

Handwritten and Printed Text Segmentation: A Signature Case Study

Sina Gholamian
Thomson Reuters AI Labs
Toronto, Canada

sina.gholamian@thomsonreuters.com

Ali Vahdat
Thomson Reuters AI Labs
Toronto, Canada

ali.vahdat@thomsonreuters.com

Abstract

While analyzing scanned documents, handwritten text can overlap with printed text. This overlap causes difficulties during the optical character recognition (OCR) and digitization process of documents, and subsequently, hurts downstream NLP tasks. Prior research either focuses solely on the binary classification of handwritten text or performs a three-class segmentation of the document, i.e., recognition of handwritten, printed, and background pixels. This approach results in the assignment of overlapping handwritten and printed pixels to only one of the classes, and thus, they are not accounted for in the other class. Thus, in this research, we develop novel approaches to address the challenges of handwritten and printed text segmentation. Our objective is to recover text from different classes in their entirety, especially enhancing the segmentation performance on overlapping sections. To support this task, we introduce a new dataset, SignaTR6K, collected from real legal documents, as well as a new model architecture for the handwritten and printed text segmentation task. Our best configuration outperforms prior work on two different datasets by 17.9% and 7.3% on IoU scores. The SignaTR6K dataset is accessible for download via the following link: <https://forms.office.com/r/2a5RDg7cAY>.

1. Introduction

For various purposes, the digitization of hard-copy documents and associated challenges are an active area of research in both academia [25, 11, 27, 10, 6, 20] and industry [18]. This digital transformation involves scanning paper documents through an OCR process, making their text accessible for downstream natural language processing (NLP) tasks, such as named entity recognition (NER). The documents of interest can originate from a variety of domains, including historical documents [27], legal and court-issued documents [18], business contracts [25], and medical records and prescriptions [5]. Although various studies, as cited above, have been conducted, there is yet a considerable gap between current approaches and

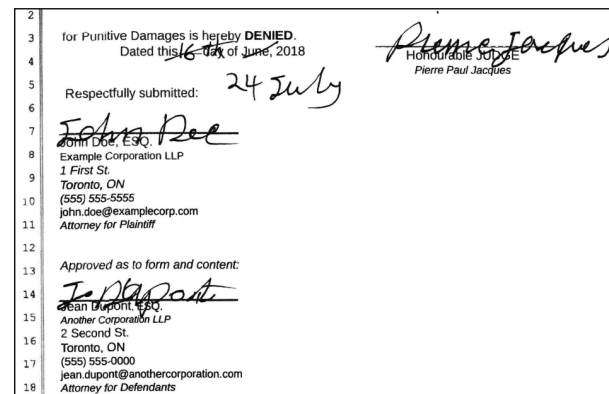


Figure 1: Court documents are first printed and then signed or annotated by various parties. This results in handwritten text overlapping the underlying printed information, leading to performance degradation in downstream tasks, such as named entity recognition (NER). This is a fabricated example, however very similar to the original documents, to protect personally identifiable information (PII).

human-level performance in mixed-text scenarios (i.e., when handwritten and printed text overlap). For instance, attorneys frequently sign legal documents, resulting in their signatures overlapping with their information. This overlap hampers the performance of OCR tools in character recognition, subsequently making it challenging for downstream tasks to accurately identify information linked to the attorneys and their associated law firms. Figure 1 provides an illustration of handwritten text (HT) overlapping with printed text (PT) in court documents. Extracting parties names is a crucial step in the named-entity recognition (NER) task for legal and court documents [24]. When attorneys and other involved parties sign these documents, which are later scanned using OCR tools, their signatures often obscure the details of names and law firms. Consequently, the semantic segmentation of handwritten elements, such as lawyers' signatures and accompanying handwritten notes, and printed text detailing the lawyer and the law firm's information, becomes vital.

In this research, we aim to address the challenges of HT and PT segmentation, as there is still a large gap between human-level performance and the existing approaches

for this task. Our focus in this effort is to improve the segmentation performance in overlapping regions for scanned legal documents, and to aid in this endeavor, we also introduce a new dataset. In summary, our research makes the following contributions:

- We introduce a new dataset, SignaTR6K (pronounce as Signature 6K)¹ derived from 200 pixel-level manually annotated crops of images from genuine legal documents. The dataset comprises signatures, handwritten text, and printed text, which frequently overlap. With data augmentation, we have created a dataset of sizes 5169, 530, 558, for training, validation, and testing, respectively, that we release to the public, to facilitate dataset availability for future research and to aid in the training and evaluation of deep-learning segmentation models.
- We propose a novel architecture that integrates both *semantic segmentation features* and *fine features*, enhancing the performance of text segmentation over previous methods. Moreover, we introduce a new loss function termed *Fusion loss*, that, comparatively, is stable and converges to optimal loss values and Intersection over Union (IoU) scores.
- Lastly, we conduct an extensive quantitative and visual evaluation of different variations of our approach against prior work on two distinct datasets, and illustrate our approach’s superior performance in the text segmentation task, especially in challenging scenarios where printed and handwritten text overlap.

2. Background and Literature Review

The text segmentation problem is defined as follows: given a scanned document possibly containing handwritten, printed, and background (i.e., blank) pixels, the task is to assign each pixel to its appropriate class. Scanned documents can originate from various sources, such as hard-copy paper documents or microfilms [20, 27]. Formally, for a given document D , assuming there exist three classes as handwritten text HT , printed text PT , and background BG , and pixel p_i : $\forall p_i \in D : p_i(c) == True \text{ if } p_i \in c \ \&\& \ c \in \{HT, PT, BG\}$ (1). A situation may arise where a pixel belongs to two classes, i.e., HT and PT classes, when handwritten text overlaps with printed text, which single-label three-class formulation cannot handle such cases.

Several previous studies have studied the segmentation of handwritten and printed text [6, 10, 20, 27], however, they have inherent limitations. Some focus solely on binary classification, determining if a pixel is handwritten or not [10], whereas others adopt a 3-class formulation of the problem, classifying pixels as handwritten, printed, or background [6, 20, 27]. This exclusive assignment of

pixels to three different classes paralyzes any machine learning segmentation model to properly detect pixels in the overlapping areas to belong to both handwritten and printed classes. In our usecase depicted in Figure 1, the OCR process achieves insufficient performance due to overlapping of handwritten text and signatures that overlay the printed text. As such, to improve the documents digitization quality, and subsequently a wide range of downstream document understanding and natural language processing (NLP) tasks, it is vital to devise image processing approaches that can understand and properly segment different layers of text, i.e., printed and handwritten text.

Models. Prior literature has applied various approaches for the identification and separation of handwritten and printed text. Early approaches [7, 12] formulated the problem as a binary classification task using KNN and SVM and focusing on connected components (CCs) (i.e., groups of pixels). More recently, Li et al. [14] employed conditional random fields (CRFs), with formulating both unary and pairwise potentials for adjacent connected components by leveraging convolutional neural networks (CNNs) architecture for the separation of CCs. The limitation of CC-based approaches is that they determine the class membership for the entire component rather than at the pixel level. Consequently, pixel-level segmentation methods were introduced, leveraging Markov random fields (MRFs) and MLPs [19, 22] for pixel-level classification of PT and HT segmentation. Following the success of encoder-decoder architectures in object segmentation [21], more recent works [10, 6, 20, 27] have predominantly adopted a U-Net based architecture [21], which comprises an encoder-decoder network, for HT and PT segmentation.

