

Projects 2025-2026

1. Digital Imaging: “The Evolution of Digital Images”

 **Objective:** Create a **visual timeline** showcasing the evolution of digital imaging, from early photography to AI-generated images.

 **Tools:** Canva, Google Slides, PowerPoint, Figma.

 **Steps:**

- Research major milestones (e.g., **early digital cameras, GIFs, JPEG compression, AI-generated art**).
- Create a **timeline with images** showing technological shifts.
- Annotate key changes (e.g., **color depth, resolution improvements**).
- **Bonus:** Compare an **old vs. new** digital image and describe quality differences.

2. Image Digitization: “Digitizing a Historical Document”

 **Objective:** Convert a **handwritten** or **printed** historical text into a **digital version** and annotate it.

 **Tools:** Google Docs (OCR), Adobe Scan, Tesseract OCR (optional for deeper analysis), or Google Colab (for more advanced work).

 **Steps:**

- Find a **short historical text** (e.g., a manuscript, letter, or book page).
- Scan or take a photo of the document.
- Use **OCR tools** to convert it into text.
- Compare the **accuracy** of the OCR result with the original.
- Add **annotations** explaining **errors, missing characters, and digitization challenges**.

☒ **Extra Challenge:** Compare **two different digitization methods** (e.g., using OCR vs. manual transcription).

3. Image Segmentation: “Spot the Difference in Art”

 **Objective:** Apply basic **image segmentation concepts** to separate foreground/background in famous paintings.

 **Tools:** Canva (for manual segmentation), GIMP/photopea.com (basic thresholding), Remove.bg (AI-based segmentation), or Google Colab (for more advanced work).

 **Steps:**

- Select a **famous painting** (e.g., a **Renaissance portrait** or **Byzantine icon**).
- Use **simple tools** to **separate the subject from the background**.
- Analyze how segmentation **affects the meaning** (e.g., does isolating a figure change how we interpret the artwork?).
- Compare **manual vs. automatic segmentation results**.

☑ **Extra Challenge:** Discuss how segmentation is used in **restoration, AI art generation, or historical analysis**.

4. Image Compression: “How Low Can You Go?”

🔗 **Objective:** Explore how **image compression affects quality** by reducing file sizes.

🔗 **Tools:** TinyPNG, Squoosh.app, Photoshop/GIMP/photopea.com, or Google Colab (for mode advanced work).

🔗 **Steps:**

- Select a **high-quality image** (e.g., a historical photograph or artwork).
- Compress it using **different levels of JPEG/PNG compression**.
- Compare the results and describe:
 - What artifacts appear? (e.g., **blurriness, blockiness**).
 - At what point does it become **unusable**?
 - How does **compression** affect **archiving digital heritage**?

☑ **Extra Challenge:** Discuss **lossy vs. lossless** compression in cultural heritage digitization.

5. Image Processing: “Restoring a Damaged Photo”

🔗 **Objective:** Simulate basic **image restoration** techniques on an old or degraded photo.

🔗 **Tools:** Photoshop/photopea.com, GIMP (Clone Tool, Noise Reduction), or Google Colab (for mode advanced work).

🔗 **Steps:**

- Find an old **damaged photo** (e.g., historical archive image).
- Apply **basic enhancement**:
 - Adjust brightness/contrast.
 - Remove small noise or scratches.
 - Compare before/after results.
- Discuss the **ethics of digital restoration** (e.g., should we **enhance historical photos**, or does it alter authenticity?).

6. Image Metadata & Cultural Heritage: “What’s Hiding in an Image?”

🔗 **Objective:** Analyze metadata stored in images and discuss its role in **digital heritage preservation**.

🔑 **Tools:** ExifTool, Metadata2Go.

📌 **Steps:**

- Select a **historical digital photograph** (or take your own).
- Use **metadata tools** to extract information (**camera model, resolution, location**).
- Discuss:
 - How can metadata **help historians or archivists**?
 - What **privacy risks** does metadata create?

☑ **Extra Challenge:** Discuss how **metadata loss** in compression can affect historical image authenticity.

7. Colorization of Historical Images: “Black & White to Color”

📌 **Objective:** Explore how **AI-generated colorization** changes our perception of history.

