UPOCR: Towards Unified Pixel-Level OCR Interface

Dezhi Peng [†] [1 2] Zhenhua Yang [†] [1 2] Jiaxin Zhang [1 2] Chongyu Liu [1 2] Yongxin Shi [1 2] Kai Ding [3 2]
Fengjun Guo [3 2] Lianwen Jin [*] [1 2]

Abstract

Existing optical character recognition (OCR) methods rely on task-specific designs with divergent paradigms, architectures, and training strategies, which significantly increases the complexity of research and maintenance and hinders the fast deployment in applications. To this end, we propose UPOCR, a simple-yet-effective generalist model for Unified Pixel-level OCR interface. Specifically, the UPOCR unifies the paradigm of diverse OCR tasks as image-to-image transformation and the architecture as a vision Transformer (ViT)-based encoder-decoder with learnable task prompts. The prompts push the general feature representations extracted by the encoder towards task-specific spaces, endowing the decoder with task awareness. Moreover, the model training is uniformly aimed at minimizing the discrepancy between the predicted and ground-truth images regardless of the inhomogeneity among tasks. Experiments are conducted on three pixel-level OCR tasks including text removal, text segmentation, and tampered text detection. Without bells and whistles, the experimental results showcase that the proposed method can simultaneously achieve state-of-the-art performance on three tasks with a unified single model, which provides valuable strategies and insights for future research on generalist OCR models. Code is available at https://github.com/shannanyinxiang/UPOCR.

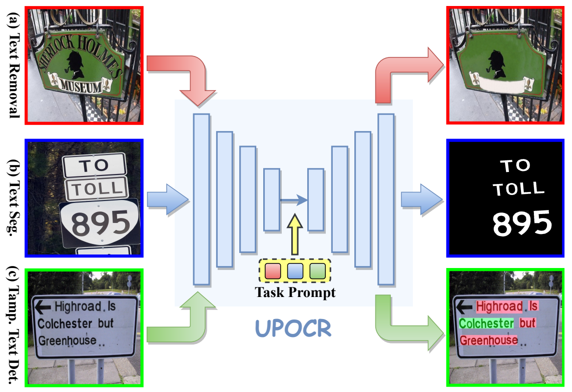


Figure 1. The proposed UPOCR is a unified pixel-level OCR interface which is simultaneously capable of diverse pixel-level OCR tasks (*e.g.*, (a) text removal, (b) text segmentation, and (c) tampered text detection) by prompting ViT-based encoder-decoder, without task- or benchmark-specific finetuning. At the bottom right, red and green colors indicate tampered and real texts, respectively.

1. Introduction

Optical character recognition (OCR) is a flourishing field with numerous real-world applications, encompassing a wide spectrum of pixel-level tasks that require dense per-pixel predictions, *e.g.*, text removal (Nakamura et al., 2017), text segmentation (Xu et al., 2021), and tampered text detection (Wang et al., 2022b). Nowadays, there have been massive methods that specialize in individual tasks, significantly contributing to the OCR advancement.

However, specialized OCR models significantly differ in aspects of paradigms, architectures, and training strategies, especially for pixel-level OCR tasks. **(1) Paradigm:** The differences in paradigms are particularly reflected by divergent input and output formats. For instance, existing approaches to text segmentation (Xu et al., 2021) and tampered text detection (Qu et al., 2023) typically transform the input image into per-pixel classification probabilities to distinguish text strokes or real/tampered texts from backgrounds. On the contrary, text removal (Zhang et al., 2019) is aimed at producing vivid text-erased images from input images. Moreover, recent studies (Liu et al., 2022a; Lee & Choi, 2022; Wang et al., 2023c) commonly consolidate text masks into inputs or outputs to promote text perception of text removal models. **(2) Architecture:** Specialized models typically elaborate dedicated modules for individual tasks. Concretely, explicit text localization modules (Tursun et al.,

[†]Equal contribution [1]South China University of Technology [2]INTSIG-SCUT Joint Lab of Document Image Analysis and Recognition [3]INTSIG Information Co. Ltd. Correspondence to: Lianwen Jin <eelwjin@scut.edu.cn>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

2020; Du et al., 2023a) and multi-step refinements (Liu et al., 2020; Lyu & Zhu, 2022; Wang et al., 2023c) are employed to resolve excessive and inexhaustive text erasure. As for text segmentation, advanced methods (Bonechi et al., 2020; Xu et al., 2021; Ren et al., 2022; Wang et al., 2023b) rely on complex attention and resampling modules as well as semantic information from text recognizers. Moreover, the fusion of frequency and RGB domains is crucial to existing tampered text detection approaches (Wang et al., 2022c; Qu et al., 2023). **(3) Training Strategy:** Specialized models are optimized in disparate manners. In general, the cross-entropy loss is adopted for text segmentation (Xu et al., 2021; Yu et al., 2023) and tampered text detection (Qu et al., 2023) while L1 distance for text removal (Zhang et al., 2019; Liu et al., 2020). Additionally, abundant dedicated losses are designed for individual tasks, *e.g.*, trimap loss (Xu et al., 2021; 2022) and Lovasz loss (Qu et al., 2023). Moreover, some studies (Xu et al., 2021; 2022; Liu et al., 2022a; Lyu et al., 2023; Peng et al., 2024) incorporate discriminators and GAN-based training strategies. The joint training with text localization modules is another trend in recent literature (Wang et al., 2021; 2023c; Yu et al., 2023), requiring auxiliary supervision. These inhomogeneities among specialized OCR models substantially raise the research complexity and increase the real-world deployment and maintenance cost. More importantly, the collaboration between OCR tasks cannot be investigated. Therefore, it is urgent to develop generalist pixel-level OCR models.

Concurrently, several studies have been devoted to establishing generalist interfaces. However, OFA (Wang et al., 2022a) and Unified-IO (Lu et al., 2022) depend on VQ-GAN (Esser et al., 2021) to decode images from discrete tokens, thus limiting the diversity and granularity in the pixel space. Although recent approaches (Alayrac et al., 2022; Liu et al., 2023a; Li et al., 2023; Ye et al., 2023c; Zhang et al., 2024; Zhu et al., 2024) investigate the combination of powerful vision Transformers (ViTs) (Dosovitskiy et al., 2021) and large language models (LLMs) (Touvron et al., 2023), they struggle with OCR tasks (Liu et al., 2023b; Shi et al., 2023) and fail to generate pixels. Painter (Wang et al., 2023a) tackles diverse tasks as inpainting problems but implicitly distinguishes tasks via example pairs, hence difficult to identify OCR tasks with inconspicuous correlation between inputs and outputs. Furthermore, existing generalist OCR models (Kim et al., 2022; Tang et al., 2023; Blecher et al., 2023; Lv et al., 2023; Ye et al., 2023b; Feng et al., 2023b; Ye et al., 2023a; Feng et al., 2023a) primarily focus on document scenarios and cannot handle pixel-level tasks. In particular, some of them (Kim et al., 2022; Tang et al., 2023) still require benchmark-specific finetuning to reach satisfactory performance.

To this end, we propose UPOCR, a simple-yet-effective generalist model for Unified Pixel-level OCR interface. For the first time, UPOCR simultaneously accomplishes multiple pixel-level OCR tasks via prompting ViT-based encoder-decoder as shown in Fig. 1. To achieve this, the proposed method unifies the paradigm, architecture, and training strategy of diverse pixel-level OCR tasks. Specifically, UPOCR unified the paradigm of different tasks as transforming RGB image inputs to RGB image outputs. Moreover, the pure ViT-based encoder-decoder architecture is uniformly adopted for all tasks. To distinguish the ongoing task, we introduce learnable task prompts into the encoder-decoder. Concretely, the encoder first extracts general OCR-related feature representations of the input image. Subsequently, the task prompt pushes the general feature towards the task-specific region, empowering the decoder to generate output images for specific tasks. During training, the model is optimized to minimize the discrepancy between predicted and ground-truth (GT) images at pixel and feature levels, eliminating dedicated designs of loss functions, adversarial learning, and auxiliary supervision.

The effectiveness of UPOCR is extensively verified on three pixel-level OCR tasks, including text removal, text segmentation, and tampered text detection. Without bells and whistles, experimental results showcase that UPOCR simultaneously achieves state-of-the-art performance on all three tasks with a unified single model, significantly surpassing specialized methods for individual tasks. In addition, UPOCR outperforms cutting-edge generalist approaches on these three tasks, demonstrating its proficiency in the pixel-level OCR field.

In summary, the contributions of this paper are as follows.

- We propose UPOCR, a simple-yet-effective generalist model for unified pixel-level OCR interface. Through the unification of paradigms, architectures, and training strategies, the proposed UPOCR is the first to simultaneously excel in diverse pixel-level OCR tasks.
- Learnable task prompts are introduced to guide the ViT-based encoder-decoder architecture. The prompts push general feature representations from the encoder towards regions of individual tasks, allowing the decoder to perform task-specific decoding.
- The generalist capacity of UPOCR is extensively verified on text removal, text segmentation, and tampered text detection tasks, significantly outperforming existing specialized models. In-depth ablation studies are also conducted to provide valuable strategies and insights for future research on generalist OCR methods.

2. Related Work

2.1. Specialized Pixel-Level OCR Model

Text Removal. Text removal is targeted at replacing text strokes with visually coherent backgrounds, primarily focusing on natural scenes. Early approaches (Nakamura et al., 2017; Zhang et al., 2019; Liu et al., 2020) follow a one-stage framework, which implicitly integrates text localization and inpainting processes into a single network in an image-to-image translation manner. However, one-stage approaches struggle with accurate text perception, leaving severe text remnants in text removal results. Therefore, two-stage methods incorporate explicit text segmentation modules (Tursun et al., 2020; Keserwani & Roy, 2021; Lyu & Zhu, 2022; Bian et al., 2022; Du et al., 2023b; Hou et al., 2022; Du et al., 2023a; Wang et al., 2023c; Lyu et al., 2023) or external text detectors (Zdenek & Nakayama, 2020; Conrad & Chen, 2021; Liu et al., 2022a; Qin et al., 2018; Tursun et al., 2019; Tang et al., 2021; Lee & Choi, 2022) for enhanced text localization capacity and have recently dominated the text removal field. Moreover, coarse-to-refine (Liu et al., 2020; Jiang et al., 2022; Tursun et al., 2020; Du et al., 2023a) and multi-step progressive refinements (Lyu & Zhu, 2022; Bian et al., 2022; Du et al., 2023b; Wang et al., 2023c) have also been intensively exploited for exhaustive text removal in recent studies. Nevertheless, ViTEraser (Peng et al., 2024) demonstrated a one-stage framework with ViTs and SegMIM pre-training can significantly outperform previous complicated methods.

Text Segmentation. Text segmentation aims to predict per-pixel classification for distinguishing text strokes from backgrounds. SMANet (Bonechi et al., 2020) inserted multi-scale attention into PSPNet (Zhao et al., 2017)-based architecture. TextFormer (Wang et al., 2023b) and ARM-Net (Ren et al., 2022) further incorporated high-level semantics from text recognizers. Moreover, TexRNet (Xu et al., 2021) dynamically reactivated low-confidence regions and adopted a character-level discriminator. Based on TexRNet, PGTSNet (Xu et al., 2022) additionally integrated a text detector for text-line cropping and employed a line-level discriminator. To fully utilize polygon annotations, Wang et al. (2021) exploited mutual interaction between polygon- and pixel-level segmentation while Yu et al. (2023) designed end-to-end hierarchical segmentation Transformers.

