IE 643 : Deep Learning - Theory and Practice **Aug-Nov 2024**

Project Report: Table and Figure extraction from scanned document

Team Name: Llama Matrix *Roll Number 23N0457*

Abstract

This project addresses the problem of extracting tables and images from scanned documents, aligning the challenge with classical deep learning problem types. The report details the methods employed to approach a potential solution, including a comprehensive exploration of various models used in the project. Key aspects such as the training procedures, inference strategies, and experimental results are discussed in depth. The inference was conducted on both the trained dataset and additional scanned images, demonstrating the system's ability to generalize across different document types. The results of the experiments provide valuable insights into the effectiveness of the proposed approach in accurately extracting structured data from scanned documents.

1 Introduction

Information extraction have been a big task since long and extracting valuable explanation, results, visual diagrams are very crucial while understanding a task. Tables or diagrams or figures present in a random page can provide very high and summarised meaning about an aspect withou providing unnessary context. Tables or tabular information provide an effective way of summarizing key aspects of statistical methods or showcasing important differences and other precise details in research papers, pay scripts, and related documents. Similarly, figures enhance visual understanding of the subject matter, making the precise extraction and retrieval of tabular and visual data critically important. Various OCR models have been used in the past to interpret and extract information from tables [6]. Subsequently, advanced deep learning techniques like image segmentation [2], semantic segmentation [3], and others have emerged to facilitate the accurate extraction of tables and figures from document images. These approaches will be discussed in detail in Section 3. The methods I attempted to implement in pursuit of a solution will be outlined in Section 4. The report also provides details on the experiments conducted in Section 6, with a description of future work in Section 8. Finally, the report concludes with a brief summary and pointers to upcoming work in Section 9.

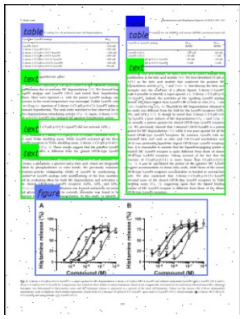

Figure 1: A screenshot of a page with the annotations of Tables, figures and text

2 Project Workflow

The problem involves developing a deep learning technique to accurately extract tabular information and figures from scanned PDF documents. The method must identify pages containing tables and figures, localize them with bounding boxes, and extract the data from tables (including titles and row-column information) and figures (along with captions) into suitable data structures. The challenge includes collecting and preparing appropriate datasets, utilizing pretrained deep learning models, and performing fine-tuning to ensure accurate extraction from diverse and unseen scanned documents. This approach specifically excludes PDFs generated through conversion software like docx2pdf or LaTeX.

Following are the steps I took to try to reach a probable solution:

- Did intensive literature review to find the deep learning area close to bounding box formation and other related problems solved using deep learning techniques.

- Collected data relevant to the project including publaynet, Table bank, Marmot dataset and other which will we explained in more details in section 5

- Considered the problem as semantic segmentation, and utilized the Encoder-Decoder Architecture: TableNet[] which typically uses a fully convolutional neural network (FCN) architecture, with an encoder (e.g., based on VGG or ResNet) for feature extraction and a decoder for segmentation to do inference.

- This implementation shifts towards a specific object localization problem by using bounding box estimation to precisely identify and segment various elements in a document, such as text blocks, tables, and figures. The task is not only about detecting the presence of objects (like tables) but also accurately estimating their location and boundaries through bounding boxes.

- The model used for this purpose is based on Mask R-CNN with an X-101 backbone and FPN architecture. It predicts both bounding boxes and segmentation masks for various document elements, including text, tables, figures, and lists. This model leverages bounding box estimation to accurately localize and categorize these elements, enabling precise extraction of structured information from documents.

- Another model used to simultaneously do the same problem is the Receptive Field Block Net (RFBNet)

- The OCR model used for this purpose is Google Cloud Vision (GCV), which is initiated with specific credentials and configured to recognize texts in specified languages, such as English. This model performs text recognition by extracting textual content from images and returns it in a structured format, including positional information. With its ability to process both simple and complex documents, the GCV model ensures accurate extraction of text blocks, which can be further refined into structured data through post-processing techniques.

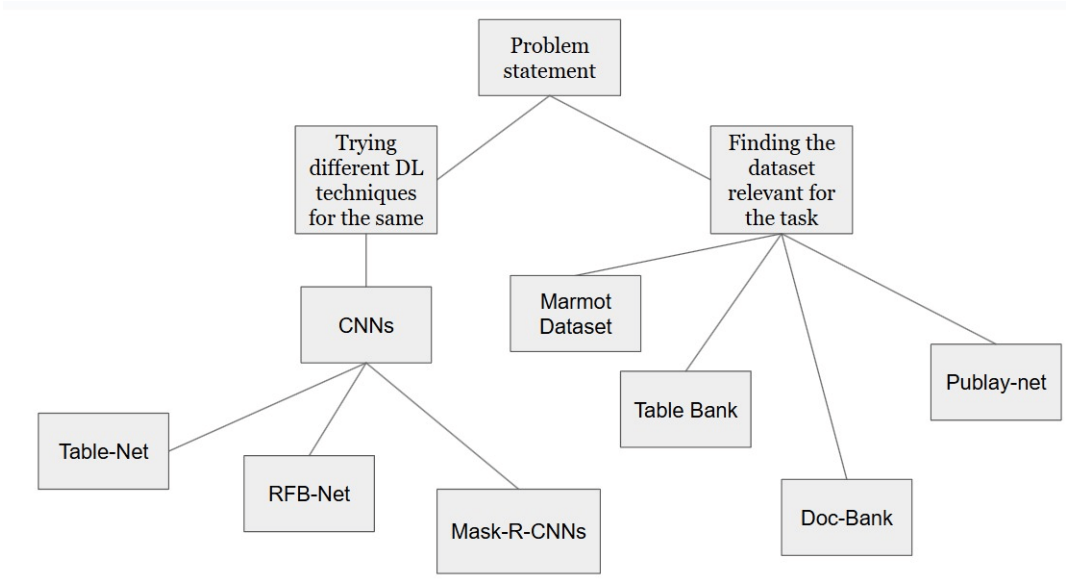

Figure 2: Workflow followed for the project

3 Literature Survey

In document image processing, accurately segmenting elements such as tables or figures involves two main techniques: **object detection** and **instance segmentation**. Object detection focuses on localizing and classifying objects within an image, while instance segmentation extends this by providing pixel-level segmentation of each detected object. This combined approach is essential in applications like document analysis, where structures such as tables need precise localization and boundary delineation. Below, we delve into the technical details, model architectures, and the motivation for choosing **Mask R-CNN** as the primary framework.

