🧠 UNIT I – Introduction to Digital Image Processing (DIP)

📚 **Total Topics**:

1. What is Digital Image Processing?
2. Origin of DIP
3. Fields using DIP
4. Fundamental Steps in DIP
5. Components of an Image Processing System
6. Digital Image Fundamentals:
    - Visual Perception
    - Light & Electromagnetic Spectrum
    - Image Sensing and Acquisition
    - Image Sampling & Quantization
    - Pixel Relationships
    - Linear & Nonlinear Operations

🔷 1. What is Digital Image Processing?

👉 **Definition**:

Digital Image Processing (DIP) involves using **computers** to process digital images using **mathematical algorithms**.

🧠 **Example**:

- Editing a photo on your mobile app (like Snapseed or Photoshop).
- Removing blur, changing brightness = DIP in action!

🔷 2. Origin of DIP

📅 **History**:

- Started in the 1960s with NASA.
- First used to improve **moon photos** from space.
- Then used in **medical imaging**, **television**, and **military**.

🔷 3. Fields using DIP

👨‍⚕️ **Medical Field** – MRI & CT scan enhancement

🛰️ **Remote Sensing** – Satellite image analysis

🚗 **Automobiles** – Object detection in self-driving cars

📸 **Photography** – Noise removal, editing

👮 **Forensics** – Enhancing CCTV footage

🔷 4. Fundamental Steps in DIP

Here's a diagram-style explanation:

```
[Image Acquisition] → [Preprocessing] → [Segmentation] →
[Representation & Description] → [Recognition] → [Knowledge Base]
```

✔️ **Explanation**:

- **Image Acquisition** – Taking the image (camera or scanner)
- **Preprocessing** – Removing noise, adjusting brightness
- **Segmentation** – Dividing image into parts (like face, background)
- **Representation** – Describing parts (shapes, colors)
- **Recognition** – Identifying objects (like "This is a car")
- **Knowledge Base** – Already known data to help in processing

---

🔷 5. Components of Image Processing System

🖥️ Here's a simple diagram:

```
+--------------------+
| Image Sensor       | → Takes image
+--------------------+
          ↓
+--------------------+
| Digital Processor | → Enhances/Analyzes image
+--------------------+
          ↓
+--------------------+
| Storage            | → Saves image
+--------------------+
          ↓
+--------------------+
| Display            | → Shows image
+--------------------+
          ↓
+--------------------+
| Hardcopy Device    | → Prints image (optional)
+--------------------+
```

🖼️ Digital Image Fundamentals

🔷 6. Elements of Visual Perception

👁️ The **human eye** can only see a limited range of colors and brightness.

- Human eye is **more sensitive to brightness** than color.
- **Illusions** show how our brain interprets images differently.

🔷 7. Light and Electromagnetic Spectrum

Light = Electromagnetic wave 🌈

Type	Wavelength (nm)	Seen?
X-rays	< 10	❌
Ultraviolet (UV)	10–400	❌
Visible Light	**400–700**	✅
Infrared (IR)	700–1,000	❌

We only use the **visible part** of spectrum in DIP (Red, Green, Blue – RGB model).

🔷 8. Image Sensing and Acquisition

📸 Sensors (like phone camera) capture the image using **light** and convert it into **digital format**.

👀 Example: A scanner scanning a paper.

🔷 9. Image Sampling and Quantization

Sampling = selecting **pixels** from the image.

Quantization = assigning a **gray level** to each pixel.

🖼️ Example:

If you zoom into a photo, you see **pixels** (tiny squares).

Each pixel has a value like 0–255 for grayscale (0 = black, 255 = white).

📊 Diagram:

```
Original Image → Grid Overlay → Each grid gets a number → Digital image
```

🔷 10. Basic Relationship Between Pixels

- **Neighboring Pixels**: Pixels next to each other (4-neighbors or 8-neighbors)
- **Adjacency**: If two pixels share an edge, they are adjacent
- **Connectivity**: Determines shapes in the image

👀 Example: Black dot surrounded by white = edge of an object.

