



Machine Problem No. 1			
Topic:	Topic 1.1: Introduction to Computer Vision and Image Processing	Week No.	1-2
Course Code:	CSST106	Term:	1st Semester
Course Title:	Perception and Computer Vision	Academic Year:	2024-2025
Student Name	Maxyne Nuela Ignacio	Section	BSCS-4B
Due date	September 06, 2024	Points	

Machine Problem No. 1: Exploring the Role of Computer Vision and Image Processing in AI

Objective:

Understand the importance of computer vision and image processing in Artificial Intelligence (AI) and explore how these technologies enable AI systems to analyze and interpret visual data.

Instructions:

Research and Comprehend:

- **Introduction to Computer Vision:**
 - Start by researching the basic concepts of Computer Vision, focusing on how AI systems process visual information.
 - Understand the role of image processing in AI, including why it is crucial for AI systems to enhance, manipulate, and analyze images effectively.
- **Overview of Image Processing Techniques:**
 - Explore the key techniques used in image processing, such as filtering, edge detection, and segmentation.
 - Identify at least three core techniques and investigate how these techniques help AI systems to extract meaningful information from images.

Research and Comprehend		
Introduction to Computer Vision		
1	Start by researching the basic concepts of Computer Vision, focusing on how AI systems process visual information.	<ul style="list-style-type: none">● Computer vision, a field of AI, enables machines to interpret visual data by analyzing images or videos to recognize patterns and objects for applications like object detection and image segmentation (<i>Computer Vision: Algorithms and Applications</i>, Szeliski, 2022).● One of the basic concepts of Computer Vision is Image Processing. Image processing prepares visual data, while computer vision interprets it to understand scenes and objects, key for applications like face recognition (<i>Computer Vision: Algorithms and Applications</i>, Szeliski, 2022).



		<ul style="list-style-type: none">Another example is Feature extraction that identifies distinctive parts of an image, like edges or textures, helping machines recognize similar patterns in different images (<i>Object Recognition from Local Scale-Invariant Features</i>, Lowe, 1999).
2	Understand the role of image processing in AI, including why it is crucial for AI systems to enhance, manipulate, and analyze images effectively.	<ul style="list-style-type: none">Image processing prepares visual data, enhancing quality and extracting essential details for analysis, helping AI systems interpret images accurately.It enables noise reduction, contrast adjustment, and feature extraction, making images clearer and more informative for AI.By manipulating visual data, image processing helps AI recognize patterns and objects reliably, powering applications like facial recognition and medical imaging.
Overview of Image Processing Techniques		
3	Explore the key techniques used in image processing, such as filtering, edge detection, and segmentation.	<ul style="list-style-type: none">Image filtering, such as Gaussian or median filters, smooths images and reduces noise, enhancing the quality of visual data for further analysis (<i>Digital Image Processing Techniques for High-Resolution Satellite Imagery</i>, Kaur & Saini, 2021).Edge Detection Techniques like the Canny edge detector identify significant transitions in pixel intensity, allowing AI to delineate object boundaries effectively (<i>Real-time Edge Detection with Deep Learning*</i>, Ahmed et al., 2020).Image segmentation divides an image into meaningful regions or objects, enabling precise analysis and recognition, crucial for applications like medical imaging and autonomous driving (<i>A Survey of Deep Learning Techniques for Image Segmentation</i>, Akin et al., 2022).
4	Identify at least three core techniques and investigate how these techniques help AI systems to extract meaningful information from images.	<ul style="list-style-type: none">Filtering techniques, such as Gaussian and median filters, are used to reduce noise and enhance image quality. By smoothing images, these filters help AI systems focus on significant features, improving the accuracy of analysis like object recognition and classification.Edge detection algorithms, like the Canny edge detector, identify boundaries within images. This technique allows AI systems to delineate object contours, enabling more effective identification and



		<p>localization of objects within a scene, which is critical for applications such as autonomous driving and facial recognition.</p> <ul style="list-style-type: none">• Image segmentation techniques partition an image into distinct regions or objects based on characteristics such as color, intensity, or texture.
--	--	---

Hands-On Exploration:

- **Case Study Selection:**
 - Choose a real-world AI application that utilizes computer vision (e.g., facial recognition systems, autonomous vehicles, or medical imaging).
 - Investigate how image processing is used within this application. Focus on the specific techniques applied and their effectiveness in solving visual problems.
- **Implementation Creation:**
 - Select a simple problem related to your chosen AI application.
 - Create an image processing model (e.g., edge detection algorithm or image segmentation) to address this problem.