Tampered Text Detection. Tampered text detection is defined as the segmentation (or detection) of tampered (or both real and tampered) texts. In the deep learning era, plenty of approaches (Zhou et al., 2018; Bappy et al., 2019; Kwon et al., 2021; 2022; Dong et al., 2022) have been developed for natural image manipulation detection. Inspired by these methods, Qu et al. (2023) proposed DTD which combines frequency and visual perception for document tampered text detection. Furthermore, Wang et al. (2022c) enhanced Faster R-CNN with RGB and frequency relationship modeling for tampered text detection in receipts. As for natural scenes, Wang et al. (2022b) equipped scene text detectors with the proposed S3R strategy to localize both real and tampered texts.

2.2. Generalist Model

The emergence of Transformer (Vaswani et al., 2017; Dosovitskiy et al., 2021) breaks the boundary between different modalities (Jaegle et al., 2022; Zhang et al., 2023), fostering a broad variety of generalist models. One category of these models unifies both the input and output as sequences and bridges them with a *sequence-to-sequence* learning framework. Pix2Seq (Chen et al., 2022a) pioneered in unifying the output vocabulary of natural language and spatial coordinates and demonstrated effectiveness in object detection. Subsequently, Pix2Seq v2 (Chen et al., 2022b) simultaneously tackled multiple tasks with the guidance of specific prompts. Furthermore, OFA (Wang et al., 2022a) discretized both the input and output as token sequences, accomplishing various uni-modal and cross-modal vision-language tasks. Similarly, Unified-IO (Lu et al., 2022) extended the framework to a wider range of tasks and modalities. With the rise of LLMs (OpenAI, 2023; Touvron et al., 2023), numerous studies (Alayrac et al., 2022; Li et al., 2023; Liu et al., 2023a; Ye et al., 2023c; Zhang et al., 2024; Zhu et al., 2024) connected pretrained ViTs and LLMs for generalist models with stronger reasoning and robustness. Following an *image-to-image* translation pipeline, MAE-VQGAN (Bar et al., 2022) treated diverse tasks as inpainting problems. Painter (Wang et al., 2023a) further investigated visual in-context learning which allows it to adapt to unseen tasks.

In the OCR field, several generalist models have been studied following *sequence-to-sequence* paradigms (Kim et al., 2022; Tang et al., 2023; Blecher et al., 2023; Lv et al., 2023). Moreover, recent approaches (Ye et al., 2023b;a; Feng et al., 2023b;a) augment large multimodal models using OCR-related data and fine-grained visual perception. However, these approaches primarily focused on document scenarios and failed to generate pixels. Moreover, some of them (Tang et al., 2023; Kim et al., 2022) still required benchmark-specific finetuning.

3. Methodology

The proposed UPOCR is a unified pixel-level OCR interface as depicted in Fig. 2. Specifically, UPOCR unifies the paradigm, architecture, and training strategy of diverse pixel-level OCR tasks. In this paper, the UPOCR is particularly verified on simultaneously handling text removal, text segmentation, and tampered text detection tasks.

3.1. Unified Paradigm

As illustrated in Fig. 2(a), despite their divergent targets (*e.g.*, image generation and segmentation), the paradigm of various pixel-level OCR tasks can be unified to translate RGB images to RGB images. As the inputs are inherently RGB images, detailed output formats of selected tasks are

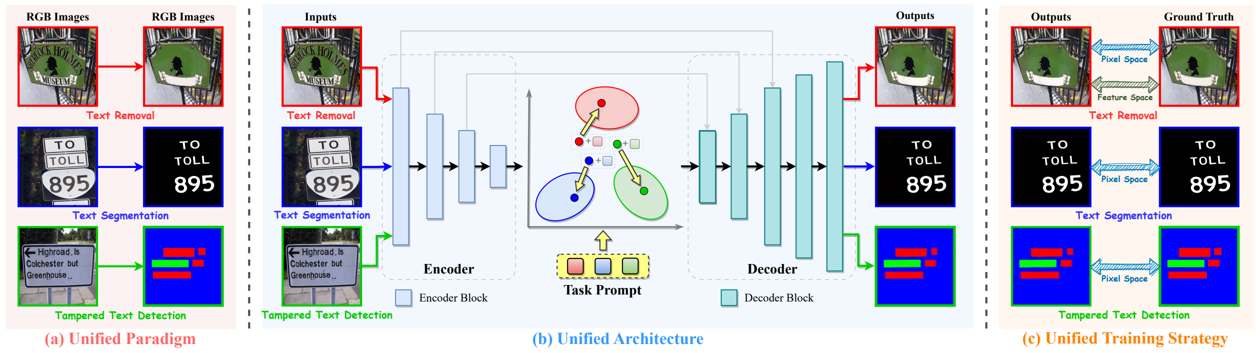


Figure 2. UPOCR is a generalist OCR model which unifies the paradigm, architecture, and training strategy of diverse pixel-level OCR tasks. (a) The paradigm is unified as RGB image to RGB image translation. (b) The ViT-based encoder-decoder architecture is employed for all tasks. Learnable task prompts are inserted to shift general hidden representations to task-specific regions. (c) During training, the model is optimized to minimize the discrepancy between predicted and GT images at pixel and feature spaces.

described as follows.

Text Removal. For the text removal task, the output is the text-erased image corresponding to the input, which is also RGB pictures.

Text Segmentation. Text segmentation aims to assign each pixel to foreground (*i.e.*, text stroke) or background. Existing methods typically conduct per-pixel binary classification. However, under the unified image-to-image translation paradigm, UPOCR predicts RGB images with white and black colors. Specifically, the RGB values for foreground and background pixels are $(255, 255, 255)$ and $(0, 0, 0)$, respectively. During inference, the category is determined by thresholding the distance between the generated RGB value and the pre-defined foreground RGB value.

Tampered Text Detection. Following recent studies (Wang et al., 2022b), we define tampered text detection as per-pixel classification of tampered text, real text, and background categories. Similar to text segmentation, we adopt different RGB values in the output image to represent different categories. Concretely, we assign red $(255, 0, 0)$, green $(0, 255, 0)$, and blue $(0, 0, 255)$ colors to tampered texts, real texts, and backgrounds, respectively. During inference, we compare the distance of predicted RGB values with these three colors to determine the per-pixel category.

3.2. Unified Architecture

As shown in Fig. 2(b), we implement the unified image-to-image translation paradigm by prompting a ViT-based encoder-decoder network. Concretely, the task prompts shift the general representation extracted by the encoder into task-specific regions at feature space, empowering the decoder to produce output images for individual tasks.

Encoder-Decoder. We adopt a ViT-based architecture to implement the encoder-decoder of UPOCR. Specifically, the encoder consists of four sequential blocks, yielding four feature maps with strides of $\{4, 8, 16, 32\}$ *w.r.t* the input image. Each encoder block encapsulates a patch embedding layer for downsampling and multiple Swin Transformer v2 blocks (Liu et al., 2022b). Subsequently, the decoder hierarchically upsamples the final feature of the encoder to strides of $\{16, 8, 4, 2, 1\}$ *w.r.t* the input size through five blocks. Each decoder block concatenates multiple Swin Transformer v2 blocks (Liu et al., 2022b) and a patch splitting layer (Peng et al., 2024) for upsampling. Based on the final feature of the decoder, the output image is predicted using a 3×3 convolution.

Task Prompt. To effectively handle multiple tasks, we introduce learnable task prompts into the encoder-decoder architecture. Retrospectively, recent generalist models (Chen et al., 2022b; Kim et al., 2022; Tang et al., 2023) commonly prepend task prompts comprising natural language or predefined tokens to the decoder for task-specific sequence generation. In contrast, we insert learnable task prompts into the shared feature space of the encoder and decoder as shown in Fig. 2(b). Specifically, for each task, its prompt is formulated as a learnable embedding with the same dimension as the hidden feature from the encoder. To perform a certain task, the corresponding prompt is simply added to every pixel of the hidden feature, pushing the general OCR-related presentations generated by the encoder towards the task-specific region. Subsequently, the decoder translates the adjusted hidden feature into the output image for this specific task. With negligible parameter and computation overhead, the UPOCR can simultaneously deal with diverse tasks in a simple yet effective fashion. See Sec. A.1 of the appendix for details.

3.3. Unified Training Strategy

Thanks to the unified image-to-image paradigm, the training of UPOCR consistently aims to minimize the discrepancy

Table 1. Ablation study on the insertion of task prompts. The **bold** and underline indicate the best and second best, respectively.

Prompt Insert Position			Text Removal				Text Segmentation		Tampered Text Det.	
Encoder	Shared Feature	Decoder	PSNR↑	MSSIM↑	MSE↓	FID↓	fgIoU↑	F↑	mIoU↑	mF↑
✓	✓		36.93	**97.64**	0.0430	**10.45**	88.61	93.96	70.19	82.48
	✓	✓	36.96	**97.64**	**0.0428**	10.46	88.78	94.06	71.13	83.13
✓	✓	✓	36.86	**97.64**	0.0436	10.55	**88.84**	**94.09**	71.40	83.31
	✓		**37.14**	97.62	**0.0428**	10.47	88.76	94.04	**71.71**	**83.53**

between the predicted and GT images at pixel and feature spaces regardless of the inhomogeneity among tasks, as shown in Fig. 2(c). For conciseness, we define the input, output, and GT images as I_{in}, I_{out}, and I_{gt}, respectively.

Pixel Space. The discrepancy in pixel space is measured by the L1 distance between output and GT images. Furthermore, a multi-scale L1 loss is involved to enhance the perception at multiple granularities during training. Specifically, multi-scale images $I_{out}^{\frac{1}{4}}$ and $I_{out}^{\frac{1}{2}}$ are predicted based on the features from the 3*rd* and 4*th* decode blocks, each through a 3×3 convolution. Denoting $\mathbb{I}_{out} = \{I_{out}^{\frac{1}{4}}, I_{out}^{\frac{1}{2}}, I_{out}\}$ and $\mathbb{I}_{gt} = \{I_{gt}^{\frac{1}{4}}, I_{gt}^{\frac{1}{2}}, I_{gt}\}$ ($I_{gt}^{\frac{1}{4}}$ and $I_{gt}^{\frac{1}{2}}$ is resized from I_{gt}), the pixel loss is formulated as:

$$L_{pix} = \sum\nolimits_{i=1}^{3} \alpha_i ||\mathbb{I}_{out}^{i} - \mathbb{I}_{gt}^{i}||_1, \tag{1}$$

where the balancing factor α is empirically set to $\{5, 6, 10\}$ and the superscript i indicates the i-th element of the array.

Feature Space. For tasks associated with realistic image generation, the similarity at high-level feature space is critical. Therefore, we additionally align the output and GT images at the feature space for the text removal task. The feature loss L_{feat} is composed of perceptual loss L_{per} (Johnson et al., 2016) and style loss L_{sty} (Gatys et al., 2016):

$$L_{feat} = \alpha_{per} L_{per} + \alpha_{sty} L_{sty}, \tag{2}$$

where L_{per} and L_{sty} are calculated following previous studies (Liu et al., 2020; 2022a) using a pretrained VGG-16 (Simonyan & Zisserman, 2015) network. In addition, the α_{per} and α_{sty} are heuristically set to 0.01 and 120, respectively.

