

BTLN - Nhóm 6 - Computer Honda Vision

- Trí thông minh là gì?

Trí thông minh có thể được định nghĩa là khả năng của con người hoặc máy móc trong việc học hỏi, hiểu và áp dụng kiến thức, giải quyết vấn đề, và thích ứng với những tình huống mới.

Trong bối cảnh của AI, trí thông minh thường liên quan đến việc xử lý thông tin, phân tích dữ liệu, và đưa ra quyết định dựa trên dữ liệu đó.

- Giải thích "Intelligence is the ability to adapt to change"

Câu này nói rằng bản chất của trí thông minh không chỉ là học hỏi hoặc giải quyết vấn đề, mà còn là khả năng thích ứng với những thay đổi. Điều này đặc biệt quan trọng trong thế giới ngày càng phức tạp và liên tục biến đổi, nơi mà khả năng thích ứng và phản ứng nhanh chóng với những tình huống mới là yếu tố then chốt để thành công.

- Sự khác nhau giữa "thông minh" và "thông tuệ" (thầy dùng từ "wisdom")

Thông minh (Intelligence): Thường liên quan đến khả năng học hỏi, giải quyết vấn đề, và suy nghĩ logic. Đây là việc áp dụng tri thức và kỹ năng để đối mặt với tình huống cụ thể.

Thông tuệ (Wisdom): Được xem như một bước tiến xa hơn, liên quan đến sự hiểu biết sâu sắc, đánh giá đúng đắn, và khả năng xem xét vấn đề trong một bối cảnh rộng lớn hơn. Thông tuệ bao gồm việc sử dụng kiến thức một cách khôn ngoan, tích lũy kinh nghiệm sống, và hiểu rõ về đạo đức và giá trị.

- Giải thích sơ đồ mạng neural

Mạng neural là một mô hình trong máy học, lấy cảm hứng từ cách thức hoạt động của não người. Nó bao gồm các đơn vị (neurons) được sắp xếp thành các lớp. Mỗi neuron nhận đầu vào, xử lý thông tin, và truyền tín hiệu đến các neuron khác. Có ba loại lớp chính:

1. Lớp đầu vào (Input Layer): Nhận dữ liệu đầu vào.
2. Lớp ẩn (Hidden Layers): Xử lý dữ liệu thông qua các hàm toán học và truyền dữ liệu giữa lớp đầu vào và lớp đầu ra.
3. Lớp đầu ra (Output Layer): Đưa ra kết quả cuối cùng dựa trên thông tin từ các lớp ẩn.

- AI giúp giải quyết gì cho semantic gap?

Trong xử lý ảnh và video số, "semantic gap" là khoảng cách giữa thông tin mà máy móc có thể trích xuất từ dữ liệu (như ảnh, video) và ý nghĩa mà con người gán cho dữ liệu đó. AI, đặc biệt là học sâu và mạng neural, giúp giảm thiểu khoảng cách này bằng cách học cách nhận diện và hiểu các mẫu dữ liệu phức tạp, giúp máy móc có thể "hiểu" và phản ứng với dữ liệu một cách gần giống con người hơn.

- Tìm một ví dụ gồm: Tên ứng dụng, input, output (5 cấp bậc)

Tên ứng dụng: Google Photos

Input: Ảnh và video do người dùng tải lên.

Output:

1. Cấp 1 (Cơ bản): Lưu trữ ảnh/video.
2. Cấp 2: Sắp xếp và phân loại ảnh/video dựa trên ngày tháng.
3. Cấp 3: Nhận diện khuôn mặt và đối tượng trong ảnh.
4. Cấp 4: Gợi ý chỉnh sửa và cải thiện chất lượng ảnh.
5. Cấp 5: Tạo album tự động dựa trên sự kiện, địa điểm, hoặc những người xuất hiện trong ảnh.

