Swami Keshvanand Institute of Technology, Management &Gramothan, Ramnagaria, Jagatpura, Jaipur-302017, INDIA
Approved by AICTE, Ministry of HRD, Government of India
Recognized by UGC under Section 2(f) of the UGC Act, 1956
Tel.: +91-0141- 3500300 Fax: +91-0141-2759555
E-mail: info@skit.ac.in Web: www.skit.ac.in

Faculty Kit

Here is a fully customized **Faculty Kit** for your **Image Segmentation and Recognition**, adapted precisely in the same format and length as the original reference, but focused on your project's objectives, tech stack, and deliverables:

Faculty Kit: Image Segmentation and Classification for Personal Identification System

1. Requirements Specification Evaluation

Objective: Ensure the system requirements are clearly defined, technically sound, and focused on biometric recognition.

- **Clarity**: Requirements around image preprocessing, segmentation techniques, classification flow, and facial data storage must be documented clearly.
- **Feasibility**: Key assumptions (e.g., grayscale conversion, edge detection models, preprocessing limits) must be justified.
- **Team Understanding:** All members must clearly understand their assigned modules (e.g., segmentation algorithm, training classifier, face database).
- **Presentation Quality**: Must visually describe the need for robust recognition under noise, occlusions, and real-world conditions.

2. Technology Familiarization

Objective: Evaluate the team's understanding of their chosen tech stack: Python, OpenCV, TensorFlow/Keras, Flask, SQLite.

- **Presentation**: Each member explains the component they handled (e.g., CNN architecture, image filters, Flask API, face dataset setup).
- **Quiz**: Faculty may ask about segmentation logic (thresholding, watershed), CNN layers, loss functions, model evaluation metrics, etc.

3. Database Design Evaluation

Objective: Assess the structure for storing user images, extracted features, model logs, and classification results.

- **Clarity & Redundancy**: All table fields (e.g., user ID, label, feature vector, accuracy) must be well-structured and normalized.
- **Backup & Recovery**: Project must explain how images and metadata are backed up, with version control or export methods.

4. High-Level and Detailed Design Evaluation

Objective: Review architectural and logical designs for all system components.

- **Requirements Coverage**: Designs should span all modules: face detection, segmentation, classification, and result generation.
- **Pseudocode & Flowcharts**: Must describe flows like facial image input, processing through CNN, and label prediction.
- **Error Handling & Alternatives**: Should cover low-quality images, undetected faces, classification errors, and response fallbacks.

5. Front-End Implementation

Objective: Evaluate how effectively the UI captures image input and presents classification results.

- **UI/UX**: Interface must allow easy image upload and display classification output (identity, probability/confidence score).
- **Error Handling**: Alerts for unsupported file types, blank inputs, or detection failures must be clearly shown.

- **Help Documentation**: Include help text on proper image formats, lighting conditions, and usage instructions.

6. Backend Integration & Functionality

Objective: Ensure that the segmentation-classification pipeline works seamlessly via Flask API calls.

- **Full Demo**: Demonstrate image upload → segmentation → feature extraction → classification → output display.
- **Stability & Robustness**: The app must handle high-resolution images, simultaneous requests, and invalid formats without crashing.
- **Major Feature Demonstration**: Test API flow using Postman or React UI that sends images and displays processed results.

7. Test-Plan Review

Objective: Validate comprehensive testing of all possible scenarios in image-based identification.

- **Coverage**: Must include both normal use cases (clear face image) and edge cases (blurred, occluded, or partial faces).
- **Clarity & Completeness**: Describe testing for each module: preprocessing, segmentation, classification, and UI.
- **Exception Handling**: Include tests for missing files, poor resolution, incorrect face detection, and faulty predictions.

8. Final Demo & Review

Objective: Evaluate full working of the system and its documentation.

- **Demo**: Showcases live identification from image input to result, with logs or model stats.
- **Report**: Final documentation includes abstract, SRS, data flow diagrams, algorithm description, architecture, and screenshots.
- **Intermediate Submissions**: Grading also considers the quality of mid-phase submissions like SRS, design plan, and testing sheets.