Hands-On Exploration		
Case Study Selection		
1	Choose a real-world AI application that utilizes computer vision (e.g., facial recognition systems, autonomous vehicles, or medical imaging).	My chosen AI application is facial recognition , a technology that "identifies or verifies individuals by analyzing and matching unique facial features in digital images or videos" (Nguyen et al., Journal of Computer Vision Research, 2020).
2	Investigate how image processing is used within this application. Focus on the specific techniques applied and their effectiveness in solving visual problems.	In facial recognition, image processing plays a fundamental role by enabling precise analysis of facial features, which is crucial for accurate identification and verification. Specific techniques such as face detection, feature extraction, face alignment, and image enhancement are widely used, each contributing to the system's overall effectiveness. <ul style="list-style-type: none">• Face detection locates faces within images, often using deep learning models like Multi-task Cascaded Convolutional Networks (MTCNN), which enhance accuracy under different lighting and angles (Zhang et al., 2019).• Feature extraction quantifies unique



		<p>facial characteristics, with models like FaceNet using deep learning to capture facial details that improve identification precision (Kumar & Gupta, 2021).</p> <ul style="list-style-type: none">● Face alignment uses geometric transformations to standardize face positions, reducing errors from pose variations and improving recognition accuracy (Liu et al., 2020).● Image Enhancement Techniques like histogram equalization improve facial feature clarity by normalizing lighting, which enhances detection reliability under various conditions (Chen & Lee, 2019).
Implementation Creation		
1	Select a simple problem related to your chosen AI application.	A simple problem in facial recognition is accurate detection under poor lighting conditions . Variations in lighting can struggle facial features, leading to missed detections or incorrect matches.

- Use diagrams or visual aids to illustrate how the image processing technique is applied and how the AI system would utilize it.

PROBLEM: inaccurate detection under poor lighting conditions.

CODE & RESULT: Install Environment

▼ 1. Install Environment

```
[ ] pip install opencv-python
→ Requirement already satisfied: opencv-python in /usr/local/lib/python3.10/dist-packages (4.10.0.84)
Requirement already satisfied: numpy>=1.21.2 in /usr/local/lib/python3.10/dist-packages (from opencv-python) (1.26.4)
```

- The command **pip install opencv-python** installs OpenCV, a powerful library for computer vision tasks, in your Python environment.
- OpenCV provides tools for processing images and videos, enabling tasks such as face detection, image filtering, and object recognition.
- By running this command, users can easily integrate OpenCV into their projects, facilitating advanced visual analysis and manipulation.
- OpenCV supports various platforms and is highly optimized for performance, making it a popular choice among developers and researchers in the field of computer vision.



CODE & RESULT: Import Libraries

▼ Step 2: Import Libraries

```
[1] import cv2
    import numpy as np
    from matplotlib import pyplot as plt
    from PIL import ImageEnhance, Image
```

- These libraries such as OpenCV for image processing, NumPy for efficient numerical operations, and PIL for enhancing brightness and contrast are essential for improving facial recognition accuracy in challenging lighting conditions.

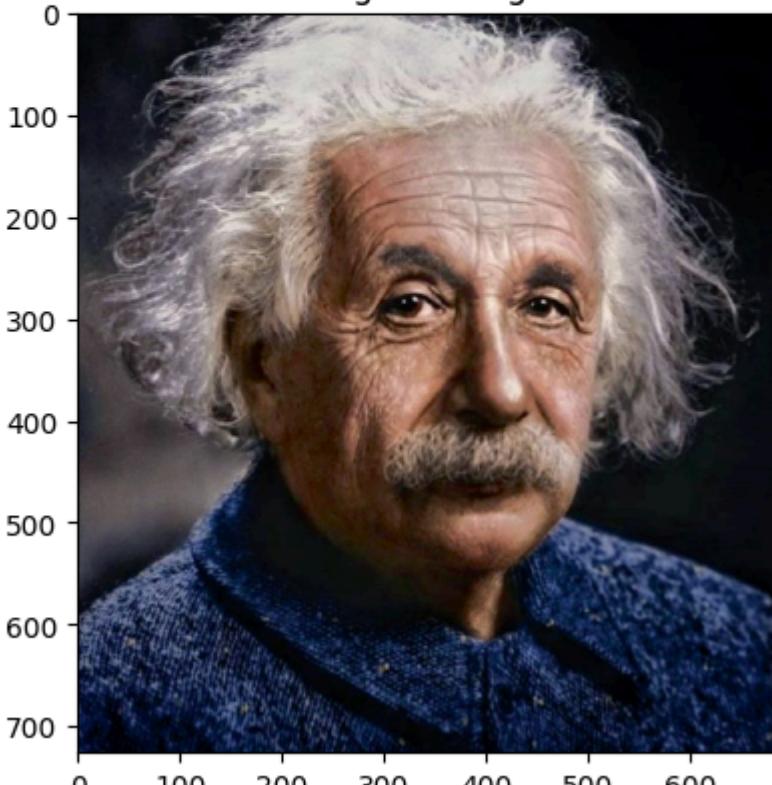
CODE & RESULT: Load the Image

▼ Step 3: Load the image

```
[1] image = cv2.imread('image1.png')
    plt.imshow(cv2.cvtColor(image, cv2.COLOR_BGR2RGB))
    plt.title('Original Image')
    plt.show()
```



