GCIT Students ID Card

CSA401 Advanced Deep Learning
Bachelor of Science in Computer Science
AI Development and Data Science
Year III, Semester II

Group Member(S)

Jigme Wangyel Wangchuk(12210012)
Pema Yangchen (12210025)
Deki Lhazom (12210048)

GUIDED BY

Ms Ugyen Choden

Gyalpozhing College of Information Technology
Kabesa : Thimphu

1 Abstract

This deep learning project aims to develop an advanced deep learning-based web application to assist gatekeepers at Gyalpozhing College of Information Technology (GCIT) in efficiently managing student entries and exits. By utilizing Optical Character Recognition (OCR) technology, the system will accurately scan and extract information from student identification cards. The application is designed to enhance the accuracy and efficiency of attendance recording, thereby reducing manual workload and minimizing errors.

Keywords: Deep Learning, Optical Character Recognition (OCR), Tesseract, Student Identification Cards, YOLO

2 Introduction

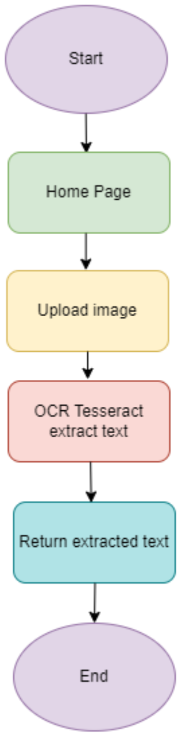

Figure 1: System Overview

In this system, users need to upload image, and the OCR tesseract will extract the text in image and return as output. In educational institutions, managing student attendance is a vital yet often cumbersome task. At Gyalpozhing College of Information Technology (GCIT), the current manual process of recording student entries and exits poses significant challenges, especially during peak times when multiple students enter or leave simultaneously. The traditional method, which involves gatekeepers manually entering each student's information, is not only time-consuming but also prone to errors.

This project addresses these issues by developing a web application that leverages advanced deep learning techniques to automate the attendance recording process. By employing Optical Character Recognition (OCR) technology, the system can scan and extract information from GCIT student ID cards accurately and efficiently.

3 Literature Review

Ojas Kumar Barawal and Dr.Yojna Arora in "Text Extraction from Image" discusses about the challenging task of extracting text from images due to various factors such as different text sizes, styles, orientations, and complex background structures. The text extraction process includes detection, localization, segmentation, and enhancement of text from input images. It presents a comparative study and performance evaluation of various text extraction techniques, including region-based, edge-based, and texture-based methods. The goal is to extract regions containing text without character recognition capabilities. Region method uses the properties of color or gray-scale in a text region and their differences with the corresponding properties of the background. Whereas edge-based technique is quick and effectively localizes and extracts text from both documents and images by detecting edges, which represent significant intensity variations and discontinuities in depth. In texture-based technique, it extract using texture-based methods relies on the observation that texts in images have distinct textural properties that distinguish them from the background[1].

Jiijul, Tuscano and Badgujar presented about the development of a system to extract text from images of bills or invoices using OpenCV, Tesseract OCR, and Google Firebase. The system involves using edge detection and contour tracking to identify the bill or invoice in the image, followed by text extraction using Tesseract OCR. The research also includes the development of an Android app to test the system, where users can upload bill images and the system extracts the total amount from the bill, rewarding the user with cash back points based on the total amount. This research also highlights the limitations of the system, such as difficulty in extracting text from handwritten bills and the possibility of detecting non-bill objects as bills. It also discusses the tools used in the research, including OpenCV, Android Studio, Tesseract OCR, and Google Firebase[2].

The research "Text Extraction and Recognition from image using Neural Network" discusses the use of neural networks for text extraction and recognition from images. The primary objective is to create an unconstrained image indexing and retrieval system using neural networks. The approach involves HSV-based color reduction, feature extraction, and a feature-based classifier to determine if a region of interest (ROI) contains text or non-text blocks. Text blocks are then input to an Optical Character Recognition (OCR) system, and the output is stored in a database for future retrieval. It also presents training of the neural network using the backpropagation algorithm and the use of morphological and geometrical constraints for text detection[3].

