2.VideoCapture(0).
* Sử dụng Cascade Classifier của OpenCV để phát hiện khuôn mặt trong video.
  + Dùng file Haar Cascade haarcascade\_frontalface\_default.xml để phát hiện khuôn mặt trong ảnh xám (cv2.cvtColor(frame, cv2.COLOR\_BGR2GRAY)).
* Khi phát hiện khuôn mặt, vẽ một hình chữ nhật bao quanh khuôn mặt (cv2.rectangle).
* Cắt và resize vùng khuôn mặt được phát hiện về kích thước 48x48, sau đó chuẩn bị dữ liệu đầu vào cho mô hình (dùng np.expand\_dims để thêm chiều cho dữ liệu).
  + - * **Hiển thị Video**
* cv2.imshow('Video', cv2.resize(frame,(1200,860),interpolation = cv2.INTER\_CUBIC)): Hiển thị video với khung hình đã được thay đổi kích thước.
* cv2.waitKey(1) & 0xFF == ord('q'): Dừng video khi nhấn phím 'q'.

### 2.3.3 haarcascade\_frontalface\_default

Tệp haarcascade\_frontalface\_default.xml là một bộ phân loại Haar Cascade được sử dụng trong thư viện OpenCV để phát hiện khuôn mặt trong ảnh hoặc video. Haar Cascade là một kỹ thuật nhận diện đối tượng dựa trên các đặc trưng hình học gọi là đặc trưng Haar, giúp phân loại các đối tượng như khuôn mặt người. Tệp XML này chứa mô hình học máy đã được huấn luyện với hàng triệu ví dụ về khuôn mặt và không phải khuôn mặt, cho phép nhận diện khuôn mặt trong ảnh với độ chính xác khá cao. Để sử dụng tệp này trong OpenCV, ta tải bộ phân loại Haar Cascade qua hàm CascadeClassifier và sử dụng phương thức detectMultiScale để phát hiện khuôn mặt. Tệp này thường có sẵn trong thư mục cài đặt của OpenCV và có thể được truy cập thông qua cv2.data.haarcascades. Tuy nhiên, vì kỹ thuật Haar Cascade có thể không đạt được độ chính xác cao trong điều kiện ánh sáng yếu hoặc với khuôn mặt có góc nghiêng, nên các mô hình nhận diện khuôn mặt hiện đại như DNN (Deep Neural Networks) hoặc MTCNN thường được ưa chuộng hơn trong các ứng dụng yêu cầu độ chính xác cao hơn.

### 2.3.4 Train.ipynb

Xây dựng và huấn luyện mô hình nhận diện cảm xúc từ khuôn mặt sử dụng Keras và OpenCV.

* + - * **Tải Dữ Liệu từ Kaggle**

Sử dụng Kaggle API để tải bộ dữ liệu FER-2013, chứa các hình ảnh khuôn mặt và các nhãn cảm xúc liên quan. Bộ dữ liệu này có thể được sử dụng để huấn luyện một mô hình nhận diện cảm xúc từ khuôn mặt. Các bước thực hiện:

* Cài đặt kaggle và upload tệp kaggle.json chứa API key để truy cập Kaggle.
* Tạo thư mục .kaggle và sao chép tệp kaggle.json vào đó.
* Tải bộ dữ liệu FER-2013 từ Kaggle bằng lệnh !kaggle datasets download -d msambare/fer2013.
* Giải nén tệp dữ liệu fer2013.zip.
  + - * **Nhập Các Thư Viện Cần Thiết**

Các thư viện cần thiết bao gồm:

* numpy: Xử lý mảng và phép toán số học.
* cv2: Thư viện OpenCV để xử lý ảnh.
* **Keras**: Thư viện học sâu để xây dựng và huấn luyện mô hình CNN.
* ImageDataGenerator: Dùng để chuẩn hóa và xử lý dữ liệu hình ảnh trong quá trình huấn luyện.
  + - * **Chuẩn Bị Bộ Dữ Liệu Train và Test**
* ImageDataGenerator: Được sử dụng để chuẩn hóa lại giá trị pixel trong các bức ảnh (giảm giá trị của mỗi pixel từ 0-255 về khoảng 0-1).
* train\_generator và validation\_generator: Được sử dụng để tải dữ liệu từ thư mục train và test, tự động phân chia theo lớp cảm xúc và thay đổi kích thước hình ảnh về 48x48 pixel.
* Tham số color\_mode="grayscale" chỉ định rằng ảnh sẽ được xử lý dưới dạng ảnh xám (grayscale).
  + - * **Xây Dựng Mạng Tính Chập (CNN)**