Total Loss. The total loss L_{total} is the sum of pixel loss L_{pix} and feature loss L_{feat} (if applicable):

$$L_{total} = L_{pix} + L_{feat}. \tag{3}$$

4. Experiment

4.1. Experiment Setting

Task and Dataset. We investigate three pixel-level OCR tasks, including text removal, text segmentation, and tampered text detection, to demonstrate the effectiveness of UPOCR on generalist pixel-level OCR processing. The SCUT-EnsText (Liu et al., 2020), TextSeg (Xu et al., 2021), and Tampered-IC13 (Wang et al., 2022b) datasets are employed for these three tasks, respectively.

Network Architecture. The encoder-decoder architecture of UPOCR inherits from ViTEraser-Small (Peng et al., 2024) but incorporates three learnable prompts for multi-task processing, totally containing 108M parameters. The encoder, decoder, and task prompts comprise 49.6M, 58.6M, and 2,304 parameters, respectively. During training, UPOCR are initialized using pretrained ViTEraser-Small (with SegMIM pre-training). The input size is set to 512×512.

Other Details. *See Sec. A of the appendix for implementation details.*

4.2. Evaluation Metrics

Text Removal. Following previous methods (Liu et al., 2020; 2022a), the evaluation for text removal involves image- and detection-eval metrics. The image-eval metrics include PSNR, MSSIM, MSE, AGE, pEPs, pCEPs, and FID while the detection-eval metrics are precision (P), recall (R), and f-measure (F) using the pretrained text detector CRAFT (Baek et al., 2019). Note that MSSIM and MSE are presented in percent (%).

Text Segmentation. The intersection over union (IoU), P, R, and F of foreground pixels are employed for evaluation on text segmentation (Xu et al., 2021).

Tampered Text Detection. The evaluation metrics include the P, R, F, and IoU of both real and tampered pixels. Additionally, the mean f-measure (mF) and mean IoU (mIoU) of real and tampered pixels are calculated for overall evaluation following Wang et al. (2022b).

4.3. Ablation Study

Insertion of Task Prompts. The crucial component of the generalist proficiency of UPOCR is the learnable task prompt. As specified in Sec. 3.2 and Fig. 2(b), the task prompt is inserted to the shared feature space between the encoder and decoder, following the idea that the encoder extracts versatile OCR-related features and the decoder decodes task-specific output images from adjusted hidden features. Nevertheless, the task prompts can also be inserted

Table 2. Ablation study on task collaboration. (TR: text removal, TS: text segmentation, TTD: tampered text detection)

TR	TS	TTD	Text Removal				Text Segmentation		Tampered Text Det.	
			PSNR↑	MSSIM↑	MSE↓	FID↓	fgIoU↑	F↑	mIoU↑	mF↑
✓			36.97	97.57	0.0500	10.50	-	-	-	-
	✓		-	-	-	-	88.83	94.08	-	-
		✓	-	-	-	-	-	-	67.73	80.72
✓	✓		36.91	**97.64**	0.0450	**10.40**	88.64	93.98	-	-
	✓	✓	-	-	-	-	**89.07**	**94.22**	70.78	82.89
✓		✓	37.08	97.60	0.0452	10.47	-	-	69.26	81.83
✓	✓	✓	**37.14**	97.62	**0.0428**	10.47	88.76	94.04	**71.71**	**83.53**

Table 3. Comparison with **specialized** models for text removal on the SCUT-EnsText dataset. For a fair comparison, MTRNet++ uses empty coarse masks and GaRNet uses text masks from pretrained CRAFT (Baek et al., 2019) instead of leveraging GT text masks.

Method	Image-Eval							Detection-Eval		
	PSNR↑	MSSIM↑	MSE↓	AGE↓	pEPs↓	pCEPs↓	FID↓	R↓	P↓	F↓
Original	-	-	-	-	-	-	-	69.5	79.4	74.1
Pix2pix (Isola et al., 2017)	26.70	88.56	0.37	6.09	0.0480	0.0227	46.88	35.4	69.7	47.0
STE (Nakamura et al., 2017)	25.47	90.14	0.47	6.01	0.0533	0.0296	43.39	5.9	40.9	10.2
EnsNet (Zhang et al., 2019)	29.54	92.74	0.24	4.16	0.0307	0.0136	32.71	32.8	68.7	44.4
MTRNet++ (Tursun et al., 2020)	29.63	93.71	0.28	3.51	0.0305	0.0168	35.68	15.1	63.8	24.4
EraseNet (Liu et al., 2020)	32.30	95.42	0.15	3.02	0.0160	0.0090	19.27	4.6	53.2	8.5
SSTE (Tang et al., 2021)	35.34	96.24	0.09	-	-	-	-	3.6	-	-
PSSTRNet (Lyu & Zhu, 2022)	34.65	96.75	0.14	1.72	0.0135	0.0074	-	5.1	47.7	9.3
CTRNet (Liu et al., 2022a)	35.20	97.36	0.09	2.20	0.0106	0.0068	13.99	1.4	38.4	2.7
GaRNet (Lee & Choi, 2022)	35.45	97.14	0.08	1.90	0.0105	0.0062	15.50	1.6	42.0	3.0
MBE (Hou et al., 2022)	35.03	97.31	-	2.06	0.0128	0.0088	-	-	-	-
PEN (Du et al., 2023b)	35.72	96.68	0.05	1.95	0.0071	0.0020	-	2.1	**26.2**	3.9
PERT (Wang et al., 2023c)	33.62	97.00	0.13	2.19	0.0135	0.0088	-	4.1	50.5	7.6
SAEN (Du et al., 2023a)	34.75	96.53	0.07	1.98	0.0125	0.0073	-	-	-	-
FETNet (Lyu et al., 2023)	34.53	97.01	0.13	1.75	0.0137	0.0080	-	5.8	51.3	10.5
ViTEraser-Base (Peng et al., 2024)	37.11	97.61	0.0474	**1.70**	0.0066	0.0035	**10.15**	**0.389**	29.7	**0.768**
UPOCR (Ours)	**37.14**	**97.62**	**0.0428**	1.72	**0.0064**	**0.0034**	10.47	0.614	36.6	1.208

into the intermediate features of the encoder and decoder, after each encoder block and decoder block. In Tab. 1, we investigate different inserting positions of task prompts. If task prompts are supposed to be integrated into the encoder or decoder, they will go through linear layers to match the required dimensions of intermediate features. The experimental results suggest simply inserting task prompts into the shared feature space can effectively prompt the decoder to generate images for specific tasks, Moreover, it also brings extremely small overhead on parameters (2.3K) and computational costs (only an element-wise addition to a $16 \times 16 \times 768$ feature map).

Task Collaboration. In Tab. 2, we conduct the experiments with different task compositions. Based on these results, We come to the following insights. (1) *The task with insufficient samples can easily benefit from the joint training with other tasks.* For example, the Tampered-IC13 (Wang et al., 2022b) dataset used for tampered text detection contains only 229 training samples, which is inadequate to train a network with 108M parameters. Therefore, the performance of tampered text detection is consistently improved because the fundamental text localization capacity can be boosted by other tasks. (2) *The joint training of tasks leads to mutual collaboration if with similar targets but negative effect if with exclusive targets.* For instance, text removal aims at erasing the texts while text segmentation needs to highlight them at a fine-grained stroke level. Therefore, additional text removal task always leads to downgraded text segmentation performance. However, comparing the 2*nd* and 5*th* rows, tampered text detection helps improve the accuracy of text segmentation, because the former provides auxiliary polygon-level text location supervision. Due to the large model capacity, the final model can achieve a good balance of three tasks.

Table 4. Comparison with **specialized** models for text segmentation on the TextSeg dataset. The performances of U-Net and SegFormer are cited from (Ren et al., 2022) and (Yu et al., 2023), respectively.

Method	fgIoU↑	P↑	R↑	F↑
U-Net (Ronneberger et al., 2015)	-	89.00	77.40	82.80
DeepLabV3+ (Chen et al., 2018)	84.07	-	-	91.40
HRNetv2-W48 (Wang et al., 2020a)	85.03	-	-	91.40
HRNetv2-W48+OCR (Wang et al., 2020a)	85.98	-	-	91.80
TexRNet+DeepLabV3+ (Xu et al., 2021)	86.06	-	-	92.10
TexRNet+HRNetv2-W48 (Xu et al., 2021)	86.84	-	-	92.40
SegFormer (Xie et al., 2021)	84.59	-	-	91.60
ARM-Net (Ren et al., 2022)	-	92.80	92.60	92.70
TFT (Yu et al., 2023)	87.11	-	-	93.10
UPOCR (Ours)	**88.76**	**94.55**	**93.55**	**94.04**

Table 5. Comparsion with **specialized** models for tampered text detection on the Tampered-IC13 dataset. CAT-Net can only perform binary classification between tampered texts and backgrounds. The **mIoU** and **mF** metrics indicate the overall effectiveness on this task.

Method	Real Text				Tampered Text				mIoU↑	mF↑
	IoU↑	P↑	R↑	F↑	IoU↑	P↑	R↑	F↑		
Detection-based Methods										
S3R (Wang et al., 2022b)+ContourNet (Wang et al., 2020b)	-	77.88	54.80	64.33	-	86.68	**91.45**	88.99	-	76.66
ViTEraser (Peng et al., 2024)+ContourNet (Wang et al., 2020b)	-	56.84	**75.82**	64.97	-	**92.62**	85.77	**89.06**	-	77.02
Segmentation-based Methods										
DeepLabV3+ (Chen et al., 2018)	48.12	79.83	54.78	64.98	72.21	89.75	78.71	83.86	60.17	74.42
HRNetv2 (Wang et al., 2020a)	43.26	76.35	49.95	60.39	73.12	89.98	79.60	84.47	58.19	72.43
Swin-Uper (Liu et al., 2021)	61.82	87.82	67.62	76.41	77.28	89.67	84.83	87.18	69.55	81.80
SegFormer (Xie et al., 2021)	53.22	86.39	58.09	69.47	**77.78**	91.78	83.60	87.50	65.50	78.49
BEiT-Uper (Bao et al., 2022)	57.07	81.23	65.74	72.67	70.88	82.27	83.66	82.96	63.98	77.82
CAT-Net (Kwon et al., 2022)	-	-	-	-	28.31	31.45	73.91	44.13	-	-
UPOCR (Ours)	**71.80**	**93.31**	75.70	**83.59**	71.62	79.76	87.53	83.46	**71.71**	**83.53**