The "Research Paper on Text Extraction using Optical Character Recognition (OCR)" from the International Journal of Advanced Research in Science, Communication and Technology (IJARSCT) discusses the challenges and applications of extracting text from images. The challenges include handling complex backgrounds, varying fonts, and distorted or degraded text. These challenges can make it difficult to accurately and efficiently extract text from images. On the other hand, the applications of text extraction from images are vast and include document digitization, information retrieval, automated data entry, and text analysis. This process plays a vital role in enabling automated processes and facilitating the analysis of large volumes of visual data. Traditional methods for text extraction from images involve preprocessing steps such as binarization and noise removal, followed by techniques like connected component analysis and optical character recognition (OCR)[4].

The “Text Extraction from Images Using OCR” discusses the development of an application for text recognition in scanned documents and images. The proposed system aims to classify images of documents such as identity proofs into different categories, extract text data from these images, and store the extracted credentials in a database. The methodology process involves the use of Tesseract OCR package for text extraction from images. The process includes uploading the image, performing image scaling or gray scale conversion, and storing the record in the database. The Tesseract OCR package contains an optical character recognition (OCR) engine and a command line program, and it focuses on line recognition and character pattern recognition using a Long Short-Term Memory (LSTM) based OCR engine. The research addressed detection of text from documents in which text is embedded in complex colored document images as a challenging problem[5].

4 Aims and Objectives

The aim of the project is to create an web application that can precisely scan student ID cards, extract data from them. This system's main goal is to automate the tracking of students' arrival and departure times using their college ID cards, which will accelerate the process of registering attendance. This application avoids potential errors and drastically decreases administrative workload by doing away with the requirement for sign-in papers or manual entry. In addition, the system aims for a ninety percent model accuracy, which guarantees dependable and effective attendance management.

5 Methodology

The dataset consist of 71 ID card collected from GCIT students. The dataset was annotated using "Label Studio" annotaion tool. The dataset consists of five labels that is name, I.D. number, course, gender and CID number.

Label Studio
Community Edition

Figure 2: Label Studio

The annotated data was then exported in the YOLO format since this study employs YOLO architecture for text detection. The respective data image and annotated file was kept in one folder. The data is then divided into 80 percent and 20 percent of data respectivley into train and test.

(a) GCIT student ID card

(b) Annotated student ID card

Figure 3: Dataset

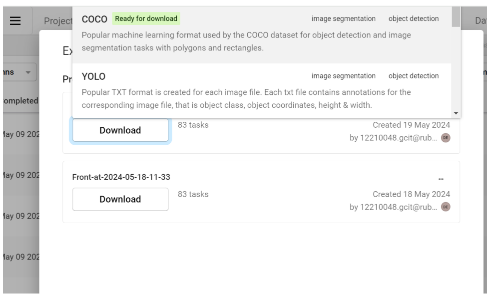

Figure 4: Annotate Data Exported

5.1 Architecture Employed

5.1.1 Text Detection

For the text detection, this research will implement YOLOv2 to train the model. YOLOv2, or "You Only Look Once version 2," is a state-of-the-art detection model used for general detection tasks. It is known for its ability to achieve a good trade-off between speed and accuracy, outperforming other advanced techniques like Faster R-CNN and SSD while still running faster than them[6].