🔷 11. Linear vs Nonlinear Operations

🧮 **Linear Operation** = Output is directly proportional

📈 **Nonlinear Operation** = Output is not directly proportional

🛠️ **Examples**:

- Linear: Averaging filter (blurring)
- Nonlinear: Median filter (used for noise removal)

🧠 UNIT II – Intensity Transformations & Spatial Filtering (12 hrs)

📚 Topics Covered:

1. Background
2. Basic Intensity Transformation Functions
3. Histogram Processing
4. Fundamentals of Spatial Filtering
5. Smoothing Spatial Filters
6. Sharpening Spatial Filters
7. Combining Spatial Enhancement Methods

🔷 1. Background

Image enhancement is the process of making an image **better for visual interpretation**.

👉 Example: Increasing brightness, contrast, or removing noise in a photo to make it look better.

🔷 2. Basic Intensity Transformation Functions

These change the **pixel intensity** values of an image.

✅ Types:

1. **Linear Transformation**
    - Contrast stretching: Makes darks darker and lights lighter.
    - Formula: `s = a * r + b`
    - Where `r = input pixel`, `s = output pixel`, `a`, `b` = constants
2. **Log Transformation**
    - Enhances dark regions
    - Formula: `s = c * log(1 + r)`
3. **Power-law (Gamma) Transformation**
    - Used in printing, TV, etc.
    - Formula: `s = c * r^γ`

📊 Diagram:

```
Linear:      Log:          Gamma < 1:
|            |              |
|           ___             /
|           /              /
|___       /                |
   \     /                \
    \___/                  \___
```

🔷 3. Histogram Processing

🖼️ **Histogram**: A graph showing how many pixels have a certain intensity (brightness level).

✅ Types:

- **Histogram Equalization**:
    - Spreads intensity values equally to enhance contrast.
    - Best for **dark/blurry images**.

📊 Example:

Before: Many pixels are dark (clustered).

After Equalization: Pixel intensities are spread out more evenly.

🔷 4. Fundamentals of Spatial Filtering

📌 Spatial filtering means modifying a pixel using **neighboring pixels**.

We use a small matrix called **Kernel** or **Mask** (like 3×3 or 5×5).

Example:

```
Original Image (3×3):
[10 20 30]
[40 50 60]
[70 80 90]

Mask (3×3 Average Filter):
[1/9 1/9 1/9]
[1/9 1/9 1/9]
[1/9 1/9 1/9]
```

👉 Multiply & add = gives smoothed center value.

🔷 5. Smoothing Spatial Filters

🧹 Removes noise, softens image.

✅ Types:

- **Average Filter** (Mean Filter)
- **Gaussian Filter**

📊 **Effect**:

Edges become smooth, noise is reduced. But image may get blurred.

🔷 6. Sharpening Spatial Filters

🗡️ Enhances edges and fine details.

✅ Types:

- **Laplacian Filter**
- **High-pass Filter**
- **Gradient Filters** (Sobel, Prewitt)

📊 **Effect**:

Edges pop out, image becomes sharper and more detailed.

🔷 7. Combining Spatial Enhancement Methods

🧠 You can **combine smoothing + sharpening** to get the best result.

📌 Example Flow:

1. First remove noise using smoothing.
2. Then sharpen edges using Laplacian or Sobel.
3. Use histogram equalization to enhance contrast.

🧠 UNIT III – Image Restoration and Reconstruction (12 hrs)

📚 Topics Covered:

1. Model of Image Degradation/Restoration

2. Noise Models
3. Restoration with Noise Only (Spatial Filtering)
4. Periodic Noise Reduction using Frequency Domain
5. Estimating the Degradation Function
6. Inverse Filtering
7. Minimum Mean Square Error (MMSE) Filtering
8. Constrained Least Squares Filtering
9. Geometric Mean Filter

🔷 1. Model of Image Degradation / Restoration Process

📉 When an image gets **blurred or noisy**, we need to restore it.