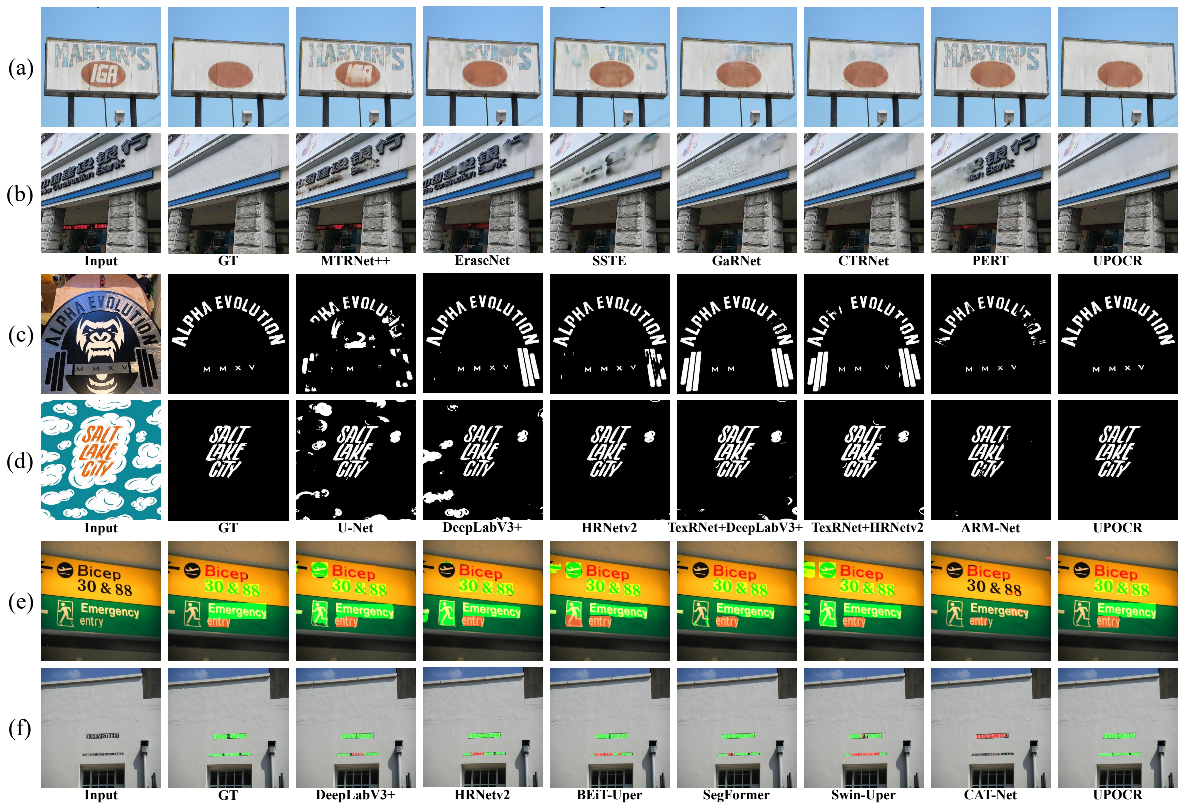


Figure 3. Qualitative comparison of UPOCR and existing **specialized** models on (a)-(b) text removal, (c)-(d) text segmentation, and (e)-(f) tampered text detection (red: tampered, green: real). Zoom in for a better view.

4.4. Comparison with Specialized Models

The comparisons of UPOCR with existing specialized methods for text removal, text segmentation, and tampered text detection are presented in Tabs. 3, 4, and 5, respectively. Furthermore, the visualization results on three tasks are shown in Fig. 3. Without bells and whistles, the generalist UPOCR with shared parameters can simultaneously outperform existing specialized models for individual tasks. (1) ***Text Removal***: The UPOCR eschews the complicated text localization modules, external text detectors, and multi-step refinements. Furthermore, UPOCR discards the GAN-based training strategy and mask branch of ViTEraser during training. Following a concise pipeline, UPOCR achieves state-of-the-art image-eval performance on SCUT-EnsText (Liu et al., 2020), outperforming ViTEraser-Base (Peng et al., 2024) with a lighter-weight ViTEraser-Small architecture. As for detection-eval metrics, the 0.614% recall of UPOCR is comparable to ViTEraser, demonstrating nearly all texts are exhaustively erased. It is worth noting that compared

Table 6. Comparison with **generalist** models on pixel-level OCR tasks.

Method	Text Removal							Text Segmentation				Tampered Text Det.	
	PSNR↑	MSSIM↑	MSE↓	AGE↓	pEPs↓	pCEPs↓	FID↓	fgIoU↑	P↑	R↑	F↑	mIoU↑	mF↑
Painter (Wang et al., 2023a)	27.13	91.67	0.2942	8.68	0.0898	0.0425	21.90	86.36	93.40	91.97	92.68	69.26	81.83
UPOCR (Ours)	**37.14**	**97.62**	**0.0428**	**1.72**	**0.0064**	**0.0034**	**10.47**	**88.76**	**94.55**	**93.55**	**94.04**	**71.71**	**83.53**

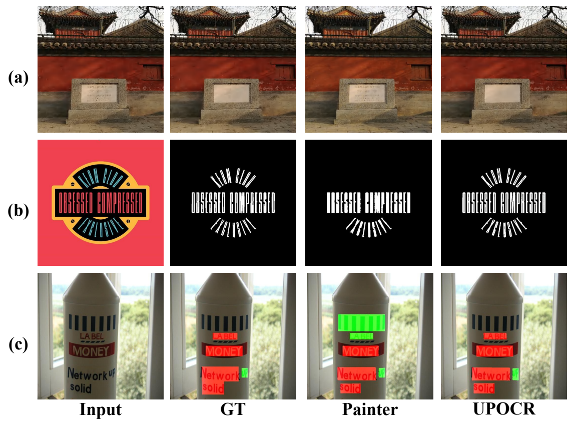


Figure 4. Qualitative comparison of UPOCR and Painter on (a) text removal, (b) text segmentation, and (c) tampered text detection (red: tampered, green: real). Zoom in for a better view.

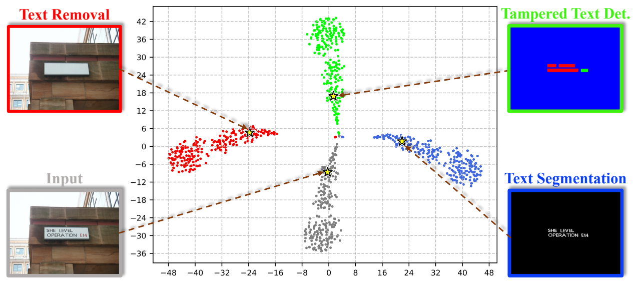


Figure 5. The t-SNE visualization of general features (gray circles) extracted by the encoder and task-specific features integrated with task prompts. Red, green and blue circles are for text removal, tampered text detection, and text segmentation, respectively. The example input image and three-task outputs are visualized along with corresponding features (yellow stars).

with earlier approaches, UPOCR significantly outperforms them in terms of both image- and detection-eval metrics. (2) ***Text Segmentation.*** Although previous methods utilize attentive modules for refinements, text detectors for coarse text localization, and text recognizers for semantic supervision, UPOCR significantly outperforms them on TextSeg (Xu et al., 2021) using a single encoder-decoder without extra modules and annotations. (3) ***Tampered Text Detection.*** The UPOCR achieves the best performance of 71.71% mIoU and 83.53% mF on Tampered-IC13 (Wang et al., 2022b), eliminating the need for frequency domain fusion (Qu et al., 2023; Kwon et al., 2022; Wang et al., 2022c) and adaptation based on text detectors (Wang et al., 2022b). Note that the segmentation-based methods are reimplemented using MMSegmentation[1] and CAT-Net's official codes[2].

4.5. Comparison with Generalist Models

To demonstrate the effectiveness of UPOCR over existing generalist models, we re-train Painter (Wang et al., 2023a) using the official implementation[3] and the same dataset as ours. The performances of UPOCR and Painter are listed in Tab. 6 while the visualizations are illustrated in Fig. 4. Because Painter learns the task target from an example input-output pair, it may hardly grasp the inconspicuous correlation such as tiny text erasing (Fig. 4(a)) and a strong ability to distinguish texts from text-like patterns (Fig. 4(c)). Moreover, as the image is predicted through inpainting, the output cannot guarantee consistent colors as inputs, *e.g.*, red vs. orange walls in Fig. 4(a). Therefore, UPOCR surpasses Painter on all three tasks, especially the text removal task.

4.6. Interpretability of Task Prompts

To investigate the mechanism of task prompts, we visualize the features before and after integrated with task prompts in Fig. 5. Specifically, we perform all three tasks on the 233 testing samples of Tampered-IC13 using corresponding task prompts. We can observe that there are clear boundaries between the general features (gray circles) extracted by the encoder and the task-specific features (red, green, and blue circles), indicating the proposed task prompts can effectively adapt the general features into task-specific regions to generate output images for individual tasks.

5. Conclusion

In this paper, we propose UPOCR, a first-of-its-kind simple-yet-effective unified pixel-level OCR interface. To acquire generalist capability, UPOCR unifies the paradigm, architecture, and training strategy of diverse pixel-level OCR tasks. Specifically, existing divergent paradigms are unified as RGB image to RGB image transformation. To implement this paradigm, UPOCR uniformly adopts a ViT-based

[1]https://github.com/open-mmlab/mmsegmentation
[2]https://github.com/mjkwon2021/CAT-Net
[3]https://github.com/baaivision/Painter

encoder-decoder with learnable task prompts to handle various tasks. During training, the strategy is unified to minimize the discrepancy between the predicted and ground-truth images at pixel and feature spaces. Extensive experiments are conducted on text removal, text segmentation, and tampered text detection to verify the generalist proficiency of UPOCR. The experimental results demonstrate that UPOCR simultaneously achieves state-of-the-art performance with shared parameters, significantly surpassing specialized OCR models. Comprehensive ablation studies and visual analyses are also presented to provide in-depth insights. We believe this work could be extended to broader tasks and spark more research on generalist OCR models.

Acknowledgement

This research is supported in part by National Natural Science Foundation of China (Grant No.: 62441604, 61936003).

Impact Statement

This paper presents work whose goal is to advance the fields of optical character recognition and machine learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Alayrac, J.-B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Mensch, A., Millican, K., Reynolds, M., et al. Flamingo: A visual language model for few-shot learning. *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 35:23716–23736, 2022.

Baek, Y., Lee, B., Han, D., Yun, S., and Lee, H. Character region awareness for text detection. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pp. 9365–9374, 2019.

Bao, H., Dong, L., Piao, S., and Wei, F. BEiT: BERT pre-training of image Transformers. In *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2022.

Bappy, J. H., Simons, C., Nataraj, L., Manjunath, B., and Roy-Chowdhury, A. K. Hybrid LSTM and encoder–decoder architecture for detection of image forgeries. *IEEE Trans. Image Process.*, 28(7):3286–3300, 2019.

Bar, A., Gandelsman, Y., Darrell, T., Globerson, A., and Efros, A. Visual prompting via image inpainting. *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 35:25005–25017, 2022.

Bian, X., Wang, C., Quan, W., Ye, J., Zhang, X., and Yan, D.-M. Scene text removal via cascaded text stroke detection and erasing. *Comput. Vis. Media*, 8:273–287, 2022.

Blecher, L., Cucurull, G., Scialom, T., and Stojnic, R. Nougat: Neural optical understanding for academic documents. *arXiv preprint arXiv:2308.13418*, 2023.

Bonechi, S., Bianchini, M., Scarselli, F., and Andreini, P. Weak supervision for generating pixel–level annotations in scene text segmentation. *Pattern Recognit. Lett.*, 138: 1–7, 2020.

Cao, H., Wang, Y., Chen, J., Jiang, D., Zhang, X., Tian, Q., and Wang, M. Swin-Unet: Unet-like pure Transformer for medical image segmentation. In *Proc. Eur. Conf. Comput. Vis. Worksh. (ECCVW)*, pp. 205–218, 2022.

Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., and Adam, H. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proc. Eur. Conf. Comput. Vis. (ECCV)*, pp. 801–818, 2018.

Chen, T., Saxena, S., Li, L., Fleet, D. J., and Hinton, G. Pix2Seq: A language modeling framework for object detection. In *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2022a.

Chen, T., Saxena, S., Li, L., Lin, T.-Y., Fleet, D. J., and Hinton, G. E. A unified sequence interface for vision tasks. *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 35: 31333–31346, 2022b.