Object detection's primary objective is to accurately localize objects by creating bounding boxes around them and classifying them into predefined classes. Each bounding box is represented by four values:

$$(x, y, w, h)$$

where:

- x: The x-coordinate of the bounding box center.
- y: The y-coordinate of the bounding box center.
- w: The width of the bounding box.
- h: The height of the bounding box.

Alternatively, bounding boxes can be represented by the coordinates of their top-left and bottom-right corners:

$$(x_{\min}, y_{\min}, x_{\max}, y_{\max})$$

Each bounding box is associated with a class label vector C, representing the probability distribution over K classes:

$$C = [c_1, c_2, \ldots, c_K]$$

where c_i is the probability that the object belongs to the i-th class. The model is trained using a combined loss function, comprising **classification loss** and **localization loss**.

Classification Loss For multi-class classification, cross-entropy loss is used to compare the true class distribution, $\hat{C}$, and the predicted distribution, C:

$$\text{Classification Loss} = -\sum_{i=1}^{K} \hat{c}_i \log(c_i)$$

where $\hat{c}_i$ is the true label (1 for the true class, 0 for others), and c_i is the predicted probability for each class.

Localization Loss Bounding box regression is typically performed using **Smooth L1 Loss**:

$$\text{Localization Loss} = \sum_{\text{bbox coords}} \text{Smooth}_{L1}(p - \hat{p})$$

where p represents the predicted box parameters and $\hat{p}$ are the ground truth box parameters. The Smooth L1 Loss is defined as:

$$\text{Smooth}_{L1}(x) = \begin{cases} 0.5 \cdot x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise} \end{cases}$$

Intersection over Union (IoU) Loss IoU measures the overlap between predicted and true bounding boxes:

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

IoU Loss is calculated as:

$$\text{IoU Loss} = 1 - \text{IoU}$$

The total loss function L combines classification and localization losses:

$$L = \alpha \cdot \text{Classification Loss} + \beta \cdot \text{Localization Loss}$$

where α and β are weighting factors that balance classification and localization accuracy.

During inference, **Non-Maximum Suppression (NMS)** is applied to refine the bounding boxes by selecting the most confident box for each object:

- Sort the bounding boxes by confidence score.
- Select the box with the highest score and remove boxes with an IoU above a set threshold.

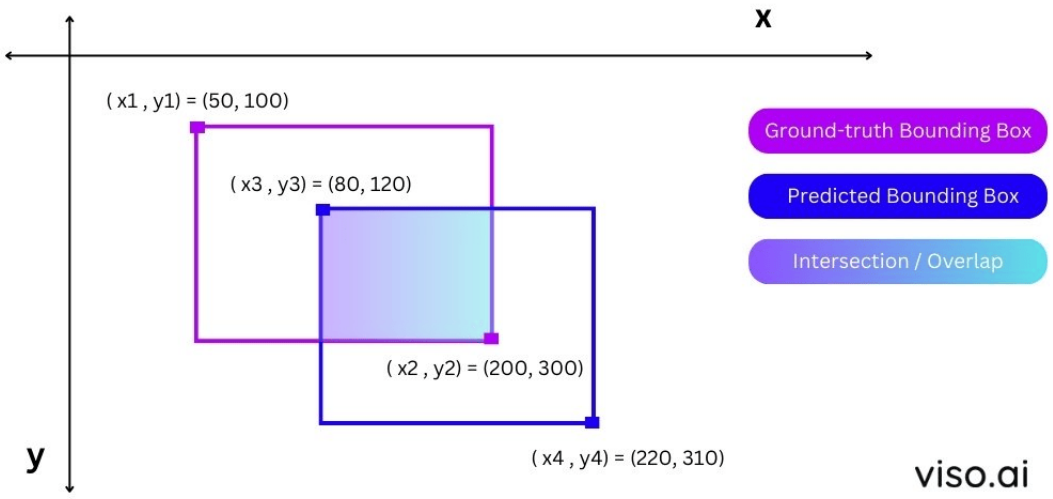

Figure 3: Visual example to understand Ground truth bounding box, predicted bounding box and intersection among them.

Instance segmentation combines object detection with pixel-level segmentation, enabling the model to distinguish individual objects within a class. In this task, each object instance is assigned a separate mask, highlighting the pixels associated with it. This approach is critical for accurately segmenting complex structures in document images, such as tables and figures.

Several architectures have been explored for document image segmentation, each designed to address specific challenges. The following overview discusses three notable models: **TableNet**, **CascadeNet**, and **RBF-Net**, highlighting their applications and limitations.