🔑 **Tools:** DeepAI Colorizer, MyHeritage Colorization, Photoshop.

📌 **Steps:**

- Take a **historical black-and-white photo**.
- Use **AI tools** to colorize it.
- Compare AI-generated vs. manually adjusted results.
- Discuss: **Does colorization make history feel more “real” or distort authenticity?**

☑ **Extra Challenge:** Research how colorization is used in **film restoration**.

8. Image Segmentation Techniques

Image segmentation is a fundamental task in image processing and computer vision, where an image is partitioned into meaningful regions. It is widely used in medical imaging, autonomous driving, satellite image analysis, object detection, and more. Traditional segmentation methods have evolved into AI-based techniques, significantly improving accuracy and automation.

This project will provide an in-depth review of various image segmentation techniques, comparing their effectiveness, advantages, and limitations across different applications.

The primary goals of this project are:

- To understand the fundamentals of image segmentation and its significance.
- To review and compare traditional and deep learning-based segmentation techniques.
- To analyze performance metrics used to evaluate segmentation models.
- To identify research gaps and future challenges in the field.

The optional goals of this project are:

- To implement indicative image segmentation techniques and test their performance on known datasets

Key areas of research:

A. Traditional segmentation methods

Traditional segmentation methods rely on mathematical and statistical techniques to segment images without deep learning.

1. Thresholding Methods
 - Uses intensity levels to separate objects from the background.
 - Examples: Otsu's method, Adaptive Thresholding.
2. Region-Based Segmentation
 - Divides an image into regions based on pixel similarities.
 - Techniques: Region Growing, Watershed Algorithm.
3. Edge-Based Segmentation
 - Detects object boundaries using gradients and edges.
 - Examples: Canny Edge Detector, Sobel Operator, Laplacian of Gaussian (LoG).

B. Deep learning methods

With the rise of AI, deep learning methods now dominate image segmentation.

4. Clustering-Based Segmentation
 - Uses unsupervised learning to group similar pixels.
 - Examples: K-Means Clustering, Fuzzy C-Means (FCM).
5. Deep Learning-Based Segmentation
 - Uses Convolutional Neural Networks (CNNs) for highly accurate segmentation.
 - Models include:
 - Fully Convolutional Networks (FCN) – Converts traditional CNNs into segmentation models.
 - U-Net – A popular architecture in medical image segmentation.
 - Mask R-CNN – Used for instance segmentation in autonomous driving and object detection.

Methodology:

1. Literature Collection:
 - Review research papers from IEEE Xplore, Springer, ScienceDirect, Google Scholar.
2. Comparison of Different Techniques:
 - Identify strengths, weaknesses, and suitable applications for each method.
3. Analysis of Recent Trends:
 - Explore the shift from traditional methods to deep learning-based segmentation.
4. (Optional) Implementation of indicative techniques in Python with open CV/tensorflow/pytorch etc.
5. Conclusion & Research Gaps:
 - Identify challenges such as dataset limitations, real-time performance, and model generalization.

Expected Outcomes:

1. A detailed comparison of the different techniques with real-world applications.
2. A discussion on which methods work best for different use cases
3. Identification of research challenges and suggestions for future improvements.
4. Well-structured report and presentation summarizing key findings.

9. Optical Character Recognition (OCR) Systems

Optical Character Recognition (OCR) is a crucial technology that enables machines to convert printed or handwritten text into digital formats. It plays a vital role in document digitization, automated data entry, assistive technology for the visually impaired, and historical document preservation. OCR systems have evolved significantly, transitioning from traditional rule-based techniques to modern deep learning approaches that achieve high accuracy.

This project aims to explore different OCR techniques, their strengths and limitations, and their applications in various domains.

The primary goals of this project are:

To understand OCR technology and its evolution.

- To review and compare traditional and deep learning-based OCR techniques.
- To analyze challenges in OCR systems such as noise, handwriting recognition, and multilingual text processing.
- To explore real-world applications of OCR technology.
- To identify research gaps and future directions in OCR development.

The optional goals of this project are:

- To implement indicative OCR techniques and test their performance on known datasets

Key areas of research:

A. Traditional OCR Techniques

Earlier OCR systems relied on handcrafted feature extraction and rule-based pattern matching.

