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 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].

2 Proposed Methods

2.1 System Overview

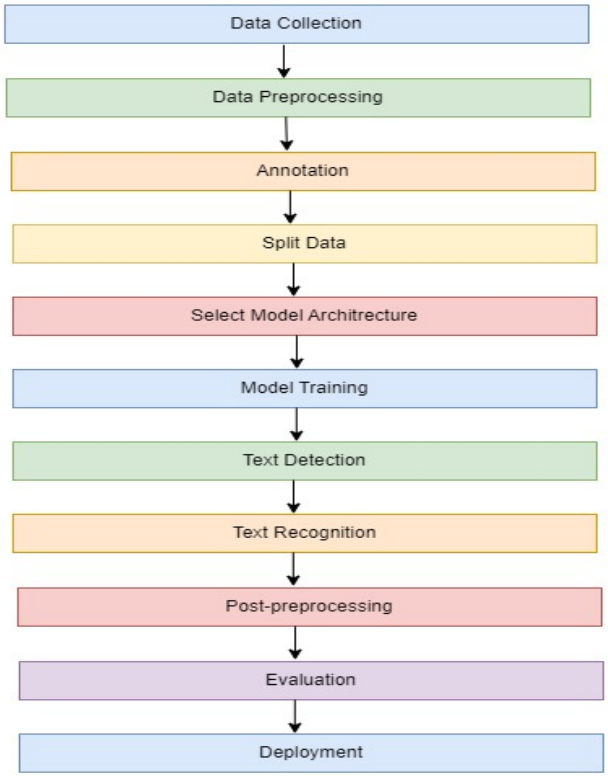

Figure 1: System Overview

- **Data Collection:** The initial stage of the process involves the acquisition of a dataset of student ID cards. Specifically targeting ID cards of students of Gyalpozhing College of Information Technology, a manual data collection approach is employed, soliciting the cooperation of students. Subsequently, the student ID cards will be systematically captured and stored as images.
- **Data Preprocessing:** Preprocess the images to enhance text visibility and standardize them for training. Steps may include resizing, normalization, noise reduction, and contrast enhancement.
- **Annotation:** Annotate the images with bounding boxes or masks to indicate the regions containing text. This annotated data will serve as the ground truth for training the model.
- **Split Data:** Divide the dataset into training, validation, and testing sets into 70 percent, 15 percent and 15 percent of collected data respectively. The training set is used to train the model, the validation set helps optimize hyperparameters, and the testing set evaluates the model's performance.
- **Select Model Architecture:** Select a deep learning architecture suitable for text extraction from images. Common choices include Convolutional Neural Networks (CNNs), Region-Based CNNs (R-CNNs), Single Shot Detectors (SSDs), or the more recent models like YOLO (You Only Look Once) and EfficientDet.
- **Model Training:** Train the selected model using the annotated image dataset. This research will implement techniques like transfer learning if applicable, where we leverage pre-trained models (e.g., on ImageNet) and fine-tune them for text extraction tasks.
- **Text Detection:** Train the model to detect text regions within images. This involves predicting bounding boxes or masks around text instances.
- **Text Recognition:** Implement a text recognition component to extract the actual text content from the detected regions. This can be achieved using techniques like Optical Character Recognition (OCR) or sequence-to-sequence models (e.g., Long Short-Term Memory networks - LSTMs).
- **Post Processing:** Apply post-processing techniques to refine the extracted text, such as language model integration, spell checking, and removing false positives.
- **Evaluation:** Evaluate the model's performance on the validation and testing datasets using metrics like precision, recall, F1 score, and accuracy. Fine-tune the model based on evaluation results to improve performance.
- **Deployment:** Once satisfied with the model's performance, deploy it in a production environment.

2.2 Algorithm

2.2.1 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].