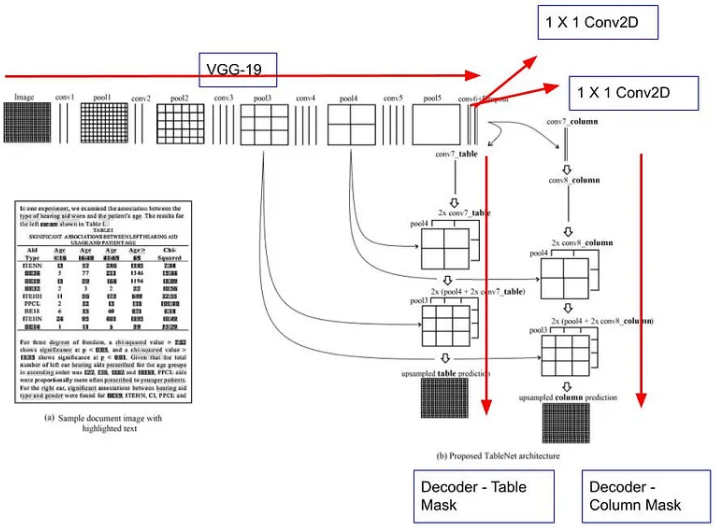

Figure 4: TableNet architecture

- **TableNet**: TableNet is a specialized architecture for segmenting tables in document images, following an encoder-decoder structure. The encoder comprises convolutional layers that extract features from the document, while the decoder generates a binary segmentation mask. TableNet is effective in identifying tables but faces challenges with variable table structures, noise, overlapping elements, and scale variability. Its reliance on annotated data is another limitation, as such datasets are often scarce.

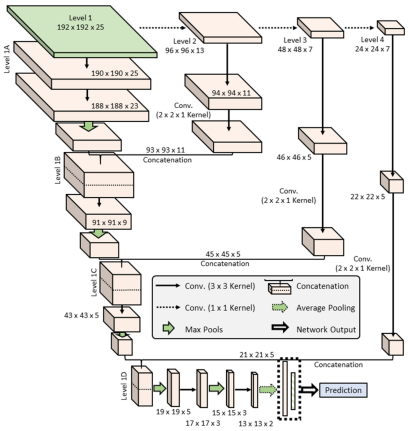

Figure 5: Cascadenet architecture

- **CascadeNet**: CascadeNet is another architecture for table segmentation in document images. It employs a multi-stage process to improve segmentation accuracy by refining results across stages. However, CascadeNet often struggles with varying document layouts, noise, and overlapping elements, and requires significant computational resources and annotated data.

- **RBF-Net**: RBF-Net leverages radial basis functions to enhance segmentation accuracy. In my exploration, I incorporated modifications to RBF-Net, focusing on improving precision by optimizing its radial basis function-based segmentation layers. These modifications retain the architecture's speed and flexibility, addressing some of the limitations of previous models and enabling adaptable segmentation across diverse document types.

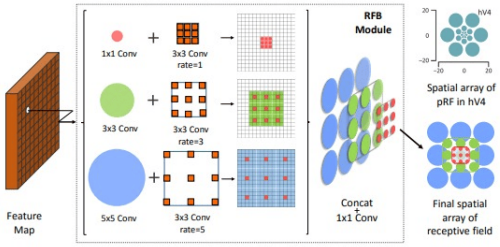

Figure 6: Construction of the RFB module by combining multiple branches with different kernels and dilated convolution layers. Multiple kernels are analogous to the pRFs of varying sizes, while dilated convolution layers assign each branch with an individual eccentricity to simulate the ratio between the size and eccentricity of the pRF.

While the above models provide useful solutions, **Mask R-CNN** addresses their limitations by integrating object detection with high-quality instance segmentation. Earlier models like Fast R-CNN introduced

bounding box detection but lacked segmentation capabilities. TableNet and CascadeNet, although specialized, struggled with complex layouts and overlapping elements. Mask R-CNN introduced a dedicated mask prediction branch and the **Region of Interest (RoI) Align layer** to address these challenges, preserving spatial information for accurate segmentation of tables and figures.

Mask R-CNN extends the Faster R-CNN framework, which utilizes a Region Proposal Network (RPN) to propose bounding boxes for potential objects. Mask R-CNN improves on this by adding a mask prediction branch that performs pixel-level segmentation for each detected object. The architecture consists of three main components:

- **Region Proposal Network (RPN)**: Generates bounding box proposals for objects in the image, filtering through a confidence threshold.
- **Bounding Box Regression Branch**: Predicts refined bounding box coordinates for each proposed region.
- **Mask Branch**: Adds a segmentation mask for each object instance, differentiating between individual instances of the same class.

Additionally, Mask R-CNN uses **RoI Align**, an improvement over RoI Pooling that addresses misalignment issues by preserving spatial information within proposed regions. This feature enhances Mask R-CNN's performance in complex images and documents with intricate layouts.

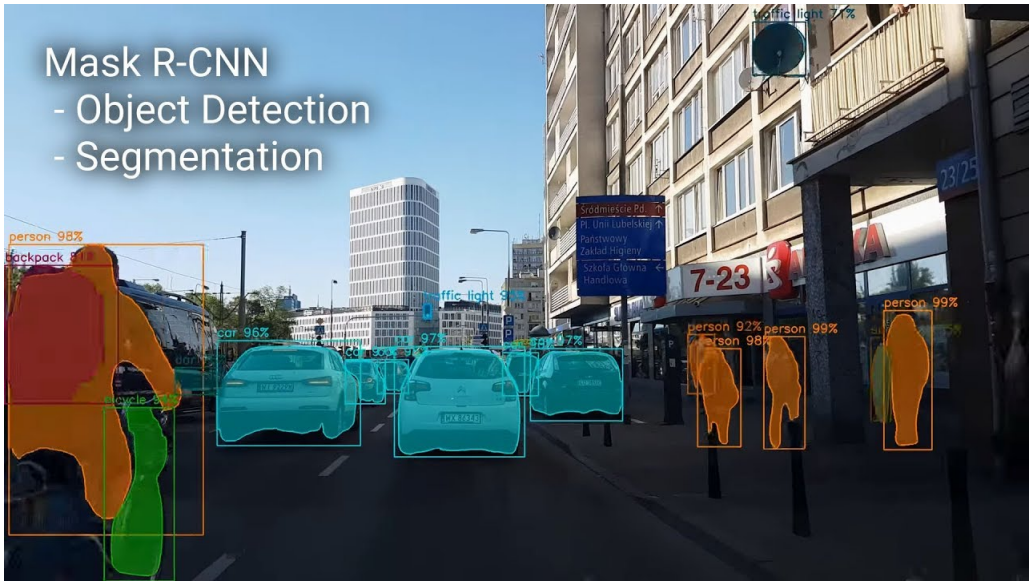

Figure 7: Example of bounding box creation using Mask-R-CNN

The combination of object detection and instance segmentation offers an effective approach for precise table and figure segmentation in document images. Mask R-CNN's adaptable framework addresses the limitations of previous architectures, integrating detection and segmentation tasks with precision. Incorporating modifications in RBF-Net further enhances its performance, achieving higher accuracy and faster processing times. In future work, I will discuss specific modifications made to RBF-Net and their impact on the model's performance.