9. Project Evaluation Rubrics:

Category	Sub-Category / Factors	Marks	Criteria for Marks Allocation
1. Requirements Specification	Clarity of Requirements	5	Use cases around input, processing, and classification are defined.
	Validity of Assumptions	5	Well-explained assumptions (image types, face visibility) are valid.
	Understanding by Team Members	5	Team clearly explains algorithms, dataset usage, and expected outputs.
	Presentation Quality	5	Clear and diagram-backed presentation of problem and solution scope.
2. Technology Familiarization	Knowledge of Tech Stack Components	5	Members explain OpenCV ops, CNN layers, Flask routing, SQLite usage.
	Presentation of Technology	5	Tool responsibilities and libraries (Keras, TensorFlow, etc.) well-demonstrated.
	Quiz / Discussion	5	Faculty may quiz CNN logic, segmentation filters, testing accuracy.
3. Database Design	Clarity of Database Fields	5	Image data, extracted vectors, and model predictions are structured.

Swami Keshvanand Institute of Technology, Management &Gramothan, Ramnagaria, Jagatpura, Jaipur-302017, INDIA
Approved by AICTE, Ministry of HRD, Government of India
Recognized by UGC under Section 2(f) of the UGC Act, 1956
Tel.: +91-0141- 3500300 Fax: +91-0141-2759555
E-mail: info@skit.ac.in Web: www.skit.ac.in

Category	Sub-Category / Factors	Marks	Criteria for Marks Allocation
	Constraints and Relationships	5	Tables correctly relate to user, label, and classification result.
	Backup and Recovery Plan	5	Includes data retention plan and periodic export of processed logs.
4. Design Specifications	Requirement Coverage in Design	5	Segmentation pipeline fully mapped in design diagrams.
	Use Case / DFD / Activity Diagrams	5	Diagrams match stages: face input → segment → classify → result.
	Error Handling & Alternatives	5	Designs include image quality issues, undetectable face fallback.
5. Front-End Implementation	UI Design (Role-Based Dashboards)	5	UI built with HTML/CSS or React must be clean and intuitive.
	Form Validation and Error Messaging	5	Warnings for incorrect files, missing images, or invalid inputs.
	Help / Guidance Content	5	Brief user guidance on image quality and submission flow.
6. Integration with Backend	RESTful API Integration (Frontend ↔ Backend)	5	Flask API correctly processes image and returns classification.
	Application Stability & Responsiveness	5	App should not fail with high image resolution or stress load.
	Feature Completion	5	All modules (input, segmentation, classification) work end-to-end.

Swami Keshvanand Institute of Technology, Management &Gramothan, Ramnagaria, Jagatpura, Jaipur-302017, INDIA
Approved by AICTE, Ministry of HRD, Government of India
Recognized by UGC under Section 2(f) of the UGC Act, 1956
Tel.: +91-0141- 3500300 Fax: +91-0141-2759555
E-mail: info@skit.ac.in Web: www.skit.ac.in

Category	Sub-Category / Factors	Marks	Criteria for Marks Allocation
7. Testing & Edge Cases	Test Plan Coverage	5	All valid/invalid face scenarios tested and documented.
	Execution of Test Scenarios	5	Testing demoed on multiple image types and face categories.
	Exception & Edge Case Handling	5	Handles face obstructions, rotations, and misdetections.
8. Final Demo & Project Report	Final Demo Quality	5	Complete walkthrough of image input to classified identity output.
	Final Report Content	5	Includes system design, flow diagrams, sample inputs, and accuracy results.
	Quality of Intermediate Deliverables	5	Quality maintained in SRS, design review, and integration updates.

Additional Resources:

- Python & OpenCV official docs
- TensorFlow/Keras model building tutorials
- Diagrams.net / Lucidchart / draw.io for UML
- SQLite DB Browser for table setup and testing
- Flask for REST API integration

(Signature):

Faculty Name: Mrs. Shalini Singhal