Semantic Segmentation of PDF Document Characteristics for Rapid Printer Configuration

Shakil Ahmed Sumon
EECS
Oregon State University
Corvallis, OR
sumons@oregonstate.edu

Soon Song Cheok
EECS
Oregon State University
Corvallis, OR
cheoks@oregonstate.edu

Abstract

While printing large volume of documents with varying ink concentrations, printing presses often resort to manual trial and error to adjust printer settings, resulting in significant paper and time wastage. The limited automated systems available require computational simulation for precise counting of ink droplets to measure page ink concentration, a process that is highly time-intensive. In this work, we leverage deep learning techniques to speed up the analysis to measure various ink concentrations. In particular, we treat the different ink concentrations on a document page as distinct features and apply semantic segmentation to identify the presence of these characteristics in the document pages. We experiment with several promising existing deep learning architectures specifically designed for the task of semantic segmentation, including U-net, E-net, DeeplabV3 and DeeplabV3+. The empirical exploration suggests that U-net performs better in terms of Intersection over Union (IoU) and also is an order of magnitude faster than the available methods for measuring ink concentrations.

1 Introduction

In the modern printing industry, the demand for high-quality prints with minimal waste and efficient resource utilization is paramount. Large-scale printing operations containing diverse document types with varying ink concentrations face significant challenges in optimizing printer settings. Traditionally, this optimization has been achieved through manual trial and error, which is both time-consuming and resource-intensive, leading to substantial paper and ink wastage.

Automated systems currently available for optimizing printer settings typically rely on computational simulations to count ink droplets and measure page ink concentrations accurately. While these methods can provide precise measurements, they are highly time-intensive and computationally expensive, making them impractical for high-volume, real-time printing operations.

Recent advancements in deep learning, particularly in the field of computer vision, offer appealing solutions to this problem. Deep learning techniques have shown remarkable success in various image analysis tasks, including object detection, image classification, and semantic segmentation. Semantic segmentation, which involves partitioning an image into semantically meaningful regions, can be particularly useful for analyzing document pages to identify and measure different ink concentrations.

This project aims to leverage deep learning techniques to develop a rapid and efficient method for measuring ink concentrations on document pages. By treating different ink concentrations as distinct features, we propose using semantic segmentation to identify these characteristics. This approach can significantly reduce the time and computational resources required for optimizing printer settings, thereby minimizing waste and improving efficiency in printing operations. To achieve this, we

Preprint. Under review.

experiment with several state-of-the-art deep learning architectures specifically designed for semantic segmentation. Among these, we focus on U-net, E-net, DeeplabV3, and DeeplabV3+. Each of these architectures has been selected for its proven effectiveness in various segmentation tasks, and we evaluate their performance in the context of measuring ink concentrations on document pages.

Our empirical exploration reveals that U-net outperforms the other architectures in terms of Intersection over Union (IoU), a standard metric for evaluating segmentation quality. Additionally, E-net demonstrates a significantly faster processing time, making it an ideal candidate for real-time applications in the printing industry.

The paper is structured as follows: Section 2 provides a brief background of the problem, section 3 reviews related work in the fields of printer optimization and semantic segmentation, section 4 details the deep learning architectures we explore, section 5 outlines the experimental setup, section 6 presents the results of our experiments, and, section 7 concludes the paper.

2 Problem Background and Related Work

In the printing industry, document pages can be categorized into two types based on the printing process: simplex and duplex. A simplex page is printed on one side only, whereas a duplex page is printed on both sides. Understanding the characteristics of these pages is crucial for optimizing printer settings, especially when dealing with varying ink concentrations.

