PROJECT REPORT

NOV 2025

THE SHAIKH AYAZ UNIVERSITY SHIKARPUR

Deep Learning

# MARKNET

## END-TO-END AUTOMATED MARKSHEET INFORMATION EXTRACTION USING YOLOV8 + CRNN

Assigned by: **Dr. Attaullah Sahito**

Submitted by: **Maria Sony** 23-BS(CS)-38**, Muhammad Kaif** 23-BS(CS)-26

DEPARTMENT OF COMPUTER SCIENCE

# ACKNOWLEDGEMENT

We take immense pleasure in presenting this project report, the fruition of our dedicated efforts, collaboration, and continuous learning. We are deeply grateful to **Dr Attaullah** for his guidance, support, and for assigning us this project, which has been a valuable learning experience in our **deep learning** journey.

We would also like to sincerely thank **Shaikh Ayaz University**, for providing us with the platform that enabled us to apply our knowledge and complete this project successfully

01

# ABSTRACT

This project presents a deep learning pipeline for automatic extraction of key academic fields—Group, Name, Percentage, and Date of Issuance—from student marksheets to streamline university admission verification. To overcome the limitations of manual review, we employ a two-stage system combining YOLOv8 for ROI detection and a CRNN for text recognition.

A dataset of 700–750 captured marksheets was collected and annotated using Roboflow. YOLOv8 localizes text regions, which are cropped and transcribed by the CRNN using CTC decoding. This approach achieves good accuracy, significantly outperforming a baseline whole-image CRNN. The resulting system offers an efficient and scalable solution for automating marksheet verification.

# KEY OUTCOMES

- Dataset: ~700–750 marksheet images, annotated with region-level bounding boxes for each field.
- Two-stage approach: YOLOv8 for detection -> CRNN (CTC loss) for text recognition.
- The detection+recognition pipeline accuracy: ~87% (as reported by your experiments). This report details methods to reproduce, evaluate, and improve those results.

# TABLE OF CONTENTS

# 1. PROJECT BACKGROUND & MOTIVATION

Universities often manually verify marksheets when processing admissions. This is time-consuming and prone to human error. Automating field extraction and policy-based verification reduces manual workload and improves consistency. This project focuses on accurate extraction of structured fields from marksheets where layout and image quality vary.

# 2. OBJECTIVES

- Build a robust ROI detector(Yolo) to extract fields: GROUP, NAME, PERCENTAGE, DATE OF ISSUANCE.
- Build a robust sequence recognition model (CRNN) to transcribe field text from ROIs.
- Integrate both models into a unified inference pipeline that automatically extracts and returns all required fields from a given marksheet.
- Provide thorough evaluation, visualization, and reproducibility documentation.

# 3. DATASET

Both models are trained on different formats of the same dataset: YOLO is trained on full marksheet images, while the CRNN is trained on the cropped text regions extracted from those marksheets.

## 3.1 Collection Process

- **Source:** University-submitted marksheets (Collected during admission).
- **Images collected:** 700–750.
- **Capture characteristics:** All images were captured in clear, controlled conditions without intentional rotations, distortions, stamps, or stains. The team ensured high-resolution captures with proper framing and minimal noise to maintain text clarity.

## 3.2 Annotation using RoboFlow (For Yolo)

- **Format:** per image with bounding boxes for each field.
- **Example Text File:** image_filename, x_min, y_min, x_max, y_max, class_name
- **Class names:** GROUP, NAME, PERCENTAGE, DATE OF ISSUANCE.

## 3.3 Labeling (For CRNN)

- Individual ROIs cropped from the original marksheet images.
- Each cropped region is labeled with the corresponding field value in csv file.
- **Example:** image_name  GROUP: PRE-MEDICAL