Tại sao trên giấy lại dùng mực riêng (CMYK) mà không là RGB

- Phương pháp Tạo Màu:
 - RGB: Dùng ánh sáng để tạo màu (đèn, màn hình). Khi kết hợp, các màu cơ bản tạo ra ánh sáng trắng.
 - CMYK: Dùng hỗn hợp mực trên giấy. Khi kết hợp, các màu cơ bản tạo ra màu đen hoặc xám.
- Hiệu ứng Hấp thụ và Phản xạ Ánh sáng:
 - RGB: Ánh sáng phản chiếu từ màn hình, sử dụng ánh sáng để tạo màu.
 - CMYK: Giấy hấp thụ ánh sáng. Các màu mực hấp thụ các bước sóng khác nhau, phản xạ lại màu sắc đặc trưng.
- Ứng dụng:
 - RGB: Hiệu quả cho các thiết bị điện tử.
 - CMYK: Hiệu quả cho in ấn, đảm bảo chất lượng màu sắc chính xác trên giấy.

Trình bày cảm xúc về 3 ngữ cảnh khi nhìn thấy màu.

- **Màu Xanh Dương** (Cảm giác Bình Yên): Trên bãi biển, màu xanh của nước biển và bầu trời mang lại cảm giác yên bình, thư giãn.
- **Màu Đỏ** (Cảm giác Kích Động): Trong một buổi hòa nhạc, ánh sáng đỏ rực rỡ tạo cảm giác hứng khởi, đam mê.
- **Màu Xanh Lá** (Cảm giác Tươi mới): Trong một khu vườn, màu xanh của cỏ và cây cối mang lại cảm giác tươi mới, sống động.

Làm sao để có đủ thông tin khi chuyển từ ảnh 2D sang 3D

- **Nguồn Dữ liệu Phụ Trợ:** Sử dụng các nguồn thông tin bổ sung như ảnh từ nhiều góc độ khác nhau.
- **Công nghệ Quét 3D:** Sử dụng thiết bị quét 3D để thu thập dữ liệu chiều sâu và kích thước thực tế.
- **Phần mềm Mô hình hóa 3D:** Sử dụng phần mềm chuyên nghiệp để tái tạo mô hình 3D từ ảnh 2D, thêm chi tiết và kết cấu.
- **Thuật toán Xử lý Ảnh:** Áp dụng thuật toán nhận diện đặc trưng và tái tạo 3D dựa trên các đặc điểm trong ảnh 2D.

Khi nhận dạng chữ in, nên dùng ảnh dưới dạng Grayscale hay Binary

- **Grayscale:**

Ưu điểm: Lưu giữ nhiều thông tin về cường độ sáng, giúp phân biệt tốt hơn các chi tiết nhỏ.

Nhược điểm: Có thể gây nhiễu nếu chênh lệch độ sáng không rõ ràng.

- **Binary (Đen/Trắng):**

Ưu điểm: Rõ ràng, dễ phân biệt chữ và nền, tốt cho nhận dạng văn bản đơn giản.

Nhược điểm: Mất mát thông tin chi tiết, không phù hợp với văn bản có kết cấu phức tạp hoặc nền nhiễu.

Tìm quan hệ giữa Cyan - Red; Yellow - Blue; Magenta - Green

- **Cyan và Red:** Là màu bổ sung trong hệ CMYK. Khi kết hợp, chúng tạo ra màu xám hoặc đen.
- **Yellow và Blue:** Trong hệ CMYK, yellow kết hợp với blue tạo ra màu xanh lá. Trong hệ màu RGB, blue và yellow (được tạo từ red và green) tạo ra màu trắng.
- **Magenta và Green:** Là màu bổ sung. Trong CMYK, magenta kết hợp với green (kết quả của yellow và cyan) tạo ra màu đen hoặc xám.

Tìm công thức chuyển đổi RGB → HSV và ngược lại

RGB → HSV:

- Tính giá trị Max và Min của R, G, B.
- Hue (H): Dựa vào màu nào trong RGB có giá trị cao nhất.
- Saturation (S): $S = (\text{Max} - \text{Min}) / \text{Max}$.
- Value (V): $V = \text{Max}$.

HSV → RGB:

- Chia Hue thành các khu vực màu.
- Tính R, G, B dựa vào Saturation và Value.
- Điều chỉnh lại R, G, B dựa trên Hue.