Characteristics	**Description**
High Moisture Page Simplex	This type involves high moisture across the entire page. The main characteristic of interest is identifying pixels with moisture levels above a certain threshold.
High/Low Moisture Pages (Duplex Delta)	This involves a duplex page where one side has high moisture, and the other side has low moisture.
Concentrated High Moisture Simplex (0.5in sq)	This involves very high moisture areas in a concentrated object larger than 0.5 inches by 0.5 inches.
Concentrated High Moisture Simplex (1in sq)	Similar to the previous type, but with the object size threshold increased to 1 inch by 1 inch.
Concentrated High Moisture Duplex (0.8in sq)	This involves high moisture areas in a concentrated duplex object larger than 0.8 inches by 0.8 inches.

Table 1: PDF Characteristics and their Descriptions

Table 1 gives an overview of the characteristics of interest for this project. High moisture concentration in specific areas of a simplex or duplex page is problematic because it requires more ink in those areas, which can lead to uneven drying and reduced print quality. The excess moisture can cause issues such as smudging, warping, or streaking, impacting the overall smoothness and appearance of the printed document.

To address these challenges, we are approaching the problem as a semantic segmentation task. Each characteristic described in Table 1 is treated as a distinct feature that needs to be detected within the document. We have converted each PDF page to an image and provided these images as inputs to our segmentation model. Unlike traditional models that use three-channel RGB inputs, our model takes a five-channel input. These channels include the CMYK color channels plus an additional transparent color channel used by HP printers to balance ink distribution across the pages.

To the best of our knowledge, this is the first initiative to address ink concentration detection across a document page as a semantic segmentation problem. However, semantic segmentation has been previously applied to detect other document-level features. For example, (1) used semantic segmentation to classify document elements into categories such as body text, comments, decoration, and background. They extracted features using VGG-M (2) and then leveraged SVM to classify these features into the respective classes. Additionally, (3) employed hierarchical semantic segmentation to extract document structures, focusing exclusively on documents containing forms.

3 Architectures

In our project, we have experimented with four architectures to address the task of semantic segmentation: U-Net, E-Net, DeepLabV3, and DeepLabV3+. This set of architectures were chosen for their diverse strengths and capabilities, offering a comprehensive evaluation of different approaches to segmentation.

U-Net (4) is a convolutional neural network architecture primarily designed for biomedical image segmentation. It was introduced in 2015 by Olaf Ronneberger et al. The architecture consists of a contracting path to capture context and a symmetric expanding path that enables precise localization. The contracting path follows a typical convolutional network structure. The expanding path consists of upsampling and concatenation operations that help recover spatial information lost during downsampling. U-Net is chosen for its ability to produce high-resolution segmentation maps, making it well-suited for our task where fine details are crucial.

On the other hand, E-Net (5), a variant of U-net, is designed for real-time semantic segmentation. It focuses on efficiency and speed while maintaining a reasonable level of accuracy. E-Net achieves this through a lightweight architecture with fewer parameters compared to more complex networks. It utilizes various techniques such as early downsampling, bottleneck layers, and skip connections to balance speed and performance. E-Net is particularly useful for applications where real-time processing is essential, such as autonomous driving or mobile applications. In our project, we have chosen this architecture to investigate the trade-off between speed and accuracy.