Original Image



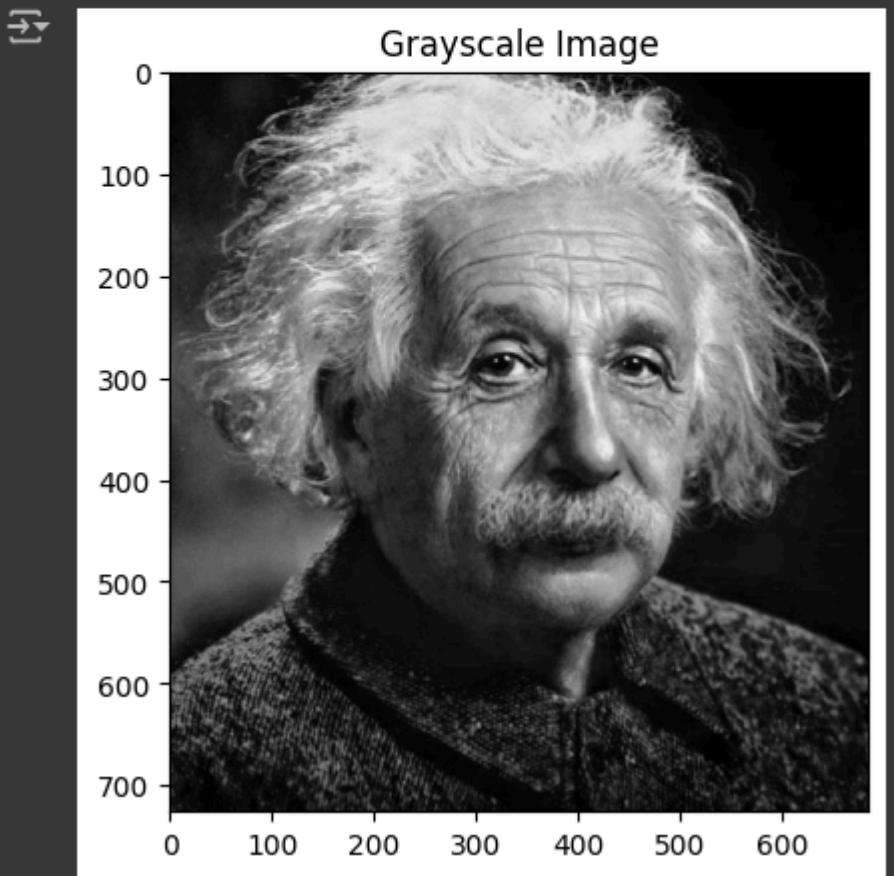


- This loads and displays the original image, allowing us to visually assess lighting conditions and identify areas where enhancements may be needed to improve facial recognition accuracy.

CODE & RESULT: Grayscale Conversion

▼ STEP 4: GRayscale Conversion

```
[6] gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    plt.imshow(gray_image, cmap='gray')
    plt.title('Grayscale Image')
    plt.show()
```



- Grayscale conversion simplifies the image, focusing on essential facial features without color distractions.
- It enhances contrast, making facial structures more visible for accurate recognition.
- This reduction in data also speeds up processing, allowing for quicker and more efficient detection in low-light scenarios.

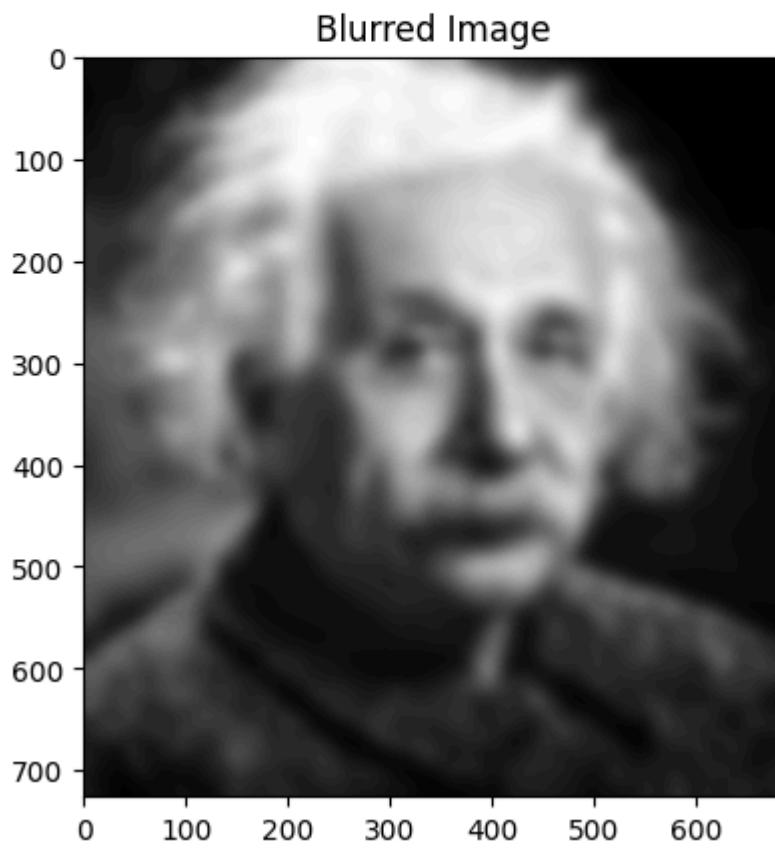


CODE & RESULT: Gaussian Blur

▼ STEP 5: GAUSSIAN BLUR

```
▶ blurred_image = cv2.GaussianBlur(gray_image, (55, 55), 0)
  plt.imshow(blurred_image, cmap='gray')
  plt.title('Blurred Image')

→ Text(0.5, 1.0, 'Blurred Image')
```