In addition to our contribution of releasing a manually annotated dataset of legal documents with overlapping text, SignaTR6K, our methodology distinguishes itself from prior works in several significant ways. Firstly, we approach the segmentation problem with a four-class formulation, allowing overlapping pixels to be assigned to a new distinct class (OV), which signifies the presence of both HT and PT layers, leading to enhanced segmentation performance. Secondly, we introduce a novel architecture, the Mixed Feature Model (MFM), that combines a Fine Feature Path (FFP) with a Semantic Segmentation Path (SSP) and improves the performance by capitalizing on both high-level and low-level features. Notably, existing U-Net style architectures [10, 6, 20, 27] are limited to leveraging only the SSP path. Further, we present a new loss function, termed Fusion Loss, that converges faster and is more stable compared to prior losses for HT and PT segmentation task. In addition, we introduce a post-processing heuristic based on Conditional Random Fields (CRFH) to carry out relabeling, resulting in further enhancement of text segmentation performance. Additional details related to background and

¹Available at <https://forms.office.com/r/2a5RDg7cAY>.

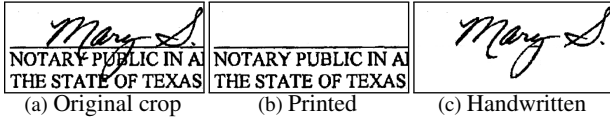


Figure 2: This figure depicts a real crop from a legal document. (a) The original image containing printed text and overlaying handwritten signature; (b) Printed layer pixels annotated only; (c) Handwritten layer pixels annotated only. Any pixels not present in the printed or handwritten layers are marked as background.

relevant works can be found in the supplementary.

3. SignaTR6K Dataset

Successfully training and testing a segmentation model requires access to high-quality labeled datasets. Due to the limited availability of public data that contains both handwritten and printed text, previous research has predominantly sought to synthetically generate such data. To achieve this, researchers have combined datasets that either exclusively contain printed or handwritten text, or those with non-overlapping text, such as IAM [17], RIMES [1], PRImA [4], CVL [13], Scanned Questionnaire [2], and WGM-SYN [27]. The scanned documents can originate from diverse domains, each possessing its unique fonts, characters, and quality. Whether the original documents having poor quality in the case of archival documents [27], or the scanning process resulted in lower resolution or loss of contrast, these factors also compound the complexities of the text segmentation task. In addition, in some cases errors are introduced during the automated text labeling process [20]. Given these challenges, the absence of a precisely and manually annotated and verified dataset from real sources that contains various patterns of overlapping printed and handwritten text, significantly impedes the supervised machine learning (ML) process. To address this gap, we introduce a new dataset, **SignaTR6K** (pronounced as Signature 6K), that is derived from 200 original legal documents from Thomson Reuters Legal Content Services [3]. The dataset features overlapping printed and handwritten text and hand-drawn signatures, as depicted in Figure 1. The documents originate from different organizations, including law firms and courts, each having its distinct fonts and document formats. Furthermore, the annotations are made by different individuals, ensuring a diverse range of printed and handwritten styles. Importantly, each document has been manually labeled and verified.

Figure 2 displays an original crop from SignaTR6K, which includes both a signature and the information of the signing party. Due to the presence of personally identifiable information (PIIs), we cropped the images to ensure no visible PIIs. Figure 2a illustrates the original image with both printed text and overlaying handwritten signature. Figure 2b presents only the printed layer pixels, while Figure 2c contains only the handwritten layer pixels that are manually annotated. Any pixels absent from either the printed or

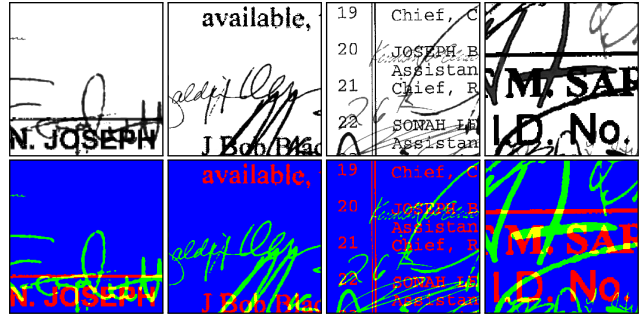


Figure 3: Examples from the SignaTR6K dataset, with the top row showing the crops and the bottom row their ground truth annotations. Red: class *PT*, printed, Green: class *HT*, handwritten, and Blue: class *BG*, background.

handwritten channels are designated as background pixels. It is also evident that in overlapping scenarios, some pixels belong to both handwritten and printed layers.

Dataset Generation. As hand-annotating a vast collection of real documents is time-consuming and expensive, in order to expand on the size of the dataset and make it adequate for training a deep learning model, we turn to data synthesis and augmentation techniques [10]. The synthetic approaches include general augmentation of crops, along with shifting, magnifying, and rotating operations. We also overlay handwritten and printed pixels from different crops to generate new real-like examples of overlapping text. For this purpose, from the 200 distinct original document samples, we set aside 16 crops, ensuring mutually exclusive samples in the test set. These crops are augmented only for the test set, and we create the training and validation sets from the remaining 184 samples. Following this approach and after excluding generated samples with visible PIIs, we have curated a dataset for training, validation, and testing sets of sizes 5169, 530, and 558 samples, respectively. Each sample is a pair of grayscale crop and its manually annotated ground truth. Each image is 256 by 256 pixels in size, with three channels (RGB) and typically contains several HT and PT overlaps. Four examples from the SignaTR6K dataset are presented in Figure 3, with their grayscale crops and corresponding ground truth (GT) labels. HT is indicated in green, PT in red, and BG in blue. Overlapping HT and PT pixels combine green and red channel values, resulting in a yellow appearance. We envision this dataset can be utilized for model training from scratch or further fine-tuning of a pre-trained classification or segmentation model for specific tasks. The SignaTR6K dataset is freely available for public download.

4. Approach and Methodology

In this section, we detail our approach, the rationale behind the chosen architecture, the Semantic Segmentation Path (SSP) and Fine Feature Path (FFP), the various loss functions employed, and our novel *Fusion* loss.

