

Generate a 40-slide presentation titled "Mid Semester Evaluation" based on the following structure and content from the Mid-Semester Internship Report. The presentation should have a modern, professional look and include clear headings, bullet points, and visuals where appropriate. Each slide should focus on a key aspect of the report. Below is the detailed slide breakdown:

Slide 1: Title Slide

- Title: "Mid Semester Evaluation"
- Subtitle: "Mid-Semester Internship Report"
- Include: Submitted by Hemant Singh Sidar, Roll no. 2103310; Submitted to Dr. Saurabh Trivedi, IIT Goa; Date of Submission: 03.03.2025

Slide 2: Internship Overview

- Internship Duration: 26-12-2024 to 25-6-2025
- Organization: Process Venue (Predusk Technology Private Limited), Jaipur
- Project Title: "Development of an LLM-Based Optical Character Recognition (OCR) System"

Slide 3: Presentation Agenda

- Overview of Month 1 and Month 2 progress
- Detailed Weekly Breakdown
- Experiments, Approaches, and Web Application Integration
- Key Learnings, Challenges, and Future Work

Slide 4: Month 1 Overview – Introduction

- Summary of progress in Month 1: Environment setup, AI tools exploration, multi-agent system, deep learning model exploration, and OCR pipeline integration.

Slide 5: Month 1 – Setting Up and Initial Experiments (Overview)

- Focus on setting up the environment and initial experiments.
- Emphasis on tool installation and multi-model PDF processing.

Slide 6: Week 1 – Environment & Tool Setup

- Installed Python, VS Code, and other essential tools.
- Explored various AI tools and their capabilities.

Slide 7: Week 1 – Multi AI Agent System Development

- Developed a system using two models:
  - "llama-3.1-8b-instant" for extracting brief information.
  - "gemma2-9b-it" for summarization.
- Extended to process multiple PDFs using reusable functions.

Slide 8: Week 2 – Overview of Deep Learning & OCR Pipelines

- Introduction to deep learning model exploration and OCR integration.
- Highlight the use of YOLOv8 and Tesseract OCR.

Slide 9: Week 2 – Prompt Engineering & Model Exploration

- Completed a prompt engineering course.

- Explored YOLOv8 for object detection, classification, and image analysis.

Slide 10: Week 2 – Image Processing & OCR Integration

- Used OpenCV for image scaling (gray, binary, RGB).
- Experimented with Tesseract for text extraction.
- Integrated YOLOv8, OpenCV, and Tesseract in a unified pipeline with Gemini summarization.

Slide 11: Week 2 – Pipeline Debugging & Optimization

- Debugged preprocessing techniques.
- Initiated integration of PaddleOCR with LLMs for enhanced extraction.

Slide 12: Week 3 – Overview of Dataset Preparation and Fine-Tuning

- Focus on dataset creation, fine-tuning, and model training.
- Introduction to OCR outputs consolidation and labeling.

Slide 13: Week 3 – Dataset Creation and OCR Outputs

- Classified images for dataset creation.
- Generated OCR outputs using PaddleOCR and produced Markdown summaries.

Slide 14: Week 3 – Labeling Process

- Initiated and completed the labeling of Markdown outputs based on OCR results.

Slide 15: Week 3 – Dataset Format & Script Development

- Developed Python scripts to convert OCR outputs (JSON) and labeled Markdown into a unified format.
- Defined supported dataset formats for fine-tuning.

Slide 16: Week 3 – Fine-Tuning Process & Troubleshooting

- Initiated fine-tuning by selecting a model and loading custom datasets.
- Documented challenges: minimal trainer progress, Colab GPU success vs. VS Code stalling.

Slide 17: Week 3 – Dataset Statistics & Further Experimentation

- Documented dataset statistics (e.g., 152 rows, validation loss, epochs).
- Experimented with models (e.g., "prithivMLmods/Reasoning-SmolLM2-135M") noting runtime challenges.

Slide 18: Week 3 – System Development & Research

- Developed system for fine-tuning across platforms.
- Studied deep learning system requirements, DGX GPU servers, Huggingface fine-tuning, and local LLM deployment.
- Evaluated local deployment benefits: privacy, latency, customization, cost predictability.

Slide 19: Week 4 – Overview of Advanced OCR and Table Extraction

- Introduction to advanced OCR techniques and table extraction methods.