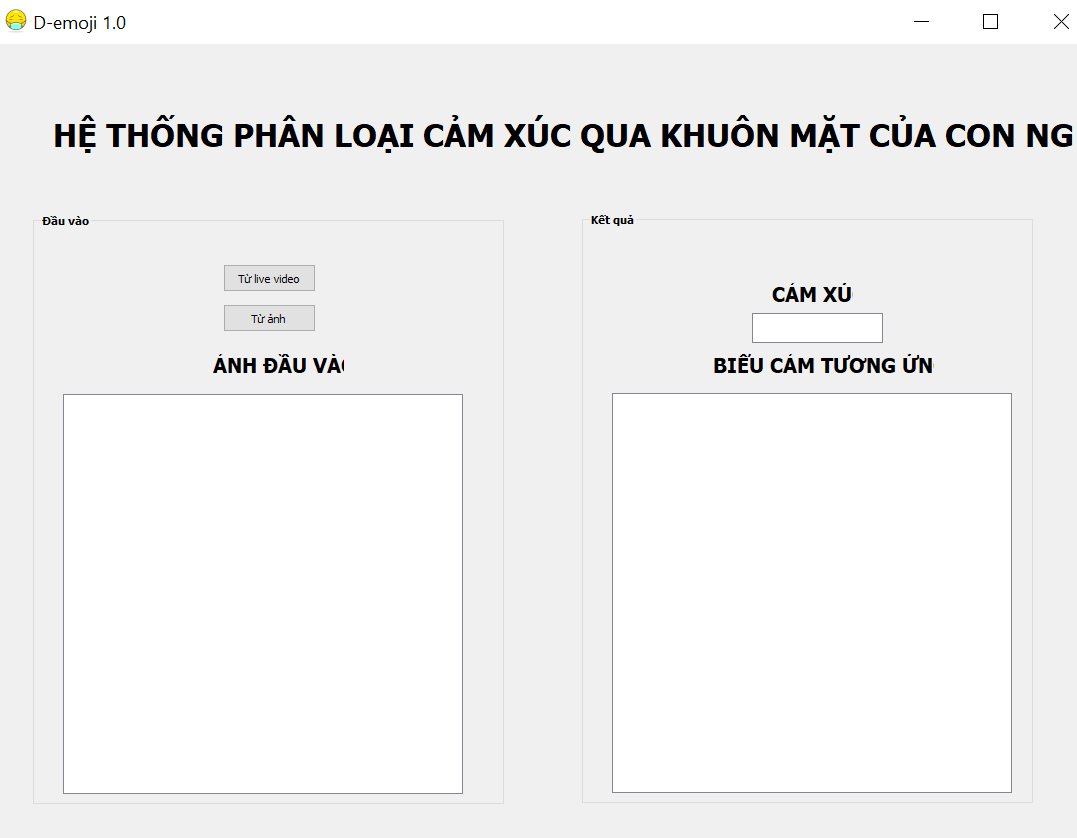
Mạng CNN được xây dựng với các lớp sau:

* **Conv2D (32, 64, 128 filters)**: Các lớp tích chập (Convolutional Layer) để trích xuất các đặc trưng từ ảnh. Mỗi lớp có các bộ lọc (filters) khác nhau với kích thước 3x3.
* **MaxPooling2D**: Các lớp Pooling giúp giảm kích thước của đặc trưng, làm giảm số lượng tính toán và tránh overfitting.
* **Dropout**: Được sử dụng để ngắt kết nối ngẫu nhiên các nơ-ron trong quá trình huấn luyện để giảm overfitting.
* **Flatten**: Chuyển đổi đầu ra từ các lớp tích chập thành một vector 1D, để có thể sử dụng trong các lớp fully connected.
* Dense: Các lớp fully connected với kích thước 1024 nơ-ron, giúp mô hình học các đặc trưng tổng quát. Lớp cuối cùng có 7 nơ-ron tương ứng với 7 lớp cảm xúc khác nhau.
* Activation softmax: Dùng để phân loại theo 7 cảm xúc, với mỗi nơ-ron biểu thị xác suất của một cảm xúc cụ thể.
  + - * **Huấn Luyện Mô Hình**
* loss='categorical\_crossentropy': Sử dụng hàm mất mát categorical\_crossentropy cho bài toán phân loại đa lớp.
* optimizer=Adam(lr=0.0001, decay=1e-6): Sử dụng bộ tối ưu Adam với tốc độ học ban đầu là 0.0001 và giảm dần qua các epoch.
* fit\_generator: Tiến hành huấn luyện mô hình với dữ liệu từ train\_generator và đánh giá với validation\_generator. Mô hình sẽ chạy 50 epoch và mỗi epoch sẽ xử lý 28709/64 bước.
  + - * **Lưu Trọng Số Mô Hình**
* emotion\_model.save\_weights('model.h5'): Sau khi huấn luyện xong, mô hình sẽ lưu trọng số vào tệp model.h5 để có thể tải lại và sử dụng sau này.

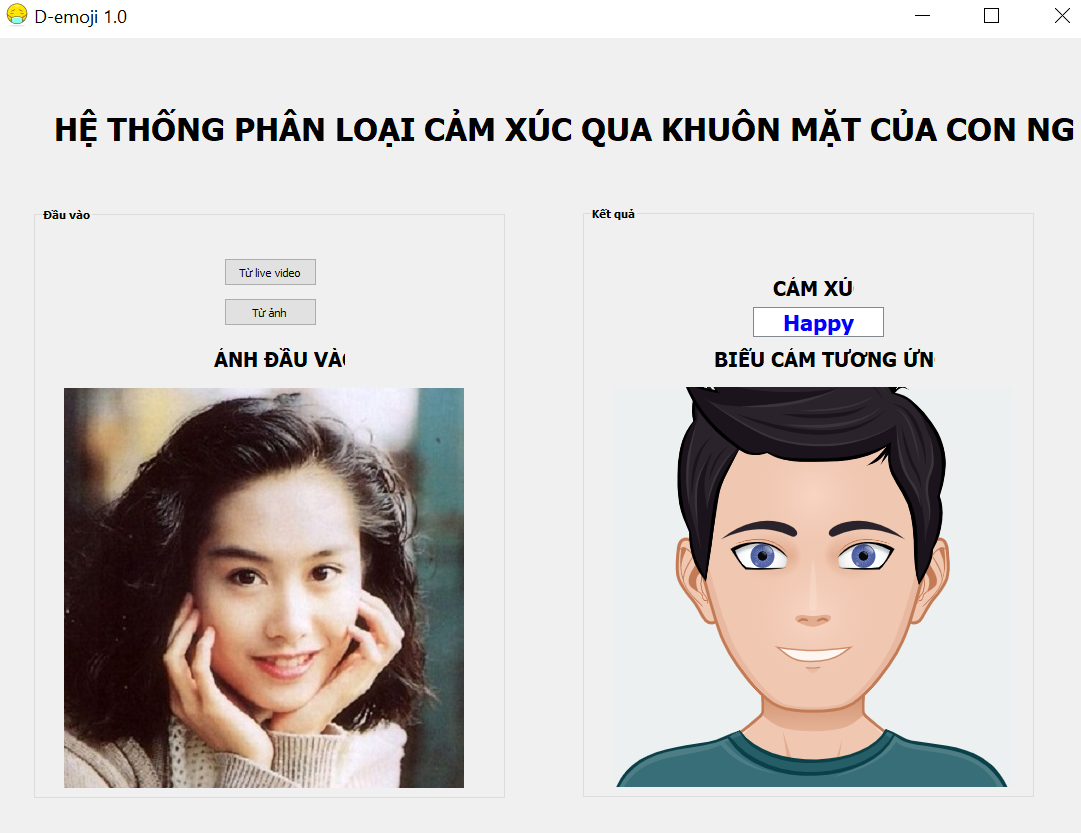
Cung cấp một giải pháp đầy đủ từ việc tải dữ liệu, chuẩn bị dữ liệu, xây dựng mô hình học sâu, huấn luyện và lưu mô hình cho bài toán nhận diện cảm xúc từ khuôn mặt, có thể dễ dàng xây dựng mô hình nhận diện cảm xúc và cải thiện hiệu quả với các kỹ thuật tối ưu khác.