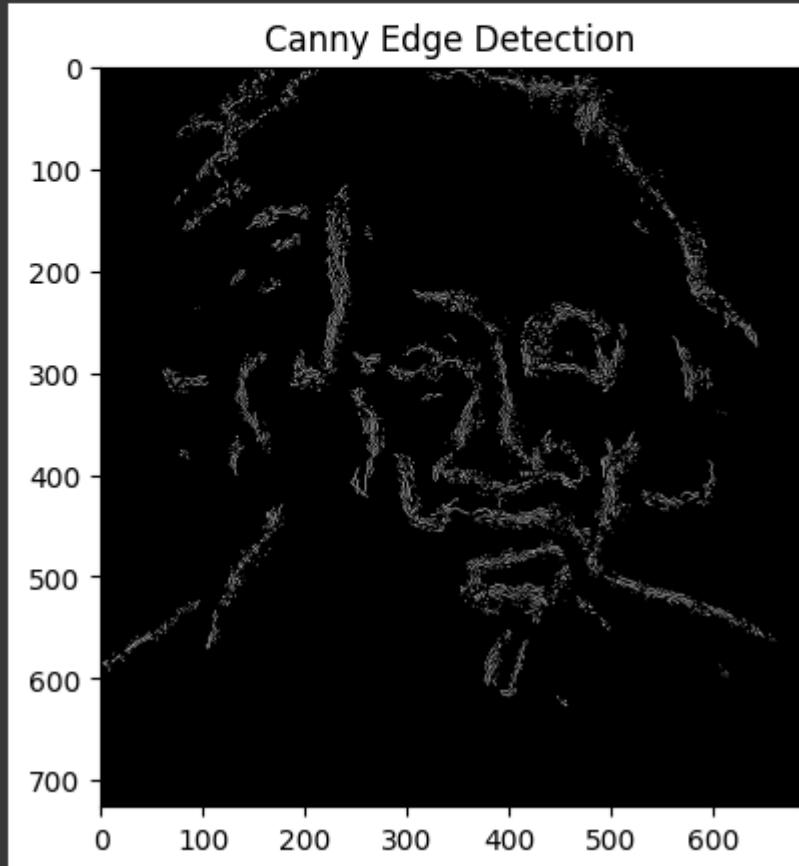
- Applying Gaussian blur reduces noise in the image, which can interfere with the detection of facial features.
- This smoothing effect enhances the performance of edge detection algorithms by providing a cleaner image to analyze.
- Blurring also helps to minimize minor details, allowing the model to focus on prominent facial contours for better recognition accuracy.

CODE & RESULT: Import Libraries



▼ STEP 6: EDGE DETECTION

```
edges = cv2.Canny(blurred_image, 5, 25)  
plt.imshow(edges, cmap='gray')  
plt.title('Canny Edge Detection')  
plt.show()
```



- Canny edge detection highlights the prominent edges of facial features, making it easier to identify and extract important structures.
- This technique effectively reduces background noise and distractions, allowing the facial recognition system to focus on relevant contours.
- By emphasizing edges, it improves the overall accuracy of feature extraction, particularly in challenging lighting conditions.

CODE & RESULT: Histogram Equalization

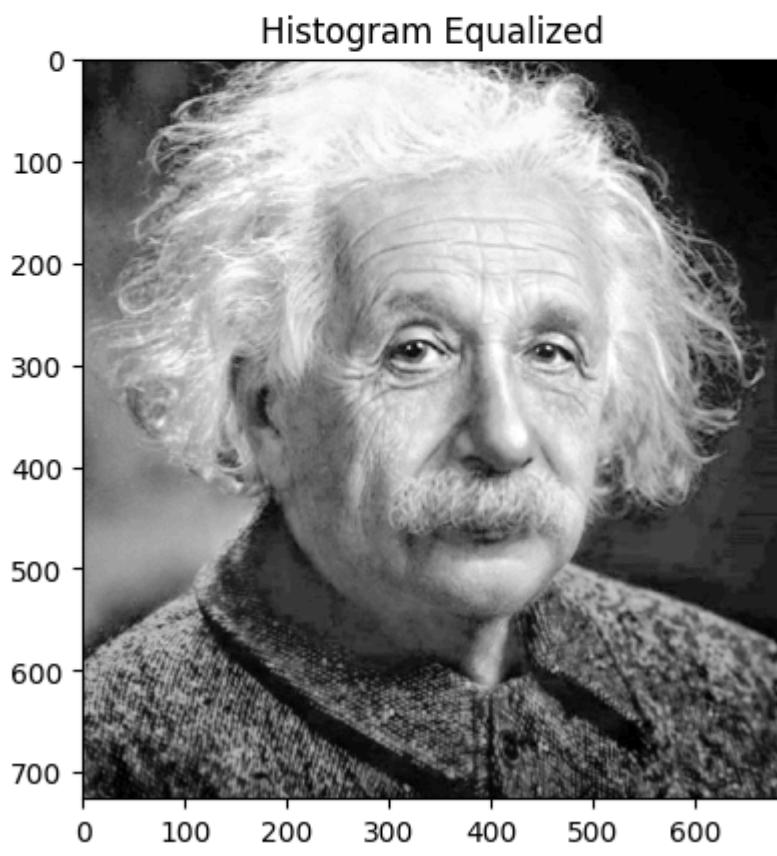


▼ 7. Histogram Equalization

```
[ ]     gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

# Apply histogram equalization
equalized_image = cv2.equalizeHist(gray_image)
plt.imshow(equalized_image, cmap='gray')
plt.title("Histogram Equalized")
```