4 Proposed Approach or Approaches

For the specific task I have used Mask R-CNN architecture[] specifically designed for document layout analysis, trained on the PubLayNet dataset. At its core, the architecture begins with a backbone network, typically a ResNet-101, integrated with Feature Pyramid Networks (FPN).

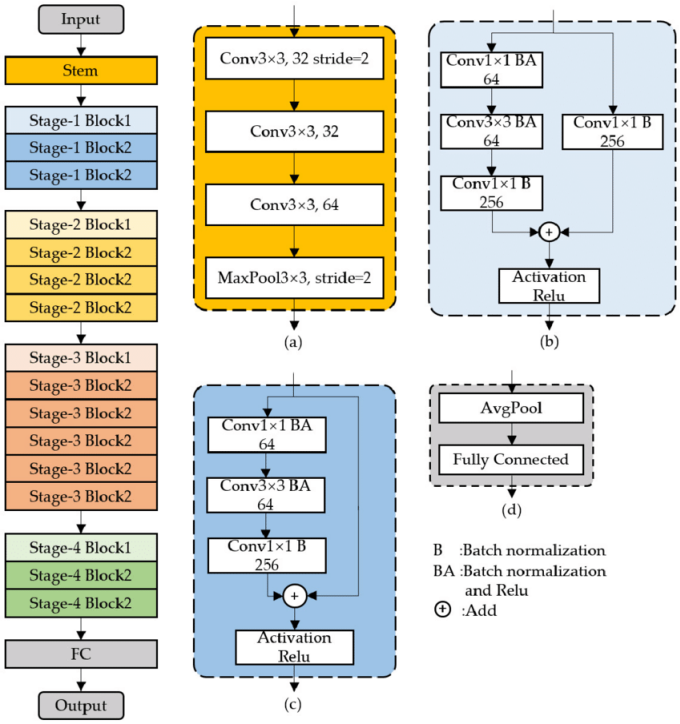

Figure 8: Resnet Backbone Architecture

This backbone serves to extract rich feature representations from input images at multiple scales, allowing the model to capture both fine and coarse details essential for accurately identifying different layout components. Following the backbone is the Region Proposal Network (RPN), which generates region proposals by applying a series of convolutional layers. The RPN predicts potential bounding boxes for objects within the image along with their corresponding objectness scores, effectively identifying regions that likely contain instances of interest. One of the key enhancements of Mask R-CNN over earlier models is the ROI Align layer, which improves upon the traditional RoI Pooling method by preserving spatial information.

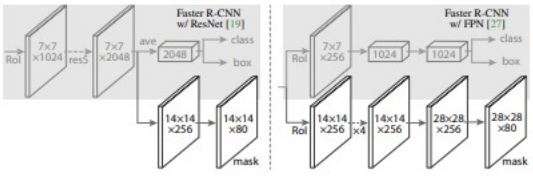

Figure 9: Head Architecture for Mask-R-CNN

ROI Align employs bilinear interpolation to accurately align the extracted feature maps with the original input image, resulting in significantly improved mask predictions and better localization of detected objects.

The architecture also includes ROI heads, which consist of two branches: one dedicated to classification, predicting the class labels for the proposed regions, and another focused on mask prediction, generating high-resolution segmentation masks for each detected object instance. Finally, the model outputs bounding boxes, class labels, and segmentation masks for each detected object, enabling comprehensive layout analysis that includes text, titles, lists, tables, and figures. For a deeper understanding of this architecture and its applications, the original paper titled "Mask R-CNN" by Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross B. Girshick, published in 2018, provides an extensive analysis of the architecture, its components, and experimental results showcasing its effectiveness across various tasks.

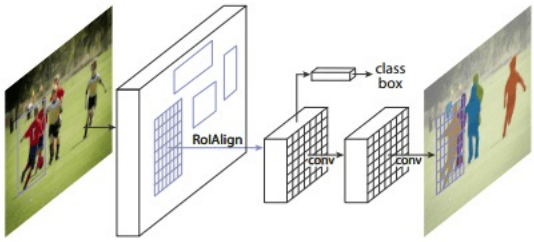

Figure 10: The Mask R-CNN framework for instance segmentation.

5 Dataset Details

- **TableBank** is a large-scale dataset designed specifically for table detection and recognition tasks in documents. It contains over 417,000 samples, with tables extracted from both scientific papers (LaTeX) and Word documents, providing a diverse range of table styles and formats. The dataset supports both object detection (identifying table boundaries) and structural recognition tasks, such as converting tables into structured formats like CSV or HTML. TableBank's vast size and variation make it useful for training models to generalize across multiple document types.
- **Marmot**is a smaller but influential dataset aimed at table detection in PDFs, focusing on the complexities involved in identifying table structures within a document layout. It contains around 2,000 annotated PDF documents from scientific papers in both English and Chinese, emphasizing the challenge of distinguishing between tabular and non-tabular layouts. Marmot serves as a benchmark for fine-tuning table detection models and works well for testing models in academic publishing contexts, where tables may not always be easily distinguishable from surrounding text or figures.

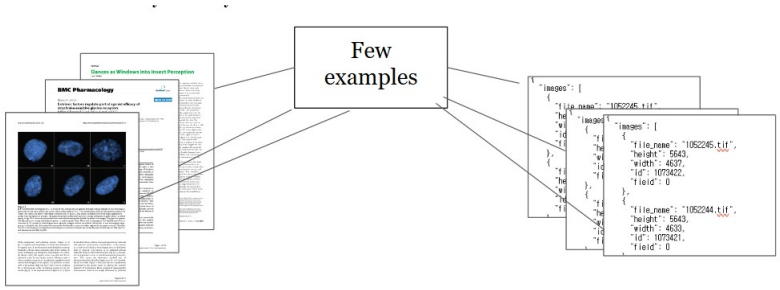

Figure 11: Some samples from the Publaynet dataset along with their annotations in COCO format.