1. Template Matching (Early OCR Systems)
 - Uses predefined character templates for recognition.
 - Example: Used in early postal address and bank check processing.
2. Feature-Based OCR (Statistical Approaches)
 - Extracts features such as edges, strokes, and character shapes to classify text.
 - Techniques: Histogram Projection, Zoning, Structural Analysis.

B. Machine Learning-Based OCR Techniques

With the rise of AI, OCR systems started using statistical models for better recognition accuracy.

3. Support Vector Machine (SVM) & K-Nearest Neighbors (k-NN)
 - Used for character classification based on extracted features.
 - Example: License plate recognition in early AI-based OCR.
4. Hidden Markov Models (HMM) for Handwriting Recognition
 - Models sequences of characters for recognizing cursive handwriting.
 - Used in postal mail sorting and bank cheque processing.

C. Deep Learning-Based OCR Techniques

Recent advancements in Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) have significantly improved OCR accuracy.

5. Convolutional Neural Networks (CNNs) for OCR
 - Used for character recognition and feature extraction.
 - Models: LeNet, ResNet, and VGG for digit classification.
6. Recurrent Neural Networks (RNNs) & Long Short-Term Memory (LSTM)
 - Used for sequence-based OCR tasks, such as recognizing handwritten text in different styles.
 - Example: Google's Tesseract OCR v4 uses LSTM networks for text recognition.
7. Transformer-Based OCR (Vision Transformers - ViTs)
 - Uses attention mechanisms to detect characters in complex layouts.
 - Example: OCR in scanned historical manuscripts.

Methodology:

1. Literature Collection:
 - Review research papers from IEEE Xplore, Springer, ScienceDirect, Google Scholar.
2. Comparison of different Techniques:
 - Identify strengths, weaknesses, and suitable applications for each method.
3. Analysis of Recent Trends:
 - Explore the shift from traditional methods to deep learning-based OCR.
4. (Optional) Implementation of indicative techniques in Python with open CV/tensorflow/pytorch etc.
5. Conclusion & Research Gaps:

- Identify challenges such as dataset limitations, real-time performance, and model generalization.

Expected Outcomes:

1. A detailed comparison of the different techniques with real-world applications.
2. A discussion on which methods work best for different use cases
3. Identification of research challenges and suggestions for future improvements.
4. Well-structured report and presentation summarizing key findings.

10. Face Detection and Recognition

Face detection and recognition are critical tasks in computer vision, widely used in security systems, biometric authentication, surveillance, social media, and human-computer interaction. Over the years, various techniques have evolved from simple rule-based methods to advanced deep learning models that achieve high accuracy and real-time performance.

This project aims to explore the different approaches, challenges, and advancements in face detection and recognition, comparing traditional methods with modern AI-driven techniques.

The primary goals of this project are:

- Understanding the fundamentals of face detection and recognition.
- Reviewing and comparing traditional, machine learning-based, and deep learning-based methods.
- Exploring real-world applications of face recognition systems.
- Analyzing challenges such as pose variation, occlusion, and adversarial attacks.
- Identifying research gaps and future trends in face detection and recognition.

The optional goals of this project are:

- To implement indicative techniques and test their performance on known datasets

Key areas of research:

A. Traditional Face Detection Methods

Face detection aims to locate human faces in an image or video. Earlier methods were based on handcrafted features and statistical models.

1. Haar Cascade Classifier (Viola-Jones Algorithm)
 - A classical method that uses Haar-like features and an Adaboost classifier.
2. Histogram of Oriented Gradients (HOG) + Support Vector Machine (SVM)
 - HOG extracts edge and texture information, while SVM is used for classification.

B. Machine Learning-Based Face Detection and Recognition

With the rise of machine learning, new methods improved upon traditional techniques.

3. Local Binary Patterns (LBP) for Face Detection
 - Uses local pixel intensity variations to detect faces.
4. Eigenfaces & Fisherfaces (Principal Component Analysis - PCA)
 - Reduces dimensionality and extracts important facial features for recognition.

C. Deep Learning-Based Face Detection and Recognition

Modern face detection and recognition systems are powered by deep learning techniques, achieving state-of-the-art accuracy.