Conrad, B. and Chen, P.-I. Two-stage seamless text erasing on real-world scene images. In *Proc. IEEE Int. Conf. Image Process. (ICIP)*, pp. 1309–1313, 2021.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. ImageNet: A large-scale hierarchical image database. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pp. 248–255, 2009.

Dong, C., Chen, X., Hu, R., Cao, J., and Li, X. MVSS-Net: Multi-view multi-scale supervised networks for image manipulation detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(3):3539–3553, 2022.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2021.

Du, X., Zhou, Z., Zheng, Y., Ma, T., Wu, X., and Jin, C. Modeling stroke mask for end-to-end text erasing. In *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, pp. 6151–6159, 2023a.

Du, X., Zhou, Z., Zheng, Y., Wu, X., Ma, T., and Jin, C. Progressive scene text erasing with self-supervision. *Comput. Vis. Image Underst.*, 233:103712, 2023b.

Esser, P., Rombach, R., and Ommer, B. Taming Transformers for high-resolution image synthesis. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pp. 12873–12883, 2021.

Feng, H., Liu, Q., Liu, H., Zhou, W., Li, H., and Huang, C. DocPedia: Unleashing the power of large multimodal model in the frequency domain for versatile document understanding. *arXiv preprint arXiv:2311.11810*, 2023a.

Feng, H., Wang, Z., Tang, J., Lu, J., Zhou, W., Li, H., and Huang, C. UniDoc: A universal large multimodal model for simultaneous text detection, recognition, spotting and understanding. *arXiv preprint arXiv:2308.11592*, 2023b.

Gatys, L. A., Ecker, A. S., and Bethge, M. Image style transfer using convolutional neural networks. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pp. 2414–2423, 2016.

Hou, Y., Chen, J., and Wang, Z. Multi-branch network with ensemble learning for text removal in the wild. In *Proc. Asian Conf. Comput. Vis. (ACCV)*, pp. 1333–1349, 2022.

Isola, P., Zhu, J.-Y., Zhou, T., and Efros, A. A. Image-to-image translation with conditional adversarial networks. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pp. 1125–1134, 2017.

Jaegle, A., Borgeaud, S., Alayrac, J.-B., Doersch, C., Ionescu, C., Ding, D., Koppula, S., Zoran, D., Brock, A., Shelhamer, E., et al. Perceiver IO: A general architecture for structured inputs & outputs. In *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2022.

Jiang, G., Wang, S., Ge, T., Jiang, Y., Wei, Y., and Lian, D. Self-supervised text erasing with controllable image synthesis. In *Proc. ACM Int. Conf. Multimedia (ACM MM)*, pp. 1973–1983, 2022.

Johnson, J., Alahi, A., and Fei-Fei, L. Perceptual losses for real-time style transfer and super-resolution. In *Proc. Eur. Conf. Comput. Vis. (ECCV)*, pp. 694–711, 2016.

Keserwani, P. and Roy, P. P. Text region conditional generative adversarial network for text concealment in the wild. *IEEE Trans. Circuits Syst. Video Technol.*, 32(5): 3152–3163, 2021.

Kim, G., Hong, T., Yim, M., Nam, J., Park, J., Yim, J., Hwang, W., Yun, S., Han, D., and Park, S. OCR-free document understanding Transformer. In *Proc. Eur. Conf. Comput. Vis. (ECCV)*, pp. 498–517, 2022.

Kwon, M.-J., Yu, I.-J., Nam, S.-H., and Lee, H.-K. CAT-Net: Compression artifact tracing network for detection and localization of image splicing. In *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, pp. 375–384, 2021.

Kwon, M.-J., Nam, S.-H., Yu, I.-J., Lee, H.-K., and Kim, C. Learning JPEG compression artifacts for image manipulation detection and localization. *Int. J. Comput. Vis.*, 130(8):1875–1895, 2022.

Lee, H. and Choi, C. The surprisingly straightforward scene text removal method with gated attention and region of interest generation: A comprehensive prominent model analysis. In *Proc. Eur. Conf. Comput. Vis. (ECCV)*, pp. 457–472, 2022.

Li, J., Li, D., Savarese, S., and Hoi, S. BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *Proc. Int. Conf. Mach. Learn. (ICML)*, pp. 19730–19742, 2023.

Liu, C., Liu, Y., Jin, L., Zhang, S., Luo, C., and Wang, Y. EraseNet: End-to-end text removal in the wild. *IEEE Trans. Image Process.*, 29:8760–8775, 2020.

Liu, C., Jin, L., Liu, Y., Luo, C., Chen, B., Guo, F., and Ding, K. Don't forget me: Accurate background recovery for text removal via modeling local-global context. In *Proc. Eur. Conf. Comput. Vis. (ECCV)*, pp. 409–426, 2022a.

Liu, H., Li, C., Wu, Q., and Lee, Y. J. Visual instruction tuning. In *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2023a.

Liu, Y., Li, Z., Li, H., Yu, W., Huang, M., Peng, D., Liu, M., Chen, M., Li, C., Jin, L., et al. On the hidden mystery of OCR in large multimodal models. *arXiv preprint arXiv:2305.07895*, 2023b.

Liu, Z. and He, K. A decade's battle on dataset bias: Are we there yet? *arXiv preprint arXiv:2403.08632*, 2024.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B. Swin Transformer: Hierarchical vision Transformer using shifted windows. In *Proc. Int. Conf. Comput. Vis. (ICCV)*, pp. 10012–10022, 2021.

Liu, Z., Hu, H., Lin, Y., Yao, Z., Xie, Z., Wei, Y., Ning, J., Cao, Y., Zhang, Z., Dong, L., et al. Swin Transformer v2: Scaling up capacity and resolution. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pp. 12009–12019, 2022b.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. In *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2019.

Lu, J., Clark, C., Zellers, R., Mottaghi, R., and Kembhavi, A. Unified-IO: A unified model for vision, language, and multi-modal tasks. In *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2022.

Lv, T., Huang, Y., Chen, J., Cui, L., Ma, S., Chang, Y., Huang, S., Wang, W., Dong, L., Luo, W., et al. Kosmos-2.5: A multimodal literate model. *arXiv preprint arXiv:2309.11419*, 2023.

Lyu, G. and Zhu, A. PSSTRNet: Progressive segmentation-guided scene text removal network. In *Proc. Int. Conf. Multimedia and Expo (ICME)*, pp. 1–6, 2022.

Lyu, G., Liu, K., Zhu, A., Uchida, S., and Iwana, B. K. FETNet: Feature erasing and transferring network for scene text removal. *Pattern Recognit.*, pp. 109531, 2023.

Nakamura, T., Zhu, A., Yanai, K., and Uchida, S. Scene text eraser. In *Proc. Int. Conf. Doc. Anal. Recognit. (ICDAR)*, pp. 832–837, 2017.

OpenAI. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Peng, D., Liu, C., Liu, Y., and Jin, L. ViTEraser: Harnessing the power of vision Transformers for scene text removal with SegMIM pretraining. In *Proc. AAAI Conf. Artif. Intell. (AAAI)*, 2024.

Qin, S., Wei, J., and Manduchi, R. Automatic semantic content removal by learning to neglect. In *Proc. Br. Mach. Vis. Conf. (BMVC)*, pp. 1–12, 2018.

Qu, C., Liu, C., Liu, Y., Chen, X., Peng, D., Guo, F., and Jin, L. Towards robust tampered text detection in document image: New dataset and new solution. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pp. 5937–5946, 2023.

Ren, Y., Zhang, J., Chen, B., Zhang, X., and Jin, L. Looking from a higher-level perspective: Attention and recognition enhanced multi-scale scene text segmentation. In *Proc. Asian Conf. Comput. Vis. (ACCV)*, pp. 3138–3154, 2022.

Ronneberger, O., Fischer, P., and Brox, T. U-Net: Convolutional networks for biomedical image segmentation. In *Proc. Int. Conf. Med. Image Comput. Comput.-Assisted Intervention (MICCAI)*, pp. 234–241, 2015.

Shi, Y., Peng, D., Liao, W., Lin, Z., Chen, X., Liu, C., Zhang, Y., and Jin, L. Exploring OCR capabilities of GPT-4V(ision): A quantitative and in-depth evaluation. *arXiv preprint arXiv:2310.16809*, 2023.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. In *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2015.

Tang, Z., Miyazaki, T., Sugaya, Y., and Omachi, S. Stroke-based scene text erasing using synthetic data for training. *IEEE Trans. Image Process.*, 30:9306–9320, 2021.

Tang, Z., Yang, Z., Wang, G., Fang, Y., Liu, Y., Zhu, C., Zeng, M., Zhang, C., and Bansal, M. Unifying vision, text, and layout for universal document processing. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pp. 19254–19264, 2023.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al. LLaMA: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Tursun, O., Zeng, R., Denman, S., Sivapalan, S., Sridharan, S., and Fookes, C. MTRNet: A generic scene text eraser. In *Proc. Int. Conf. Doc. Anal. Recognit. (ICDAR)*, pp. 39–44, 2019.

Tursun, O., Denman, S., Zeng, R., Sivapalan, S., Sridharan, S., and Fookes, C. MTRNet++: One-stage mask-based scene text eraser. *Comput. Vis. Image Underst.*, 201: 103066, 2020.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 30, 2017.

Wang, C., Zhao, S., Zhu, L., Luo, K., Guo, Y., Wang, J., and Liu, S. Semi-supervised pixel-level scene text segmentation by mutually guided network. *IEEE Trans. Image Process.*, 30:8212–8221, 2021.

Wang, J., Sun, K., Cheng, T., Jiang, B., Deng, C., Zhao, Y., Liu, D., Mu, Y., Tan, M., Wang, X., et al. Deep high-resolution representation learning for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 43(10):3349–3364, 2020a.

Wang, P., Yang, A., Men, R., Lin, J., Bai, S., Li, Z., Ma, J., Zhou, C., Zhou, J., and Yang, H. OFA: Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework. In *Proc. Int. Conf. Mach. Learn. (ICML)*, pp. 23318–23340, 2022a.

Wang, X., Wang, W., Cao, Y., Shen, C., and Huang, T. Images speak in images: A generalist painter for in-context visual learning. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pp. 6830–6839, 2023a.

Wang, X., Wu, C., Yu, H., Li, B., and Xue, X. TextFormer: Component-aware text segmentation with Transformer. In *Proc. Int. Conf. Multimedia and Expo (ICME)*, pp. 1877–1882, 2023b.

Wang, Y., Xie, H., Zha, Z.-J., Xing, M., Fu, Z., and Zhang, Y. ContourNet: Taking a further step toward accurate arbitrary-shaped scene text detection. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pp. 11753–11762, 2020b.

Wang, Y., Xie, H., Xing, M., Wang, J., Zhu, S., and Zhang, Y. Detecting tampered scene text in the wild. In *Proc. Eur. Conf. Comput. Vis. (ECCV)*, pp. 215–232, 2022b.

Wang, Y., Zhang, B., Xie, H., and Zhang, Y. Tampered text detection via RGB and frequency relationship modeling. *Chin. J. Netw. Inf. Secur.*, 8(3):29–40, 2022c.

Wang, Y., Xie, H., Wang, Z., Qu, Y., and Zhang, Y. What is the real need for scene text removal? Exploring the background integrity and erasure exhaustivity properties. *IEEE Trans. Image Process.*, 32:4567–4580, 2023c.