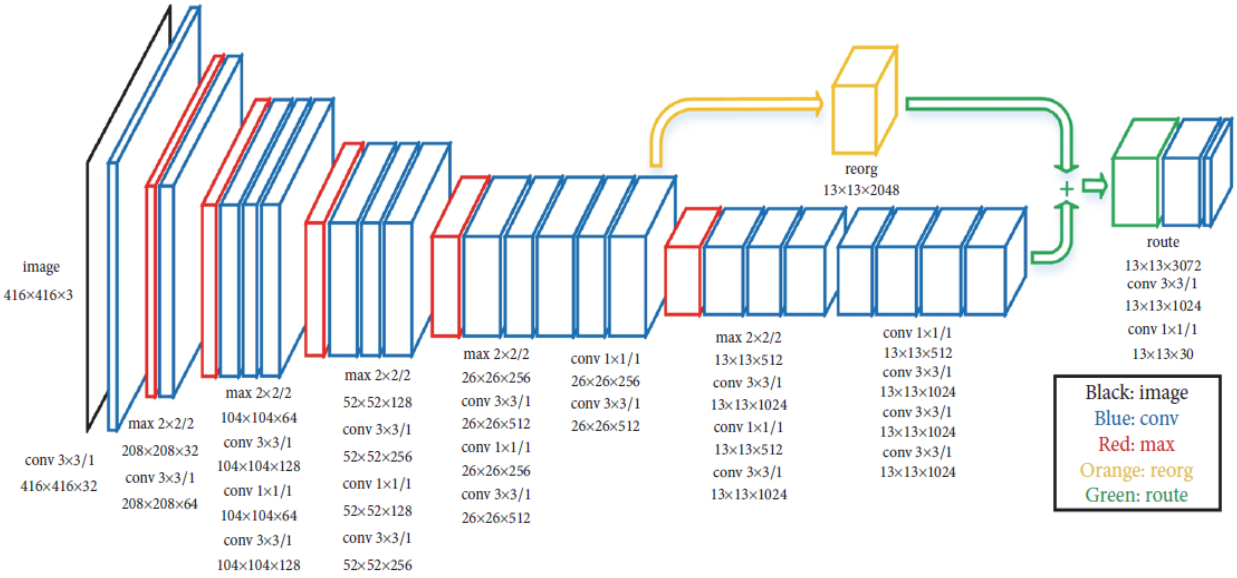

Figure 2: Network Architecture of YOLOv2

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].

2.2.2 Text Recognition

For text recognition, this research will implement OCR variants that is CRNN architecture. The CRNN (Convolutional Recurrent Neural Network) is a neural network architecture that combines the functionalities of Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) along with transcription layers and Connectionist Temporal Classification (CTC) for text recognition. In the proposed OCR system, the CRNN architecture is used for text recognition without the need for character segmentation[7].

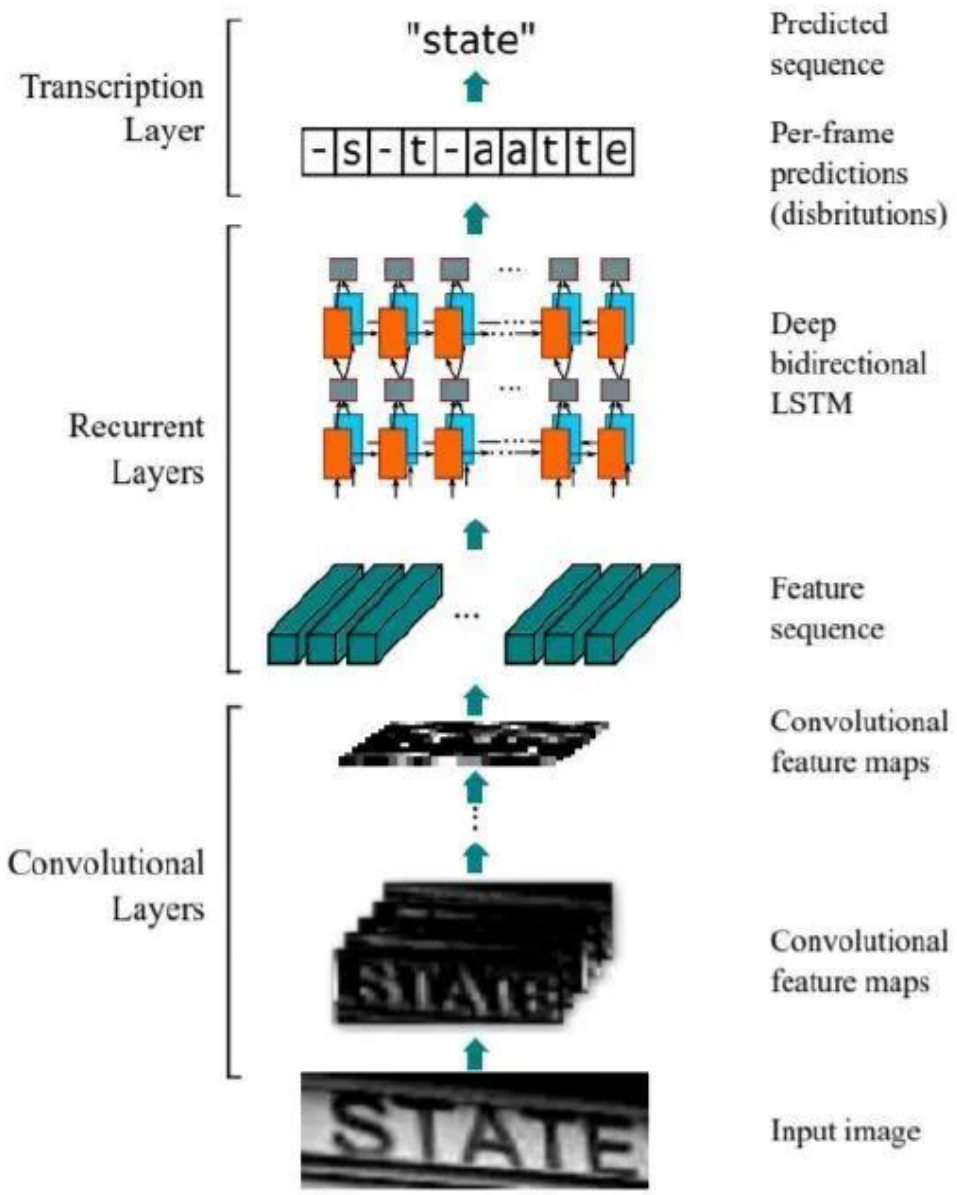

Figure 3: CRNN Architecture

The CRNN architecture flow includes as follows[7]:

- The preprocessed image given as input to the CRNN architecture.
- This input image is passed on to the convolution component for feature extraction.
- The convolutional feature maps generated are used to form receptive fields resulting in a sequence of feature vectors.
- The output from the CNN component is passed as input to the RNN component for sequence modeling.
- The layers in RNN predict label distribution for every frame in the feature sequence.
- The per-frame label distributions generated from the bidirectional LSTM layers are passed through the transcription component.

