Họ và tên: Nguyễn Công Hiếu

MSSV: 22110124

BÁO CÁO THỊ GIÁC MÁY TÍNH

**I. Giới thiệu**

Ứng dụng “Digital Imaging Processing” là một web app dựa trên Streamlit cho phép thao tác và trình diễn các thuật toán xử lý ảnh số từ cơ bản đến nâng cao trong một giao diện trực quan. Người dùng có thể chọn chức năng trước, tải ảnh lên và quan sát ngay kết quả xử lý cùng các biểu đồ/hình ảnh minh họa liên quan.

Ứng dụng hỗ trợ cả ảnh grayscale và RGB, nhiều định dạng phổ biến (PNG, JPG/JPEG, BMP, TIF/TIFF), và bao phủ các nhóm thuật toán chính:

* Biến đổi cường độ: Resolution, Quantization (4/8/16/256 levels), RGB channels, Negative, Thresholding, Logarithmic, Power-law (Gamma), Contrast Stretching, Piecewise Linear, Gray-level Slicing, Bit-plane Slicing.
* Histogram: Equalization, Matching/Specification (Uniform, Gaussian, hoặc từ ảnh tham chiếu).
* Tương quan chuẩn hóa (Normalized Correlation): Tự động trích mask từ ảnh gốc, dùng template upload, hoặc kernel tùy chỉnh; hiển thị correlation map và vị trí khớp tốt nhất.
* Lọc không gian: Correlation/Convolution, Smoothing (Average/Gaussian/Box), Median, Sharpening (Laplacian/Unsharp/High-boost), Spatial filter mở rộng.
* Fourier: Phân tích 1-D theo hàng/cột và 2-D với phổ biên độ (fftshift, log-scale).
* PCA Face Detection: Phát hiện khuôn mặt bằng mô hình PCA tiền huấn luyện (nếu có), tự động fallback Haar cascade khi không có model; tùy chọn huấn luyện nhanh từ ảnh upload.
* Restoration: Mô hình nhiễu (Gaussian/Uniform/Impulse/Salt & Pepper/Periodic), khử nhiễu không gian (Median/Average/Gaussian/Min/Max/Midpoint/Alpha-Trimmed Mean), giảm nhiễu tuần hoàn (notch filtering), mô phỏng suy giảm tuyến tính (Gaussian/Motion blur), và lọc nghịch đảo (Inverse filtering).
* Morphology: Erosion, Dilation, Opening, Closing, Gradient, Top-hat, Black-hat với nhiều dạng kernel.
* Segmentation: Ngưỡng toàn cục (Mean), Otsu, K-Means (mask hoặc bản đồ nhãn màu).

Mục tiêu là cung cấp một công cụ gọn nhẹ để học tập, minh họa và thử nghiệm nhanh các kỹ thuật xử lý ảnh, với kiến trúc mã tách theo module, dễ bảo trì và mở rộng.

**II. Cơ sở lý thuyết và chức năng (công thức và chức năng của thuật toán)**

1. **Biến đổi cường độ (Intensity Transformations)**

* Resolution: Thay đổi kích thước ảnh theo tỉ lệ, sử dụng nội suy phù hợp (nearest/bilinear/area).
* Quantization: Lượng tử hóa mức xám/màu theo số levels cố định (4/8/16/256), tạo hiệu ứng posterization.
* RGB Channels: Bật/tắt các kênh R/G/B để quan sát đóng góp của từng kênh.
* Negative Images: s = 255 − r (đảo màu để làm nổi bật biên và vùng tối).
* Thresholding: s = 0 hoặc 255 theo ngưỡng T cố định hoặc điều chỉnh; có thể dùng Otsu (tự động ngưỡng tối ưu).
* Logarithmic: s = c·log(1 + r), mở rộng vùng tối và nén vùng sáng; thường hiển thị phổ/log để tăng độ tương phản.
* Gamma (Power-law): s = r^γ (γ < 1 ảnh sáng hơn, γ > 1 ảnh tối hơn); ứng dụng hiệu chỉnh màn hình/hiệu chỉnh gamma.
* Contrast Stretching:
  + Min–Max: s = (r − r\_min)/(r\_max − r\_min) × 255
  + Percentile Stretching: dùng phần trăm (p\_low, p\_high) thay vì giá trị cực trị để giảm ảnh hưởng outliers.
* Piecewise Linear: Biến đổi theo các đoạn tuyến tính định nghĩa bởi các cặp điểm (r\_i, s\_i).
* Gray-level Slicing: Làm nổi bật một dải mức xám [a, b], giữ hoặc bỏ nền tùy chọn.
* Bit-plane Slicing: Phân tích ảnh theo từng bit-plane (b0…b7) để quan sát thông tin cấu trúc; tái tạo từ các bit-plane được chọn.