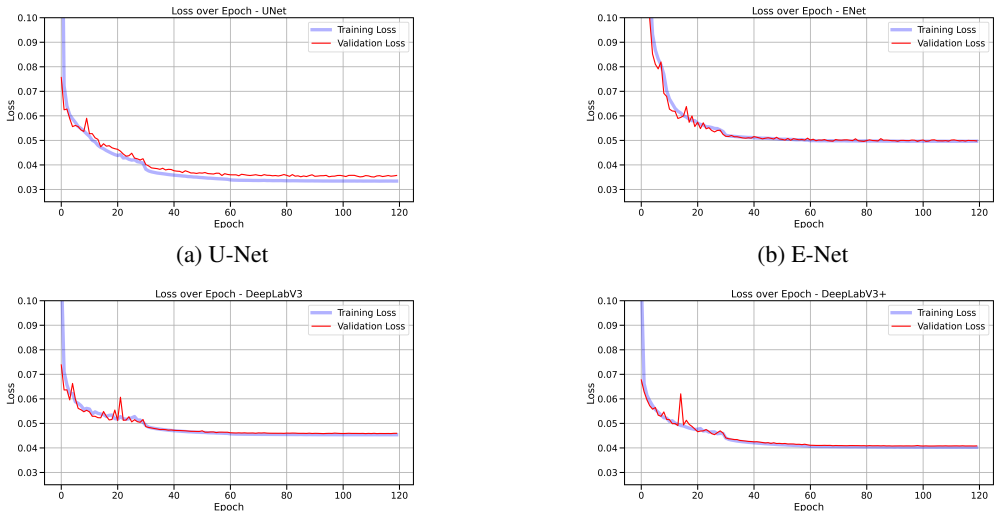
Additionally, we experimented with DeepLabV3 (6), a convolutional neural network architecture that employs atrous convolution (also known as dilated convolution) (8) to capture multi-scale context by adjusting the receptive field. Introduced by Chen et al. in 2017, DeepLabV3 uses atrous spatial pyramid pooling (ASPP) (9) to incorporate multi-scale information and improve segmentation performance. ASPP combines features from different atrous rates, effectively capturing context at various scales. DeepLabV3 is selected for its capability to capture multi-scale context effectively through atrous convolutions and ASPP. This architecture balances accuracy and computational efficiency, making it suitable for our segmentation task as it requires capturing fine details at various scales.

The last architecture we used is DeepLabV3+ (7), that builds upon the success of DeepLabV3 by integrating an encoder-decoder structure. Introduced in 2018, it combines the strengths of atrous convolutions and the spatial pyramid pooling module with the benefits of a decoder path that refines segmentation results. The encoder captures context through atrous convolution, while the decoder recovers spatial details through upsampling and convolutional layers. DeepLabV3+ enhances the localization ability and accuracy of the segmentation, making it highly effective for applications requiring detailed segmentation maps.

4 Experimental Setup

In this work, PDFs are collected and converted into input images and labeled images. The criteria for selecting PDFs are that they must be accessible and come from public domain websites, such as government websites. They are collected using Scrapy and the Google search engine. A Python script retrieves hyperlinks to PDF files from the Google search engine, these links are then passed to Scrapy to access and download the PDFs through the links. Once the data is collected, it is processed with HP's algorithm to generate input and label images from raw PDFs. HP has a very precise method for generating the bitmaps that takes .47 seconds on average to process one page. The input images are grayscale images for different channels, and the label images represent the ink droplet concentration in the image.

After collecting PDFs, the number of samples in our dataset is approximately 11,000, with each sample representing a page in a PDF file. The dataset is further tripled by applying data augmentation techniques such as horizontal and vertical flips, bringing the total number of samples to approximately 33,000. The dataset is then split in a 7 to 3 ratio for the training (which includes the validation dataset) and testing datasets. The training dataset is further divided into training and validation datasets at a ratio of 9 to 1.



(a) U-Net

(b) E-Net

(c) DeepLabV3

(d) DeepLabV3+

Figure 1: Training and validation loss graphs for the four models.

There are two primary metrics used in this work to measure the model's performance: IoU and F1 score. Both of these metrics are commonly used to assess semantic segmentation performance. IoU measures the overlap between the predicted area and the ground truth area as a ratio.

$$IoU = \frac{\text{Number of pixels in the Overlap}}{\text{Number of pixels in the Union}}$$

The F1 score is defined as the equation below.

$$\text{F1 score} = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Both metrics are great for measuring image segmentation models. While the F1 score is more versatile and can be used for any binary classification task, IoU is used when spatial information is crucial.

For the training hyperparameters, the models are trained with a learning rate of 1e-4, a batch size of 8, and 120 epochs. In addition to these hyperparameters, L2 regularization with a weight decay factor of 0.1 is employed to avoid model overfitting. Learning rate decay, with a decay factor of 0.1 that occurs every 30 epochs, is employed to ensure better convergence. ADAM is used as the optimization technique for training the models.

