

AI VIET NAM – COURSE 2022

Statistic - Exercise

Ngày 4 tháng 9 năm 2022

Note: Từ **problem 1** đến **problem 7** các bạn khuyến khích nên trình bày bằng latex, doc, ... rồi convert thành file pdf để nộp. Ngoài ra các bạn cũng có thể trình bày ra giấy và chụp ảnh, sau đó đặt tên ảnh và sắp xếp theo thứ tự câu hỏi rồi nén lại và gửi file.

Mean

- Data: $X = \{x_1, \dots, x_N\}$
- Mean: $\mu = \frac{1}{N} \sum_{i=1}^N x_i$

Problem1: Cho Data $X = \{2, 0, 2, 2, 7, 4, -2, 5, -1, -1\}$ Tìm mean μ của Data X đã cho?

Median

- Data: $X = \{x_1, \dots, x_N\}$
- Median:
 - Sort $X \longrightarrow S$ (tăng dần)
 - If N is odd: $m = \frac{S_{N+1}}{2}$, If N is even: $m = \frac{1}{2} \left(\frac{S_N}{2} + \frac{S_{N+1}}{2} \right)$

Problem2: Cho Data $X = \{1, 5, 4, 4, 9, 13\}$ Tìm median m của Data X đã cho ở trên?

Variance

- Data: $X = \{x_1, \dots, x_N\}$
- Mean: $\mu = \frac{1}{N} \sum_{i=1}^N x_i$
- Variance: $var(X) = \frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2$
- Standard Deviation: $\sigma = \sqrt{var(X)}$

Problem3: Cho Data $X = \{171, 176, 155, 167, 169, 182\}$. Tìm variance $var(X)$ và σ của Data X đã cho?

Correlation Coefficient

- Random variables X, Y: $X = \{x_1, \dots, x_N\}$ $Y = \{y_1, \dots, y_N\}$
- Correlation Coefficient: $p_{xy} = \frac{E[(X - \mu_X)(Y - \mu_Y)]}{\sigma_X \sigma_Y}$

$$= \frac{N(\sum_i x_i y_i) - (\sum_i x_i)(\sum_i y_i)}{\sqrt{N \sum_i x_i^2 - (\sum_i x_i)^2} \sqrt{N \sum_i y_i^2 - (\sum_i y_i)^2}}$$

Problem4: Cho $X = \{-2, -5, -11, 6, 4, 15, 9\}$ và $Y = \{4, 25, 121, 36, 16, 225, 81\}$
 Tìm correlation coefficient của Data X và Y đã cho?

Eigenvector and eigenvalue

- $\mathbf{A} \in R^{n \times n}$, \mathbf{I} (identity matrix) $\in R^{n \times n}$, $\mathbf{v} \in R^n$
- Eigenvalue (λ): $\det(\mathbf{A} - \lambda \mathbf{I}) = 0$
- Eigenvector (\mathbf{v}): $\mathbf{A}\mathbf{v} = \lambda \mathbf{v} \iff (\mathbf{A} - \lambda \mathbf{I})\mathbf{v} = 0$
- Normalize vector: $\frac{\mathbf{v}}{\|\mathbf{v}\|}$, $v_i = \frac{v_i}{\sqrt{\sum_1^n v_i^2}}$

Problem5: $\mathbf{A} = \begin{bmatrix} 0.9 & 0.2 \\ 0.1 & 0.8 \end{bmatrix}$ Tìm Eigenvector (\mathbf{v}) đã được normalize và eigenvalue λ của \mathbf{A} ?

Cosine Similarity

- Data (vector \mathbf{x} , \mathbf{y}): $\mathbf{x} = \{x_1, \dots, x_N\}$ $\mathbf{y} = \{y_1, \dots, y_N\}$
- Cosine Similarity: $cs(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\| \|\mathbf{y}\|} = \frac{\sum_1^n x_i y_i}{\sqrt{\sum_1^n x_i^2} \sqrt{\sum_1^n y_i^2}}$

Problem6: $\mathbf{x} = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}$ $\mathbf{y} = \begin{bmatrix} 1 \\ 0 \\ 3 \\ 0 \end{bmatrix}$ Tìm Cosine similarity $cs(\mathbf{x}, \mathbf{y})$?