1. **Histogram**

* Equalization: Cân bằng histogram để tăng tương phản; dùng phân phối tích lũy CDF để ánh xạ mức xám:
  + s = T(r) = CDF(r) × 255
* Matching/Specification: Khớp histogram ảnh đầu vào với phân bố mục tiêu:
  + Uniform: phân bố đều
  + Gaussian: N(μ, σ) với μ, σ mặc định
  + Custom Image: khớp theo histogram ảnh tham chiếu

1. **Tương quan chuẩn hóa (Normalized Correlation – NCC)**

* Mục tiêu: Đo độ tương đồng giữa template và các vùng trong ảnh.
* Công thức NCC (triển khai thủ công):
  + NCC(x, y) = Σ[(R − μ\_R)(T − μ\_T)] / [h·w·σ\_R·σ\_T]
  + R: vùng ảnh kích thước h×w tại vị trí (x, y); T: template; μ, σ: trung bình và độ lệch chuẩn tương ứng.
* Chế độ:
  + Template Matching (OpenCV): cv2.matchTemplate với TM\_CCOEFF\_NORMED
  + Auto Detect Mask: Trích mask từ ảnh gốc theo tọa độ/kích thước phần trăm
  + Custom Kernel: Tương quan với kernel tùy chỉnh, chuẩn hóa tránh ảnh hưởng độ sáng nền
* Kết quả: Correlation map (vùng sáng = tương đồng cao), vị trí khớp tốt nhất, hiển thị khung bao template.

1. **Lọc không gian (Filtering)**

* Correlation/Convolution: Áp dụng kernel 2D; convolution lật kernel 180° trước khi tích chập.
* Smoothing Linear Filters:
  + Average/Box: Làm mượt bằng trung bình miền
  + Gaussian: Làm mượt theo phân bố Gaussian; tham số σ quyết định độ mờ
* Median Filter: Bỏ nhiễu muối tiêu bằng giá trị trung vị cửa sổ; bảo toàn biên tốt hơn so với trung bình.
* Sharpening:
  + Laplacian: Tăng cường biên dựa trên đạo hàm bậc hai
  + Unsharp Masking: s = r + k(r − Gσ(r)), k điều chỉnh độ sắc
  + High-boost: Tương tự unsharp với k > 1 để tăng cường chi tiết
* Spatial Filter nâng cao:
  + Weighted average, Order-statistic (Median/Max/Min)
  + Gradient/Sobel: ước lượng biên, hướng biên
  + High-pass: nhấn mạnh chi tiết tần số cao

1. **Biến đổi Fourier**

* 1-D FFT: Trích tín hiệu theo hàng/cột, tính phổ biên độ |F(ω)|, quan sát thành phần tần số.
* 2-D FFT: F(u, v) = Σ\_x Σ\_y f(x, y) e^(−j2π(ux/M + vy/N)); hiển thị magnitude spectrum sau fftshift và log(1 + |F|).

1. **Phát hiện khuôn mặt bằng PCA**

* Ý tưởng: Biểu diễn ảnh mặt trong không gian các thành phần chính (eigenfaces); tính lỗi tái tạo để phân biệt mặt/không mặt.
* Quy trình:
  + Mặc định: tải mô hình PCA tiền huấn luyện từ models/pca\_face\_model.npz
  + Fallback: nếu không có model, dùng Haar cascade của OpenCV
  + Tùy chọn: huấn luyện từ ảnh mặt upload (grayscale, kích thước thống nhất, ví dụ 64×64)
* Tham số mặc định: window 64×64, components 20, stride 16, threshold 1500
* Kết quả: Vẽ khung phát hiện, đếm số khuôn mặt.

1. **Phục hồi ảnh (Restoration)**

* Noise Models:
* Gaussian: n ~ N(μ, σ²).
* Uniform: n ~ U(−A, +A).
* Impulse: một phần pixel bị thay thế bởi 0 hoặc 255.
* Salt & Pepper: chọn ngẫu nhiên pixel thành 0/255 theo “amount”.
* Periodic: thêm nhiễu tuần hoàn ở miền tần số theo tần số u, v.
* Spatial Denoising:
* Median, Average, Gaussian.
* Min filter: s = min(window); tương đương erosion cường độ.
* Max filter: s = max(window); tương đương dilation cường độ.
* Midpoint: s = (min + max)/2; thích hợp giảm nhiễu đối xứng.
* Alpha-Trimmed Mean: s = mean(window sau khi cắt d/2 giá trị nhỏ nhất và d/2 lớn nhất).
* Periodic Noise Reduction: Phát hiện đỉnh trong phổ |F| và tạo notch filters; giảm nhiễu tuần hoàn.
* Linear, Position-Invariant Degradations: Mô phỏng PSF Gaussian/Motion; ảnh mờ do hệ thống.
* Inverse Filtering: Ĝ(u,v) = F(u,v)/(H(u,v)+ε); ε giúp ổn định khi H gần 0.