5 Results and Discussions

This section presents the results of our experiments with the four segmentation architectures: U-Net, E-Net, DeepLabV3, and DeepLabV3+. Our objective was to evaluate and compare the performance of these models, assessing their intersection over union (IoU), efficiency, and overall effectiveness.

In our experiment, we tracked the training and validation losses of the models over a series of 120 epochs. The resulting graphs, reported in figure 1 provide insights into the learning dynamics and generalization capabilities of each model. The graphs indicate that all four models undergo an initial phase of rapid learning, as evidenced by the steep decline in losses. The fluctuations in validation loss, particularly in DeepLabV3 and U-Net, suggest moments of overfitting, which are subsequently corrected. The eventual stabilization of both training and validation losses across all models suggests that they all achieve a satisfactory level of generalization, with U-Net performing better as indicated by its superior performance in validation loss.

The superior generalizability of U-net is also evidenced by figure 2 as it achieves the lowest validation loss. DeepLabV3 and DeepLabV3+ also perform well, with slightly higher but stable validation

losses. E-Net, while efficient and quick to learn, stabilizes at a higher validation loss, suggesting that its real-time performance trade-offs somehow handicaps its ability to do well on the task.

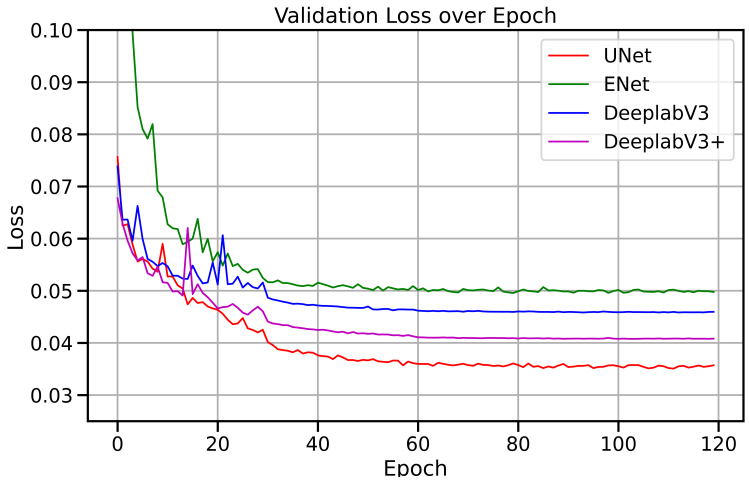


Figure 2: Validation losses over epochs for U-Net, E-Net, DeepLabV3, and DeepLabV3+

The trend of generalization ability in terms of validation loss, as observed in the previous graph, is supported by the model's performance in terms of Intersection over Union (IoU) and F1 score, as detailed in Table 2.

The data indicates that U-Net outperforms the other models, achieving the highest IoU of 78.61 and an F1 score of 87.35. This aligns with the validation loss graph, where U-Net showed the lowest final validation loss, reflecting its strong generalization ability and robustness in handling the segmentation task. DeepLabV3+ follows closely, with an IoU of 75.28 and an F1 score of 84.92, demonstrating its effectiveness in capturing fine details and spatial context, albeit slightly less efficient than U-Net.

Models	**IoU**	**F1 score**
Unet	**78.61**	**87.35**
DeeplabV3+	75.28	84.92
DeeplabV3	73.36	83.53
Enet	71.74	82.48

Table 2: IoU and F1 scores for the segmentation models

DeepLabV3, while showing a stable learning curve, achieves a slightly lower IoU of 73.36 and an F1 score of 83.53. This model's performance indicates solid generalization but highlights the improvements brought by the encoder-decoder structure in DeepLabV3+. E-Net, designed for real-time performance, shows the lowest IoU (71.74) and F1 score (84.92), reflecting the trade-off between efficiency and accuracy. Despite its quick learning and stable performance, E-Net's lightweight architecture is less capable of capturing complex segmentation details compared to the other models.