5. Convolutional Neural Networks (CNNs) for Face Detection
 - YOLO-Face, MTCNN (Multi-Task Cascaded CNN), RetinaFace are popular CNN-based detectors.
6. Deep Learning for Face Recognition
 - FaceNet, DeepFace, OpenFace – Advanced models that use deep learning for facial representation.
 - Siamese Networks – Compares two faces using a distance metric for verification.

Methodology:

1. Literature Collection:
 - Review research papers from IEEE Xplore, Springer, ScienceDirect, Google Scholar.
2. Comparison of different Techniques:
 - Identify strengths, weaknesses, and suitable applications for each method.
3. Analysis of Recent Trends:
 - Explore the shift from traditional methods to deep learning-based.

4. (Optional) Implementation of indicative techniques in Python with open CV/tensorflow/pytorch etc.
5. Conclusion & Research Gaps:
 - Identify challenges such as dataset limitations, real-time performance, and model generalization.

Expected Outcomes:

1. A detailed comparison of different techniques with real-world applications.
2. A discussion on which methods work best for different use cases
3. Identification of research challenges and suggestions for future improvements.
4. Well-structured report and presentation summarizing key findings.

11. Super-Resolution in Image Processing

Super-Resolution (SR) in image processing is a technique used to enhance the resolution of an image by reconstructing high-frequency details that are lost in low-resolution (LR) images. It has gained significant importance in fields such as medical imaging, satellite imagery, surveillance, video processing, and computer vision.

This project aims to explore the different approaches, challenges, and advancements in super-resolution techniques, comparing traditional methods with deep learning-based solutions and their applications in various industries.

The primary goals of this project are:

- Understanding the fundamentals of Super-Resolution in image processing.
- Reviewing and comparing traditional, machine learning-based, and deep learning-based SR methods.
- Exploring real-world applications of super-resolution in various fields.
- Analyzing challenges such as noise amplification, computational cost, and real-time SR.
- Identifying research gaps and future trends in super-resolution techniques.

The optional goals of this project are:

- To implement indicative techniques and test their performance on known datasets

Key areas of research:

A. Traditional Super-Resolution Methods

Early Super-Resolution techniques relied on signal processing and mathematical models to enhance image resolution.

1. Interpolation-Based Methods
 - Examples: Bicubic Interpolation, Bilinear Interpolation, Nearest Neighbor
2. Reconstruction-Based Methods
 - Uses multiple low-resolution images to reconstruct a high-resolution image using Bayesian inference and edge priors.
 - Techniques: Iterative Back Projection (IBP), Maximum a Posteriori (MAP), and Sparse Representation-based SR.

B. Machine Learning-Based Super-Resolution

With the advancement of machine learning, SR techniques have improved in efficiency and accuracy.

3. Sparse Coding-Based Super-Resolution
 - Uses dictionary learning to map low-resolution patches to high-resolution patches.
4. Random Forest Regression-Based Super-Resolution
 - Uses decision trees to predict high-resolution patches from low-resolution inputs.

C. Deep Learning-Based Super-Resolution

Recent advances in deep learning have revolutionized super-resolution, achieving state-of-the-art results.

5. Super-Resolution Convolutional Neural Networks (SRCNN)
 - One of the first deep learning-based SR models using a three-layer CNN.
6. Generative Adversarial Networks (GANs) for Super-Resolution
 - Example: SRGAN (Super-Resolution GAN)
 - Uses a generator and a discriminator to create realistic high-resolution images.
7. Transformers for Super-Resolution
 - Example: SwinIR (Swin Transformer for Image Restoration)
 - Uses self-attention mechanisms to enhance images while preserving textures.

Methodology:

6. Literature Collection:

- Review research papers from IEEE Xplore, Springer, ScienceDirect, Google Scholar.
- 7. Comparison of different Techniques:
 - Identify strengths, weaknesses, and suitable applications for each method.
- 8. Analysis of Recent Trends:
 - Explore the shift from traditional methods to deep learning-based.
- 9. (Optional) Implementation of indicative techniques in Python with open CV/tensorflow/pytorch etc.
- 10. Conclusion & Research Gaps:
 - Identify challenges such as dataset limitations, real-time performance, and model generalization.