## 3.3 Dataset Statistics

- Total images: 740
- Total crops after annotation: 2927

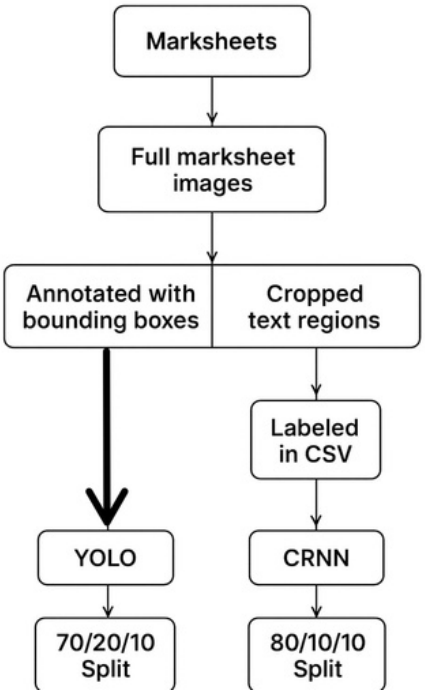
## 3.4 Train / Val / Test Split

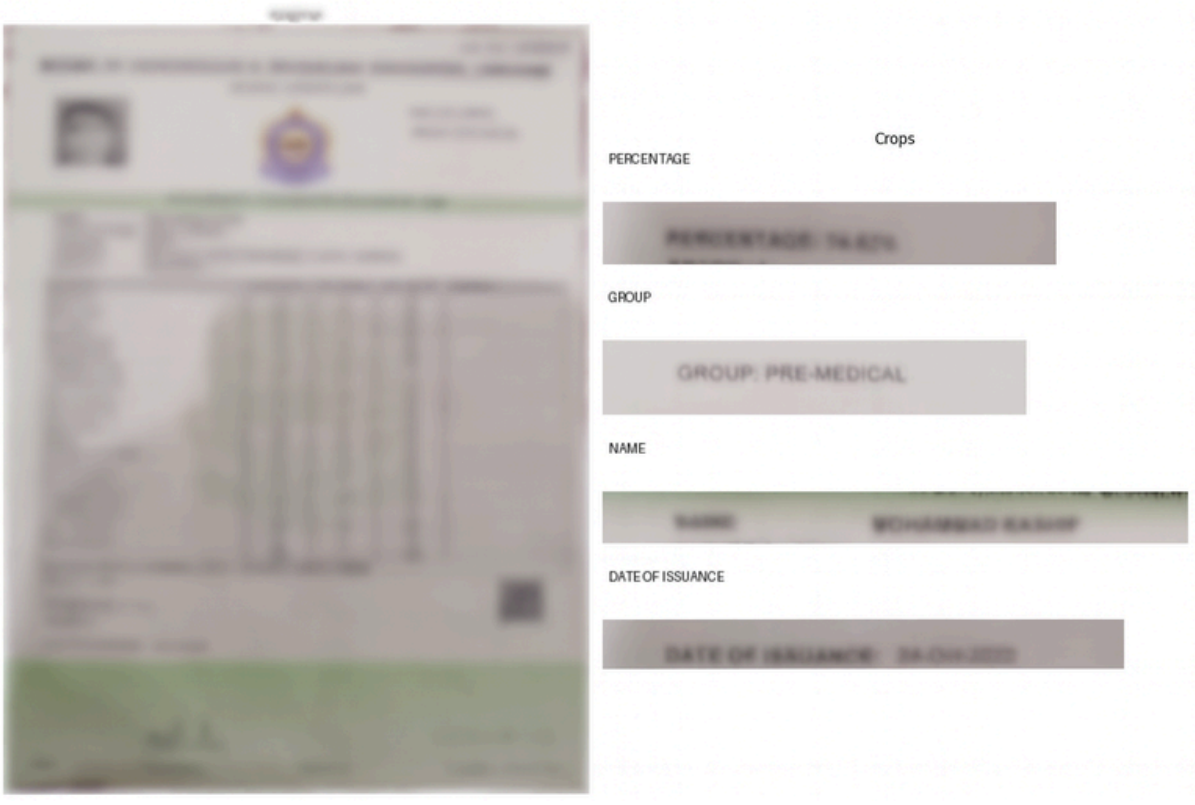
The dataset is divided separately for both models based on their training requirements:

- YOLOv8 (ROI Detection): 70% Training, 20% Validation, 10% Testing
- CRNN (Text Recognition): 80% Training, 10% Validation, 10% Testing

# 05

"In this project, two datasets are derived from the original images:
1. the full images annotated with bounding boxes, and
2. the cropped image patches with text labels used for training the CRNN model."



06

# 4. PREPROCESSING

The preprocessing stage ensures that all text-region crops (NAME, GROUP, PERCENTAGE, DATE) are converted into a uniform, model-ready format before being fed into the CRNN text-recognition network.
This stage covers vocabulary construction, text normalization, image resizing/padding, grayscale conversion, and data augmentation.