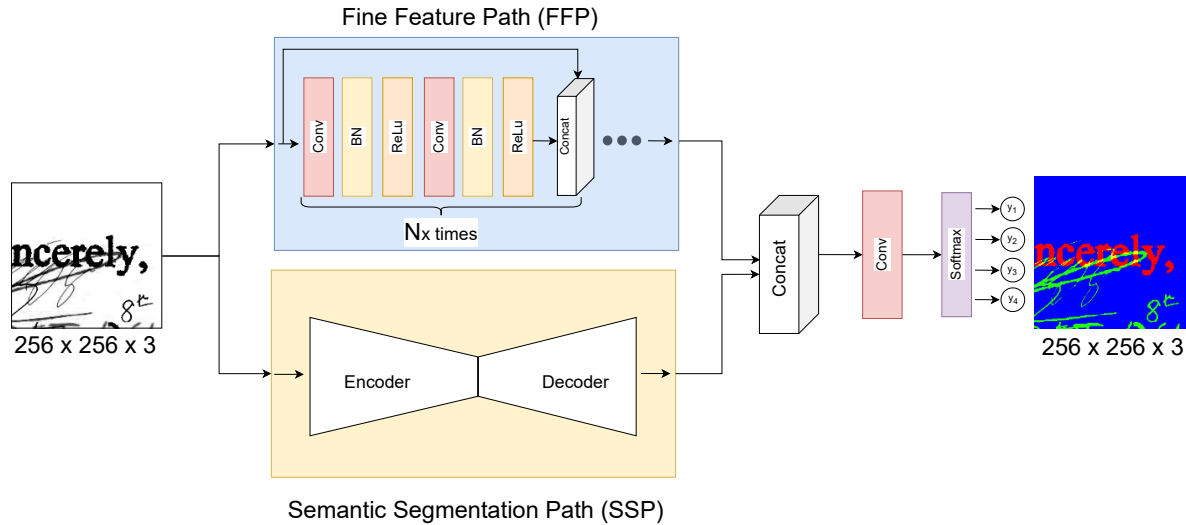


Figure 4: Our architecture proposal showing FFP and SSP model paths. The output of FFP and SSP are concatenated prior to the final round of convolutions. A Softmax activation then selects one of the four classes for each pixel. The model’s final output consists of pixel-level annotations of the input image. BN: BatchNorm.

4.1. Model Architecture

The prevalent architecture for object segmentation applies a U-Net design [21, 16]. This architecture leverages a Fully Convolutional Network (FCN), that is without fully connected layers that are typically present at the end of Convolutional Neural Networks (CNNs). The U-Net architecture consists of two main parts, an encoder and a decoder. In the encoder segment, the original image (in our context a document crop) is fed into the model. It then undergoes a series of convolutions and max-pooling layers, extracting features from the image and reducing its dimensions. Conversely, the decoder processes the down-sampled image from the encoder using convolutions and up-sampling layers that eventually restores the image back to its original input size with the same number of channels. The final output of the decoder is a pixel-level labeling of the original image. Semantic Segmentation Path (SSP) in Figure 4 shows this part of the fully convolutional network. In addition to the encoder-decoder architecture in U-Net, there are also skip-connections that bring the feature maps directly from an encoder stage to the corresponding decoder stage, i.e., the decoder stage that has the same image size of the encoder stage. These skip-connections improve segmentation performance by re-accessing early-stage features that may be lost in the encoder’s final output due to down-sampling.

4.2. Four-Class Formulation

As mentioned earlier, prior approaches have either considered binary classification of HT or a three-class formulation of the problem. Binary classification detects only one type of text, while the three-class formulation results in overlapping pixels being assigned solely to either the HT or PT class, impairing performance. Therefore, we propose a four-class formulation of the segmentation task,

and the fourth class, *overlap*, *OV*, models pixels that belong to both the HT and PT classes, specifically enabling the detection of overlapping areas. To expand on Formula (1) for this four-class formulation, for a given document D , assuming there exist four classes as handwritten text *HT*, printed text *PT*, background *BG*, and overlap *OV*, and pixel p_i : $\forall p_i \in D : p_i(c) == True$ if $p_i \in c$ and $c \in \{HT, PT, BG, OV\}$.

Overlapping pixels are highlighted in yellow in the ground truth, as seen in Figure 3. Our four-class single-label classification employs a Softmax activation function in the final layer, which ensures that only one output for each pixel is activated (Figure 4). Since the output image comprises three channels, when the *OV* class is predicted, during a post-processing step we turn on pixels for both *HT* and *PT* channels, resulting in the yellow color in the output image. In addition to the four-class formulation, we also explored three-class and multi-label formulations. In this scenario, instead of a Softmax activation, we applied Sigmoid activations for each class (i.e., three separate sigmoids). However, formulating the problem as a multi-label classification introduces added complexity and degrees of freedom. This can lead to undesirable scenarios, such as the simultaneous activation of pixels for *HT* and *BG*. Consequently, the multi-label approach did not yield results comparable to those of the four-class formulation.

4.3. Semantic Segmentation Path (SSP)

The semantic segmentation path (SSP) of our model leverages a U-Net based architecture, with down-sampling in the encoder stages (i.e., backbone) and up-sampling in the decoder stages. The U-net architecture works well in capturing high-level image features. In this architecture, the encoder and decoder maintain a symmetrical architec-

ture with a similar number of stages. For example, in the SSP, if the encoder goes through four down-sampling stages, i.e., the input image size changes from 256×256 to 16×16 ($256 \rightarrow 128 \rightarrow 64 \rightarrow 32 \rightarrow 16$), correspondingly, the decoder undergoes four up-sampling stages, from 16×16 to 256×256 . Thus, the final output retains the original input image’s pixel dimensions. For the SSP, we explored a variety of network sizes. As a baseline and for comparison, we implemented FCN-light [6, 20, 27]. We then improved on the SSP architecture by using VGG16 [23], InceptionV3 [26], and ResNet34 [8] as backbones of the SSP and observed improvement in the performance. In particular, the ResNet34 and InceptionV3 backbones outperform the prior work (FCN) due to their larger number of learnable parameters. Additionally, the inclusion of residual connections and varied convolution sizes allows to better carry over the low-level features of text to the later stages of the segmentation network. This observation further inspired us to introduce the FFP network, which similarly incorporates residual connections and without down-sampling.

4.4. Fine Feature Path (FFP)

The semantic segmentation path excels when segmenting distinct objects. In this path, down-sampling layers are rapidly applied to the input image to capture high-level features and patterns. However, this rapid processing can lead to the loss of fine, or low-level, features. For our application, low-level features are crucial due to the intertwined nature of printed and handwritten text. To address this, we introduce a parallel path to the SSP, termed the fine feature path (FFP), which avoids down-sampling and instead incorporates a convolution block with residual connections. Note that, while the FFP aids in capturing fine features without down-sampling, on its own, i.e., without the SSP that includes down-sampling, it is insufficient for text segmentation, as the absence of high-level features means the model will not be able to detect high-level patterns irrespective of their pixel locations in the image. In Figure 4, the fine feature block of the FFP is repeated N_x times; in our implementation, $N_x = 4$. In addition, each stage of the FFP itself implements residual connections as it provides the flexibility to either bypass the block or use its output, leading to improved results, as we will discuss later in the paper. Similar architectures employing residual blocks have shown improved performance in fine object segmentation tasks, e.g., road segmentation [29]. Table 1 details the FFP architecture for four stages ($N_x = 4$) and the connections between layers and convolution sizes.

4.5. Mixed Feature Model (MFM)

The Mixed Feature Model (MFM) serves as the umbrella model containing two parallel paths: SSP and FFP, as depicted in Figure 4. The objective of MFM is to capture the image’s low-level features alongside its high-level features.