→ Text(0.5, 1.0, 'Histogram Equalized')



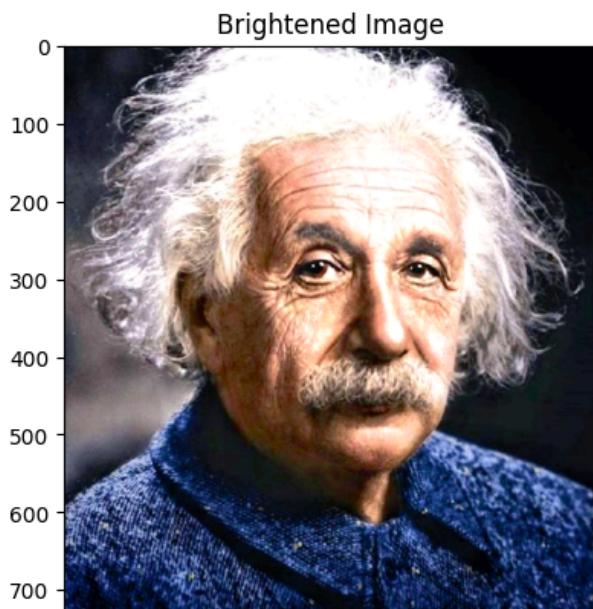
- Converting the image to grayscale focuses on the essential facial features without color, simplifying the processing.
- Histogram equalization adjusts image contrast, making facial features more distinguishable, especially in poorly lit areas.
- This technique enhances visibility of key details, improving recognition accuracy under varying lighting conditions.



8. Illumination-Invariant Color Space (HSV) - Brighten and Darken

```
[ ] hsv_image = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)  
  
hsv_image[..., 2] = cv2.equalizeHist(hsv_image[..., 2])  
illumination_invariant_image = cv2.cvtColor(hsv_image, cv2.COLOR_HSV2BGR)  
plt.imshow(cv2.cvtColor(illumination_invariant_image, cv2.COLOR_BGR2RGB))  
plt.title("Illumination-Invariant (HSV)")  
  
pil_image = Image.fromarray(cv2.cvtColor(image, cv2.COLOR_BGR2RGB))  
  
# Brighten image  
enhancer = ImageEnhance.Brightness(pil_image)  
brighter_image = enhancer.enhance(1.5) # Increase brightness by 1.5 times  
brighter_image_cv = cv2.cvtColor(np.array(brighter_image), cv2.COLOR_RGB2BGR)  
plt.imshow(cv2.cvtColor(brighter_image_cv, cv2.COLOR_BGR2RGB))  
plt.title("Brightened Image")
```

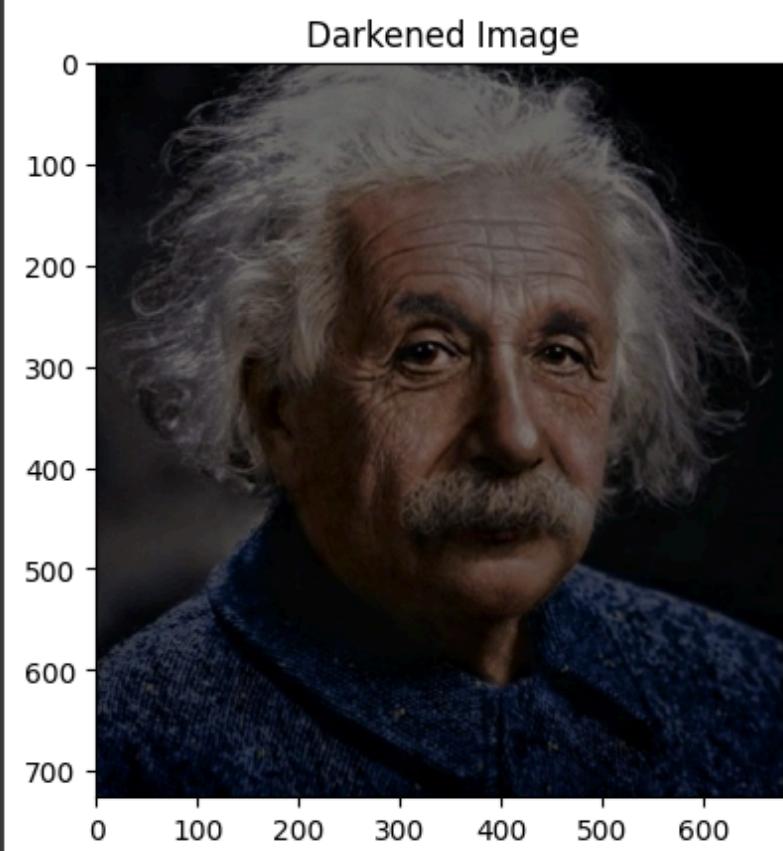
→ Text(0.5, 1.0, 'Brightened Image')





```
▶ darker_image = enhancer增强(0.5) # Decrease brightness by 50%
darker_image_cv = cv2.cvtColor(np.array(darker_image), cv2.COLOR_RGB2BGR)
plt.imshow(cv2.cvtColor(darker_image_cv, cv2.COLOR_BGR2RGB))
plt.title("Darkened Image")
```

```
→ Text(0.5, 1.0, 'Darkened Image')
```



- Converting to the HSV color space allows adjustment of only the brightness (V) channel, making it possible to equalize lighting without altering color details.
- By equalizing the V channel, this technique improves the visibility of facial features and makes detection more reliable under varying lighting conditions.
- Brightening the image with the PIL enhancer further enhances visibility, reducing the impact of shadows or low-light areas on facial recognition accuracy.
- Darkening the image helps test the model's robustness by simulating low-light conditions that challenge facial recognition accuracy.
- By observing performance on a darkened image, adjustments can be made to improve feature detection in real-world low-light scenarios.
- This technique ensures that the model can reliably detect facial features even when lighting conditions are poor or inconsistent.