- Image processing giúp được gì
 - Cải thiện chất lượng hình ảnh: Làm rõ, tăng cường độ sáng, độ tương phản.
 - Phát hiện và phân tích đối tượng: Nhận dạng khuôn mặt, phát hiện biển số xe.
 - Chỉnh sửa hình ảnh: Cắt, xoay, thay đổi kích thước.
 - Phân loại và phân tích dữ liệu ảnh: Phân loại ảnh dựa trên đặc điểm, phân tích cảm xúc.
 - Y học và khoa học: Phân tích hình ảnh y tế, nghiên cứu vũ trụ.
- Làm sao để chuyển đổi ảnh từ mức xám [f1, f2] sang [g1, g2]
Sử dụng phép biến đổi tuyến tính: $g = g1 + \frac{(f - f1) * (g2 - g1)}{f2 - f1}$ cho mỗi pixel f trong ảnh.
- Trên hàm exponential mapping, ảnh biến thiên vùng sáng như thế nào
Hàm exponential thường tăng cường độ sáng cho các vùng tối hơn, làm cho chúng trở nên rõ ràng hơn, trong khi vùng sáng có thể trở nên quá sáng.
- Vùng tối sẽ thay đổi độ sáng ít hơn vùng sáng
Trong nhiều phương pháp xử lý ảnh, đặc biệt là những phương pháp tập trung vào việc cân bằng độ tương phản, vùng tối thường được cải thiện ít hơn so với vùng sáng để tránh làm mất đi chi tiết trong vùng tối.
- Vì sao không dùng được Linear / non-Linear để biến đổi cột trái sang phải (như sau)
Cần thông tin cụ thể hơn về "cột trái sang phải" mà bạn đang nói tới. Trong xử lý ảnh, các biến đổi tuyến tính và phi tuyến tính thường áp dụng theo cách khác nhau tùy thuộc vào bối cảnh và mục đích xử lý.
- Viết giải thuật cho Probability Density function
PDF trong xử lý ảnh thường được sử dụng để biểu diễn phân bố độ sáng hoặc màu sắc của ảnh.

Pseudocode:

```
function calculatePDF(image):
    histogram = calculateHistogram(image)
    totalPixels = sum(histogram)
    pdf = histogram / totalPixels
    return pdf
```

- Biến 1 tập ảnh từ 1000 ảnh sang 50000 ảnh
Sử dụng các kỹ thuật augmentation để tạo ra các biến thể của ảnh gốc: xoay, thay đổi kích thước, làm méo, thay đổi độ sáng, thêm nhiễu.
Kỹ thuật GANs (Generative Adversarial Networks) để sinh ra ảnh mới.
Nếu cần giữ nguyên nội dung: Tái sử dụng ảnh với các biến thể nhẹ như thay đổi màu sắc, cắt ngẫu nhiên, hoặc thay đổi góc nhìn
- Tìm phương pháp nội suy giá trị màu bằng cách lấy giá trị trung bình trọng số của hàng xóm
Cách 1: Xét lân cận gần nhất
Cách 2: Lấy trung bình cộng 4 pixel lân cận
Cách 3: Lấy trung bình có trọng số
- Viết giải thuật thực hiện cách "Trung bình có trọng số"
Giải thuật "Trung bình có trọng số" (Weighted Average) thường được sử dụng trong xử lý ảnh để làm mịn, nơi mà mỗi giá trị điểm ảnh mới được tính dựa trên trung bình có trọng số của các giá trị điểm ảnh lân cận.

Pseudocode:

```
for each pixel in the image:
    initialize a weighted sum variable
    initialize a total weight variable
    for each neighboring pixel:
        add the value of the neighbor pixel multiplied by its weight to the weighted sum
        add the weight to the total weight
    set the value of the current pixel to the weighted sum divided by the total weight
```

Trong đó, trọng số có thể phụ thuộc vào khoảng cách từ pixel hiện tại đến các pixel lân cận hoặc các yếu tố khác.