- **DocBank** is a large-scale, multi-format dataset created for document layout analysis. It consists of more than 500,000 document pages, each annotated with layout elements such as titles, paragraphs,

tables, and figures. DocBank's focus is on understanding the semantic structure of documents to enable better extraction of meaningful content.

- **PubLayNet** is a dataset curated from research papers sourced from PubMed Central. It contains around 1 million annotated pages, with labels for tables, figures, text, and equations. PubLayNet's size and scope make it ideal for training deep learning models to detect various document elements and for enhancing the performance of object detection models used in academic or biomedical research.

6 Experiment Setup/ Image preprocessing

The image preprocessing workflow begins by gathering metadata about each image in the dataset, including file path and dimensions. This metadata is essential for linking images with their corresponding annotations and understanding spatial details for further processing. After collecting the image metadata, each image is then associated with its respective annotations, which describe the locations of tables, figures, or other relevant elements within the document. This linkage ensures that every image has an accurate map of document elements, forming the basis for effective training.

Next, the images are resized to a standard dimension, which promotes consistency across the dataset and optimizes memory usage during training. Resizing helps the model process images of uniform size, leading to a more stable training process. Adjustments are made to the bounding boxes associated with each image, ensuring they still accurately represent the locations of annotated elements within the resized dimensions.

To further enhance uniformity, pixel values are normalized, often scaling them between 0 and 1. This normalization ensures consistent lighting and contrast across images, helping the model treat all document parts equally, which in turn improves its ability to detect and differentiate elements regardless of brightness or contrast variations.

Color adjustments are applied to address lighting and document quality differences. Gamma correction adjusts brightness levels, while hue adjustment modifies color tones. These adjustments improve model robustness by simulating various lighting conditions and color casts, accommodating the typical variability found in scanned documents or images captured in different environments.

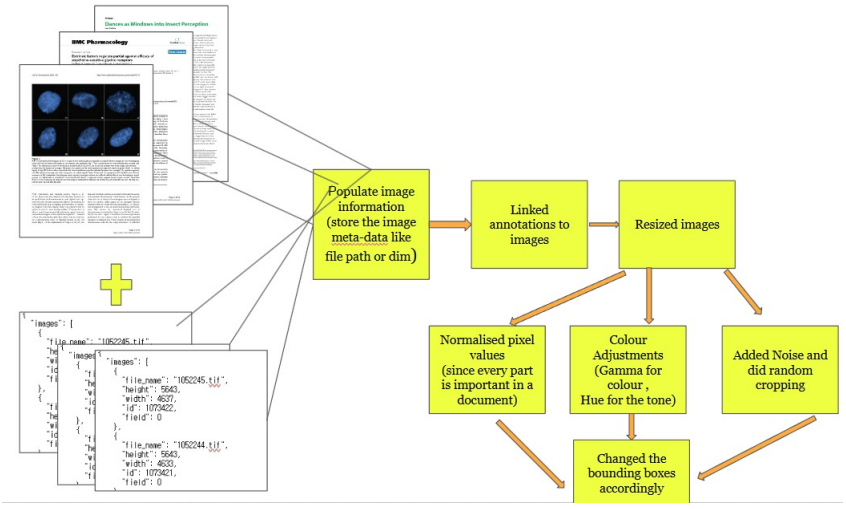

Figure 12: Workflow of the Image processing for all the experiments

To prepare the model for real-world conditions, noise is added to the images, and random cropping is performed. Adding noise simulates common document artifacts, such as scanner dust or pixelation, while random cropping trains the model to handle partial document views. These techniques increase the model's ability to generalize to noisy or incomplete images, which are often encountered in real-world applications.

Finally, since transformations like resizing and cropping affect the image's spatial structure, bounding boxes are recalculated to ensure they accurately encompass annotated elements in their new positions. This ensures that annotations remain precise and aligned with the document's layout even after preprocessing transformations. Through these carefully designed steps, the dataset is refined to enhance diversity and quality, equipping the model to handle various document types, lighting conditions, and real-world document imperfections effectively.

7 Experiments

- The first experiment was conducted on the TableNet architecture, which is specifically designed for table detection and localization tasks in images and scanned documents. To enhance the model's performance, TableBank, a comprehensive and widely recognized dataset for table recognition, was utilized for training. TableBank contains a variety of table formats from academic papers, reports, and documents, providing the necessary diversity and complexity to help the model generalize across different real-world scenarios. The architecture was thoroughly trained on this dataset to ensure it could effectively identify tables, regardless of their size, structure, or format. After the model was trained, a scanned image was selected as the input for testing the model's ability to predict and localize the table within the figure. The goal of this test was to evaluate the model's performance under conditions typically found in real-world applications, such as noise, skewing, or uneven lighting that can occur in scanned images. Through this process, the model attempted to accurately identify the table's presence and determine its precise location within the scanned figure, demonstrating the potential of the TableNet architecture for table detection and localization tasks in challenging scenarios.

- The second experiment involved training the RFBNet architecture from scratch, using a ResNet backbone for feature extraction on the PubLayNet dataset. PubLayNet, a large-scale dataset created for document layout analysis, contains a diverse set of document images with annotated layouts, including tables, text, and images, making it well-suited for object detection and segmentation tasks in document processing. Training RFBNet on this dataset aimed to enable the model to capture complex spatial dependencies and fine-grained features within document layouts, leveraging ResNet's strong feature extraction capabilities. After training, the model was evaluated on unseen document images to assess its ability to accurately detect tables and other layout elements under varied document styles, layouts, and print qualities. This experiment was instrumental in understanding how effectively RFBNet could perform document layout analysis tasks, such as table detection, across heterogeneous document types.