Models	**SimplexPage**	**SimplexObject1.3cm**	**SimplexObject2.5cm**	**DuplexObject2.2cm**	**DuplexPageDelta**
Unet	**73.31**	**93.62**	**95.45**	90.09	**58.45**
DeeplabV3+	66.21	92.14	94.77	**90.71**	54.90
DeeplabV3	63.20	89.96	93.46	89.90	52.98
Enet	65.53	86.14	86.46	82.87	51.15

Table 3: Per-class IoU metrics for different segmentation models

To gain a more granular understanding of how each model performs on specific segmentation tasks, we examine the per-class IoU metrics. As IoU and F1 score generally correlate with each other, we

will report only the per-class IoU metrics, which we believe will serve as a suitable proxy for the F1 score as well. Table 3 provides a detailed breakdown of the per-class IoU for each of the segmentation models: U-Net, DeepLabV3+, DeepLabV3, and E-Net.

From the table, it is evident that U-Net consistently achieves the highest IoU across most classes, reinforcing its overall strong performance. Specifically, U-Net excels in segmenting SimplexObject2.5cm and SimplexObject1.3cm with IoU scores of 95.45 and 93.62, respectively. This high performance in detailed object segmentation indicates U-Net's robustness and ability to capture fine details, which translates to higher accuracy in these categories. DeepLabV3+ follows closely behind U-Net, particularly excelling in SimplexObject2.5cm with an IoU of 94.77 and SimplexObject1.3cm with 92.14. While it performs slightly lower than U-Net, DeepLabV3+ still demonstrates strong generalization and precise segmentation capabilities, particularly in classes requiring high spatial detail.

DeepLabV3 shows competitive performance but falls short of U-Net and DeepLabV3+ in most categories. Its best performance is also in SimplexObject2.5cm with an IoU of 93.46, showing that while it is effective, the additional enhancements in DeepLabV3+ provide a notable boost. E-Net, designed for efficiency, shows the lowest IoU scores across all categories. Although it performs reasonably well in SimplexObject2.5cm and SimplexObject1.3cm with scores of 86.46 and 86.14, respectively, its performance in DuplexObject2.2cm and DuplexPageDelta is considerably lower. This highlights the trade-off between computational efficiency and segmentation accuracy.

It is evident that U-Net outperforms all other models, which can be explained through the differences in their architecture. U-Net's unique design effectively captures both local and global information, thanks to its contracting path (encoder) and symmetric expanding path (decoder) connected by extensive skip connections. These connections help preserve spatial information, allowing U-Net to maintain fine details while understanding the broader context, resulting in superior segmentation performance. In contrast, DeepLabV3+ includes a simpler decoder that, while enhancing spatial detail recovery compared to DeepLabV3, does not match U-Net's effectiveness. DeepLabV3, although employs dilated convolution to preserve spatial resolution - lacks a dedicated decoder. Absence of a dedicated decoder results in losing local information as convolutional layers stack on each other, impacting its ability to retain fine details. E-Net, designed for real-time performance, is not powerful enough to capture and retain detailed information required for high-accuracy segmentation tasks. Thus, the architectural strengths of U-Net, particularly its sophisticated encoder-decoder structure with skip connections, explain its superior performance in our experiments.

To further investigate the performance of the models, we present a visual comparison of the segmentation outputs from each model in figure 3. From these visualizations, it is evident that U-Net captures finer details more effectively than the other models. Additionally, the visual comparison reveals that none of the models are sufficiently capable of capturing the necessary details to accurately generate the segmentation mask for the duplex page class.

Additionally, we report the inference speed of each model, as it is a critical factor, especially for applications requiring real-time processing. Table 4 compares the inference speeds of U-Net, DeepLabV3+, DeepLabV3, and E-Net in terms of pages processed per second.