Expected Outcomes:

1. A detailed comparison of different techniques with real-world applications.
2. A discussion on which methods work best for different use cases
3. Identification of research challenges and suggestions for future improvements.
4. Well-structured report and presentation summarizing key findings.

12. Deep Learning in Image Classification

Image classification is one of the most fundamental tasks in computer vision, where an algorithm assigns a label or category to an image based on its content. Traditional computer vision methods relied on handcrafted features, but deep learning has revolutionized the field by learning features directly from data.

Deep learning models, particularly Convolutional Neural Networks (CNNs), have achieved state-of-the-art performance in image classification, outperforming traditional machine learning techniques. This project explores different CNN architectures, their evolution, challenges, and real-world applications.

The primary goals of this project are:

- Understand the fundamentals of deep learning-based image classification.
- Review and compare CNN architectures such as AlexNet, VGG, ResNet, EfficientNet, and Transformers.
- Analyze real-world applications of deep learning in image classification.
- Discuss challenges such as overfitting, interpretability, and computational cost.
- Identify research gaps and future trends in deep learning-based image classification.

The optional goals of this project are:

- To implement indicative techniques and test their performance on known datasets

Key areas of research:

A. Traditional Image Classification Methods

Before deep learning, image classification relied on:

1. Feature Extraction + Classifiers
 - Handcrafted features such as SIFT, HOG, LBP were used for pattern recognition.
 - Classifiers like Support Vector Machines (SVM), Random Forest, and k-Nearest Neighbors (k-NN) were used for classification.

B. Deep Learning-Based Image Classification

With the rise of deep learning, CNNs (Convolutional Neural Networks) became the dominant approach for image classification.

1. Convolutional Neural Networks (CNNs) - Core Concept

CNNs automatically learn features (edges, textures, shapes) from images through convolutional layers, pooling layers, and fully connected layers.

C. Evolution of CNN Architectures

1. AlexNet (2012)

- First deep CNN to win the ImageNet competition, demonstrating deep learning's superiority.
- Used ReLU activation, dropout, and data augmentation to improve performance.

2. VGG (2014)

- Introduced deeper networks (VGG-16, VGG-19) with small 3×3 filters.

3. ResNet (2015) - Deep Networks with Skip Connections

- Residual learning allowed for extremely deep networks (up to 152 layers) without vanishing gradients.
- Used in medical imaging, autonomous vehicles, and industrial inspection.

4. *EfficientNet (2019) - Scaling Networks Efficiently*

- Introduced compound scaling to balance depth, width, and resolution.
- Achieves better accuracy with fewer parameters, making it suitable for mobile devices and edge computing.

5. *Vision Transformers (ViTs) (2020-Present)*

- Unlike CNNs, ViTs use self-attention mechanisms to process images as a sequence of patches.
- Achieves state-of-the-art performance on large datasets but requires significant computing power.

Methodology:

1. Literature Collection:
 - Review research papers from IEEE Xplore, Springer, ScienceDirect, Google Scholar.
2. Comparison of different Techniques:
 - Identify strengths, weaknesses, and suitable applications for each method.
3. Analysis of Recent Trends:
 - Explore the shift from traditional methods to deep learning-based.
4. (Optional) Implementation of indicative techniques in Python with open CV/tensorflow/pytorch etc.
5. Conclusion & Research Gaps:
 - Identify challenges such as dataset limitations, real-time performance, and model generalization.

Expected Outcomes:

1. A detailed comparison of different techniques with real-world applications.
2. A discussion on which methods work best for different use cases
3. Identification of research challenges and suggestions for future improvements.
4. Well-structured report and presentation summarizing key findings.

13. Image Compression: “A 30-Year Decline in File Size”

Objective:

Investigate how image compression techniques (JPEG, PNG, WebP, HEIC, AVIF) have evolved over the last three decades, and create a comparative visual report showing how identical images change under each method.

Tools:

Squoosh.app, TinyPNG, ImageMagick, Google Colab, Canva/Figma.

Steps:

1. **Historical Research:**

- Create a timeline of compression formats (JPEG 1992 → AVIF 2019).
- Identify key innovations (DCT, entropy coding, wavelets, neural compression).