- Chứng minh tổng độ dị biệt của Median Filter là $n \times n$
Median Filter là một phương pháp làm mịn ảnh, nơi mà giá trị của mỗi pixel được thay thế bằng giá trị trung vị của các giá trị pixel trong một cửa sổ lân cận. Đối với một cửa sổ kích thước $n \times n$, tổng số pixel sẽ là n^2 . Để chứng minh tổng độ dị biệt (tức là số lượng các giá trị khác nhau có thể) của Median Filter là n^2 , chúng ta cần xem xét sự phân bố và số lượng giá trị khác nhau có thể trong cửa sổ lọc này. Tuy nhiên, một chứng minh cụ thể yêu cầu thông tin chi tiết hơn về cách định nghĩa "độ dị biệt" trong trường hợp này.
- So sánh 3 phương pháp sau và giải thuật
→ Input và Output của Nearest-Neighbor Interpolation:
Input: Một ảnh với độ phân giải thấp.

Output: Một ảnh với độ phân giải cao hơn, nơi mỗi pixel mới được gán giá trị của pixel gần nhất từ ảnh gốc.

→ Input và Output của Linear Interpolation:

Input: Một ảnh với độ phân giải thấp.

Output: Một ảnh với độ phân giải cao hơn, nơi mỗi pixel mới được tính bằng cách sử dụng một tổ hợp tuyến tính của các pixel lân cận.

- Input và Output của hai giải thuật Nearest-Neighbor Interpolation và Linear Interpolation
- Viết giải thuật cho phép biến đổi ảnh:

Nearest-Neighbor Interpolation

Mục đích: Tăng độ phân giải của ảnh mà không thay đổi nội dung. Mỗi pixel trong ảnh mới được gán giá trị của pixel gần nhất trong ảnh gốc.

Pseudocode:

```
function nearestNeighborInterpolation(inputImage, newWidth, newHeight):
```

```
    oldWidth, oldHeight = dimensions of inputImage
```

```
    outputImage = new image of size newWidth x newHeight
```

```
    for x from 0 to newWidth - 1:
```

```
        for y from 0 to newHeight - 1:
```

```
            // Tính tọa độ tương ứng trong ảnh gốc
```

```
            oldX = round(x * oldWidth / newWidth)
```

```
            oldY = round(y * oldHeight / newHeight)
```

```
            // Gán giá trị của pixel gần nhất
```

```
            outputImage[x, y] = inputImage[oldX, oldY]
```

```
    return outputImage
```

Linear Interpolation

Mục đích: Tăng độ phân giải của ảnh một cách mượt mà hơn. Sử dụng tổ hợp tuyến tính của các pixel lân cận để xác định giá trị cho mỗi pixel mới.

Pseudocode:

```
function linearInterpolation(inputImage, newWidth, newHeight):
```

```
    oldWidth, oldHeight = dimensions of inputImage
```

```
    outputImage = new image of size newWidth x newHeight
```

```
    for x from 0 to newWidth - 1:
```

```
        for y from 0 to newHeight - 1:
```

```
            // Tính tọa độ tương ứng và các trọng số cho interpolation
```

```
            oldX = x * oldWidth / newWidth
```

```
            oldY = y * oldHeight / newHeight
```

```

x1, y1 = floor(oldX), floor(oldY)
x2, y2 = ceil(oldX), ceil(oldY)

// Lấy các giá trị pixel
Q11 = inputImage[x1, y1]
Q12 = inputImage[x1, y2]
Q21 = inputImage[x2, y1]
Q22 = inputImage[x2, y2]

// Tính giá trị interpolated
outputImage[x, y] = Q11 * (x2 - oldX) * (y2 - oldY) +
                    Q21 * (oldX - x1) * (y2 - oldY) +
                    Q12 * (x2 - oldX) * (oldY - y1) +
                    Q22 * (oldX - x1) * (oldY - y1)

return outputImage

```

Trong Linear Interpolation, tỷ lệ của tổ hợp tuyến tính phụ thuộc vào khoảng cách tương đối giữa pixel đầu ra và bốn pixel đầu vào lân cận.