Group	Layer type	Filter	Input(s)	Output(s)	Output size			
Stage 1	G 1	Conv	$3 \times 3 / 64$	img_input	s1.g1.c.o	$256 \times 256 \times 64$		
		BatchNorm	-	s1.g1.c.o	s1.g1.b.o	$256 \times 256 \times 64$		
		ReLU	-	s1.g1.b.o	s1.g1.r.o	$256 \times 256 \times 64$		
	G 2	Conv	$3 \times 3 / 64$	s1.g1.r.o	s1.g2.c.o	$256 \times 256 \times 64$		
		BatchNorm	-	s1.g2.c.o	s1.g2.b.o	$256 \times 256 \times 64$		
		ReLU	-	s1.g2.b.o	s1.g2.r.o	$256 \times 256 \times 64$		
		Concat	-	img_in, s1.g2.r.o	s1.o	$256 \times 256 \times 67$		
		Conv	$3 \times 3 / 64$	s1.o	s2.g1.c.o	$256 \times 256 \times 64$		
		BatchNorm	-	s2.g1.c.o	s2.g1.b.o	$256 \times 256 \times 64$		
		ReLU	-	s2.g1.b.o	s2.g1.r.o	$256 \times 256 \times 64$		
Stage 2	G 1	Conv	$3 \times 3 / 64$	s2.g1.r.o	s2.g2.c.o	$256 \times 256 \times 64$		
		BatchNorm	-	s2.g2.c.o	s2.g2.b.o	$256 \times 256 \times 64$		
		ReLU	-	s2.g2.b.o	s2.g2.r.o	$256 \times 256 \times 64$		
	G 2	Concat	-	s1.o, s2.g2.r.o	s2.o	$256 \times 256 \times 64$		
		Conv	$3 \times 3 / 64$	s2.o	s3.g1.c.o	$256 \times 256 \times 64$		
		BatchNorm	-	s3.g1.c.o	s3.g1.b.o	$256 \times 256 \times 64$		
		ReLU	-	s3.g1.b.o	s3.g1.r.o	$256 \times 256 \times 64$		
		Stage 3	G 1	Conv	$3 \times 3 / 64$	s3.g1.r.o	s3.g2.c.o	$256 \times 256 \times 64$
				BatchNorm	-	s3.g2.c.o	s3.g2.b.o	$256 \times 256 \times 64$
				ReLU	-	s3.g2.b.o	s3.g2.r.o	$256 \times 256 \times 64$
Concat	-		s2.o, s3.g2.r.o	s3.o	$256 \times 256 \times 195$			
Stage 4	G 1	Conv	$3 \times 3 / 64$	s3.o	s4.g1.c.o	$256 \times 256 \times 64$		
		BatchNorm	-	s4.g1.c.o	s4.g1.b.o	$256 \times 256 \times 64$		
		ReLU	-	s4.g1.b.o	s4.g1.r.o	$256 \times 256 \times 64$		
	G 2	Conv	$3 \times 3 / 64$	s4.g1.r.o	s4.g2.c.o	$256 \times 256 \times 64$		
		BatchNorm	-	s4.g2.c.o	s4.g2.b.o	$256 \times 256 \times 64$		
		ReLU	-	s4.g2.b.o	s4.g2.r.o	$256 \times 256 \times 64$		
		Concat	-	s3.o, s4.g2.r.o	s4.o	$256 \times 256 \times 259$		
		Conv	$1 \times 1 / 4$	s4.o	FFP.o	$256 \times 256 \times 4$		

Table 1: The architecture of the Fine Feature Path (FFP). In our design, we use four stages (Stages 1-4) of fine feature blocks. We include BatchNorm and ReLU activation after each convolution layer, and unlike SSP, down-sampling layers (e.g., max pooling) are absent from FFP. The output of each stage is concatenated with its input to create residual connections.

The outputs of SSP and FFP are concatenated before the final layer convolution, producing the output of the MFM model. Additional details regarding the architecture, inputs, outputs, and convolution layer sizes of the MFM can be found in the supplementary.

4.6. CRF Post-Processing and CRF Heuristic

Prior research [6, 20, 27] has leveraged dense Conditional Random Fields (CRFs) as a post-processing step to re-label pixels based on their neighboring pixels. In our architectural exploration, we found that, while CRF post-processing is intended to improve segmentation performance, in practice it often hurts the segmentation performance by aggressively re-labeling pixels to incorrect classes. One issue arises due to the imbalanced nature of pixels across classes, with background pixels being predominant. Consequently, many pixels are mistakenly annotated from HT or PT to BG, or from HT to PT, which is undesirable. Based on this observation, we designed a heuristic for CRF post-processing to contain this unfavourable behaviour by only allowing the BG pixels to be relabelled to HT or PT pixels and not vice versa. As we will discuss in the experimentation section, this heuristic further improves segmentation performance.

4.7. Loss Functions

Due to the nature of the scanned documents and the amount of text on each page, the number of background pixels (i.e., white blank pixels) surpasses that of handwritten and printed pixels. This leads to a class imbalance problem [30], posing the risk of predicting the majority of pixels as background while still minimizing the loss value. As a result, we investigated various loss functions and weight as-

signments to different classes in the loss function to evaluate its impact on segmentation performance. Additionally, during this exploration, we observed that different loss functions achieve different IoU (Intersection over Union) scores. Consequently, we introduce a new loss function, Fusion Loss, to incorporate the benefits of various losses. In the following, we discuss the different loss functions we explored.

Cross-Entropy Loss. The multi-class cross-entropy loss is used for classification tasks involving more than two classes. It assumes that for each data point, only one class can be the ground truth, i.e., multi-class single-label classification. The standard form of cross-entropy loss for a single data point in multi-class segmentation with M classes is defined as: $\mathcal{L}_{CE}(gt, pr) = -\sum_{m=1}^M gt \cdot \log(pr)$, where pr and gt represent the prediction and ground truth, respectively. The final loss is the average of the losses for all data points in a training set. Similarly, the weighted version (WCE) is computed as $\mathcal{L}_{WCE}(gt, pr) = -\sum_{m=1}^M w_m \cdot gt \cdot \log(pr)$, with w_m being the weight assigned to each class in the loss calculation.

Focal Loss. The focal loss [15] was introduced to address the class-imbalance problem during training of object detection tasks. The focal loss puts the focus of the learning algorithm on the incorrectly classified examples by applying a modulating term $(1 - pr)^\gamma$ to the cross-entropy loss. This scaling factor dynamically weighs down the contribution of easy and correctly classified samples, allowing the training loop to concentrate on the harder and incorrectly classified samples. The focal loss is calculated as: $\mathcal{L}_{Focal}(gt, pr) = -\sum_{m=1}^M gt \cdot (1 - pr)^\gamma \log(pr)$, where γ is the modulating/focusing factor, and we set $\gamma = 2$ as per the original paper. Similarly, the weighted version of the focal loss can be expressed as: $\mathcal{L}_{WFL}(gt, pr) = -\sum_{m=1}^M gt \cdot \alpha_m \cdot (1 - pr)^\gamma \log(pr)$, where α_m is analogous to w_m , assigning weights to each class such that: $\forall \alpha_m \mid 0 < \alpha_m < 1 \ \&\& \ \sum_{m=1}^M \alpha_m = 1$.

Dice Loss. Dice loss is given by $\mathcal{L}_{Dice}(precision, recall) = 1 - \frac{2}{M} \sum_{m=1}^M \frac{precision_m \cdot recall_m}{precision_m + recall_m}$. Intuitively, it aims to maximize the F-Score while minimizing the loss, i.e., $\mathcal{L}_{Dice} = 1 - F_{Score}$. Accordingly, the weighted version of the dice loss is defined as $\mathcal{L}_{WD}(precision, recall) = 1 - \frac{2}{M} \sum_{m=1}^M \frac{w_m \cdot precision_m \cdot recall_m}{precision_m + recall_m}$.