Slide 20: Week 4 – PaddleOCR – Text Detection Algorithms

- Discuss EAST, DB, and SAST algorithms.
- Key points: End-to-end detection, differentiable binarization, context attention blocks.

#### Slide 21: Week 4 – PaddleOCR – Text Recognition Algorithms

- Outline CRNN, Rosetta, STAR-Net, RARE, and SRN.
- Emphasize CNN-RNN models, scalable extraction, spatial transformers, and self-attention frameworks.

#### Slide 22: Week 4 – Table Extraction & Layout Parsing Techniques

- Structured extraction with PDF Plumber.
- Unstructured extraction exploration with PyMuPDF, PyPDF, Camelot, and Tabula-py.
- Layout parsing using Detectron2LayoutModel, AutoLayoutModel, and EfficientDetLayoutModel.

#### Slide 23: Week 4 – Final Pipeline Development

- Integrated PDF Plumber to extract both tables and text from multiple files.
- Highlighted the comprehensive and generalized extraction approach.

#### Slide 24: Transition to Month 2

- Summary of Month 1 achievements.
- Introduction to Month 2 focus: Enhancing structured content extraction and web application integration.

#### Slide 25: Month 2 – Introduction

- Overview of research and development in PDF processing and structured content extraction.
- Focus on TATR, YOLO-based table detection, and PyMuPDF methods.

#### Slide 26: Month 2 – Experiments and Approaches

- List techniques tried: TATR, YOLO-based detection, unstructured extraction, PyMuPDF-based text, table, and image extraction.
- Highlight pros and cons of each approach.

#### Slide 27: Month 2 – Development of the Extraction Pipeline (Overview)

- Outline the planned approach: table detection & cropping, OCR-based extraction, text and image extraction using PyMuPDF.

#### Slide 28: Month 2 – Execution Steps for Table Extraction

- Convert PDF pages into images.
- Identify tables with bounding boxes.
- Crop tables and address coordinate mismatches.
- Use image partitioning to improve accuracy.

#### Slide 29: Month 2 – Enhancing Table Extraction

- Applied PaddleOCR and PyMuPDF for structured extractions.
- Utilized TensorFlow and Non-max function for table structure creation.
- Use Intersection over Union (IoU) for accurate cell placement.

Slide 30: Month 2 – Approaches for Cropping Table Elements

- Method 1: Directly process PDF using the Unstructured Library (limitations).
- Method 2: Convert PDF pages to images and crop using bounding boxes (improved quality).

Slide 31: Month 2 – Execution Steps for Text and Image Extraction

- Text Extraction: Exclude table areas, extract remaining text.
- Image Extraction: Use PyMuPDF (fitz) to extract images.

Slide 32: Month 2 – Integration with Streamlit-Based Web Application

- Hierarchical code structure for modularity.
- Temporary output storage.
- Interactive frontend: display, download, copy, and edit extracted data.
- Backend-frontend connectivity and multi-file handling.

Slide 33: Month 2 – Additional Enhancements in Streamlit App

- Session management and improved user experience.
- Integration of an image processing pipeline for handwritten text extraction using user prompts.

Slide 34: Month 2 – Building the Django-Based Web Application

- Unified platform integrating PDF and image processing pipelines.
- Backend: Seamless integration and efficient handling.
- Frontend: Interactive UI with features for downloading, copying, and editing content.

Slide 35: Month 2 – Features of the Django App

- Supports typed and handwritten text extraction.
- User-prompt-based extraction option.
- Enhanced usability with smooth session management and multi-file handling.

Slide 36: Month 2 – Overall Benefits of the Web Applications

- Scalable and comprehensive solution for structured document processing.
- Improved user experience and customization options.

Slide 37: Future Work & Enhancements

- Plans for further optimization of table extraction and data retrieval.
- Enhance OCR performance with PaddleOCR improvements.
- Expand dataset for fine-tuning.
- Ensure deployment readiness for production.

Slide 38: Key Learnings & Challenges

- Integration of OCR-based table extraction into web applications.
- Structured approach to handling scanned and digital PDFs.
- Identified limitations in existing techniques and areas for improvement.

Slide 39: Final Reflections

- Summary of achievements over the two months.
- Importance of iterative development and multi-approach experimentation.

- Readiness to tackle real-world document processing challenges.

Slide 40: Q&A and Closing Remarks

- Invite questions from the audience.
- Thank you slide with contact details for further queries.

Please generate the slides using slideSpeak with the above structure and content. Ensure each slide is visually engaging and succinctly summarizes the key points.