- Tại sao Laplace bất biến với phép xoay
Toán tử Laplace, được biểu diễn qua phương trình $\Delta f = \nabla^2 f$ là đạo hàm bậc hai của hàm số. Đặc điểm quan trọng của toán tử Laplace là nó bất biến với phép xoay. Điều này có nghĩa là không phụ thuộc vào hướng của hình ảnh, kết quả của phép toán Laplace luôn nhất quán. Điều này là do trong toán tử Laplace, sự đóng góp của các đạo hàm theo mọi hướng là đồng nhất và không phụ thuộc vào hệ tọa độ.
- Viết giải thuật phát hiện biên cạnh bằng Gradient Operator
Gradient Operator là một phương pháp cơ bản trong phát hiện biên cạnh. Nó hoạt động bằng cách tìm độ lớn và hướng của gradient tại mỗi điểm ảnh. Một ví dụ thông dụng là Sobel Operator.
Pseudocode:
for each pixel in the image:
 compute the horizontal gradient (Gx)
 compute the vertical gradient (Gy)

combine G_x and G_y to find the magnitude of the gradient
if the magnitude is above a threshold, mark it as an edge

- Viết giải thuật phát hiện biên cạnh bằng Laplace
Toán tử Laplace tập trung vào việc phát hiện các vùng có sự thay đổi đột ngột trong độ sáng, tức là các cạnh.

Pseudocode:

for each pixel in the image:
 apply the Laplace filter to the pixel
 if the result is above a threshold, mark it as an edge

- Viết giải thuật phát hiện biên cạnh bằng Laplace of Gaussian
Laplace of Gaussian kết hợp làm mịn Gaussian với toán tử Laplace. Đầu tiên, ảnh được làm mịn để giảm nhiễu, sau đó áp dụng toán tử Laplace để phát hiện cạnh.

Pseudocode:

apply Gaussian blur to the image
for each pixel in the image:
 apply the Laplace filter to the blurred image
 if the result is above a threshold, mark it as an edge

- Thực hiện so sánh các phương pháp
 - **Gradient Operator (Sobel, Prewitt, etc.):** Tốt trong việc phát hiện hướng và cường độ của cạnh. Nhạy với nhiễu.
 - **Laplace:** Phát hiện cạnh dựa trên độ lớn của đổi sáng. Không nhạy với hướng cạnh nhưng dễ bị ảnh hưởng bởi nhiễu.
 - **Laplace of Gaussian (LoG):** Kết hợp lợi ích của làm mịn Gaussian (giảm nhiễu) và phát hiện cạnh mạnh mẽ của Laplace. Có thể không nhạy bằng Gradient Operator ở cạnh nhỏ hoặc mờ.

Mỗi phương pháp có điểm mạnh và điểm yếu riêng, và lựa chọn phương pháp phụ thuộc vào bối cảnh cụ thể của ứng dụng xử lý ảnh.

- Tìm kiếm thêm các luật kiểm tra tính thuần nhất của vùng.
Khi nói đến việc kiểm tra tính thuần nhất của một vùng trong xử lý ảnh, có một số phương pháp và tiêu chí được sử dụng. Các luật hoặc tiêu chí này thường dựa trên các đặc điểm thống kê của vùng, như độ sáng trung bình, phân tán, texture, hoặc màu sắc. Dưới đây là một số phương pháp phổ biến:
 - **Phương Pháp Độ Sáng Trung Bình và Phương Sai:** Một vùng được coi là thuần nhất nếu độ sáng trung bình và phương sai của nó nằm trong một ngưỡng nhất định.