Wang, Z., Cun, X., Bao, J., Zhou, W., Liu, J., and Li, H. Uformer: A general U-shaped Transformer for image restoration. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pp. 17683–17693, 2022d.

Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J. M., and Luo, P. SegFormer: Simple and efficient design for semantic segmentation with Transformers. In *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, pp. 12077–12090, 2021.

Xu, X., Zhang, Z., Wang, Z., Price, B., Wang, Z., and Shi, H. Rethinking text segmentation: A novel dataset and a text-specific refinement approach. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pp. 12045–12055, 2021.

Xu, X., Qi, Z., Ma, J., Zhang, H., Shan, Y., and Qie, X. BTS: A bi-lingual benchmark for text segmentation in the wild. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pp. 19152–19162, 2022.

Ye, J., Hu, A., Xu, H., Ye, Q., Yan, M., Dan, Y., Zhao, C., Xu, G., Li, C., Tian, J., et al. mPLUG-DocOwl: Modularized multimodal large language model for document understanding. *arXiv preprint arXiv:2307.02499*, 2023a.

Ye, J., Hu, A., Xu, H., Ye, Q., Yan, M., Xu, G., Li, C., Tian, J., Qian, Q., Zhang, J., Jin, Q., He, L., Lin, X. A., and Huang, F. UReader: Universal OCR-free visually-situated language understanding with multimodal large language model. In *Conf. Empir. Methods Nat. Lang. Process. (EMNLP)*, 2023b.

Ye, Q., Xu, H., Xu, G., Ye, J., Yan, M., Zhou, Y., Wang, J., Hu, A., Shi, P., Shi, Y., et al. mPLUG-Owl: Modularization empowers large language models with multimodality. *arXiv preprint arXiv:2304.14178*, 2023c.

Yu, H., Wang, X., Niu, K., Li, B., and Xue, X. Scene text segmentation with text-focused Transformers. In *Proc. ACM Int. Conf. Multimedia (ACM MM)*, pp. 2898–2907, 2023.

Zdenek, J. and Nakayama, H. Erasing scene text with weak supervision. In *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, pp. 2238–2246, 2020.

Zhang, R., Han, J., Zhou, A., Hu, X., Yan, S., Lu, P., Li, H., Gao, P., and Qiao, Y. LLaMA-Adapter: Efficient fine-tuning of language models with zero-init attention. In *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2024.

Zhang, S., Liu, Y., Jin, L., Huang, Y., and Lai, S. EnsNet: Ensconce text in the wild. In *Proc. AAAI Conf. Artif. Intell. (AAAI)*, pp. 801–808, 2019.

Zhang, Y., Gong, K., Zhang, K., Li, H., Qiao, Y., Ouyang, W., and Yue, X. Meta-Transformer: A unified framework for multimodal learning. *arXiv preprint arXiv:2307.10802*, 2023.

Zhao, H., Shi, J., Qi, X., Wang, X., and Jia, J. Pyramid scene parsing network. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pp. 2881–2890, 2017.

Zhou, P., Han, X., Morariu, V. I., and Davis, L. S. Learning rich features for image manipulation detection. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pp. 1053–1061, 2018.

Zhu, D., Chen, J., Shen, X., Li, X., and Elhoseiny, M. MiniGPT-4: Enhancing vision-language understanding with advanced large language models. In *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2024.

A. Implementation Details

In this section, we supplement every detail to implement the proposed UPOCR, which may not be thoroughly specified in the main paper due to the page limit.

A.1. Network Architecture

As described in Secs. 3.2 and 4.1 of the main paper, the UPOCR employs a vision Transformer (ViT)-based encoder-decoder with learnable task prompts. The detailed network architecture of UPOCR is presented in Tab. 7 where the notations are defined as follows.

- D_i^{enc}: The downsampling ratio of the patch embedding layer in the *i*-th block of the encoder.
- E_i^{enc}: The output dimension of the patch embedding layer in the *i*-th block of the encoder.
- W_i^{enc}: The window size of the Swinv2 (Liu et al., 2022b) blocks in the *i*-th block of the encoder.
- H_i^{enc}: The number of heads of the Swinv2 blocks in the *i*-th block of the encoder.
- C_i^{enc}: The feature dimension of the Swinv2 blocks in the *i*-th block of the encoder.
- N^{tp}: The number of task prompts which is equal to the number of tasks that the model is supposed to simultaneously deal with.
- C^{tp}: The dimension of task prompts.
- U_i^{dec}: The upsampling ratio of the patch splitting layer in the *i*-th block of the decoder.
- E_i^{dec}: The output dimension of the patch splitting layer in the *i*-th block of the decoder.
- W_i^{dec}: The window size of the Swinv2 blocks in the *i*-th block of the decoder.
- H_i^{dec}: The number of heads of the Swinv2 blocks in the *i*-th block of the decoder.
- C_i^{dec}: The feature dimension of the Swinv2 blocks in the *i*-th block of the decoder.

As shown in Tab. 7, supposing the shape of the input RGB image is $H \times W \times 3$, the encoder hierarchically produces features $\{f_i^{enc} \in \mathbb{R}^{\frac{H}{2^{(i+1)}} \times \frac{W}{2^{(i+1)}} \times C_i^{enc}}\}_{i=1}^4$. Then the learnable prompt of the target task is integrated into the final feature f_4^{enc} of the encoder, yielding a task-specific feature for decoding. Subsequently, the decoder hierarchically generates features $\{f_i^{dec} \in \mathbb{R}^{\frac{H}{2^{(5-i)}} \times \frac{W}{2^{(5-i)}} \times C_i^{dec}}\}_{i=1}^5$ based on the task-specific feature. Finally, the output image is predicted based on feature f_5^{dec} through a 3×3 convolution.

Task Prompt Insertion. The task prompts are formulated as a set of learnable embeddings $f^{tp} \in \mathbb{R}^{N^{tp} \times C^{tp}}$, where $N^{tp} = 3$ and $C^{tp} = 768$ are the number of tasks and embedding dimension, respectively. To perform the *i*-th task, the correspond prompt $f_i^{tp} \in \mathbb{R}^{1 \times 768}$ is first repeated by $\frac{H}{32} \times \frac{W}{32}$ times, yielding a feature $\hat{f}_i^{tp} \in \mathbb{R}^{\frac{H}{32} \times \frac{W}{32} \times 768}$. Then the $\hat{f}_i^{tp}$ is element-wise added to the final feature $f_4^{enc} \in \mathbb{R}^{\frac{H}{32} \times \frac{W}{32} \times 768}$ of the encoder, pushing the general representations towards task-specific regions.

Patch Embedding. Given an input feature map or image $f_{in} \in \mathbb{R}^{h \times w \times c_{in}}$, a patch embedding layer with a downsampling ratio of r and an output dimension of c_{out} first flattens each $r \times r$ patch, yield an intermediate feature $f' \in \mathbb{R}^{\frac{h}{r} \times \frac{w}{r} \times r^2 c_{in}}$. Then a linear layer is adopted to transform the dimension of feature f', producing the output feature $f_{out} \in \mathbb{R}^{\frac{h}{r} \times \frac{w}{r} \times c_{out}}$.

Patch Splitting. Supposing the input feature is $f_{in} \in \mathbb{R}^{h \times w \times c_{in}}$, a patch splitting layer with r upsampling ratio and c_{out} output dimension first generates an intermediate feature $f' \in \mathbb{R}^{rh \times rw \times \frac{c_{in}}{r^2}}$ by decomposing each pixel of the input feature into a $r \times r$ patch. Subsequently, the feature f' goes through a linear layer to transform the feature dimension, yielding the output feature $f_{out} \in \mathbb{R}^{rh \times rw \times c_{out}}$.

Lateral Connection. As shown in Fig. 2(b) of the main paper, lateral connections are built between the encoder and decoder blocks to shortcut the transmission of fine-grained representations. Specifically, the encoder features $\{f_i^{enc}\}_{i=1}^3$ are laterally

Table 7. Detailed network architecture of the proposed UPOCR.

	Block	Output Size	Layer Name	Details
Encoder	Block 1	$\frac{H}{4} \times \frac{W}{4}$	Patch Embedding	$D_1^{enc} = 4$, $E_1^{enc} = 96$
			Swinv2 block	$\begin{bmatrix} W_1^{enc} = 16 \\ H_1^{enc} = 3 \\ C_1^{enc} = 96 \end{bmatrix} \times 2$
	Block 2	$\frac{H}{8} \times \frac{W}{8}$	Patch Embedding	$D_2^{enc} = 2$, $E_2^{enc} = 192$
			Swinv2 block	$\begin{bmatrix} W_2^{enc} = 16 \\ H_2^{enc} = 6 \\ C_2^{enc} = 192 \end{bmatrix} \times 2$
	Block 3	$\frac{H}{16} \times \frac{W}{16}$	Patch Embedding	$D_3^{enc} = 2$, $E_3^{enc} = 384$
			Swinv2 block	$\begin{bmatrix} W_3^{enc} = 16 \\ H_3^{enc} = 12 \\ C_3^{enc} = 384 \end{bmatrix} \times 18$
	Block 4	$\frac{H}{32} \times \frac{W}{32}$	Patch Embedding	$D_4^{enc} = 2$, $E_4^{enc} = 768$
			Swinv2 block	$\begin{bmatrix} W_4^{enc} = 16 \\ H_4^{enc} = 24 \\ C_4^{enc} = 768 \end{bmatrix} \times 2$
Task Prompts		$\frac{H}{32} \times \frac{W}{32}$	$N^{tp} = 3, C^{tp} = 768$	
Decoder	Block 1	$\frac{H}{16} \times \frac{W}{16}$	Swinv2 block	$\begin{bmatrix} W_1^{dec} = 8 \\ H_1^{dec} = 24 \\ C_1^{dec} = 768 \end{bmatrix} \times 2$
			Patch Splitting	$U_1^{dec} = 2$, $E_1^{dec} = 384$
	Block 2	$\frac{H}{8} \times \frac{W}{8}$	Swinv2 block	$\begin{bmatrix} W_2^{dec} = 8 \\ H_2^{dec} = 12 \\ C_2^{dec} = 384 \end{bmatrix} \times 18$
			Patch Splitting	$U_2^{dec} = 2$, $E_2^{dec} = 192$
	Block 3	$\frac{H}{4} \times \frac{W}{4}$	Swinv2 block	$\begin{bmatrix} W_3^{dec} = 8 \\ H_3^{dec} = 6 \\ C_3^{dec} = 192 \end{bmatrix} \times 2$
			Patch Splitting	$U_3^{dec} = 2$, $E_3^{dec} = 96$
	Block 4	$\frac{H}{2} \times \frac{W}{2}$	Swinv2 block	$\begin{bmatrix} W_4^{dec} = 8 \\ H_4^{dec} = 3 \\ C_4^{dec} = 96 \end{bmatrix} \times 2$
			Patch Splitting	$U_4^{dec} = 2$, $E_4^{dec} = 48$
	Block 5	$H \times W$	Swinv2 block	$\begin{bmatrix} W_5^{dec} = 8 \\ H_5^{dec} = 2 \\ C_5^{dec} = 48 \end{bmatrix} \times 2$
			Patch Splitting	$U_5^{dec} = 2$, $E_5^{dec} = 24$

connected to the decoder features $\{f_{4-i}^{dec}\}_{i=1}^{3}$. As for the architecture, if the feature $f_1 \in \mathbb{R}^{h \times w \times c}$ is connected to feature $f_2 \in \mathbb{R}^{h \times w \times c}$, the lateral connection processes the feature f_1 sequentially using a 1×1 convolution with c channels for non-linear transformation, two 3×3 convolutions with $2c$ channels for expanding, and a 1×1 convolution with c channels for shrinking, following EraseNet (Liu et al., 2020). Then the resulting feature is element-wise added to feature f_2.