## 4.1 Vocabulary Construction

To enable sequence modeling, a unified vocabulary (set of characters used across all fields) is created from the annotated text labels.

**Steps**

1. Default character set  A predefined set of characters expected in marksheets is initialized:
2. Normalize labels to uppercase for consistency Each row is converted into a unified string such as:
   "NAME: ZAIB ALI SHAH" or "PERCENTAGE: 77.27"

## 4.2 Image Preprocessing

CRNN requires uniform-sized grayscale images. All crops (YOLO-detected ROIs) are therefore passed through the following pipeline.
Image Dimensions

- Height = 64
- Width = 512

**1) Training transformations :**

- Grayscale conversion
- Aspect-ratio preserving resize and padding (ResizePad)
- Small random rotations (±3°)
- Slight brightness and contrast adjustments
- Tensor conversion and normalization (mean=0.5, std=0.5)

**(2) Validation transformations**

- Grayscale conversion
- Resize and padding (ResizePad)
- Tensor conversion and normalization

### 4.3 Data Augmentation

To increase CRNN robustness to real-world variations, controlled augmentation is applied on-the-fly.

# 5.1 YOLO

**YOLO for Detection of Regions of Interest (ROIs)**

Using transfer learning from COCO-pretrained weights. To localize the four required fields (**NAME, GROUP, PERCENTAGE, DATE OF ISSUANCE**) from full marksheet images, a YOLOv8 object-detection model was trained on the annotated dataset. This model performs region-of-interest (ROI) extraction before text recognition.
Model Selection
YOLOv8 variants differ in speed and accuracy trade-offs.
For this project, YOLOv8n (nano) was selected because:

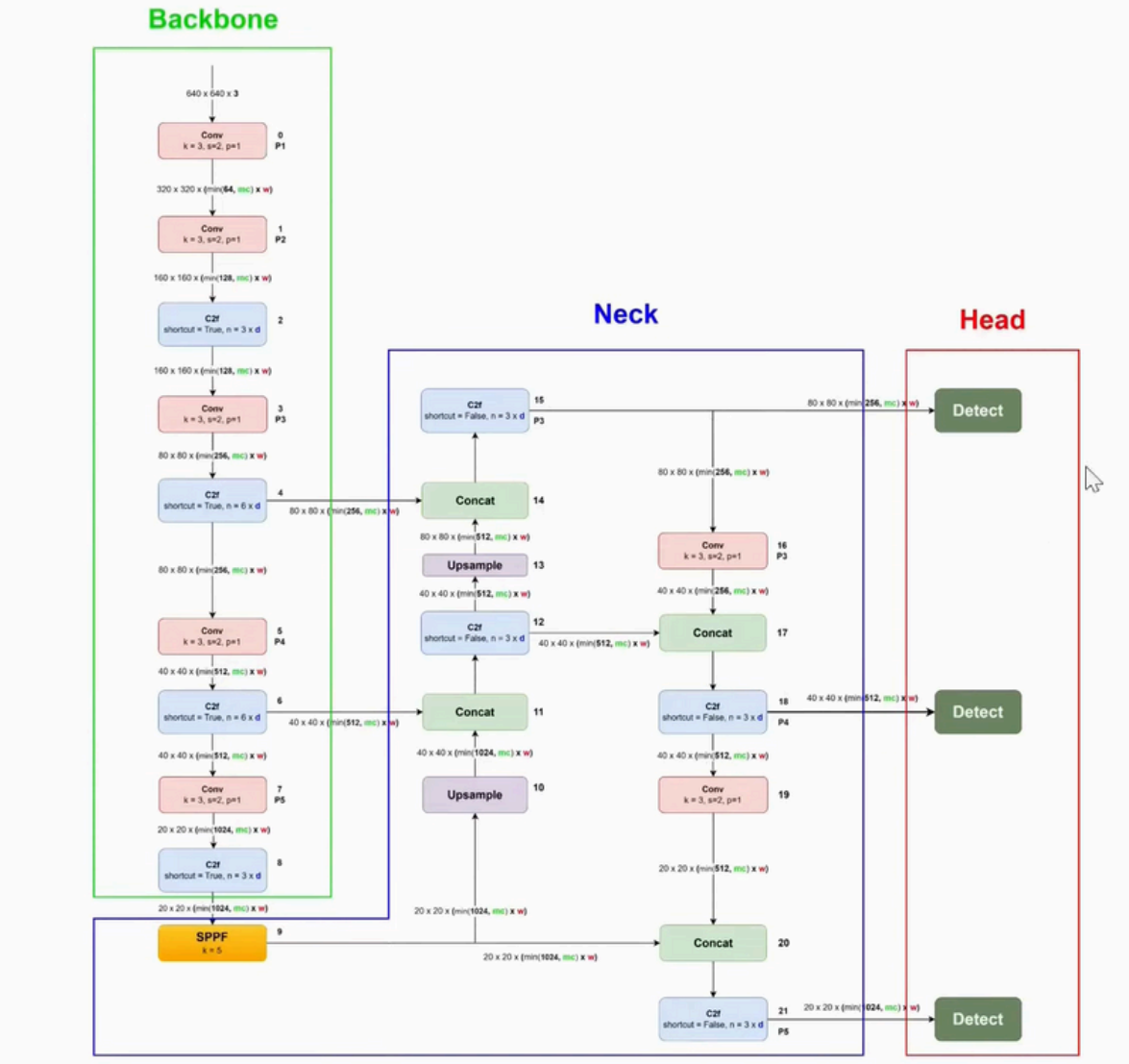
- It provides fast inference suitable for deployment.
- It achieves high accuracy on well-structured documents.
- It can be trained efficiently on Google Colab (T4 GPU).