Fusion Loss. Given our observation that different loss functions adversely affect different classes, we introduce a new loss targeted for maximizing the performance of all the classes. For example, the dice loss performs better on the background class, as in its formulation, it aims to maximize the F-Score. Because the majority of pixels are attributed

to the background class, having higher values of correctly classified background pixels achieves a higher F-Score and lower loss values. However, this does not necessarily yield higher IoU scores for PT and HT classes. In contrast, the weighted versions of the cross-entropy and focal losses focus on the handwritten and printed classes. As such, with the Fusion loss, we aim to combine the behaviors of various losses, and we define the Fusion loss as the sum of the three weighted losses: $\mathcal{L}_{Fusion} = \mathcal{L}_{WFL} + \mathcal{L}_{WCE} + \mathcal{L}_{WD}$.

5. Experiments and Results

Evaluation Metric. Intersection over Union (IoU) is a commonly used metric to measure the performance of a segmentation task [28], and in particular text segmentation [25, 10, 6, 27]. In our context, we use IoU to measure the pixel-level performance of the model output versus the ground truth. For a class c , let TP_c represent the correctly predicted pixels for that class, FP_c denote the pixels incorrectly predicted as belonging to class c , and FN_c represent the pixels that are incorrectly not predicted for class c , then Intersection over Union, IoU, for class c is calculated as: $IoU_c = \frac{TP_c}{TP_c + FP_c + FN_c}$. For each experiment, we calculate the IoU for all three classes, HT, PT, and BG. In the four-class formulation, pixels attributed to the overlap (OV) class are converted in a post-processing step to both HT and PT classes, contributing to the IoU calculations for both. We also calculate the mean IoU, which is the average of IoUs for these three classes: $IoU_{Mean} = \frac{IoU_{HT} + IoU_{PT} + IoU_{BG}}{3}$.

Experiments. We perform 50 epochs of training with a batch size of 8. For problem formulation, we use both three-class and four-class implementations. The experiments are conducted on two datasets: SignaTR6K and WGM-SYN [27]. The IoU values for FCN-light and WGM-MOD are as reported in [27] for the three-class formulation which has been performed on the same dataset, WGM-SYN. However, for the SignaTR6K dataset, we retrain an FCN-based model for three-class formulation to ensure a fair comparison on the new dataset. We run experiments for a variety of architectures and in three post-processing configurations. We structure experiments incrementally, i.e., adding one improvement at a time. This approach aids in understanding and isolating the impact of each added improvement, i.e., ablation study. Due to space constraints, the table detailing the complete list of experiment parameters and configurations is relegated to the supplementary.

Experimented Architectures. We start with FCN-light which has approximately 295K parameters. This architecture aligns with prior research and serves as our baseline [6, 27]. Next, we improve the performance of FCN-light by adopting a four-class implementation. We then experiment with SSP architecture with three different back-

Formulation	Backbone	# Parameters	Loss function	IoU %				With CRF (IoU %)				With CRFH (IoU %)			
				PT	HT	BG	Mean	PT	HT	BG	Mean	PT	HT	BG	Mean
3-Class	FCN-light [6]	~295K	Weighted CE	24.00	26.00	72.00	41.00	11.00	23.00	72.00	36.00	-	-	-	-
	WGM-MOD [27]	~295K	Weighted CE	42.00	36.00	74.00	50.00*	41.00	32.00	74.00	49.00	-	-	-	-
4-Class (Ours)	FCN-light	~295K	CE	46.49	41.98	73.77	54.08	38.95	29.35	73.92	47.41	46.57	41.59	73.87	54.01
			Focal	46.02	42.01	73.80	53.95	32.96	24.50	74.00	43.82	46.11	41.23	73.98	53.77
			Dice	48.01	47.28	71.22	55.55	48.59	48.48	71.22	56.10	48.07	47.79	71.29	55.72
			Weighted CE	46.07	42.18	73.82	54.02	40.54	30.07	73.98	48.20	46.31	41.68	73.95	53.98
			Weighted Focal	43.71	41.77	73.95	53.14	32.67	24.40	74.07	43.71	44.06	41.12	74.11	53.10
			Weighted Dice	47.97	47.22	71.02	55.40	48.69	48.34	71.02	56.02	48.04	47.70	71.10	55.61
	SSP - VGG16	~24M	Fusion	48.12	47.25	71.93	55.77	46.34	44.25	71.86	54.15	48.19	47.89	72.38	56.15
			CE	35.14	32.40	73.96	47.17	30.97	19.14	73.88	41.33	35.21	32.08	74.13	47.14
			Focal	34.56	32.22	74.02	46.93	29.11	19.42	73.91	40.81	34.67	31.79	74.22	46.89
			Dice	40.64	39.64	71.13	50.47	43.30	40.16	71.21	51.55	40.81	40.25	71.25	50.77
			Weighted CE	35.73	33.44	74.43	47.87	32.00	20.70	74.12	42.28	35.93	32.95	74.65	47.84
			Weighted Focal	34.90	34.35	74.36	47.87	29.18	20.72	74.06	41.32	35.11	33.80	74.56	47.82
	MFM (FFP + SSP) - InceptionV3	~30M	Weighted Dice	41.80	40.64	70.67	51.03	44.14	41.39	70.70	52.07	42.52	41.19	70.97	51.56
			Fusion	39.99	39.56	72.31	50.62	35.39	27.93	72.35	45.22	40.10	39.94	72.63	50.89
			CE	52.51	44.09	73.91	56.84	46.03	35.94	74.21	52.06	52.55	42.17	74.35	56.36
			Focal	51.56	43.74	73.89	56.40	35.89	27.15	74.41	45.82	51.63	41.74	74.43	55.93
			Dice	20.91	19.28	<u>81.52</u>	40.57	25.51	24.61	<u>79.94</u>	43.35	20.92	19.29	81.54	40.58
			Weighted CE	51.58	43.72	73.95	56.42	44.78	34.04	74.32	51.05	51.64	41.85	74.40	55.96
	MFM (FFP + SSP) - ResNet34	~24M	Weighted Focal	51.01	43.62	73.89	56.18	35.76	26.60	74.22	45.53	51.08	41.60	74.35	55.68
			Weighted Dice	22.30	21.25	80.73	41.42	29.11	29.48	<u>77.97</u>	45.52	22.32	21.26	80.74	41.44
			Fusion	<u>53.22</u>	51.25	71.84	58.77	50.05	48.83	72.03	56.97	53.32	49.83	72.72	58.62
			CE	52.40	44.02	73.91	56.78	47.52	36.40	74.12	52.68	52.43	42.12	74.34	56.30
			Focal	51.95	43.98	73.89	56.61	36.73	27.17	74.31	46.07	51.90	41.99	74.33	56.08
			Dice	29.68	29.43	77.06	45.39	31.62	25.39	75.15	44.05	29.84	29.63	77.07	45.52
MFM (FFP + SSP) - ResNet34	~24M	Weighted CE	52.63	44.12	73.94	56.90	47.60	37.25	74.12	52.99	52.56	42.22	74.36	56.38	
		Weighted Focal	51.68	43.95	73.90	56.51	35.73	27.22	74.30	45.75	51.72	41.98	74.34	56.01	
		Weighted Dice	51.44	50.60	71.96	58.00	<u>52.97</u>	<u>51.42</u>	71.90	<u>58.76</u>	51.54	49.37	72.75	57.89	
		Fusion	52.99	51.72	71.69	<u>58.80</u>	51.15	50.79	71.83	57.92	53.01	<u>51.29</u>	72.49	58.93	