Models	**Inference Speed (pages per second)**
Unet	239.15
DeeplabV3+	346.22
DeeplabV3	268.38
Enet	**415.5**

Table 4: Inference Speed for the segmentation models

From the table, we observe that E-Net has the highest inference speed, processing 415.5 pages per second. This makes E-Net the most efficient model in terms of speed, aligning with its design for real-time performance. DeepLabV3+ follows with an inference speed of 346.22 pages per second, while DeepLabV3 processes 268.38 pages per second. U-Net, despite being the most accurate in terms of segmentation performance, has the lowest inference speed at 239.15 pages per second.

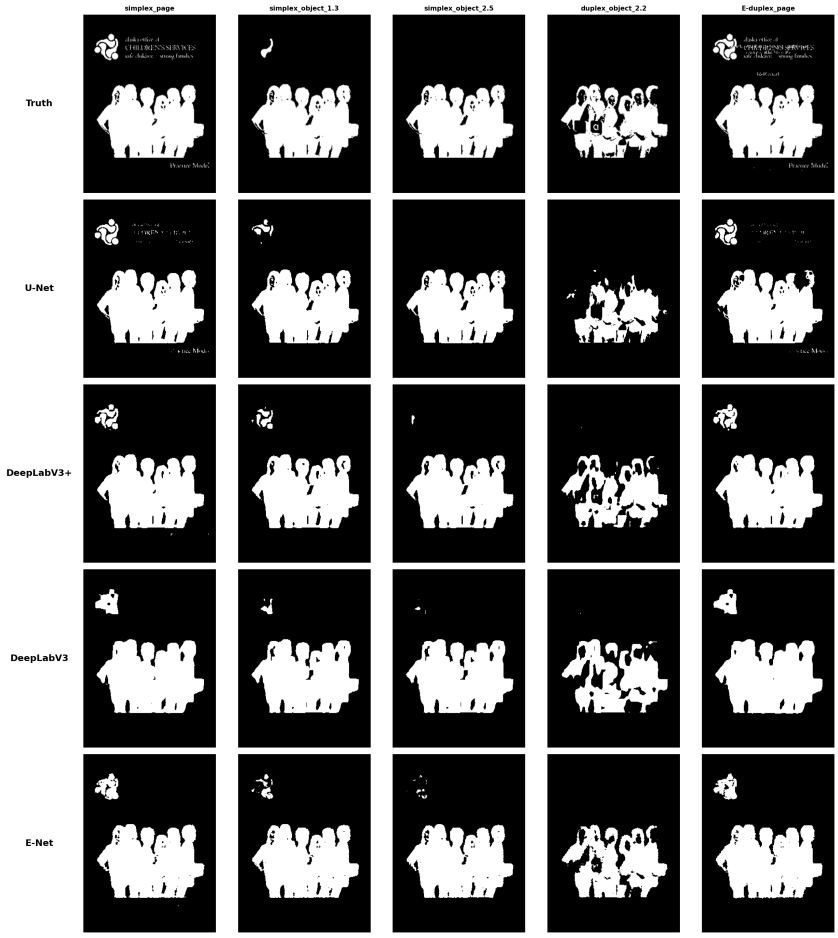


Figure 3: Outputs from the models

These results highlight the trade-offs between model complexity, accuracy, and inference speed. While U-Net excels in segmentation accuracy, its processing speed is the slowest among the models tested. Conversely, E-Net, although less accurate, demonstrates superior efficiency in processing speed, making it suitable for scenarios where real-time performance is critical. DeepLabV3+ and DeepLabV3 offer a balance between accuracy and speed, making them versatile choices depending on the specific requirements of the task.

6 Future Improvements and Lessons Learnt

One critical lesson learned is the importance of model architecture in determining the performance of segmentation tasks. U-Net's architecture, which effectively captures both local and global information, outperformed other models, highlighting the need for careful architectural design in segmentation tasks. The limitations observed in DeepLabV3+ and DeepLabV3 underscore the importance of having a robust decoder to preserve spatial details.