Key Hyperparameters

- Epochs: 100
- Image Size: 640 × 640 (recommended for YOLOv8)
- Batch Size: 16
- Device: GPU (Colab T4)
- Loss Functions:
  - Bounding Box Loss: CIoU/GIoU
  - Class Loss
  - Objectness Loss

08

# YOLO V8 ARCITECTRUE

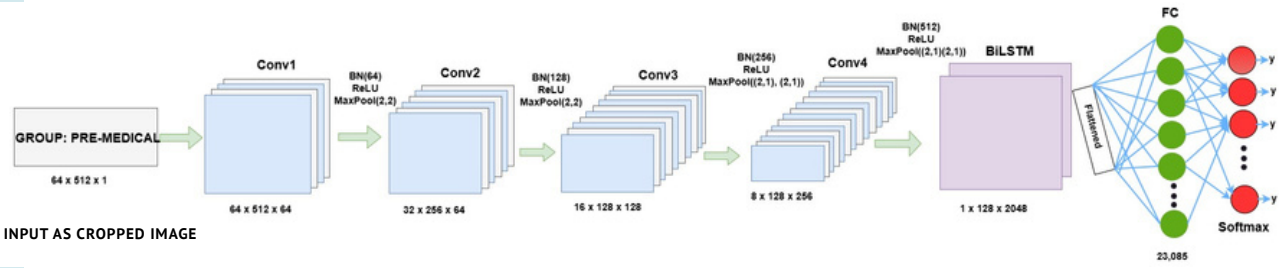


# 5.2 CRNN (CONVOLUTIONAL RECURRENT NEURAL NETWORK)

The CRNN architecture is employed to perform sequence-based text recognition on the cropped Regions of Interest (ROIs) extracted by the YOLO detector. CRNN combines convolutional, recurrent, and transcription layers to efficiently handle variable-length text in images, making it highly suitable for OCR tasks on structured documents such as marksheets.
CRNN (Convolutional Recurrent Neural Network)

**Model Architecture (CRNN)**
The CRNN uses a three-stage pipeline for text recognition:

- **Convolutional Feature Extraction:**
- A lightweight CNN backbone (Conv + BatchNorm + ReLU) extracts spatial features, with max-pooling layers reducing size while retaining important text details.
- **Sequence Modeling (BiLSTM):**
- The CNN feature map is converted into a sequence and processed by a 2-layer Bidirectional LSTM to learn contextual character dependencies.
- **Transcription (CTC Layer):**
- A final linear layer outputs per-timestep character probabilities, and CTC loss handles alignment without requiring character-level labels.



# 6. TRANING

## *6.1 YOLOv8 Training*

The YOLOv8 model was trained on full marksheet images annotated with bounding boxes for all required fields. The model achieved extremely high detection accuracy and was subsequently used to generate precise ROI crops for the OCR/CRNN stage.