Table 2: IoU performance (%) on the WGM-SYN dataset [27]. The maximum value of a column (representing a class) is underlined. The overall maximum of a class across different post-processing methods is both bolded and underlined. For example, the best mean IoU for the WGM-SYN dataset is achieved with Fusion loss, using CRFH, and the MFM-ResNet34 architecture, stands at **58.93**. The best performing configuration from prior work, denoted with (*), stands at 50.00.

bones: VGG16, InceptionV3, and ResNet34. Our experimentation also includes three variations of MFM, which incorporates both FFP and SSP, and with three SSP variations: VGG16, InceptionV3, and ResNet34. For our implementations, we use the Segmentation Models Library available in TensorFlow [9].

Post-Processing Configurations. We experiment with three distinct configurations: without post-processing, with CRF post-processing, and with CRFH post-processing. In the scenario without post-processing, the IoU calculation is performed directly on the output of the model (FCN-light, SSP, and MFM). For the CRF post-processing approach, all classes are permissible for relabeling. However, with the CRF post-processing with heuristic (CRFH), as explained in Section 4.6, only the background class pixels (BG) are allowed to be relabeled as HT or PT classes

Weight Initialization. For both FCN-light and SSP models, model weights start from their initial random values at the beginning of the training. We train MFM configurations at last to reuse the weights from the SSP configurations. Thus, for the MFM trainings, we apply transfer learning by initializing with weights from the corresponding, previously trained SSP. For example, when training an MFM (FFP+SSP) with an SSP using the ResNet34 backbone, we initialize its SSP path with the available weights from the

SSP - ResNet34. Lastly, although we have aimed for consistent IoU results by fixing the random seeds, the inherent non-determinism associated with GPU execution remains. As such, we have run our experiments multiple times to validate the IoU value improvements across configurations.

Results. Tables 2 and 3 show the IoU values for the WGM-SYN and SignaTR6K datasets. The overall trend across the results indicates that transitioning from the three-class formulation to the four-class formulation improves the IoU scores. Furthermore, employing larger model backbones generally, but not always, improves the segmentation performance. Among all the model architectures evaluated, the ResNet34 and InceptionV3 backbones achieve the highest performance, which we attribute to their residual connections and different size convolutions as they can better capture fine features from the image. We provide a summary of the trends in the results here, while a more detailed version of the result tables can be found in the supplementary material.

- Prior work [6, 27], which implements the three-class formulation of HT and PT segmentation, generally exhibits the lowest performance. Transitioning from the three-class to the four-class formulation with the same backbone, i.e., FCN-light, with the same number of model parameters and WCE loss, improves the average IoU values by

Formulation	Backbone	# Parameters	Loss function	IoU %				With CRF (IoU %)				With CRFH (IoU %)			
				PT	HT	BG	Mean	PT	HT	BG	Mean	PT	HT	BG	Mean
3-Class	FCN-based [6, 27]	~295K	CE	62.56	88.09	98.40	83.02*	52.68	89.68	99.26	80.46	62.72	90.58	99.05	84.11
			Focal	62.34	88.00	98.45	82.93	44.60	84.86	99.28	76.25	62.57	90.83	99.21	84.21
			Dice	60.45	87.29	97.85	81.86	61.24	88.83	98.17	82.74	60.52	87.88	97.97	82.12
			Weighted CE	60.58	84.89	97.78	81.09	53.45	90.73	99.52	81.23	60.84	86.33	98.24	81.80
			Weighted Focal	61.25	85.74	98.02	81.67	44.27	85.77	99.50	76.51	61.55	87.42	98.58	82.52
			Weighted Dice	60.21	87.00	97.74	81.65	60.72	88.17	97.98	82.29	60.27	87.46	97.83	81.86
			Fusion	61.52	88.39	98.38	82.76	58.94	92.55	99.37	83.62	61.67	90.45	98.91	83.68
4-Class	FCN-based	~295K	CE	64.55	89.21	98.39	84.05	54.60	89.68	99.23	81.17	64.87	91.81	99.06	85.25
			Focal	64.10	88.86	98.32	83.76	46.64	86.01	99.26	77.30	64.34	91.65	99.11	85.03
			Dice	64.37	88.68	98.37	83.81	65.17	90.13	98.59	84.63	64.57	89.14	98.46	84.06
			Weighted CE	63.78	87.52	98.16	83.15	54.77	90.50	99.42	81.56	64.12	89.43	98.71	84.09
			Weighted Focal	63.68	87.71	98.19	83.20	48.18	87.10	99.45	78.24	64.05	89.72	98.80	84.19
			Weighted Dice	64.21	88.57	98.30	83.69	65.08	90.10	98.53	84.57	64.44	89.06	98.39	83.96
			Fusion	64.68	88.48	98.33	83.83	59.83	91.77	99.28	83.63	65.00	90.72	98.91	84.87
4-Class	MFM (FFP + SSP) - InceptionV3	~30M	CE	73.10	<u>92.66</u>	98.77	<u>88.18</u>	63.48	92.89	99.55	85.31	73.05	94.89	99.36	89.10
			Focal	72.80	92.50	98.75	88.01	57.47	90.58	99.54	82.53	72.77	94.74	99.35	88.95
			Dice	72.56	92.20	98.70	87.82	72.38	93.60	99.00	88.33	72.52	94.73	99.37	88.87
			Weighted CE	72.66	91.90	98.54	87.70	62.65	93.04	99.62	85.10	72.63	93.31	98.94	88.29
			Weighted Focal	72.55	92.13	98.62	87.77	59.35	91.64	99.59	83.53	72.50	93.77	99.07	88.45
			Weighted Dice	72.59	92.31	98.70	87.87	72.60	94.00	99.07	<u>88.56</u>	72.54	94.60	99.32	88.82
			Fusion	72.55	92.25	98.71	87.83	65.57	93.49	99.53	86.19	72.49	94.62	99.35	88.82
4-Class	MFM (FFP + SSP) - ResNet34	~24M	CE	72.81	92.56	<u>98.78</u>	88.05	63.04	92.94	99.55	85.17	72.75	94.93	99.39	89.02
			Focal	73.04	92.46	98.75	88.08	53.02	89.04	99.55	80.54	73.00	94.75	99.35	89.03
			Dice	72.96	92.38	98.72	<u>88.02</u>	<u>73.16</u>	93.35	98.93	88.48	72.93	94.79	99.36	89.03
			Weighted CE	72.96	91.96	98.69	87.87	64.49	92.47	99.56	85.51	72.90	94.19	99.27	88.79
			Weighted Focal	73.18	92.10	98.63	87.97	54.96	89.71	99.60	81.42	73.16	93.72	99.06	88.65
			Weighted Dice	72.78	92.32	98.71	87.94	72.95	93.66	99.00	88.53	72.75	94.67	99.34	88.92
			Fusion	73.26	92.45	98.73	88.15	68.38	<u>94.85</u>	99.56	87.60	<u>73.21</u>	94.59	99.31	89.04

Table 3: IoU performance (%) on the SignaTR6K dataset. The maximum value of a column (i.e., class) is underlined, and the overall maximum for a class across different post-processing is both bolded and underlined. For example, the best mean IoU for the SignaTR6K dataset is for CE loss, with CRFH, and the MFM-InceptionV3 architecture at **89.10**. The best performing configuration of prior work, i.e., excluding results for Fusion loss and CRFH, is marked with (*) and is 83.02.