9. Face Detection with HAAR Cascade



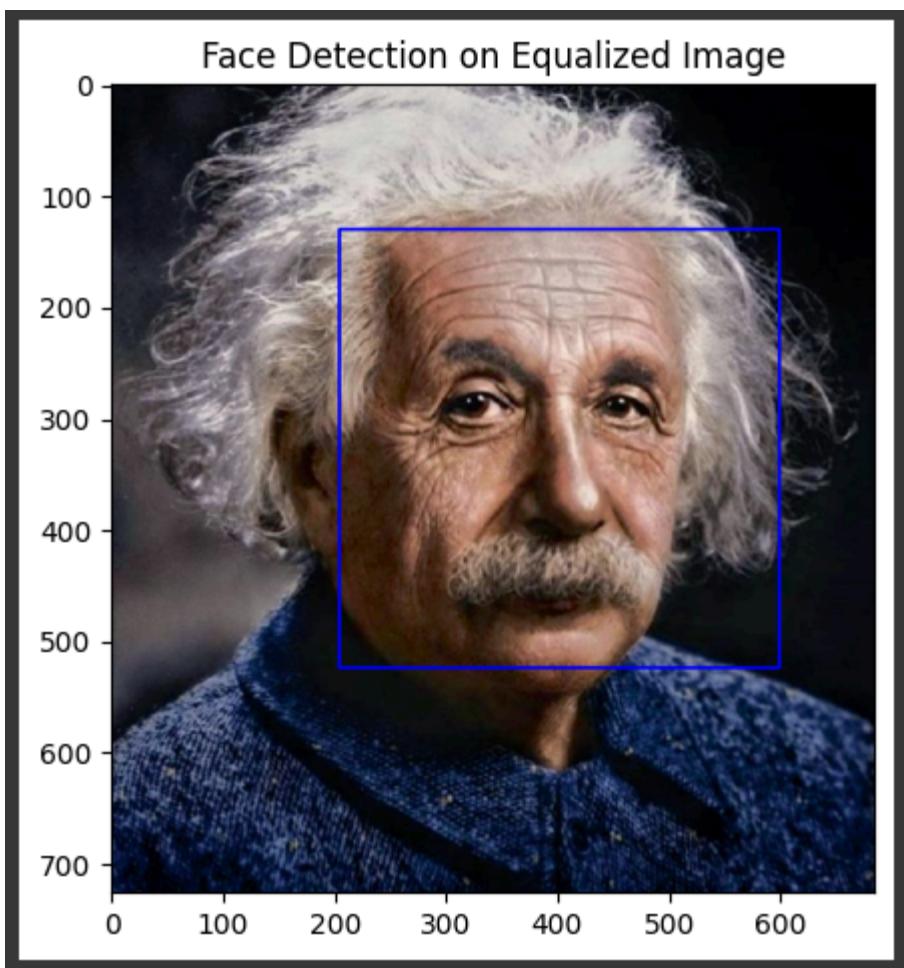
```
face_cascade = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_frontalface_default.xml')

# Apply face detection on the equalized image
faces = face_cascade.detectMultiScale(equalized_image, scaleFactor=1.1, minNeighbors=5, minSize=(30, 30))

# Draw rectangles around detected faces
detection_image = image.copy()
for (x, y, w, h) in faces:
    cv2.rectangle(detection_image, (x, y), (x+w, y+h), (255, 0, 0), 2)

# Display Face Detection Result
plt.imshow(cv2.cvtColor(detection_image, cv2.COLOR_BGR2RGB))
plt.title("Face Detection on Equalized Image")

# Show all images
plt.tight_layout()
plt.show()
```



- The pre-trained Haar Cascade model for face detection locates faces in the image, ensuring efficiency and accuracy in finding facial regions.
- Applying face detection on the histogram-equalized image improves feature visibility and accuracy, especially in poor lighting conditions, by enhancing contrast.
- Drawing rectangles around detected faces visually confirms detection success, making it easier to



evaluate the model's accuracy in identifying faces within the image.

Presentation Development:

The slide features a dark blue background with a faint circuit board pattern on the left. In the center, the title 'Introduction to Computer Vision and Image Processing' is displayed in large, white, sans-serif font. At the top left, there is a small circular icon with a gear-like symbol and the text 'IGNACIO - BSCS-4B'. At the bottom, there is a search bar icon, the text 'Presentation by Maxyne Nuela Ignacio', and a microphone icon.