1. **Training Data:**
   Full marksheet images labeled for NAME, GROUP, DATE OF ISSUANCE, and PERCENTAGE.
2. **Training Setup:**

- Epochs: 100
- Image Size: 640×640
- Batch Size: 16
- Hardware: NVIDIA Tesla T4 (15GB VRAM)
- Ultralytics Version: 8.3.229
- Model Size: YOLOv8n/s (72 layers, ~3M parameters, 8.1 GFLOPs)

## *6.2 CRNN Training*

The CRNN model was trained on YOLO-extracted text crops to recognize student names, percentages, groups, and dates using a sequence-learning architecture based on CNN + BiLSTM + CTC.

**Training Setup:**

- Optimizer: Adam
- Learning Rate: 5e-4 (steady learning, stable CTC convergence)
- Epochs: 20
- Batching: Variable-length label sequences handled via CTC rules.
- Loss: CTC loss with blank token handling and collapsing repeated predictions.

# 7. APPLY CROSS-VALIDATION ON THE VAL DATASET TO TUNE HYPERPARAMETERS

*To identify the optimal training configuration for the CRNN, a 3-fold cross-validation procedure was performed on the validation set. The search space included learning rate, batch size, and LSTM hidden dimension.*

***Search Space:***

- *Learning Rate: 1e-4, 3e-4*
- *Batch Size: 32, 64*
- *Hidden Size: 128, 256*

*Each hyperparameter combination was evaluated across all folds, and the average validation CTC loss was used as the selection criterion.*

***Cross-Validation Results (Summary):***

- *Smaller batch sizes generally produced more stable convergence.*
- *Higher hidden dimensions (256) consistently outperformed 128 in terms of sequence modeling fidelity.*
- *Learning rate 3e-4 yielded the lowest losses across almost all configurations.*

***Best Hyperparameter Configuration:***

- *Learning Rate: 3e-4*
- *Batch Size: 32*
- *Hidden Size: 256*

*This combination achieved the lowest mean validation loss, indicating the most effective balance between optimization stability and model capacity.*

# 8. MODEL EVALUATION

## 8.1 YOLOv8 Evaluation

The YOLOv8 detector was evaluated on an independent test set of 74 marksheet images containing 291 labeled text fields. The model demonstrated highly reliable detection performance, with near-perfect precision and recall across all classes.

**Overall Test Metrics**

- Precision: 0.9966
- Recall: 0.9964
- mAP50: 0.9921
- mAP50-95: 0.7636

**Per-Class Detection Performance**

| Class | Precision | Recall | mAP50 | mAP50-95 |
|---|---|---|---|---|
| DATE OF ISSUANCE | 1.000 | 1.000 | 0.995 | 0.737 |
| GROUP | 1.000 | 1.000 | 0.995 | 0.757 |
| NAME | 0.986 | 1.000 | 0.986 | 0.785 |
| PERCENTAGE | 1.000 | 0.986 | 0.992 | 0.775 |

## 8.2 CRNN Evaluation

The tuned CRNN model was evaluated on YOLO-extracted text crops from the test set. Metrics were computed at character-level, string-level, and field-level to assess real-world OCR reliability.

**Overall Recognition Metrics**

- Character Accuracy: 0.943≈
- String Accuracy: 0.84≈
- Character Error Rate (CER): 0.078≈

14

# RESULTS

Our **two-stage pipeline** using **YOLOv8** for field detection and a **CRNN with CTC decoding** for text recognition performed significantly better than a baseline whole-image CRNN. Using a dataset of 700–750 annotated marksheets, the system achieved strong accuracy.

# CONCLUSION

We have completed this project by following all provided guidelines and applying our best efforts to learn, design, and implement a deep learning system for automating marksheet verification.
The goal was to extract key academic fields—**Group, Name, Percentage, and Date of Issuance**—more accurately and efficiently than manual review.
This project strengthened our understanding of deep learning workflows, dataset preparation, model evaluation, and real-world automation challenges. It demonstrates how AI (deep learning) can streamline admission processes and reduce human error in document verification.

***"Dedication turns plans into progress, and teamwork turns progress into success."***

# ENVIRONMENT USED

- *Google Colab*

# FRAMEWORK

- *Pytorch*

# GITHUB

- *https://github.com/Kaif-Sasoli/MarkNet-DL*
- *https://github.com/Maria-Dahar/MarkNet-DL*

# WEBSITE

- *DEMO UI Interface*
  *https://saus-omega.vercel.app*

# REFERENCES


- *Deep Learning Specialization (Andrew Ng)*
  *https://www.coursera.org/specializations/deep-learning*