- **Phân Tích Texture:** Sử dụng các đặc trưng như co-occurrence matrix, gradient, hoặc các đặc trưng dựa trên wavelet để xác định tính đồng nhất của texture trong vùng.
 - **So Sánh Màu Sắc:** Áp dụng phân tích màu sắc để kiểm tra sự đồng nhất về màu sắc trong vùng. Các phương pháp như histogram matching hoặc phân tích thành phần chính (PCA) có thể được sử dụng.
 - **Phương Pháp Dựa Trên Hình Dạng:** Kiểm tra sự đồng nhất của hình dạng hoặc kích thước các cấu trúc trong vùng.
 - **Phân Tích Kết Nối:** Kiểm tra xem các pixel trong vùng có liên kết chặt chẽ với nhau hay không, thông qua các thuật toán như region growing hoặc connected components.
- Chứng minh giá trị trung bình của vùng tại bước $k+1$ bằng giá trị trung bình ở bước k + thêm (trong slide)
 Nếu ta có một quá trình lặp cập nhật giá trị trung bình của một vùng trong ảnh, thì giá trị trung bình ở bước **$k+1$** có thể được tính bằng cách lấy giá trị trung bình ở bước k và cộng thêm một giá trị cập nhật $\Delta\mu$. Công thức sẽ là $\mu(k+1) = \mu_k + \Delta\mu$, nơi μ_k là giá trị trung bình tại bước k , và $\Delta\mu$ là sự thay đổi được áp dụng.
 - Viết rõ ra thuật toán K-means
 - Bước 1: Khởi tạo các trọng tâm ban đầu:**
 Chọn k điểm ngẫu nhiên từ X làm các trọng tâm ban đầu.
 - Bước 2: Gán mỗi điểm dữ liệu vào cụm gần nhất:**
 Đối với mỗi điểm x_i trong X , tính khoảng cách từ x_i đến tất cả các trọng tâm. Gán x_i vào cụm của trọng tâm gần nhất.
 - Bước 3: Cập nhật trọng tâm:**
 Đối với mỗi cụm, tính trung bình của tất cả các điểm dữ liệu thuộc cụm đó. Cập nhật trọng tâm của cụm là giá trị trung bình.
 - Bước 4: Lặp lại Bước 2 và Bước 3:**
 Lặp lại Bước 2 và Bước 3 cho đến khi có sự hội tụ hoặc số lần lặp đã đạt giới hạn.
 - Kết quả:**
 Các trọng tâm và phân loại của mỗi điểm dữ liệu vào các cụm.
 - Thuật toán dừng:**
 Khi có sự hội tụ (trọng tâm không thay đổi nhiều) hoặc số lần lặp đã đạt giới hạn.
 - Ứng dụng của định lý Convolution
 Định lý Convolution thường được sử dụng trong xử lý ảnh để thực hiện các phép toán như làm trơn, làm nổi bật đặc trưng, phát hiện biên cạnh, và nhận diện đối tượng.
 - Giải thích ý nghĩa của công thức

$$(f * g)(t) = \int_{-\infty}^{\infty} f(\tau)g(t - \tau)d\tau$$

Ý nghĩa: Công thức này mô tả cách thông tin từ một vùng nhỏ của ảnh được kết hợp để tạo ra một pixel trong ảnh kết quả, giúp làm nổi bật hoặc làm mịn các đặc trưng.

- Viết giải thuật làm trơn ảnh trong miền tần số
B1: Sử dụng Fourier Transform để chuyển đổi ảnh vào miền tần số
B2: áp dụng một hàm truyền tần số thích hợp (ví dụ: Gaussian)
B3: chuyển đổi ngược để có ảnh đã được làm trơn.
- Biện luận D0 cho Butterworth HighPass Filter
Đối với bộ lọc Butterworth HighPass, D0 là tham số quyết định tần số cắt. Khi D0 lớn, bộ lọc bảo toàn các thành phần tần số cao hơn.
- Viết giải thuật phát hiện biên cạnh trong miền tần số
B1: Sử dụng Fourier Transform để chuyển đổi ảnh vào miền tần số
B2: Áp dụng bộ lọc tần số cao như Laplacian để tăng cường thành phần biên cạnh
B3: chuyển đổi Fourier ngược để quay trở lại ảnh không gian.
- Trực quan hóa thuật toán trên tập ảnh nhận diện mặt người.

Để trực quan hóa một thuật toán nhận diện khuôn mặt trên tập ảnh, bạn có thể thực hiện các bước sau đây. Giả sử chúng ta đang sử dụng một thuật toán nhận diện khuôn mặt đơn giản, dựa trên các tính năng như vị trí của mắt, mũi, miệng và hình dạng của khuôn mặt.

Bước 1: Chuẩn Bị Tập Ảnh

Chọn một tập hợp các ảnh chứa khuôn mặt người. Đảm bảo các ảnh này đa dạng về độ tuổi, giới tính, tông màu da, và các điều kiện ánh sáng.

Bước 2: Áp Dụng Thuật Toán Nhận Diện Khuôn Mặt

Sử dụng một thuật toán nhận diện khuôn mặt, chẳng hạn như Viola-Jones hoặc một mô hình học sâu như CNN (Convolutional Neural Network), để phát hiện khuôn mặt trong mỗi ảnh.