A.2. Loss Function

Pixel Space. As described in Sec. 3.3 and defined in Eq. (1) of the main paper, the pixel loss is formulated as the weighted sum of L1 distances between multi-scale output images $\mathbb{I}_{out} = \{I_{out}^{\frac{1}{4}}, I_{out}^{\frac{1}{2}}, I_{out}\}$ and ground-truth (GT) images $\mathbb{I}_{gt} = \{I_{gt}^{\frac{1}{4}}, I_{gt}^{\frac{1}{2}}, I_{gt}\}$.

In practical implementation, for samples of the text removal task, the GT text box mask M_{gt} is incorporated to focus more on the discrepancy in text regions, following most existing text removal methods (Liu et al., 2020; Zhang et al., 2019; Liu et al., 2022a). Specifically, the $M_{gt} \in \mathbb{R}^{H \times W}$ is a binary mask at the bounding-box level where 1 and 0 values indicate text and non-text pixels, respectively. Similar to $\mathbb{I}_{gt}$, we also resize M_{gt} to multiple scales denoted as $\mathbb{M}_{gt} = \{M_{gt}^{\frac{1}{4}}, M_{gt}^{\frac{1}{2}}, M_{gt}\}$. Then the pixel loss of text removal samples is calculated as

$$\begin{aligned} L_{pix}^{tr} = &\sum\nolimits_{i=1}^{3} \alpha_i ||(\mathbb{I}_{out(i)} - \mathbb{I}_{gt(i)}) \odot \mathbb{M}_{gt(i)}||_1 \\ &+ \beta_i ||(\mathbb{I}_{out(i)} - \mathbb{I}_{gt(i)}) \odot (1 - \mathbb{M}_{gt(i)})||_1, \end{aligned} \tag{4}$$

where α and β are empircally set to $\{5, 6, 10\}$ (as specified in Sec. 3.3 of the main paper) and $\{0.8, 1, 2\}$, respectively.

As for the text segmentation and tampered text detection tasks, smooth L1 loss functions are employed in Eq. (1) of the main paper to replace the standard L1 distance, because it is not required to precisely align the pixel values of output and GT images but critical to penalize the outliers for these two segmentation-oriented tasks.

Feature Space. As specified in Sec. 3.3 and Eq. (2) of the main paper, a feature loss L_{feat} containing perceptual loss L_{per} and style loss L_{sty} is adopted to ensure the visual plausibility of generated images. Concretely, the losses L_{per} and L_{sty} are formulated as

$$\begin{aligned} I_{out}^{*} =& I_{out} \odot M_{gt} + I_{in} \odot (1 - M_{gt}), && (5) \\ L_{per} =& \sum\nolimits_{i=1}^{3} ||\Phi_i(I_{out}) - \Phi_i(I_{gt})||_1 \\ &+ ||\Phi(I_{out}^{*}) - \Phi_i(I_{gt})||_1, && (6) \\ L_{sty} =& \sum\nolimits_{i=1}^{3} ||Gram(\Phi_i(I_{out})) - Gram(\Phi_i(I_{gt}))||_1 \\ &+ ||Gram(\Phi_i(I_{out}^{*})) - Gram(\Phi_i(I_{gt}))||_1, && (7) \end{aligned}$$

where I_{in} is the input image and $Gram(\cdot)$ calculates the Gram matrix of the input feature map. Moreover, the $\Phi_i(x)$ represents the feature map produced by the *i*-th pooling layer of an ImageNet (Deng et al., 2009)-pretrained VGG-16 (Simonyan & Zisserman, 2015) network fed with an input x.

A.3. Dataset Statistics

As described in Sec. 4.1 of the main paper, experiments are conducted using the SCUT-EnsText (Liu et al., 2020), TextSeg (Xu et al., 2021), and Tampered-IC13 (Wang et al., 2022b) datasets for the text removal, text segmentation, and tampered text detection tasks, respectively. The statistics of these datasets are introduced as follows.

SCUT-EnsText is a real-world scene text removal dataset, comprising 2,749 samples for training and 813 samples for testing.

TextSeg is a large-scale fine-annotated text segmentation dataset with 4,024 images of scene text and design text. The training, validating, and testing sets contain 2,646, 340, and 1,038 samples, respectively.

Tampered-IC13 is aimed at tampered scene text detection in the wild. The dataset is divided into 229 training samples and 233 testing samples. The annotation includes bounding boxes of real and tampered texts.

A.4. Training Setting

The proposed UPOCR is implemented with PyTorch[4]. During training, the parameters of UPOCR are initialized using the pretrained ViTEraser-Small weights (with SegMIM pre-training) (Peng et al., 2024). Subsequently, the model is optimized for 80,000 iterations with a batch size of 48 using an AdamW (Loshchilov & Hutter, 2019) optimizer in a multi-task fashion. Specifically, each batch consists of 16 samples from SCUT-EnsText (Liu et al., 2020) for text removal, 16 samples from TextSeg (Xu et al., 2021) for text segmentation, and 16 samples from Tampered-IC13 (Wang et al., 2022b) for tampered text detection. The size of training images is set to 512 $\times$ 512. The learning rate is initialized as 0.0005 and linearly decays per 200 iterations, finally reaching 0.00001 at the last 200 iterations. The training lasts approximately 36 hours using two NVIDIA A100 GPUs with 80GB memory. **Note that we perform no task- or benchmark-specific finetuning.**

A.5. Inference

In this section, we detail the inference procedure mentioned in Sec. 3.1 of the main paper. For instance, given an input image $I_{in} \in \mathbb{R}^{H \times W \times 3}$, the UPOCR produces an output image $I_{out} \in \mathbb{R}^{H \times W \times 3}$. In the following, we introduce how to obtain the formatted prediction for individual tasks from the I_{out} in detail.

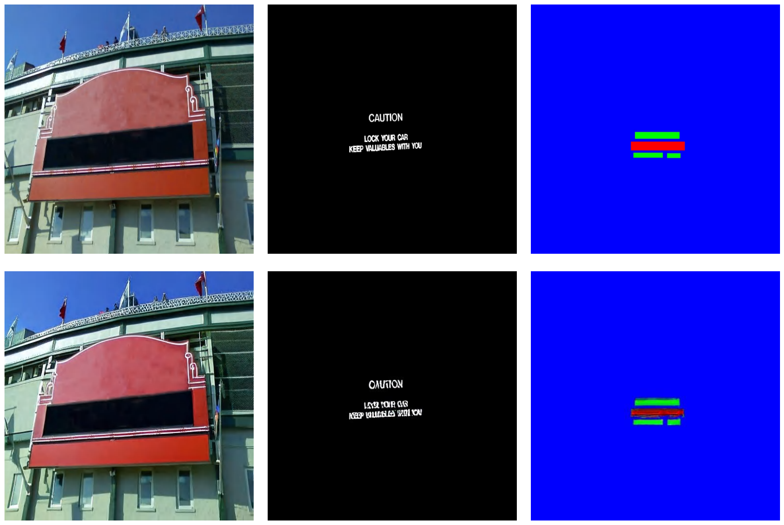
Text Removal. The output image I_{out} is exactly the text-erased image without additional processing.

Text Segmentation. As we define the RGB values of foreground and background pixels as $(255, 255, 255)$ and $(0, 0, 0)$, the text-stroke pixels are determined by setting a threshold of the distance between corresponding RGB values and $(255, 255, 255)$. To achieve this efficiently, we first normalize I_{out} to the range of $[0, 1]$ and then average the three channels,

[4]https://pytorch.org/

Table 8. Upper bound of OFA performance on three pixel-level OCR tasks. The performance of the proposed UPOCR is also provided for comparison, which has already significantly surpassed the upper bound of OFA on text removal. The **bold** and underline indicate the best and the second best, respectively. (Seq. Len.: Sequence Length)

Image Size	Seq. Len.	Text Removal							Text Segmentation		Tampered Text Det.	
		PSNR↑	MSSIM↑	MSE↓	AGE↓	pEPs↓	pCEPs↓	FID↓	fgIoU↑	F↑	mIoU↑	mF↑
512×512	4096	24.94	78.95	0.4871	8.74	0.1013	0.0125	19.81	**92.97**	**96.36**	**96.23**	**98.07**
256×256	1024	24.50	72.88	0.5110	9.43	0.1227	0.0286	41.24	84.15	91.39	94.98	97.42
UPOCR		**37.14**	**97.62**	**0.0428**	**1.72**	**0.0064**	**0.0034**	**10.47**	88.76	94.04	71.71	83.53



(a) Text Removal **(b) Text Segmentation** **(c) Tampered Text Det.**

Figure 6. Visualization of GT images (top) and corresponding reconstructed results (bottom) by VQGAN. The image size is 512×512. Zoom in for a better view.

yielding $\hat{I}_{out} \in \mathbb{R}^{H \times W}$. Then the pixel at (i, j) position is identified as the text stroke if $\hat{I}_{out}^{(i,j)} > 0.4$ and the background otherwise.

Tampered Text Detection. As described in Sec. 3.1 of the main paper, we compare the distance of the generated RGB values to $(255, 0, 0)$ (red for tamper texts), $(0, 255, 0)$ (green for real texts), and $(0, 0, 255)$ (blue for backgrounds) to determine per-pixel categories. In practical implementation, it is equivalent to finding the color with the maximum value in the RGB triplet and assigning the corresponding category to the pixel, getting rid of the complex calculation of distances.

B. Further Comparison with Generalist Model

In Sec. 4.5 and Tab. 6 of the main paper, we compare the proposed UPOCR with the cutting-edge generalist model (*i.e.*, Painter (Wang et al., 2023a)) that is based on image-to-image paradigms. However, existing generalist models (Wang et al., 2022a; Lu et al., 2022) with sequence-to-sequence paradigms are also able to perform image-to-image translation. For instance, given an input image, OFA (Wang et al., 2022a) produces a sequence composed of discrete tokens from the VQGAN (Esser et al., 2021) codebook. After that, the output image is reconstructed from the generated sequence through the decoder of VQGAN. In this way, the OFA accomplishes the image-to-image transformation in a sequence-to-sequence manner. Therefore, based on the OFA model, we further conduct experiments and in-depth analyses to verify the effectiveness of sequence-to-sequence generalist models on pixel-level OCR tasks.