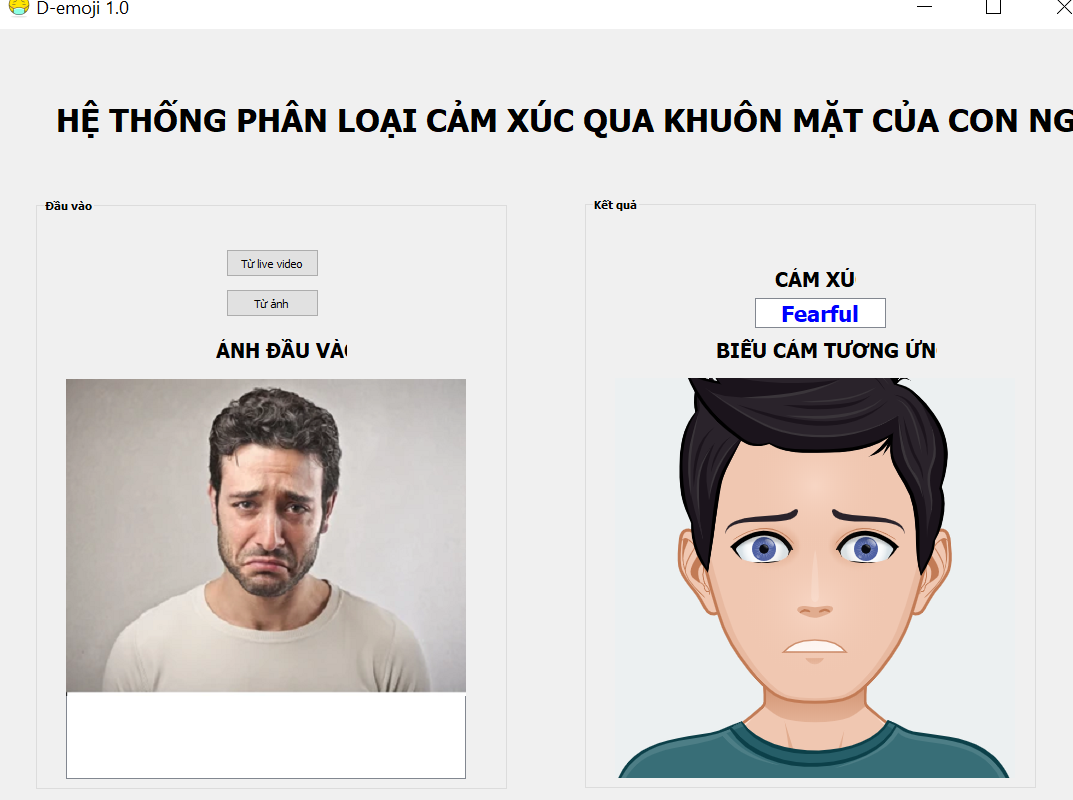
# CHƯƠNG 3 : THỰC NGHIỆM CHƯƠNG TRÌNH

## 3.1 Trang chủ

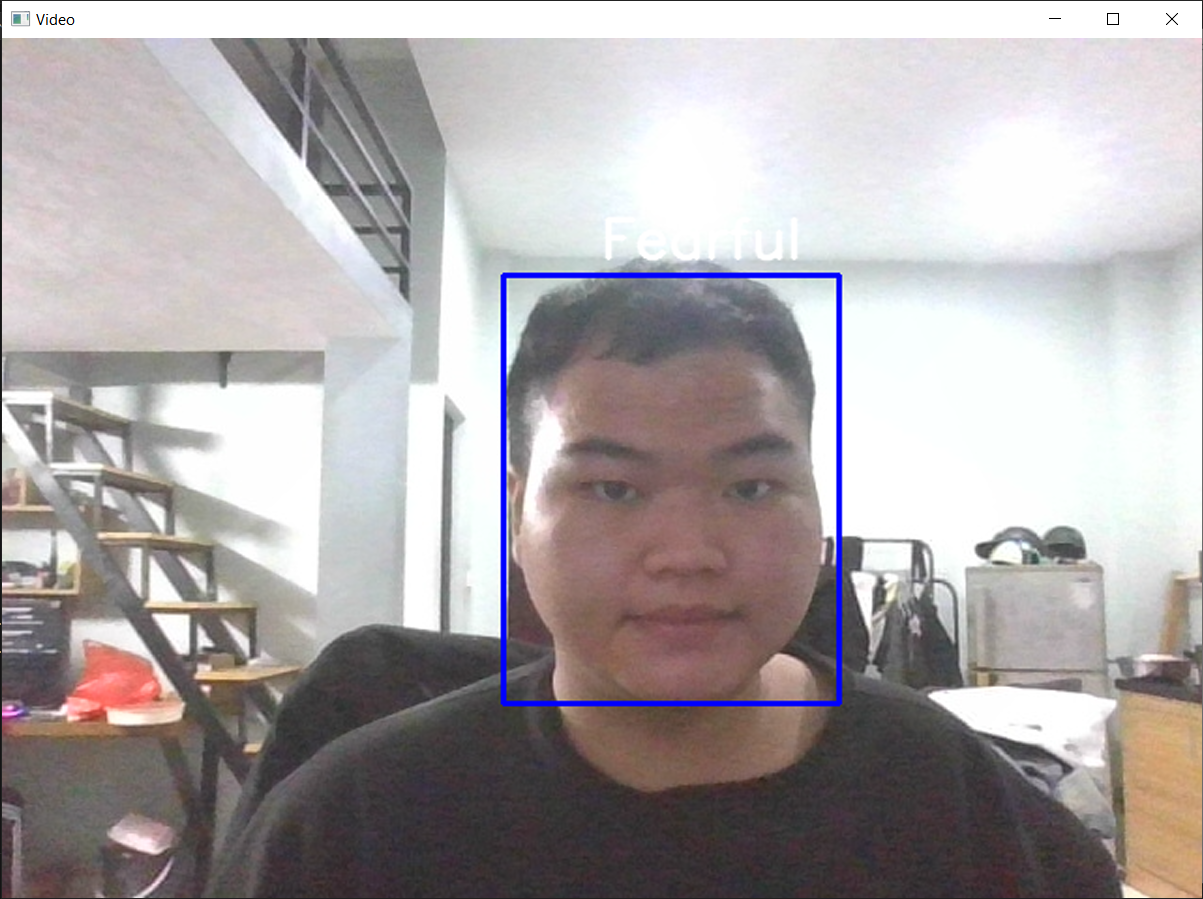


## 3.2 Nhận diện cảm xúc từ ảnh





## 3.3 Nhận diện cảm xúc qua video



# KẾT LUẬN

-   Ưu điểm: Nhận diện tốt những cảm xúc dễ nhận thấy như tức giận, vui vẻ hay ngạc nhiên, hiển thị kết quả nhanh, độ trễ cực thấp

-   Khuyết điểm: Những cảm xúc như sợ hãi, khinh bỉ,.. thì nhận diện chưa được tốt bởi vì trong thực tế những cảm xúc này mỗi người biểu cảm sẽ có phần khác biệt, nên khi sử dụng trong thực tế, tính ứng dụng chưa cao, cần phải cải tiến thêm. Ngoài ra với những ảnh có góc chụp nghiêng hay có phụ kiện che đáng kể khuôn mặt thì ứng dụng vẫn chưa xử lý tốt được, sẽ cải tiến sau.

# TÀI LIỆU THAM KHẢO

* <https://www.techsource-asia.com/resources/convolutional-neural-network>
* [EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks](https://arxiv.org/abs/1905.11946)
* [Facial expression and attributes recognition based on multi-task learning of lightweight neural networks](https://arxiv.org/pdf/2103.17107.pdf)
* [ResNeSt: Split-Attention Networks](https://arxiv.org/abs/2004.08955)
* [Context-Aware Emotion Recognition Networks](https://arxiv.org/abs/1908.05913)