1. **Hình thái học (Morphology)**

* Phép toán: Erosion, Dilation, Opening, Closing, Morphological Gradient, Top-hat, Black-hat
* Structuring element: Rectangle/Ellipse/Cross, kích thước lẻ 3–31, hỗ trợ iterations cho các phép cần lặp
* Ứng dụng: Tách nền, lấp lỗ, làm nổi bật cấu trúc nhỏ, trích xuất biên/mặt nạ.

1. **Segmentation**

* Global Threshold (Mean): T = mean(r); s = 0/255 theo r < T hoặc r ≥ T.
* Otsu Thresholding: Tối đa hóa giữa-lớp, tối thiểu hóa trong-lớp; tự động tìm ngưỡng T tối ưu.
* K-Means (Grayscale): Gom cụm cường độ thành K nhóm; đầu ra:
* Binary (chọn cụm sáng nhất làm foreground), hoặc
* Label Map (bản đồ nhãn 0..K−1; trực quan hóa bằng colormap).

**III. Demo ứng dụng**

* Giao diện: Streamlit, gồm Sidebar chọn “Loại xử lý” và khu vực trung tâm hiển thị ảnh.

A screenshot of a computer

AI-generated content may be incorrect.

* Luồng thao tác:
  + Chọn chức năng trước trong Sidebar (ví dụ: Histogram → Equalization)
  + Upload ảnh; hệ thống tự chuyển đổi kênh phù hợp (RGB/GRAY) và hiển thị song song ảnh gốc/ảnh xử lý

A screenshot of a computer

AI-generated content may be incorrect.

* + Với NCC: chọn Auto Detect Mask hoặc template; hiển thị correlation map và khung kết quả

A phone and a book on a table

AI-generated content may be incorrect.

* + Với Fourier: xem phổ 2-D; với 1-D, chọn hàng/cột và xem tín hiệu cùng phổ

A screenshot of a computer

AI-generated content may be incorrect.

* + Với PCA: không cần huấn luyện runtime; hệ thống tự phát hiện dựa trên model có sẵn hoặc Haar fallback

A screenshot of a football team

AI-generated content may be incorrect.

* + Với Restoration: thêm noise, lọc không gian, khử nhiễu tuần hoàn, mô phỏng suy giảm và nghịch đảo

A screenshot of a computer

AI-generated content may be incorrect.

* + Với Morphology: chọn phép, kernel shape/size, iterations; xem kết quả tức thời.

A screenshot of a computer

AI-generated content may be incorrect.

* + Thử nghiệm Segmentation:
    - Global Mean: đạt phân tách nhanh với đối tượng/ nền khác biệt rõ.
    - Otsu: hiệu quả trên ảnh bimodal; tự động ngưỡng.
    - K-Means: K=2 cho foreground/background; Label Map hiển thị ranh giới cụm.

A screenshot of a computer

AI-generated content may be incorrect.

* Tải kết quả: Sidebar có nút “Tải ảnh đã xử lý” để lưu PNG.

**IV. Công nghệ sử dụng**

* Ngôn ngữ và thư viện:
  + Python 3.11, NumPy, SciPy
  + OpenCV (opencv-python), scikit-image cho xử lý ảnh
  + Matplotlib cho biểu đồ/phổ
  + Streamlit cho UI web
  + scikit-learn cho PCA
* Kiến trúc mã:
  + app.py: điều phối UI và gọi các hàm theo danh mục
  + src/processing\_functions.py, histogram.py, correlation.py, filtering.py, fourier.py, pca\_face.py, restoration.py, morphology.py, ui\_helpers.py
* Triển khai:
  + Môi trường ảo venv, requirements.txt để cài đặt nhanh
  + Hỗ trợ Windows PowerShell khởi chạy: python -m streamlit run app.py

**V. Kết luận**

* Ứng dụng đáp ứng mục tiêu trình diễn và thực hành các thuật toán xử lý ảnh từ cơ bản đến nâng cao với giao diện đơn giản, trực quan.
* Việc tách module theo chức năng giúp mã dễ bảo trì, mở rộng. PCA được đơn giản hóa nhờ mô hình tiền huấn luyện và fallback Haar.
* Các tính năng như NCC, Fourier, Restoration, Morphology cung cấp góc nhìn đa chiều về phân tích, lọc, phục hồi và trích đặc trưng ảnh.

**VI. Tài liệu tham khảo**

* Rafael C. Gonzalez, Richard E. Woods, Digital Image Processing, Pearson.
* OpenCV Documentation: filter2D, matchTemplate, Cascade Classifier.
* NumPy/SciPy FFT: numpy.fft, scipy.fftpack.
* scikit-image Documentation: filters, morphology.
* scikit-learn PCA: decomposition.PCA.