- The third experiment focused on training a Mask R-CNN architecture on the PubLayNet dataset for instance segmentation of document components, such as tables and figures. Mask R-CNN was chosen for its robust instance segmentation capabilities, which go beyond simple object detection by generating pixel-wise masks for each detected object. Initially, the model was trained on a curated set of images from PubLayNet, where each document element was annotated for precise segmentation. To further enhance its performance, the model was then fine-tuned using a custom-annotated dataset specifically prepared to meet unique project requirements. This additional fine-tuning step enabled the model to adapt more accurately to the specific document types and styles expected in the deployment environment. Post-training evaluations involved testing on a set of new documents to verify the model's performance in segmenting and localizing tables and figures accurately, even in cases where overlapping or complex layouts were present. This experiment demonstrated the capability of Mask R-CNN in

handling document segmentation tasks effectively, highlighting its adaptability to challenging document structures and custom requirements.

8 Results

The **RBFNet** on the publaynet using a noise-enhanced softmax and dropout for robustness. Training losses used are Multibox, DIOU, and EIOU losses to handle classification and location regression for improved detection accuracy. Ranger21 optimizer with adaptive gradient clipping and L2 regularization to control weight updates effectively. Tracked classification accuracy, recall, IoU, and bounding box metrics to assess detection quality.

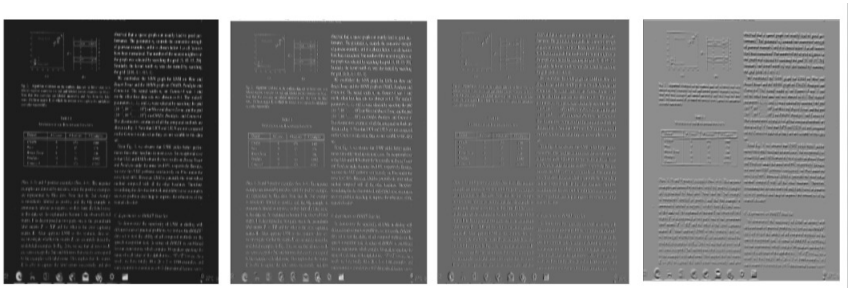

Figure 13: Some feature maps from different convolution layers of RFBnet

Learning Rate Adjustment was done using PolyLR scheduler which decays the learning rate gradually, enhancing model stability. Incorporated gamma, hue, contrast, and saturation adjustments along with noise to improve model generalization. Then the model was tested using some of the images from the test set and some picture scanned by me using Adobe scanner. Here I have included the kernel images and convolution layer feature maps using the RFBs. Also the training losses is included for this experiment using a curve.

This progression in the feature maps suggests that the network is effectively learning to emphasize key structures, like tables and figures, over other document elements as it goes deeper. By the final feature map on the right, the model appears to retain only the most prominent structural features (e.g., boundaries or grid-like patterns in tables and figures), indicating that it is filtering out less relevant details. This behavior aligns with the network's goal of detecting tables and figures, as it gradually isolates the patterns most characteristic of these elements while ignoring surrounding text and irrelevant details. Overall, these feature maps demonstrate the network's ability to distill complex document images into the essential features needed for identifying figures and tables.

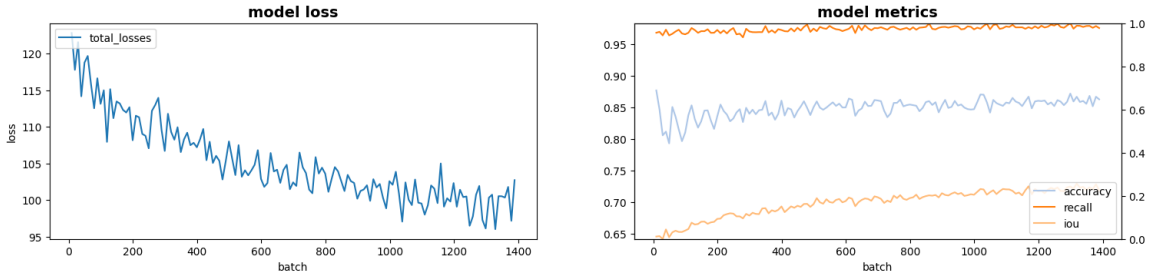

Figure 14: Loss and different metric curve for RBFnet

Figure 15: Different kernel views for different channels

The left set of kernels shows diverse patterns and intensity variations, suggesting that these kernels have effectively learned to detect various features relevant to identifying figures and tables, such as edges, textures, or distinct shapes. This variation is crucial for capturing the structural details necessary for distinguishing these elements in documents. In contrast, the right set of kernels appears uniformly dark, with no visible patterns, indicating that these kernels may not be actively contributing to the detection task. This lack of activation suggests a possible underutilization of these kernels in the learning process, which could limit the network's ability to fully capture the complexities of figures and tables in document images. The observed disparity implies that while some kernels are effectively engaged in feature detection, others might require adjustments in training or architecture to improve overall performance in figure and table detection.

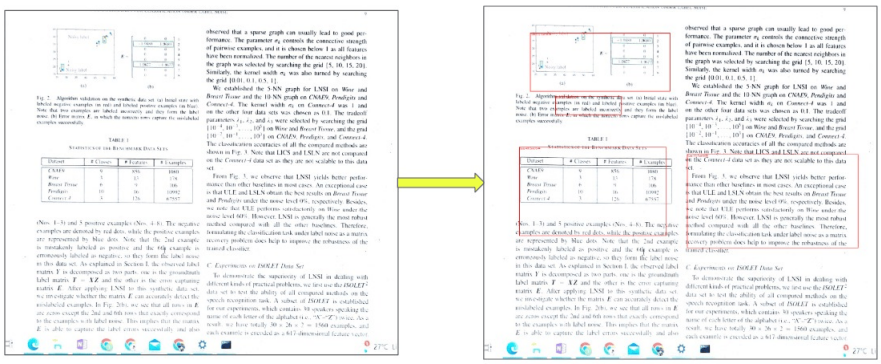

Figure 16: Bounding box detection around the figure, the tables and the texts present in the document using learned RFBnet architecture on publaynet dataset

The figure 16 shows the bounding box detection using the RBFnet architecture as discussed above.