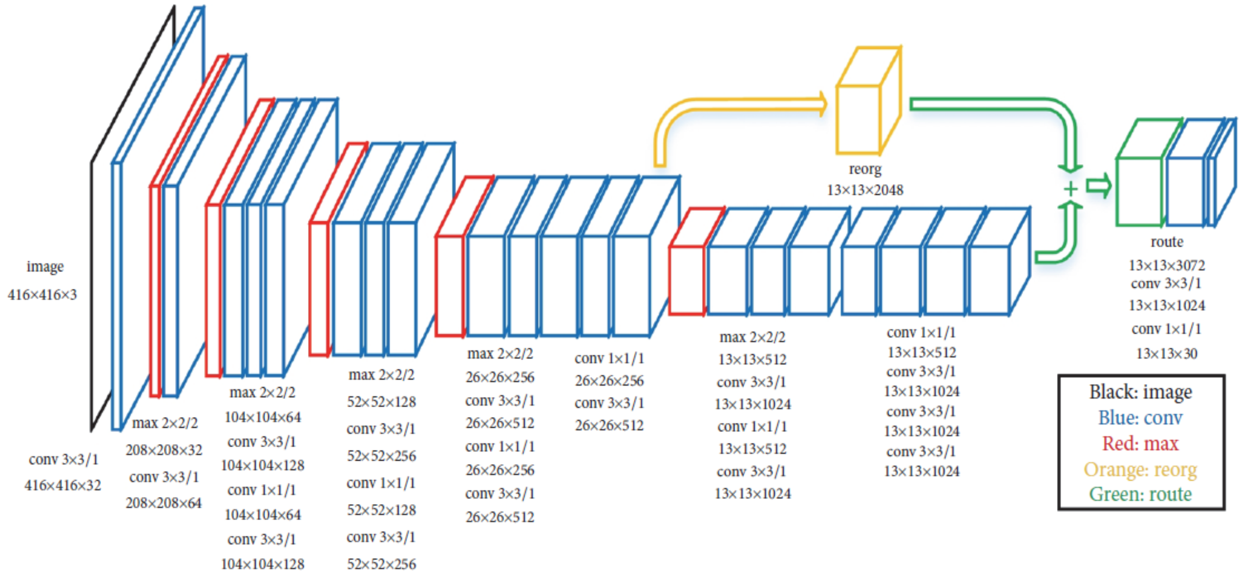

Figure 5: Network Architecture of YOLOv2

The YOLOv2 network integrates the extraction of candidate boxes, feature extraction, target classification, and target location into a single deep network. It runs at different sizes using a novel and multiscale training technique, offering a good trade-off between speed and accuracy. The network employs a grid-based approach, subdividing the image into an M x N grid, with each grid detecting an object if the center of the object falls into that grid cell. It uses initial bounding boxes of different specifications and predicts bounding boxes and confidence scores for corresponding boxes through deep convolutional layers. The network also employs a clustering algorithm for preprocessing the training dataset to obtain initial candidate boxes and introduces technologies such as multiscale, semantic fusion, and scale-aware for different data[6].

5.1.2 OCR Tool Tesseract

Optical character Recognition (OCR) is a conversion of scanned or printed text images, handwritten text into editable text for further processing. This technology allows machine to recognize the text automatically. It is like combination of eye and mind of human body. An eye can view the text from the images but actually the brain processes as well as interprets that extracted text read by eye. Tesseract is an open source optical character recognition engine. It was developed at HP in between 1984 to1994. It was modified and improved in 1995 with greater accuracy. In late 2005, HP released Tesseract for open source[7].