- These label distributions are converted into label sequences by applying the Connectionist Temporal Classification (CTC).
- The label sequences obtained are then converted into the format of required label using the CTC decoder.

2.3 Dataset

(a) Front GCIT student ID card

(b) Back GCIT student ID card

Figure 4: Dataset Samples

A manual data collection approach is utilized to capture each student ID, which is then stored in the form of an images specifically in jpg extension. This research will extract text such as name, course, gender, CID number., I.D. number and contact number to record of students entry and exit from college campus.

2.4 Evaluation Metrics

Following evaluating metrics will be used to calculate the performance of the model:

- **Precision:** The precision is calculated as the ratio between the number of Positive samples correctly classified to the total number of samples classified as Positive (either correctly or incorrectly). The precision measures the model's accuracy in classifying a sample as positive.

  Precision = TP / (TP + FP)

- **Recall:** The recall is calculated as the ratio between the number of Positive samples correctly classified as Positive to the total number of Positive samples. The recall measures the model's ability to detect Positive samples. The higher the recall, the more positive samples detected.

  Recall = TP / (TP + FN)

- **F1 score (F1):** It is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account. Intuitively it is not as easy to understand as accuracy, but F1 is usually more useful than accuracy, especially if you have an uneven class distribution. Accuracy works best if false positives and false negatives have similar cost. If the cost of false positives and false negatives are very different, it's better to look at both Precision and Recall. F1 score is a metric that combines both precision and recall. It is defined as a simple weighted average (harmonic mean) of precision and recall. If we denote precision using P and recall using R, we can represent the F1 score as:

  F1 = 2PR / (P + R)

- **Confusion Matrix:** A confusion matrix provides a detailed breakdown of true positives, false positives, true negatives, and false negatives, offering insights into the types of errors made by the text extraction model.

- **Intersection over Union (IoU):** IoU measures the overlap between predicted text regions and ground truth text regions. It's calculated as the ratio of the intersection area between the predicted and ground truth bounding boxes to the union area of the two bounding boxes. Higher IoU values indicate better spatial alignment between predicted and ground truth text regions.

- **Character-level Accuracy:** Character-level accuracy measures the accuracy of text recognition within extracted text regions. It calculates the percentage of correctly recognized characters compared to the total number of characters in the ground truth text. This metric focuses on the accuracy of the recognized text content.

- **Word-Level Accuracy:** Similar to character-level accuracy, word-level accuracy measures the accuracy of word recognition within extracted text regions. It calculates the percentage of correctly recognized words compared to the total number of words in the ground truth text.

- **Mean Average Precision (mAP):** mAP is commonly used in object detection tasks and can be adapted for text extraction evaluation. It measures the average precision across different IoU thresholds for text region predictions.

2.5 Experimental Setup

2.5.1 Programming Language:

- Python

2.5.2 Deep Learning Library:

- TensorFlow: TensorFlow is an open-source platform for fast numerical computing. It's a comprehensive and variety of tools, libraries, and other resources that provide workflows with high-level APIs.
- PyTorch: PyTorch is an open source machine learning (ML) framework based on the Python programming language and the Torch library. Torch is an open source ML library used for creating deep neural networks and is written in the Lua scripting language.
- Darknet: Darknet is the original framework for YOLO, and YOLOv2 is implemented within Darknet.

2.5.3 Platform for training the model

- Google Colab (cloud-based with GPU support)

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] Computer Science & Engineering, Gayatri Vidya Parishad College of Engineering for Women, Visakhapatnam, India., K. M. Sai*, H. C. Panuganti, Computer Science & Engineering, Gayatri Vidya Parishad College of Engineering for Women, Visakhapatnam, India., K. Bebe, Computer Science & Engineering, Gayatri Vidya Parishad College of Engineering for Women, Visakhapatnam, India., G. S. R. Pramila, Computer Science & Engineering, Gayatri Vidya Parishad College of Engineering for Women, Visakhapatnam, India., G. S. Rao, and Computer Science & Engineering, Gayatri Vidya Parishad College of Engineering for Women, Visakhapatnam, India., "Optical Character Recognition using CRNN," *International Journal of Innovative Technology and Exploring Engineering*, vol. 9, no. 8, pp. 115–120, Jun. 2020. [Online]. Available: https://www.ijitee.org/portfolio-item/H6264069820/