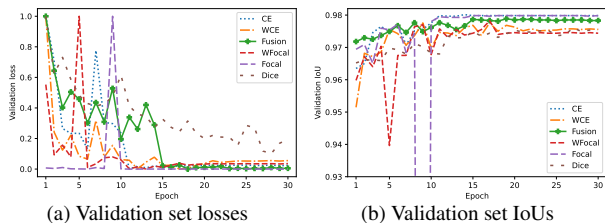


Figure 5: Convergence speed of different loss functions and normalized loss values on the validation set of the SignaTR6K dataset.

8.0% (50→54.02) for the WGM-SYN dataset, and 2.6% (81.09→83.15) for the SignaTR6K dataset.

- Applying CRF post-processing generally degrades the results. In our research, we observe that CRF post processing is dataset dependent and decreases the performance of segmentation in some cases. CRF post-processing relabels aggressively and incorrectly converts HT and PT pixels to background ones, or OV pixels to PT ones. For example, Figure 6i illustrates that how OV pixels are wrongly relabeled to PT class by CRF post-processing. Contrary to CRF post-processing, our CRF heuristic generally improves the IoU values. For example, for the SignaTR6K dataset, for FCN-light with four-class implementation and CE loss, CRFH improves HT IoUs by 2.9% (89.21→91.81).
- Moving from the FCN-light architecture to larger models, i.e., SSP and MFM models, generally improves the results, except for the VGG16 backbone. For VGG16, we ratio-

nalize that having a deep network without residual connections or varied-size convolutions, as seen in ResNet34 and InceptionV3 architectures, hurts the model’s ability in capturing fine features. In addition, we observe that adding FFP helps to improve the performance of the VGG model as we compare SSP and MFM performance values for both datasets. This also confirms the FFP’s ability to capture low-level features that are missed by the SSP path.

- The MFM model with ResNet34 and InceptionV3 backbones generally achieves the best results. MFM-ResNet34 with Fusion loss and CRFH achieves the best result for the WGM-SYN dataset (58.93%), while MFM-InceptionV3 with CE loss and CRFH achieves the best mean IoUs for the SignaTR6K dataset (89.10%). Overall, the best-performing model from our designs improves on the mean IoU performance of the best-performing prior work [27] by 17.9% (50→58.93) and 7.3% (83.02→89.10) for the WGM-SYN and SignaTR6K datasets, respectively.
- Although using skip connections in SSP helps with the segmentation performance and thus less pronounced improvement is observed in MFM, our FFP design is agnostic of the SSP implementation. This distinction becomes clear when VGG16 is employed as SSP; the addition of FFP boosts the IoU scores by 7% and 8% on the WGM-SYN and SignaTR6K datasets, respectively. The FCN-based model with 295K parameters outperforms VGG16 with 30M parameters. Thus, larger models do not necessarily outper-

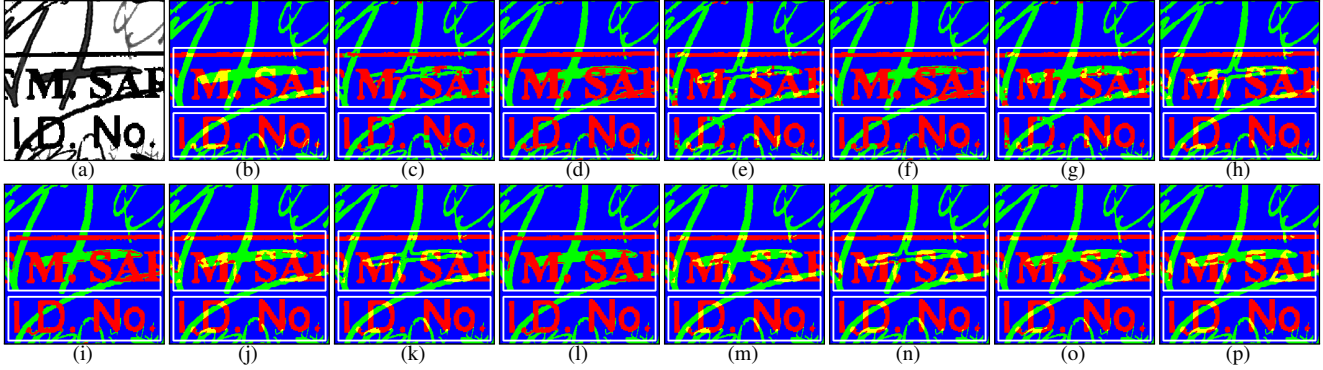


Figure 6: Example results on the test set of the SignaTR6K dataset for our approach compared to the ground truth and prior works. (a) Input image; (b) Ground truth; (c) & (d) 3-class FCN-based [6, 27] with CE loss without (c) and with (d) CRF post-processing; (e), (f), & (g) Our FCN-based 4-class formulation with CE loss without CRF (e), with CRF (f), and with CRFH (g); (h), (i), & (j) SSP-ResNet34 with CE loss without CRF (h), with CRF (i), and with CRFH (j); (k), (l), & (m) MFM-ResNet34 with CE loss without CRF (k), with CRF (l), and with CRFH (m); (n), (o), & (p) MFM-ResNet34 with Fusion loss without CRF (n), with CRF (o), and with CRFH (p).

form smaller ones consistently, and integrating FFP can bring significant improvements in IoU scores. Our findings show that, irrespective of the SSP size and architecture, incorporating FFP improves the performance, and it is essential for segmentation tasks involving fine objects, e.g., text.

Loss Functions. We also perform an analysis to confirm the stability and convergence of the Fusion loss compared to other losses. Figures 5a and 5b compare different loss functions on the validation set of the SignaTR6K dataset. In Figure 5a, we observe that the Fusion loss is generally stable while converging to its minimum loss. Additionally, in Figure 5b, we observe Fusion loss reaches its maximum IoU around Epoch 15, generally faster than most other losses and stays close to the maximum IoU values on the validation set. It is important to highlight that the maximum IoU on the validation set for Focal and CE losses does not necessarily imply the best performance on the test set. Overall, Fusion loss shows a stable behavior and achieves a better performance on the test set, whereas for comparison, Focal loss shows instability at some epochs during the training.

Visual Comparison. Figure 6 shows sample model outputs for the SignaTR6K dataset, with two rectangle regions of interest highlighted to showcase the differences between various models. Figure 6b shows the ground truth, and a visual trend indicates that the performance on the regions of interest improves from Figure 6c to Figure 6p. It is also visually noticeable that CRF post-processing aggressively relabels pixels (6i), whereas CRFH generally improves the results. More visual comparisons on the WGM-SYN and SignaTR6K datasets can be found in the supplementary.