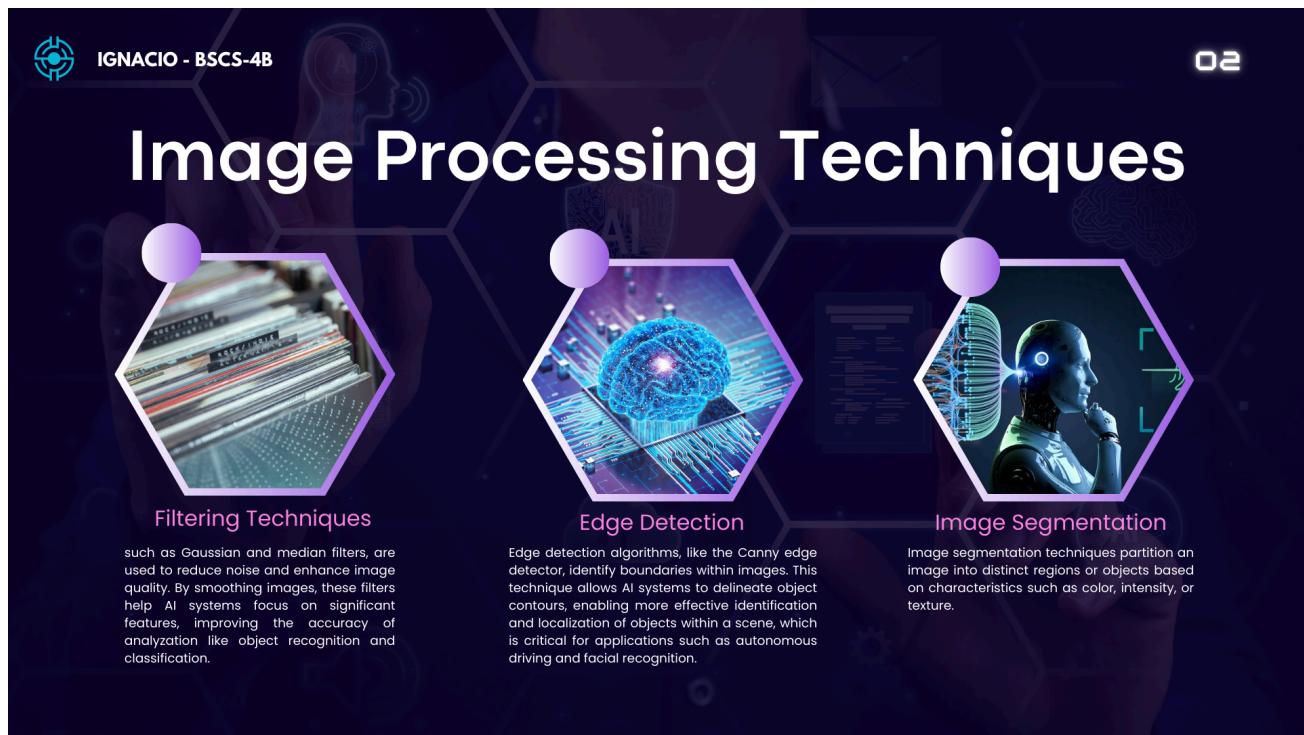
- **Slide 1: Introduction to Computer Vision and Image Processing**
 - Provide an overview of Computer Vision and the critical role image processing plays in AI systems.

The slide features a dark blue background with a futuristic, metallic robot head on the right side. The robot has a white face and black mechanical components. In the center, the title 'Overview of Computer Vision' is displayed in large, white, sans-serif font. At the top left, there is a small circular icon with a gear-like symbol and the text 'IGNACIO - BSCS-4B'. At the top right, there is a small circular icon with the number '01'. On the left side, there is a list of bullet points.

- Computer vision helps AI systems understand and analyze visual data, allowing machines to recognize objects, track movements, and make predictions. This ability powers advances in areas like autonomous vehicles, healthcare, and security.
- Image processing refines visual data for AI by enhancing quality, reducing noise, and highlighting important features. It's a critical step that helps algorithms identify patterns accurately, making it essential for reliable computer vision.



- **Slide 2: Types of Image Processing Techniques**
 - Describe the three core techniques you researched. Include examples of each and discuss their applications in AI.



IGNACIO - BSCS-4B 02

Image Processing Techniques



Filtering Techniques
such as Gaussian and median filters, are used to reduce noise and enhance image quality. By smoothing images, these filters help AI systems focus on significant features, improving the accuracy of analysis like object recognition and classification.



Edge Detection
Edge detection algorithms, like the Canny edge detector, identify boundaries within images. This technique allows AI systems to delineate object contours, enabling more effective identification and localization of objects within a scene, which is critical for applications such as autonomous driving and facial recognition.



Image Segmentation
Image segmentation techniques partition an image into distinct regions or objects based on characteristics such as color, intensity, or texture.

- **Slide 3: Case Study Overview**
 - Present the AI application you selected. Explain how image processing is used in this application and the challenges it addresses.



IGNACIO - BSCS-4B 03

Case Study Overview

- Face recognition identifies or verifies a person's identity by analyzing facial features from an image or video.
- Image processing enhances facial data by detecting faces, adjusting for lighting, and extracting features like the eyes, nose, and mouth.
- These processes address challenges such as varying lighting conditions, different poses, and complex backgrounds that could reduce accuracy.
- By refining image quality, image processing ensures that face recognition systems remain reliable and accurate across diverse settings.