🧪 Model Equation:

```
g(x, y) = h(x, y) * f(x, y) + η(x, y)
```

Where:

- `g(x, y)` = Degraded image
- `f(x, y)` = Original image
- `h(x, y)` = Degradation function (like blur)
- `η(x, y)` = Noise
- = Convolution

🎯 Goal: Find `f(x, y)` from `g(x, y)` (i.e., remove noise/blur)

🔷 2. Noise Models

Noise = Random variation in brightness or color.

✅ Common Types:

- **Gaussian Noise** – Looks like grainy fog

- **Salt-and-Pepper Noise** – Random white/black dots
- **Uniform Noise** – Constant random range

📸 **Example**: Old CCTV footage or poor-quality scans.

🔷 3. Restoration in the Presence of Noise Only (Spatial Filtering)

✅ Methods:

- **Mean Filter**: Smooths the image, reduces random noise
- **Median Filter**: Removes salt-and-pepper noise (replaces center pixel with middle value)

🧠 Use **Median Filter** if many black/white dots are in image.

🔷 4. Periodic Noise Reduction (Frequency Domain Filtering)

📈 Periodic noise appears as patterns (waves or ripples).

- Use **Fourier Transform** to go to frequency domain
- Remove noise peaks
- Then apply **Inverse Fourier Transform**

🌀 Think of it as cleaning the image using frequency cleaning.

🔷 5. Estimating the Degradation Function

📷 Sometimes we don't know how the image got degraded.

We estimate it by:

- **Observing multiple frames**
- **Using known objects (like a square/line in test images)**

🔷 6. Inverse Filtering

🧼 Try to **undo** the effect of degradation.

Formula:

```
F(u, v) = G(u, v) / H(u, v)
```

Where:

- `F` = Original image (frequency domain)
- `G` = Degraded image
- `H` = Degradation function

❗ Problem: If `H(u, v)` is close to 0, the result becomes unstable (too bright/dark)

🔷 7. Minimum Mean Square Error (MMSE) Filtering

📊 This method tries to **minimize error** between original & restored image.

Better than inverse filtering when noise is strong.

🔷 8. Constrained Least Squares Filtering

🧠 Adds constraints to make restoration more realistic (like not over-sharpening).

Used when we know little about the image but want smooth output.

🔷 9. Geometric Mean Filter

💡 Combines multiple filters like:

- Inverse filtering
- Wiener filter
- Constrained least squares

It gives **balanced restoration**.

🧠 **Summary**:

Method	Used For
Median Filter	Salt-and-pepper noise
Mean Filter	General smoothing
Inverse Filter	Known blur + low noise

MMSE Filter	Unknown noise, better quality
Frequency Filter	Periodic (wave) noise
Constrained LS Filter	Smooth results with flexibility
Geometric Mean Filter	Mix of filters for best result

🧠 UNIT IV – Image Compression (12 hrs)

📚 Topics Covered:

1. Fundamentals of Image Compression
2. Image Compression Models
3. Containers and Compression Standards
4. Some Basic Compression Methods
5. Watermarking
6. Image Formats

🔷 1. Fundamentals of Image Compression

👉 **Why compression?**

To reduce the **size of image files** while keeping good quality.

🧠 Example:

- Original JPG: 5 MB
- Compressed JPG: 500 KB
- Looks almost the same, but smaller size = easier to store/share.

🔷 2. Image Compression Models

Here's a basic block diagram:

```
[Image] → [Encoder] → [Compressed Image]
          ↓
```

```
[Decoder] ← [Compressed Image]
```

✔ **Encoder**: Removes redundancy

✔ **Decoder**: Rebuilds the image

✅ Types of Redundancy Removed:

1. **Spatial Redundancy** – Repeating pixels in neighboring areas
2. **Spectral Redundancy** – Repeating info in color channels
3. **Temporal Redundancy** – In video, repeated frames

🔷 3. Containers and Compression Standards

📦 **Container**: File type that stores compressed image/video/audio

Examples:

- **Image**: JPEG (.jpg), PNG, TIFF, BMP
- **Video**: MP4, AVI
- **Others**: PDF (contains images & text)

📏 Compression Standards:

- **JPEG** – Most common, lossy
- **PNG** – Lossless, supports transparency
- **TIFF** – High quality, used in printing
- **GIF** – Animated image format

🔷 4. Some Basic Compression Methods

✅ (i) Lossless Compression – No data lost

Examples:

- Run-Length Encoding (RLE)
- Huffman Coding

- LZW (used in GIF, PNG)

✅ (ii) Lossy Compression – Some data removed, quality may reduce

Examples:

- JPEG
- MPEG

🧠 **Example** of Run-Length Encoding (RLE):

Original: `AAAABBBCCDAA`

Compressed: `4A3B2C1D2A`

🔷 5. Watermarking

🛡️ Adding hidden text/logo to protect image ownership.

🧠 Example:

- A photographer adds a **faint name or logo** on an image.
- It's not easily removable without quality loss.

Types:

- **Visible Watermark** – You can see it
- **Invisible Watermark** – Hidden, but traceable

🔷 6. Image Formats

Format	Type	Features
JPG	Lossy	Small size, good quality
PNG	Lossless	Supports transparency
BMP	Raw	Large size, no compression
GIF	Lossless	Supports animation
TIFF	Lossless	High quality, used in print

🧠 UNIT V – Image Segmentation (12 hrs)

📚 Topics Covered:

1. Fundamentals of Image Segmentation
2. Point, Line, and Edge Detection
3. Thresholding
4. Region-Based Segmentation

🔷 1. Fundamentals of Image Segmentation

👉 **What is segmentation?**

Dividing an image into **meaningful parts or regions**.

🧠 Example:

In a photo of a face, segmentation separates the **eyes, nose, lips, etc.**

🎯 Goal: Identify and isolate objects in an image.

🔷 2. Point, Line, and Edge Detection

These are **basic features** in any image. Let's break them down:

✅ Point Detection:

- Detects **small isolated points** (like a dot)
- Uses a **mask** to highlight if any pixel differs a lot from neighbors

✅ Line Detection:

- Detects **straight lines** in different directions (horizontal, vertical, diagonal)
- Uses **directional masks**

Example:

```
Horizontal line mask:
[-1 -1 -1]
```

```
[ 2  2  2]
[-1 -1 -1]
```

✅ Edge Detection:

- Detects **boundaries** between different objects or colors

🧠 Used in:

- Face detection
- Object counting
- Road detection in self-driving cars

📐 Common Edge Detectors:

- **Sobel**
- **Prewitt**
- **Canny** (most accurate)

📊 Edge = Place where intensity changes rapidly

Example: from black to white pixel suddenly

🔷 3. Thresholding

This is the **easiest segmentation method**.

👉 Convert a grayscale image to **black & white** (binary) using a **threshold**.

✅ How?

- If pixel > threshold → make it white (1)
- If pixel < threshold → make it black (0)

🧠 Example:

```
Threshold = 100
```

```
Pixel = 120 → white
Pixel = 90 → black
```

✅ Types:

- **Global Thresholding** – One value for the whole image
- **Adaptive Thresholding** – Varies threshold in different regions

---

🔷 4. Region-Based Segmentation

Instead of checking edges, this method **groups similar pixels together**.

✅ Techniques:

- **Region Growing**:
  - Start from a seed pixel
  - Add neighboring pixels if they are similar (e.g., same color)
- **Region Splitting and Merging**:
  - Split image into blocks
  - Merge if blocks are similar

🧠 Example: In a forest image, this method will separate trees, sky, and river based on color similarity.

📌 Bonus Tip for Exam:

When writing theory:

- **Start with definition**
- **Draw a diagram** (edge detection masks, threshold chart)
- **Explain steps or algorithm**
- **Give a real-life example** (face detection, road maps, fingerprint scan)

🔥 That's it Machi! **Full syllabus completed!**

Need revision notes, diagrams, or expected 2/5/10-mark questions for any unit?