Limitations. Compared to prior work, the complexity and training cost of our approach present some limitations. Our MFM model, being larger than FCN-light, requires greater GPU memory sizes for training, and takes longer to train.

However, we believe some of the limitations on the training time can be offset through transferring weights from SSP models to MFM ones. Additionally, while our approach outperforms prior work on the WGM-SYN dataset, the mean IoU performance is lower compared to the SignaTR6K dataset. This indicates potential limitations in our method’s efficacy for lower-quality original documents, those undergoing low-quality scanning processes (like historical documents), or those with errors from automated labeling. We also attribute the improved performance on the SignaTR6K dataset to its higher quality and our thorough manual annotation, that have resulted in better model training and improved IoU results.

6. Conclusion

Segmentation of handwritten text (HT) and printed text (PT) is vital for digitization and understanding of scanned documents. The complexity increases with the overlapping of different text types. In this research, we introduced SignaTR6K, a new open-source dataset with high-quality, manual pixel-level annotations, sourced from original legal documents. Additionally, we proposed a novel four-class formulation and a new architecture for the segmentation task. Our design leverages both the Fine Feature Path (FFP) and the Semantic Segmentation Path (SSP) to create the Mixed Feature Model (MFM), and incorporates both high-level and low-level features and improves on the text segmentation performance. We also introduced a CRF-based post-processing heuristic (CRFH) that further improves the model output, and included a new loss function, Fusion loss, that combines the advantages of different loss functions and achieves faster convergence and higher stability compared to most of the evaluated losses. In conclusion, our designs outperform the prior work in mean IoU scores by 17.9% and 7.3% on WGM-SYN and SignaTR6K datasets, respectively.

References

- [1] RIMES. <http://www.a2ialab.com/doku.php?id=rimesdatabase:start>. Inaccessible: 2023-01-23.
- [2] Scanned Questionnaire. <https://github.com/jottue/HTSNet>. Accessed: 2023-01-23.
- [3] Thomson Reuters Legal Content Services. <https://www.thomsonreuters.com/en/products-services/legal.html>. Accessed: 2023-01-21.
- [4] Apostolos Antonacopoulos, David Bridson, Christos Papadopoulos, and Stefan Pletschacher. A realistic dataset for performance evaluation of document layout analysis. In *2009 10th International Conference on Document Analysis and Recognition*, pages 296–300. IEEE, 2009.
- [5] Dibyajyoti Dhar, Avishek Garain, Pawan Kumar Singh, and Ram Sarkar. Hp_docpres: A method for classifying printed and handwritten texts in doctor’s prescription. *Multimedia Tools and Applications*, 80(7):9779–9812, 2021.
- [6] Nicolas Dutly, Fouad Slimane, and Rolf Ingold. PHTI-WS: a printed and handwritten text identification web service based on FCN and CRF post-processing. In *2019 International Conference on Document Analysis and Recognition Workshops (ICDARW)*, volume 2, pages 20–25. IEEE, 2019.
- [7] Jürgen Franke and Matthias Oberlander. Writing style detection by statistical combination of classifiers in form reader applications. In *Proceedings of 2nd International Conference on Document Analysis and Recognition (ICDAR’93)*, pages 581–584. IEEE, 1993.
- [8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [9] Pavel Iakubovskii. Segmentation models. https://github.com/qubvel/segmentation_models, 2019.
- [10] Junho Jo, Hyung Il Koo, Jae Woong Soh, and Nam Ik Cho. Handwritten text segmentation via end-to-end learning of convolutional neural networks. *Multimedia Tools and Applications*, 79(43):32137–32150, 2020.
- [11] Celso AML Junior, Matheus Henrique Marques da Silva, Byron Leite Dantas Bezerra, Bruno José Torres Fernandes, and Donato Impedovo. FCN+ RL: a fully convolutional network followed by refinement layers to offline handwritten signature segmentation. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7. IEEE, 2020.
- [12] R Kandan, Nirup Kumar Reddy, KR Arvind, and AG Ramakrishnan. A robust two level classification algorithm for text localization in documents. In *Advances in Visual Computing: Third International Symposium, ISVC 2007, Lake Tahoe, NV, USA, November 26-28, 2007, Proceedings, Part II 3*, pages 96–105. Springer, 2007.
- [13] Florian Kleber, Stefan Fiel, Markus Diem, and Robert Sablatnig. CVL-database: An off-line database for writer retrieval, writer identification and word spotting. In *2013 12th international conference on document analysis and recognition*, pages 560–564. IEEE, 2013.
- [14] Xiao-Hui Li, Fei Yin, and Cheng-Lin Liu. Printed/handwritten texts and graphics separation in complex documents using conditional random fields. In *2018 13th IAPR International Workshop on Document Analysis Systems (DAS)*, pages 145–150. IEEE, 2018.
- [15] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.
- [16] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.
- [17] U-V Marti and Horst Bunke. The IAM-database: an English sentence database for offline handwriting recognition. *International Journal on Document Analysis and Recognition*, 5(1):39–46, 2002.
- [18] Milda Norkute, Nadja Herger, Leszek Michalak, Andrew Mulder, and Sally Gao. Towards explainable ai: Assessing the usefulness and impact of added explainability features in legal document summarization. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–7, 2021.
- [19] Xujun Peng, Srirangaraj Setlur, Venu Govindaraju, and Ramachandrala Sitaram. Handwritten text separation from annotated machine printed documents using markov random fields. *International Journal on Document Analysis and Recognition (IJDAR)*, 16(1):1–16, 2013.
- [20] Anastasia Prikhodina. Handwritten and printed text separation for historical documents. 2021.
- [21] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical*

- image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.
- [22] Mathias Seuret, Marcus Liwicki, and Rolf Ingold. Pixel level handwritten and printed content discrimination in scanned documents. In *2014 14th International Conference on Frontiers in Handwriting Recognition*, pages 423–428. IEEE, 2014.
- [23] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- [24] Stavroula Skylaki, Ali Oskooei, Omar Bari, Nadja Herger, and Zac Kriegman. Named entity recognition in the legal domain using a pointer generator network. *arXiv preprint arXiv:2012.09936*, 2020.
- [25] Nishant Subramani, Alexandre Matton, Malcolm Greaves, and Adrian Lam. A survey of deep learning approaches for ocr and document understanding. *arXiv preprint arXiv:2011.13534*, 2020.
- [26] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.
- [27] Mahsa Vafaie, Oleksandra Bruns, Nastasja Pilz, Jörg Waitelonis, and Harald Sack. Handwritten and printed text identification in historical archival documents. In *Archiving Conference*, volume 19, pages 15–20. Society for Imaging Science and Technology, 2022.
- [28] Zhaobin Wang, E Wang, and Ying Zhu. Image segmentation evaluation: A survey of methods. *Artificial Intelligence Review*, 53:5637–5674, 2020.
- [29] Zhengxin Zhang, Qingjie Liu, and Yunhong Wang. Road extraction by deep residual U-Net. *IEEE Geoscience and Remote Sensing Letters*, 15(5):749–753, 2018.
- [30] Zhi-Hua Zhou and Xu-Ying Liu. Training cost-sensitive neural networks with methods addressing the class imbalance problem. *IEEE Transactions on knowledge and data engineering*, 18(1):63–77, 2005.