- **Slide 4: Your Image Processing Implementation**
 - Present the model you created for the simple problem. Explain how the model works and how it helps the AI system solve the problem.

IGNACIO - BSCS-4B

Image Processing Implementation

• The model processes images using several key steps to enhance facial data for reliable recognition and solve the **Problem in inaccurate detection under poor lighting conditions**. First, it converts images to **grayscale**, simplifying the data by reducing color complexity.

• **Gaussian blur** is applied to smooth out noise, while **edge detection** highlights important facial boundaries, making it easier to identify key features.

• **Histogram equalization** improves contrast, helping the model work better under different lighting conditions. For additional robustness, an illumination-invariant color space minimizes the effect of varying light intensities on the face.

• Finally, **face detection with Haar cascades** identifies the location of faces, ensuring the model focuses only on relevant areas for recognition. This combination of steps prepares facial images effectively, enhancing accuracy in face recognition by ensuring quality and consistency in the input data.

04

- **Slide 5: Conclusion**
 - Summarize the importance of effective image processing in AI and reflect on what you learned from the activity

IGNACIO - BSCS-4B

CONCLUSION

Effective image processing is essential in AI, as it optimizes raw data for analysis, enabling more accurate and reliable performance in applications like face recognition. It enhances image quality, reduces noise, and highlights key features, addressing challenges such as lighting variations and background complexity. From this activity, I learned how critical preprocessing is to ensure that AI systems interpret visual data accurately, as well as how specific techniques like grayscale conversion, Gaussian blur, and histogram equalization contribute to improved recognition in complex, real-world conditions.

05



Extension Activity:

- **Research an Emerging Form of Image Processing:**
 - Investigate a newer or emerging technique in image processing, such as deep learning-based image analysis.
 - Prepare a brief report or additional slide discussing its potential impact on future AI systems.

Emerging Form of Image Processing

Investigate a newer or emerging technique in image processing, such as deep learning- based image analysis.

- Deep learning-based image analysis, leveraging Convolutional Neural Networks (CNNs) and advanced architectures, is transforming how AI interprets visual data. Unlike traditional methods relying on manually crafted features, CNNs automatically learn patterns from vast datasets, adapting to variations like lighting, scale, and noise. This capability has proven especially effective in applications requiring complex analysis, such as autonomous driving, where deep learning helps vehicles understand surroundings, identify obstacles, and make real-time decisions (Khare et al., 2022; Liu et al., 2022).

Investigate a newer or emerging technique in image processing, such as deep learning- based image analysis.

- The adaptability and precision of deep learning models, especially in handling noisy or variable image data, position them as critical components in future AI developments. In medical imaging, for instance, these models already support early disease detection by analyzing CT and MRI scans with higher accuracy than previous methods. This technology also promises breakthroughs in security and surveillance, facilitating real-time threat detection and enhanced situational awareness (Hussain et al., 2023). As deep learning-based image analysis continues to evolve, it could become a foundational element in AI, enabling more enhanced, accurate, and context-sensitive image understanding across sectors from healthcare to urban planning.

06

Submission Instruction:

- **Create GitHub Repository:** Create a repository for the subject (e.g., CSST106-CS4D).
- **Sample Machine Problem** (Reference: <https://github.com/leeroyvincent/MIT-504-DEPATILLO>)
- **Submission Format:** Create a PowerPoint presentation based on the "Presentation Development" section and export it to a video with 5-10 seconds transition per slide. Include the content in GitHub using Markdown Language (.md).
- **Filename Format:** [SECTION-BERNARDINO-MP1] 4D-BERNARDINO-MP1
- **Penalties:** Inability to follow this instruction will result in a 5-point deduction for filename format and a 5-point deduction per day for late submission. Cheating and plagiarism will be penalized



Rubric for Machine Problem No. 1: Exploring the Role of Computer Vision and Image Processing in AI

Criteria	Excellent (10 points)	Good (8 points)	Fair (5 points)	Poor (2 points)
Research and understanding of AI	Comprehensive understanding of AI and the role of image processing.	Good understanding with minor gaps.	Basic understanding with some inaccuracies.	Poor or incomplete understanding.
Overview of Image Processing Techniques	Thorough exploration of three techniques with clear examples.	Adequate exploration with some examples.	Limited exploration with unclear examples.	Minimal or incorrect exploration of techniques.
Case Study Selection and Analysis	Well-chosen AI application with clear analysis of image processing techniques and their effectiveness.	Appropriate application chosen with some analysis.	Basic application chosen with limited analysis.	Poorly chosen application with little to no analysis.
Creation of Image Processing Model	Clear, accurate, and well-illustrated model that effectively addresses the problem.	Good model with minor inaccuracies or unclear illustrations.	Basic model with significant errors or poor illustrations.	Inaccurate or poorly constructed model.
Presentation Quality	Well-organized, visually appealing, and professional presentation with clear explanations.	Organized presentation with minor issues in visual appeal or clarity.	Somewhat organized but lacks visual appeal or clarity.	Disorganized presentation with poor visuals and unclear content.



Republic of the Philippines
Laguna State Polytechnic University
Province of Laguna





Republic of the Philippines
Laguna State Polytechnic University
Province of Laguna