In Mask-R-CNN model configuration, I utilized three loss functions to optimize accuracy at various stages of instance segmentation. For classification loss, I employed a standard cross-entropy loss to ensure correct labeling of document elements. To achieve precise localization of bounding boxes, I used Smooth L1 loss, which effectively penalizes minor localization errors and improves bounding box accuracy. Additionally, binary cross-entropy was applied to refine pixel-level mask predictions, enabling detailed segmentation of individual objects within document images. For feature extraction, I integrated a ResNeXt-101 (X-101-32x8d) backbone, a deep network architecture that captures intricate features, particularly beneficial for extracting fine details in document elements such as text and tables. Paired with a Feature Pyramid Network (FPN), this architecture allowed the model to detect elements at multiple scales, from small text blocks to large tables, by fusing features from different resolutions. To enhance result reliability, I set a score threshold of 0.6, filtering out low-confidence detections and thereby reducing false positives.

These figure 17 and figure 18 displays a series of feature maps from different convolution layers within the Mask R-CNN architecture, showing how the model progressively extracts and refines features from the input images.

Each row represents feature maps from specific layers with varying depths, such as layers with 2048, 1024,

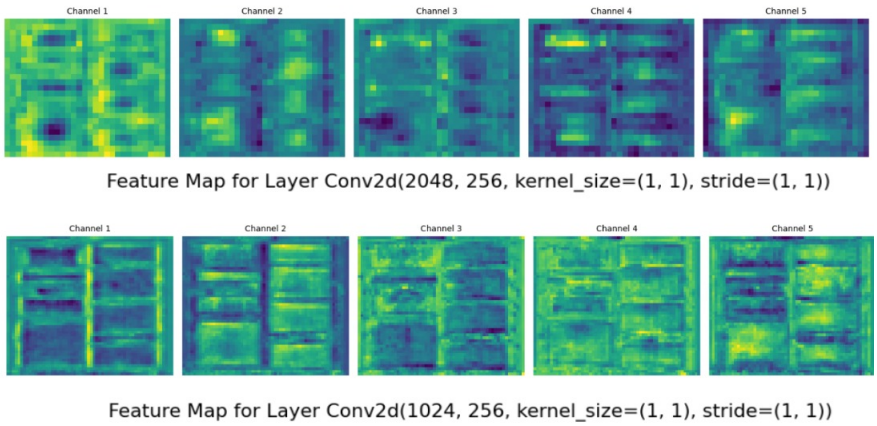

Figure 17: Some feature maps from different convolution layers of Mask-R-CNN

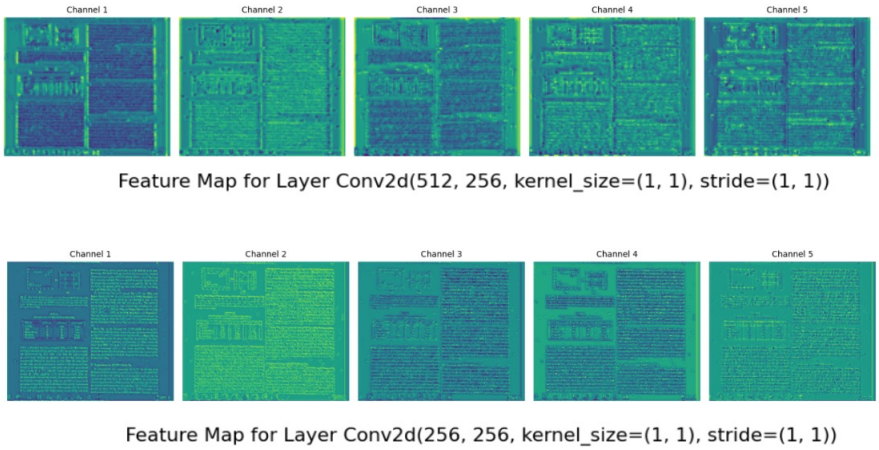

Figure 18: Some feature maps from different convolution layers of Mask-R-CNN

512, and 256 channels. At the higher levels (e.g., Conv2d(2048, 256)), the feature maps are more abstract and less detailed, capturing broad, high-level patterns and shapes. As we move to lower layers, like Conv2d(256, 256), the feature maps become more detailed, focusing on finer features that capture specific textures, edges, and structural details within the document elements. This hierarchical feature extraction is essential in Mask R-CNN, as it allows the model to recognize both large, general patterns and small, intricate details, making it well-suited for detecting and segmenting complex objects, such as text blocks, tables, and figures in document images.

The figure 19 displays the convolutional kernels (filters) from two layers, fpn-output2.weight and fpn-output4.weight, within the Feature Pyramid Network (FPN) in the Mask R-CNN architecture. Each small matrix (or kernel) represents a pattern-detecting filter that captures specific visual features in the input image, such as edges, textures, and shapes. The varying shades of gray within each kernel correspond to different weight values, with darker and lighter areas reflecting lower and higher values, respectively. These kernels slide over the feature maps, applying a convolution operation that accentuates certain patterns and details in the data.

By stacking multiple kernels, each layer captures a variety of features that contribute to the model's ability

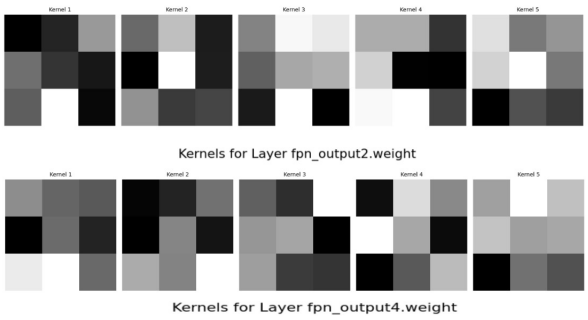

Figure 19: Different kernel views for different channels

to recognize objects at different scales and orientations. The FPN's hierarchical structure leverages these layers to create a multi-scale representation, enhancing the model's capability to detect both small and large objects across complex document layouts. This visualization illustrates how the kernels work together to build a rich feature map, ultimately improving object detection and segmentation accuracy.