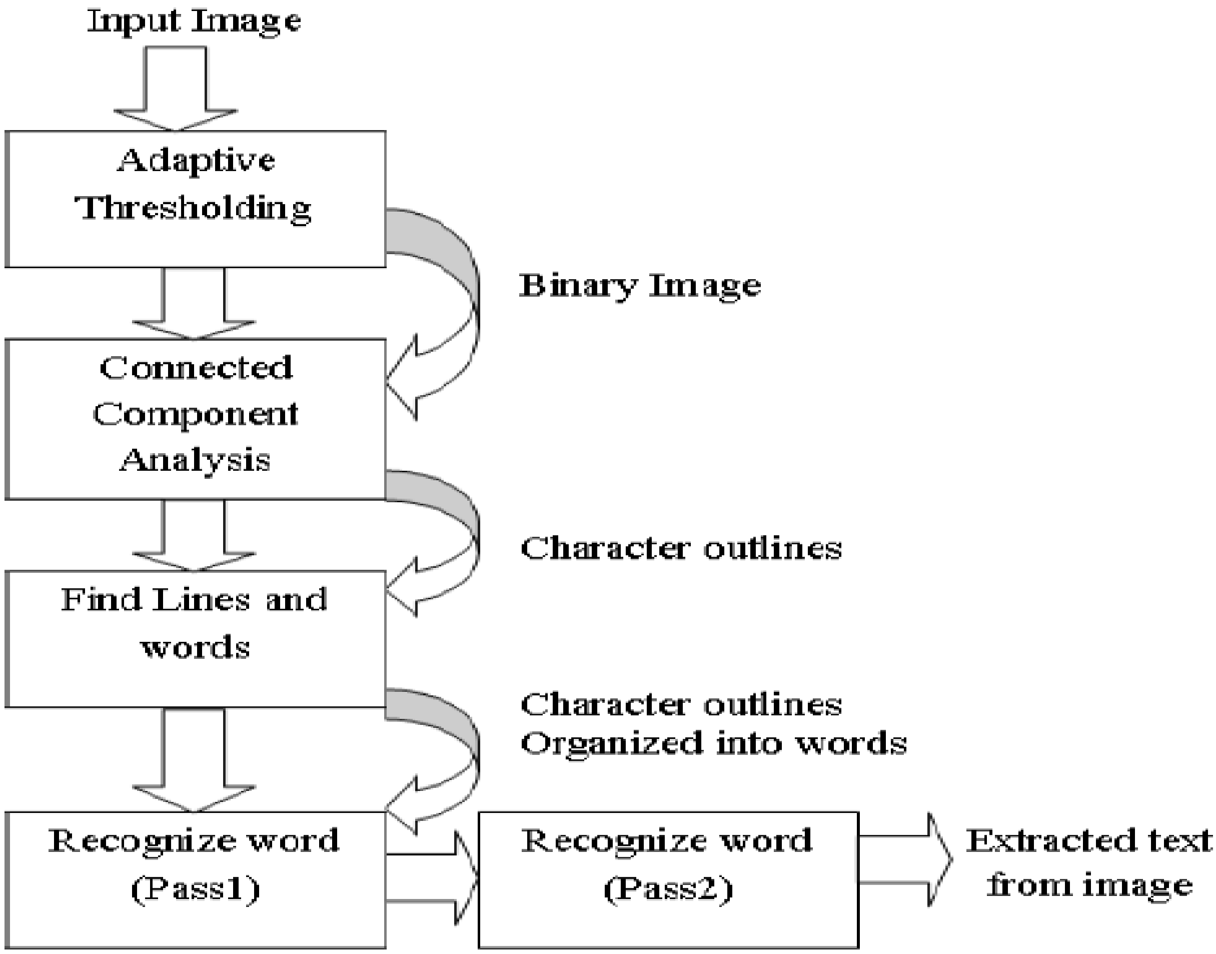

Figure 6: Architecture of Tesseract

The architecture of Tesseract OCR involves several steps in the optical character recognition process. It begins with Adaptive Thresholding to convert the image into binary images, followed by connected component analysis to extract character outlines. These outlines are then converted into blobs, which are organized into text lines. The lines and regions are analyzed for fixed areas or equivalent text sizes, and the text is divided into words using definite and fuzzy spaces. Tesseract uses a two-pass process for text recognition, with the first pass attempting to recognize each word and passing satisfactory words to an adaptive classifier as training data. The adaptive classifier then tries to recognize text more accurately. The final phase resolves various issues and extracts text from images[7].

6 Results and Discussions

6.1 Experimental Setup

6.1.1 Programming Language

- Python: The most popular programming language for deep learning applications is Python. It provides a vast array of libraries and frameworks that simplify working with big datasets and neural networks.

6.1.2 Framework

- Darknet: Darknet is an open-source neural network framework written in C and CUDA. It's primarily designed for deep learning tasks, especially in the domains of object detection and image classification.

6.1.3 Text extraction

- OCR Tesseract: Tesseract OCR is a widely used option for text recognition because of its accessibility and accuracy. For smooth text recognition, it can be incorporated into the Python environment using tools like Pytesseract.

6.2 Model Training Procedure

To employ yolo v2 for text detection, this study used "Darknet" framework for training the yolo v2 model. Darknet is an open-source neural network framework written in C and CUDA. It is the framework used to train and test the YOLO (You Only Look Once) family of models. The study clone darknet framework in repository. The "obj.data" file was created inside darknet repository which contains the number of classes, directory for train and test set of data and directory of "obj.names" which contain the list of classes name. "Yolov2-obj.cfg" file was configured based on this study requirements. The training command "./darknet detector train data/obj.data cfg/yolov2-obj.cfg yolov2.conv.23 " executed to train the yolo v2 on custom dataset.