An interesting avenue for future research involves reformulating the segmentation problem. Rather than predicting a segmentation mask directly, a two-stage approach could be employed. First, a model could predict which classes are present in the image, and then generate the segmentation masks only for those classes. This approach could potentially reduce errors where the model predicts masks for classes that are not present in the image.

Additionally, incorporating vision transformers and foundation segmentation models such as "Segment Anything" could provide significant improvements. Vision transformers have shown promise in various vision tasks due to their ability to capture long-range dependencies and global context. Evalu-

ating the performance of these advanced models on our segmentation tasks could yield interesting insights and potentially superior results.

We have made our reading and coding resources available as separate folders in our GitHub repository. This includes a step-by-step guide on how to execute the code, ensuring that others can replicate our experiments and build upon our work. The repository can be accessed at our GitHub repository.

7 Conclusion

In this study, we evaluated the performance of four prominent semantic segmentation models— U-Net, E-Net, DeepLabV3, and DeepLabV3+ to identify PDF document characteristic. Our comprehensive analysis included both high-level performance metrics such as IoU and F1 score, as well as detailed per-class IoU metrics to understand the strengths and weaknesses of each model.

The architectural differences between the models significantly influenced their performance. U-Net's ability to maintain fine details and understand broader context led to its superior results, while the simpler or absent decoders in DeepLabV3+ and DeepLabV3, and the lightweight design of E-Net, impacted their effectiveness. These findings underscore the importance of selecting an appropriate model architecture based on the specific requirements of the segmentation task, balancing the need for efficiency and generalization capability.

Furthermore, by employing deep learning, we have significantly improved the speed of inference over traditional systems. Leveraging the parallelism capabilities of GPUs, these deep learning models can process images much faster, enabling real-time or near-real-time performance, which is crucial for many practical applications.

8 Code and Resources

The details of our implementations and the code base is uploaded here: GitHub repository.

References

[1] Jobin, K. V., and Jawahar, C. V. (2018). Document Image Segmentation Using Deep Features. In Rameshan, R., Arora, C., and Dutta Roy, S. (Eds.), *Computer Vision, Pattern Recognition, Image Processing, and Graphics* (pp. 372–382). Springer Singapore.

[2] Chatfield, K., Simonyan, K., Vedaldi, A., and Zisserman, A. (2014). Return of the devil in the details: Delving deep into convolutional nets. *arXiv preprint arXiv:1405.3531*.

[3] Sarkar, M., Aggarwal, M., Jain, A., Gupta, H., and Krishnamurthy, B. (2020). Document Structure Extraction Using Prior Based High Resolution Hierarchical Semantic Segmentation. In Vedaldi, A., Bischof, H., Brox, T., and Frahm, J.-M. (Eds.), *Computer Vision – ECCV 2020* (pp. 649–666). Springer International Publishing.

[4] Ronneberger, O., Fischer, P., and Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. *arXiv preprint arXiv:1505.04597*.

[5] Paszke, A., Chaurasia, A., Kim, S., and Culurciello, E. (2016). ENet: A Deep Neural Network Architecture for Real-Time Semantic Segmentation. *arXiv preprint arXiv:1606.02147*.

[6] Chen, L.-C., Papandreou, G., Schroff, F., and Adam, H. (2017). Rethinking Atrous Convolution for Semantic Image Segmentation. *arXiv preprint arXiv:1706.05587*.

[7] Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., and Adam, H. (2018). Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. *arXiv preprint arXiv:1802.02611*.

[8] Yu, F., and Koltun, V. (2016). Multi-Scale Context Aggregation by Dilated Convolutions. *arXiv preprint arXiv:1511.07122*.

[9] Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L. (2017). DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *arXiv preprint arXiv:1606.00915*.