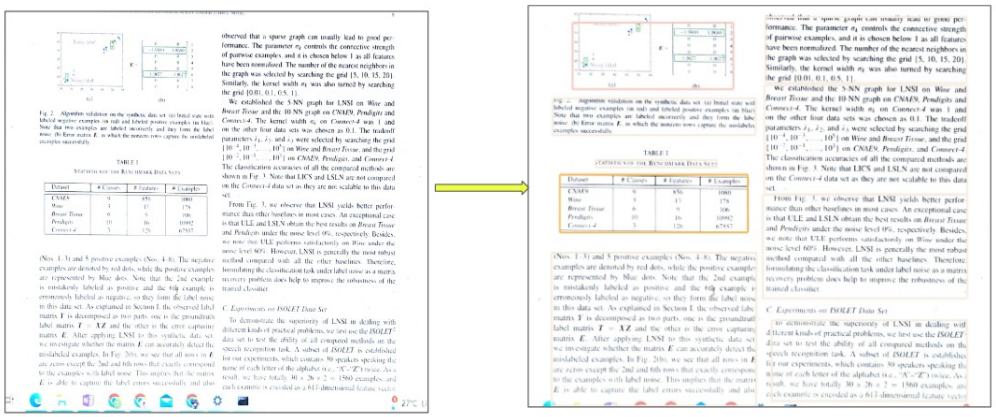

Figure 20: Final bounding box detection result using the Mask-R-CNN architecture

Figure 20 illustrates the final bounding box representation produced by the Mask R-CNN model architecture, applied to a document image with multiple pages. The left side of the figure shows the original input, a document with several columns of text, tables, and graphical elements. After processing this image through the Mask R-CNN, the model detects individual regions of interest, as seen on the right side. These regions are outlined by bounding boxes that encapsulate specific sections of the document, such as text blocks, tables, and figures.

The bounding boxes represent different channels within the model, each of which focuses on capturing particular types of content. This segmentation allows for the precise localization and categorization of various document elements, distinguishing between textual and graphical content. By leveraging multiple channels, the model captures the fine-grained structure within complex document layouts, enabling accurate identification and segmentation of key components. This feature is especially valuable for tasks like automated document analysis, where identifying individual sections (such as figures and tables) is crucial for structured data extraction. The detailed bounding box representation provided by Mask R-CNN highlights its capability to handle intricate, multi-element documents with high accuracy.

9 Plan for Novelty Assessment

I plan to approach it systematically by first creating a pipeline for all the steps involved. This structured pipeline will guide the process from start to finish, ensuring that each phase, from data preparation to model evaluation, is clearly defined and executed efficiently. Next, I will understand the model architecture in more detail, delving into the components and mechanisms of the chosen models to gain a deep, comprehensive insight. This step will allow me to make informed decisions during model configuration, fine-tuning, and evaluation. Finally, I will be confident in my approach and findings, knowing that a thorough understanding and a well-organized pipeline back my work. This confidence will enable me to present results effectively and make adjustments with assurance as the project progresses.

10 Conclusion

The Mask R-CNN model offers a slight edge in precision compared to the RFBNet, making it well-suited for tasks that require detailed segmentation and accurate localization of objects within an image. However, this precision comes at the cost of training speed, as Mask R-CNNs generally require more resources and time to train due to their complex architecture and additional mask prediction branch. In contrast, RFBNet is faster to train and more lightweight, providing a more efficient option for scenarios where speed is a priority and a slight compromise in accuracy is acceptable. Fine-tuning the RFBNet on a custom dataset could be a practical alternative, allowing for an effective balance between training efficiency and model performance. By adapting RFBNet to the specific characteristics of the dataset, it could yield satisfactory results while maintaining the benefits of faster training. This approach would make RFBNet a viable choice when working with limited resources or tighter timelines.

11 References

- V7labs. *https://www.v7labs.com/blog/image-segmentation-guide*. Accessed on 10th Oct.
- Siddhesh. *https://medium.com/@siddheshb008/vgg-net-architecture-explained-71179310050f*. Accessed on 3rd Sept.
- Qisheng Huang, Chunming Zhao, Ming Jiang, Xiaoming Li, Jing Liang. *Cascade-Net: a New Deep Learning Architecture for OFDM Detection*, 2018. *https://arxiv.org/pdf/1812.00023*.
- Shubham Paliwal, Vishwanath D, Rohit Rahul, Monika Sharma, Lovekesh Vig. *TableNet: Deep Learning Model for End-to-End Table Detection and Tabular Data Extraction from Scanned Document Images*, 2020. *https://arxiv.org/pdf/2001.01469*.
- Songtao Liu, Di Huang, Yunhong Wang. *Receptive Field Block Net for Accurate and Fast Object Detection*, 2020. *https://arxiv.org/pdf/1711.07767v3*.
- Kaiming He, Georgia Gkioxari, Piotr Dollar, Ross Girshick. *Mask R-CNN*, 2018. *https://arxiv.org/pdf/1703.06870*.
- Maryam Bahami. *https://towardsdatascience.com/generalized-iou-loss-for-object-detection-with-torchvision-9534029d1a89*. Accessed on 17th Oct, 2024.
- Firiuza. https://firiuza.medium.com/roi-pooling-vs-roi-align-65293ab741db. Accessed 15th Oct, 2024
- Kaiming He Xiangyu Zhang Shaoqing Ren Jian Sun. Deep Residual Learning for Image Recognition, 2015. https://arxiv.org/pdf/1512.03385

- Tiba Razmi. https://medium.com/@tibastar/mask-r-cnn-d69aa596761f. Accessed 20th Oct, 2024