2. **Dataset Creation:**

- Select 5–10 reference images (portrait, landscape, artwork, text-heavy image).

3. **Experimental Work:**

- Compress each image across 5 formats at various levels (10–90%).
- Log file size, compression ratio, artifacts (blocking, ringing, color banding).

4. **Quality Evaluation:**

- Compute PSNR and SSIM in Colab.
- Annotate where each format collapses or excels.

5. **Final Output:**

- A visual grid showing comparisons.
- Summary explaining which formats are best for archival, web, or mobile imaging.

14. Cultural Heritage Digitization: “Comparing OCR on Ancient Scripts”

Objective:

Evaluate how modern OCR tools handle historical scripts (e.g., Greek polytonic, medieval Latin, handwritten letters) and produce an error-taxonomy of OCR failures.

Tools:

Google OCR, Tesseract OCR, Adobe Scan

Steps:

1. Corpus Creation:

- Gather 5 document samples (printed + handwritten).
- Ensure variety (fonts, noise, stains, lighting).

2. Multi-Tool OCR Testing:

- Run each image through Google OCR or Tesseract or another tool.
- Export transcriptions + confidence scores.

3. Error Analysis:

- Identify typical OCR mistakes (diacritics, ligatures, long-s, noise misreadings).
- Categorize errors: segmentation errors, character errors, spacing errors.

4. Deep Research Add-On:

- Investigate why certain scripts fail (e.g., OCR trained on modern fonts).
- Search academic papers on OCR for ancient manuscripts.

5. Final Output:

- A comparative table + annotated examples.
- A recommended workflow for digitizing heritage texts.

15. Image Enhancement: "From Old to Modern: A Guided Restoration Pipeline"

Objective:

Create a full workflow to restore a degraded historical image using both classical and AI-based enhancement, documenting every step and evaluating the impact.

Tools:

GIMP/Photopea, Remini (AI enhancement), Colab

Steps:

1. Image Selection:

- Choose an old or damaged photograph (noise, cracks, low resolution).

2. Manual Restoration:

- Use brightness/contrast correction.
- Remove dust and scratches using clone/heal tools.
- Reduce noise using Gaussian or median filters.

3. **AI Restoration:**

- Apply AI upscaling and denoising tools.
- Compare AI vs. manual restoration.

4. **Critical Evaluation:**

- Analyze risks: hallucinated details, over-smoothing, historical distortion.
- Document both positive and negative changes.

5. **Final Output:**

- Before/after gallery.
- A written reflection on ethics and authenticity.

16. **Segmentation & Art Interpretation: “Deconstructing Meaning Through Foreground Extraction”**

Objective:

Use classical + AI segmentation techniques to isolate elements in artworks and analyze how removing or highlighting components changes interpretation.

Tools:

Remove.bg, Canva, Photopea, Colab

Steps:

1. **Artwork Selection:**

- Choose 3 artworks (e.g., Renaissance portrait, icon, modern minimalist).

2. **Segmentation Methods:**

- Manual segmentation (lasso/cut-out).
- AI segmentation (remove.bg).
- Segmentation in Colab.

3. **Interpretation Analysis:**

- Observe how composition changes when background is removed.
- Evaluate what is lost/gained visually and symbolically.

4. **Final Output:**

- Segmentation comparison chart.
- 1–2 page interpretative analysis per artwork.

17. Super-Resolution: “Rebuilding History in High Definition”

Objective:

Experiment with upscaling low-resolution historical images using classical interpolation and deep learning, and comparing impact on detail, authenticity, and information loss.

Tools: Colab, Upscale.media, GIMP.

Steps:

1. **Image Selection:**

- Pick 3 low-resolution images (scanned photos, newspaper clippings, early-web images).

2. **Baseline Enhancement:**

- Bicubic, bilinear, nearest-neighbor interpolation.

3. **Deep Learning Enhancement:**

- Use ESRGAN or online SR tools to upscale images.

4. **Artifact & Authenticity Study:**

- Detect hallucinated details, false textures, reconstructed areas.
- Evaluate usefulness for archives, research, and museum digitization.

5. **Final Output:**

- Quality comparison board.
- Report on what SR adds