Bước 3: Đánh Dấu Các Điểm Đặc Trưng

Sau khi nhận diện khuôn mặt, đánh dấu các điểm đặc trưng trên khuôn mặt như vị trí của mắt, mũi, miệng, và đường viền của khuôn mặt. Có thể sử dụng các hình dạng như hình tròn, hình chữ nhật hoặc các đường thẳng để đánh dấu.

Bước 4: Trực Quan Hóa Kết Quả

Vẽ hoặc hiển thị các đánh dấu đó trực tiếp trên ảnh. Mỗi khuôn mặt phát hiện được có thể được bao quanh bằng một hình chữ nhật, và các điểm đặc trưng có thể được hiển thị bằng các dấu đặc biệt.

Bước 5: So Sánh và Phân Tích

Hiển thị tập hợp các ảnh trước và sau khi áp dụng thuật toán nhận diện khuôn mặt để dễ dàng so sánh và phân tích hiệu suất của thuật toán.

Bước 6: Đánh Giá và Tinh Chỉnh

Đánh giá hiệu suất của thuật toán dựa trên số lượng khuôn mặt được nhận diện chính xác và sai sót. Nếu cần, tinh chỉnh các tham số của thuật toán để cải thiện hiệu suất.

Lưu ý rằng, trong thực tế, việc nhận diện khuôn mặt có thể đối mặt với nhiều thách thức như biến đổi về góc nhìn, biểu cảm, và chất lượng ảnh. Đảm bảo rằng quy trình và thuật toán của bạn tôn trọng quyền riêng tư và tuân thủ các quy định về dữ liệu cá nhân.

- Nhận xét seminar các nhóm:

Nhóm 1: Research of superpixels and their applications in image segmentation

Nội dung rõ ràng, đầy đủ, thuyết trình bằng tiếng Anh tốt nhưng hơi dài dòng.

Nhóm 2: Research of saliency maps and their applications in detecting regions of interest in images.

Nội dung đầy đủ, rõ ràng, nhưng chọn hình ảnh chưa phù hợp với cách trình bày.

Nhóm 4: Research of Object tracking based on language-image models (Tracking anything)

Nội dung đầy đủ rõ ràng, áp dụng hiệu quả lý thuyết và ứng dụng. Tuy nhiên, cần rút ngắn để tránh lan man.

Nhóm 5: Research of Image Stitching and its applications for stitching images taken from different viewpoints.

Chưa nêu được lợi ích rõ ràng về chủ đề (ví dụ: camera motion estimation).

Nhóm 7: Research of Background Removing and its applications for natural-looking image integration.

Nhóm dùng học sâu (thay vì cách truyền thống). Nhưng chưa giải thích được quy trình học sâu (input, output, loss function, kiến trúc mạng, encoder, decoder), chưa rõ ràng về kết quả.

Nhóm 8: Research of Inpainting and its applications for data restoration (e.g., repairing documents with stains and damage).

Trình bày đầy đủ về ứng dụng của Inpainting trong phục hồi dữ liệu.

Nhóm 9: Research of image Restoration (Denoise - Deblur - Demotion)

Related work chưa thấy rõ được khác nhau giữa các mô hình. Tiêu chí mơ hồ. Chưa giải thích được hàm loss tổng cho mô hình BlurGAN.

Nhóm 10: Research of Super Image Resolution and its applications for enhancing image quality.

Cung cấp thông tin đầy đủ về tăng cường độ phân giải ảnh. Diễn đạt tự nhiên và dễ hiểu

Nhóm 11: Research of Super Video Resolution and its applications for improving video quality.

Nội dung đầy đủ, nhưng chưa hiểu rõ về mô hình mạng học sâu.

Nhóm 12: Research of Image Forensics and its applications for detecting fake images.

Cung cấp thông tin quan trọng về “detecting fake images”, nhưng cần chú trọng hơn vào việc phân biệt công trình nghiên cứu liên quan.

Nhóm 14: Research of Photometry (Photogrammetry) and its applications in agriculture.

Trình bày thông tin hữu ích về ứng dụng của photometry trong nông nghiệp. Cần mô tả rõ ràng hơn về lợi ích cụ thể

Nhóm 15: Research of Deep Fake.

Nội dung đầy đủ rõ ràng về Deep Fake. Trình bày mạch lạc.