Singular Value Decomposition

Có ma trận $\mathbf{A}_{m \times n}$ so sánh m và n

- Nếu $m > n$: có nghĩa dim của \mathbf{U} sẽ lớn hơn \mathbf{V} . Cần tìm ma trận \mathbf{U} trước bằng cách tìm eigenvectors (sắp xếp các eigenvector theo singular value tương ứng từ lớn đến bé) của $\mathbf{A}\mathbf{A}^T$. Tiếp theo, sắp xếp các singular values (căn bậc 2 eigenvalue của $\mathbf{A}\mathbf{A}^T$) theo thứ tự từ lớn đến bé. Hình thành ma trận đường chéo vuông có kích thước là số lượng singular values khác không, (đường chéo là các singular values vị trí 00, 11, 22, ... các vị trí còn lại = 0). Sau đó padding = 0 để ma trận này có kích thước = ma trận \mathbf{A} . Thu được Σ . Sau đó tìm ma trận \mathbf{V} bằng công thức cho từng vector trong \mathbf{V} : $\mathbf{v}_i = \frac{1}{\sigma_i} \mathbf{A}^T \mathbf{u}_i$
- Nếu $m < n$: có nghĩa dim của \mathbf{U} sẽ bé hơn \mathbf{V} . Cần tìm ma trận \mathbf{V} trước bằng cách tìm eigenvectors (sắp xếp các eigenvector theo singular value tương ứng từ lớn đến bé) của $\mathbf{A}^T \mathbf{A}$. Tiếp theo, sắp xếp các singular values (căn bậc 2 eigenvalue của $\mathbf{A}^T \mathbf{A}$) theo thứ tự từ lớn đến bé. Hình thành ma trận đường chéo vuông có kích thước là số lượng singular values khác không, (đường chéo là các singular values vị trí 00, 11, 22, ... các vị trí còn lại = 0). Sau đó padding = 0 để ma trận này có kích thước = ma trận \mathbf{A} . Thu được Σ . Sau đó tìm ma trận \mathbf{U} bằng công thức cho từng vector trong \mathbf{U} : $\mathbf{u}_i = \frac{1}{\sigma_i} \mathbf{A} \mathbf{v}_i$
- Cách tính determinant của ma trận 3x3

$$- \mathbf{C} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$$

$$- \det(\mathbf{C}) = a(ei - hf) - b(di - gf) + c(dh - eg)$$

Problem7: Tìm Singular Value Decomposition của $\mathbf{A} = \begin{bmatrix} 7 & 1 \\ 0 & 0 \\ 5 & 5 \end{bmatrix}$ ($\mathbf{U}, \Sigma, \mathbf{V}^T$)

Problem8 (Object Tracking: Bài này chạy trên máy local và cần có webcam để chạy được chương trình) Bài toán này sẽ yêu cầu sử dụng template matching với thư viện openCV để thực hiện track object (template) từ video của webcam. Các bạn dựa vào file mẫu sau [link notebook mẫu](#) tìm vị trí các dòng có ký hiệu "#???????" để thực hiện code:

- Hàm `capture_template()`:
 - **line 14:** Thực hiện convert ảnh màu sang ảnh xám (Các bạn tìm hiểu cách convert ảnh bằng thư viện openCV)
 - **line 20:** Thực hiện vẽ bounding box (hình chữ nhật) với 2 cặp tọa độ `top_left` và `bottom_right` lên ảnh `img` (Các bạn tìm hiểu cách vẽ hình chữ nhật lên ảnh bằng thư viện openCV)
 - **line 30:** Thực hiện crop ảnh `gray` để lấy được ảnh template bằng cách sử dụng 2 cặp tọa độ `top_left` và `bottom_right` (Các bạn tìm hiểu cách slice một array (ảnh xám))

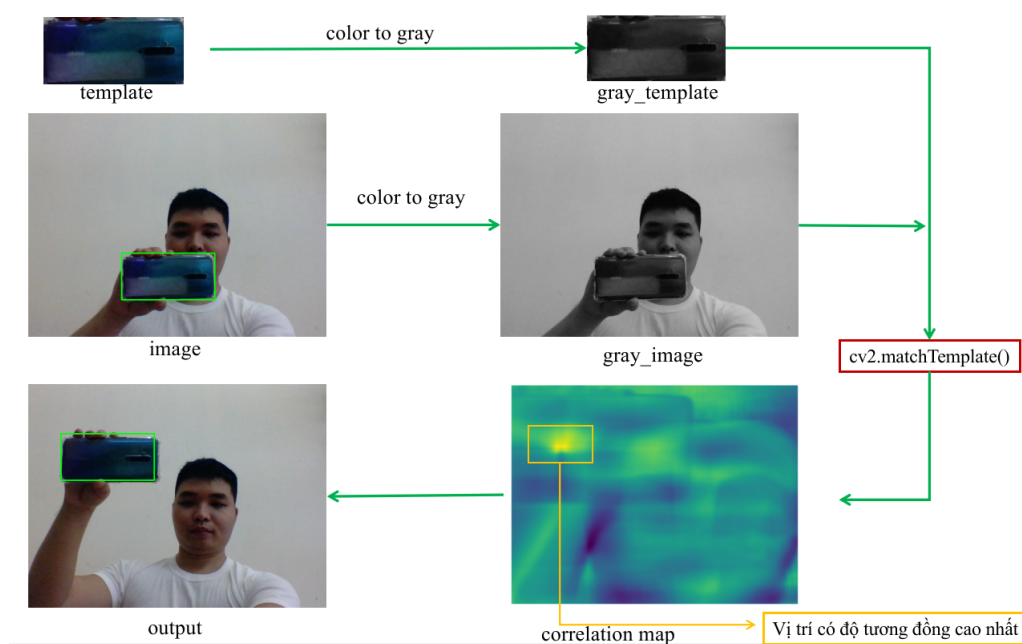
- Hàm `track_object(cap, template)`:

- **line 50**: Lấy chiều cao và chiều dài của `template`
- **line 51**: Dựa vào `top_left` (1 cặp tọa độ x, y), `h` (chiều cao), `w` (chiều dài) tính `bottom_right` (1 cặp tọa độ x, y)
- **line 54**: Thực hiện vẽ bounding box (hình chữ nhật) với 2 cặp tọa độ `top_left` và `bottom_right` lên ảnh `img` (Các bạn tìm hiểu cách vẽ hình chữ nhật lên ảnh bằng thư viện openCV)

- Sau khi các bạn hoàn thành 2 function trên thì chạy 3 function bên dưới để chạy hệ thống

```
1 cap, template = capture_template()
2 cap = track_object(cap, template)
3 release(cap)
```

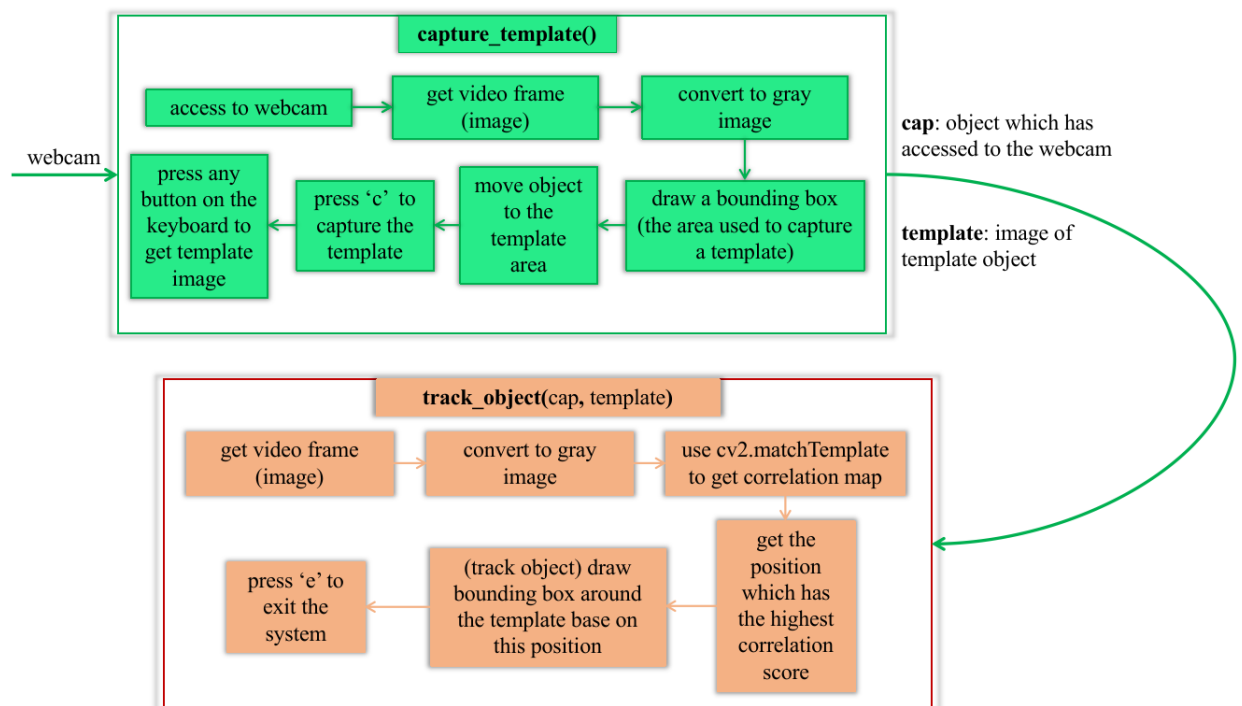
- Note:** Đây là mô tả cơ bản về cách hoạt động của hệ thống, khi chạy hệ thống sẽ sử dụng webcam của các bạn lấy thông tin và hiển thị video từ webcam thông qua một cửa sổ trên màn hình. Trong đó sẽ có một vùng bounding box các bạn đặt object (template) muốn tracking vừa với bounding box đó. Tiếp theo các bạn nhấn phím 'c' để lưu và lấy thông tin template trong bounding box. Sau đó hệ thống sẽ hiển thị ảnh template thông qua một cửa sổ khác. Khi các bạn kiểm tra đúng template mình mong muốn các bạn nhấn phím bất kỳ. Cuối cùng các bạn sẽ di chuyển object từ từ để thấy được object đang được track và và bounding box xung quanh object đó. Khi các bạn muốn kết thúc chương trình nhấn phím 'e'. **Các bạn có thể tham khảo video demo sau: [link video](#)**
- (Đọc thêm) Các bạn có thể tham khảo thêm hình 1, 2, 3 để hiểu rõ hơn các function trong template.



Hình 1: Matching template với một ảnh



Hình 2: Matching template một chuỗi ảnh (các frame). Tạo thành track object trong video



Hình 3: Cách hoạt động của chương trình