For text recognition, this study implements OCR tesseract to recognise and extract text from given input image. In tesseract, preprocessing steps include converting the image to grayscale, thresholding in which grayscale image is converted to binary image to enhance text region. This preprocessed image is then fed into tesseract tool to extract text.

6.3 Model Performance

The result of the study is displayed below which includes result from YOLO v2 and ocr tesseract.

Table 1: YOLO v2 accuracy

	Last Accuracy	Best Accuracy
mAP@0.50	69.14	73.00

mAP (mean Average Precision) is a metric that evaluates the accuracy of object detection models by considering both precision and recall. It measures how well a model identifies objects correctly while minimizing false positives. mAP@0.50 specifies that the average precision is calculated at an IoU threshold of 0.50. This means that for an object to be considered correctly detected, the overlap (IoU) between the predicted bounding box and the ground truth bounding box must be at least 50 percent. If the IoU is below 50 percent, the detection is considered a false positive.

Table 2: OCR Tesseract Accuracy

	Tesseract Accuracy
Accuracy	1
Precision	1
Recall	1
F1-score	1

The accuracy, precision, recall and f1-score achieved by OCR tesseract is 1.

- Accuracy is calculated as (TP + TN) / (TP + TN + FP + FN), where TN is true negatives and FN is false negatives.
- Precision is calculated as TP / (TP + FP), where TP is true positives and FP is false positives.
- Recall is calculated as TP / (TP + FN), where FN is false negatives.
- F1-score is calculated as 2 * (Precision * Recall) / (Precision + Recall).

7 Conclusion

Using Tesseract OCR for text recognition and YOLOv2 for text detection, the deep learning-based web application that was built at Gyalpozhing College of Information Technology (GCIT) effectively automates the management of student admissions and exits. With Tesseract OCR yielding flawless accuracy, precision, recall, and F1-score and YOLOv2 yielding a mAP@0.50 of 69.14 percent, the system greatly enhances the efficiency and accuracy of attendance recording. This solution offers a dependable and effective way to control attendance by minimizing errors and cutting down on the amount of human labor.

References

[1] O. K. Barawal and D. Y. Arora, "Text Extraction from Image," *International Journal of Innovative Research in Engineering & Management*, pp. 89–92, Jun. 2022. [Online]. Available: https://ijirem.org/DOC/12-text-extraction-from-image-1.pdf

[2] Alan Jiju, Shaun Tuscano, and Chetana Badgujar, "OCR Text Extraction," *International Journal of Engineering and Management Research*, vol. 11, no. 2, pp. 83–86, Apr. 2021. [Online]. Available: https://ijemr.vandanapublications.com/index.php/ijemr/article/view/105/105

[3] C. Misra, P. K. Swain, and J. K. Mantri, "Text Extraction and Recognition from Image using Neural Network," *International Journal of Computer Applications*, vol. 40, no. 2, pp. 13–19, Feb. 2012. [Online]. Available: http://research.ijcaonline.org/volume40/number2/pxc3877156.pdf

[4] A. Thorat, M. Zagade, S. More, and A. Narute5, "Research paper on text extraction using ocr," *International Journal of Advanced Research in Science, Communication and Technology*, vol. 3, no. 14, May 2023, p. 10–14, May 2023.

[5] J. E. Tejaswini, K, "Text Extraction from Images Using OCR," *International Journal for Research in Applied Science and Engineering Technology*, vol. 8, no. 5, pp. 1805–1810, May 2020. [Online]. Available: http://ijraset.com/fileserve.php?FID=28750

[6] Z. Liu, Z. Chen, Z. Li, and W. Hu, "An Efficient Pedestrian Detection Method Based on YOLOv2," *Mathematical Problems in Engineering*, vol. 2018, pp. 1–10, Dec. 2018. [Online]. Available: https://www.hindawi.com/journals/mpe/2018/3518959/

[7] C. Patel, A. Patel, and D. Patel, "Optical Character Recognition by Open source OCR Tool Tesseract: A Case Study," *International Journal of Computer Applications*, vol. 55, no. 10, pp. 50–56, Oct. 2012. [Online]. Available: http://research.ijcaonline.org/volume55/number10/pxc3882784.pdf