Upper Bound of OFA Performance. As described above, OFA relies on the VQGAN decoder to reconstruct the output image. However, there has already been information loss in the encoding and decoding process of VQGAN, limiting the upper bound of OFA performance. To quantify the information loss, we use VQGAN to encode the GT image I_{gt} into a sequence S_{gt} comprising discrete tokens from the codebook and then directly decode an image I_{dec} from the S_{gt}. Because

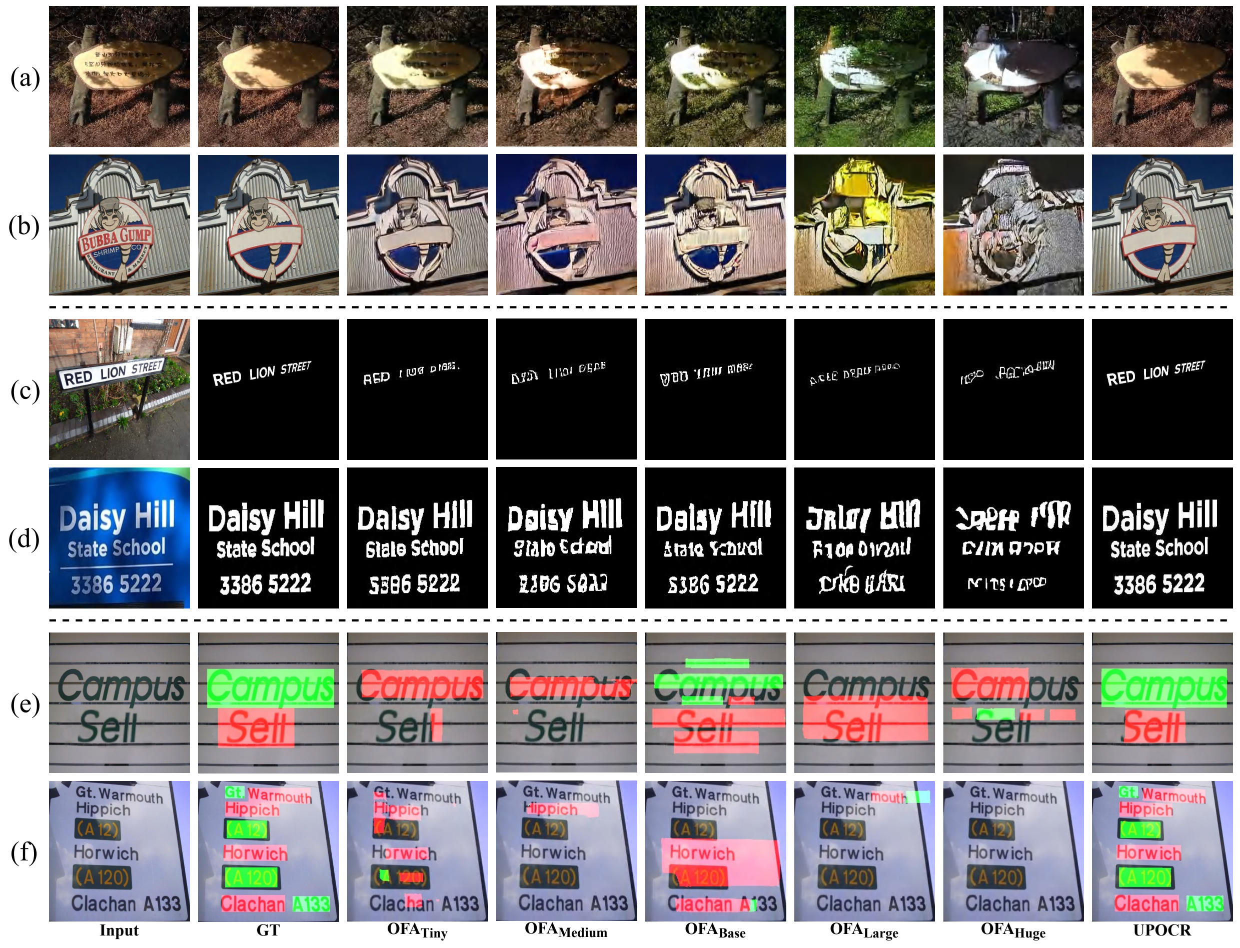


Figure 7. Qualitative comparison of different scales of OFA with UPOCR on (a)-(b) text removal, (c)-(d) text segmentation, and (e)-(f) tampered text detection. Zoom in for a better view.

the OFA is optimized to predict the sequence S_{gt} from the input image, the metrics computed using I_{dec} are the upper bound of OFA performance.

In Tab. 8, we present the upper bounds with two sizes of GT images, including 256×256 and 512×512. Using the VQGAN with a factor of 8 adopted by the OFA, they are encoded into 1024- and 4096-token sequences, respectively. The metrics listed in Tab. 8 demonstrate that the encoding and decoding procedure based on VQGAN itself has caused severe information loss. Especially for the text removal task, the proposed UPOCR has already significantly outperformed the upper bound of OFA. The reason may be that the decoder of VQGAN cannot reconstruct the details and complex patterns and also struggles to guarantee color consistency as illustrated in Fig. 6.

Comparison with OFA. Because of the overwhelming computational costs required to train an OFA with a sequence length of 4096, we set the size of input and output images to 256×256 following the original configuration of OFA. Tab. 9 presents the comparison between UPOCR and different scales of OFA. Moreover, the visualizations are shown in Fig. 7. Limited by the intrinsic sequence-to-sequence mechanism, OFA exhibits significantly inferior performance on all three pixel-level OCR tasks that require strong image-to-image translation capacity. **(1) Text Removal:** As the model size increases, the OFA tends to cause larger color deviation and miss more details of the input image as shown in Fig. 7(a)-(b), leading to decreasing evaluation metrics. **(2) Text Segmentation:** As shown in Fig. 7(c)-(d), OFA struggles to generate the complex patterns of texts at the stroke level, primarily limited by the reconstruction capacity of the VQGAN decoder. Moreover, as the network goes deeper, the less fine-grained alignment with the input image can be guaranteed. It can be seen that OFA$_{huge}$ just draws text-like but meaningless patterns. **(3) Tampered Text Detection:** With the indirect supervision of token sequences with

Table 9. Comparison with **generalist** models on pixel-level OCR tasks. (Params: Parameters)

Method	Text Removal							Text Segmentation		Tampered Text Det.		Params↓
	PSNR↑	MSSIM↑	MSE↓	AGE↓	pEPs↓	pCEPs↓	FID↓	fgIoU↑	F↑	mIoU↑	mF↑	
Sequence-to-Sequence-Based Generalist Model												
OFA$_{tiny}$ (Wang et al., 2022a)	20.07	67.34	1.2419	15.86	0.2262	0.1038	68.46	48.40	65.23	14.70	25.17	**33M**
OFA$_{medium}$ (Wang et al., 2022a)	19.13	62.38	1.4990	18.06	0.2639	0.1354	88.05	38.51	55.61	10.19	17.86	93M
OFA$_{base}$ (Wang et al., 2022a)	19.45	64.73	1.3994	16.88	0.2430	0.1172	78.85	47.77	64.66	6.34	11.90	182M
OFA$_{large}$ (Wang et al., 2022a)	16.04	58.08	3.0576	20.93	0.3115	0.1791	119.82	30.26	46.46	6.07	11.41	472M
OFA$_{huge}$ (Wang et al., 2022a)	15.73	55.80	3.2342	22.27	0.3351	0.1993	134.73	27.75	43.44	3.78	7.25	930M
Image-to-Image-Based Generalist Model												
Painter (Wang et al., 2023a)	27.13	91.67	0.2942	8.68	0.0898	0.0425	21.90	86.36	92.68	69.26	81.83	371M
UPOCR (Ours)	**37.14**	**97.62**	**0.0428**	**1.72**	**0.0064**	**0.0034**	**10.47**	**88.76**	**94.04**	**71.71**	**83.53**	108M

Table 10. Ablation study on different model sizes. (Params: Parameters)

Model Size	Text Removal							Text Segmentation		Tampered Text Det.		Params↓
	PSNR↑	MSSIM↑	MSE↓	AGE↓	pEPs↓	pCEPs↓	FID↓	fgIoU↑	F↑	mIoU↑	mF↑	
UPOCR-Tiny	36.87	97.59	0.0430	1.77	0.0065	**0.0034**	10.60	87.33	93.23	68.84	81.54	**65M**
UPOCR-Small	37.14	**97.62**	**0.0428**	1.72	**0.0064**	**0.0034**	**10.47**	**88.76**	**94.04**	71.71	83.53	108M
UPOCR-Base	**37.16**	**97.62**	0.0451	**1.68**	0.0066	0.0035	10.54	87.20	93.16	**72.95**	**84.36**	192M

condensed information, OFA can hardly distinguish the inconspicuous difference between real and tampered texts, resulting in significantly poor performance on tampered text detection (Tab. 9). Moreover, the OFA is likely to produce visually plausible patches of red and green colors, ignoring the real text distribution of the input image, as shown in Fig. 7(e)-(f). Finally, it is worth noting that the proposed UPOCR substantially outperforms existing generalist approaches based on either sequence-to-sequence or image-to-image paradigms, demonstrating its remarkable effectiveness and promising future in building unified pixel-level OCR interfaces.

Training Details of Painter and OFA. Following the training setting of UPOCR as specified in Sec. A.4, the Painter (Sec. 4.5 of the main paper) and OFA are trained for 80,000 iterations with a batch size of 48. Moreover, the text removal, text segmentation, and tampered text detection tasks each occupy 16 samples of a batch. Other training settings are kept the same as their original implementations.

C. Ablation Study on Model Size

As described in Sec. 4.1 of the main paper, the encoder-decoder architecture of UPOCR inherits from ViTEraser-Small (Peng et al., 2024). In Tab. 10, we further investigate the effect of model size on the generalist capabilities of UPOCR. In addition to UPOCR (denoted as UPOCR-Small for clarity), we build UPOCR-Tiny and UPOCR-Base following the encoder-decoder architectures of ViTEraser-Tiny and ViTEraser-Base, respectively. The UPOCR-Tiny and UPOCR-Base are trained following the same setting as UPOCR and also use the weights of corresponding ViTEraser as initialization. It can be seen that UPOCR-Small significantly outperforms UPOCR-Tiny on all three tasks. However, UPOCR-Base with a larger model size exhibits inferior performance than UPOCR-Small on text segmentation and most metrics of text removal, probably due to insufficient training samples. Therefore, we opt for UPOCR-Small as the final implementation of the proposed UPOCR.

D. Ablation Study on Weight Initialization

Retrospectively, existing generalist models (Li et al., 2023; Alayrac et al., 2022; Liu et al., 2023a; Lv et al., 2023; Zhu et al., 2024) mostly rely on pretrained weights from strong vision or language models. As described in Sec. 4.1 of the main paper, UPOCR uses the pretrained *ViTEraser* (with SegMIM pre-training) (Peng et al., 2024) for initialization during training. In Tab. 11, other two ways of weight initialization are investigated, including *ImageNet* and *SegMIM*. Specifically, for *ImageNet* manner, a Swin Transformer v2 (Liu et al., 2022b) pretrained on the ImageNet-1k (Deng et al., 2009) classification task is adopted to initialize the encoder, while for *SegMIM* manner, pretrained weights using SegMIM (Peng et al., 2024) are employed to initialize the encoder-decoder. It can be seen that the initialization is critical to the generalist model

Table 11. Ablation study on different weight initializations during training.

Initialization	Text Removal				Text Segmentation		Tampered Text Det.	
	PSNR↑	MSSIM↑	MSE↓	FID↓	fgIoU↑	F↑	mIoU↑	mF↑
ImageNet	35.67	<u>96.86</u>	0.3556	12.76	86.53	92.78	47.96	64.80
SegMIM	<u>37.04</u>	**97.62**	<u>0.0433</u>	<u>10.64</u>	<u>88.53</u>	<u>93.92</u>	**73.62**	**84.80**
ViTEraser	**37.14**	**97.62**	**0.0428**	**10.47**	**88.76**	**94.04**	<u>71.71</u>	<u>83.53</u>

Table 12. Ablation study on feature loss.

Feature Loss	Text Removal				Text Segmentation		Tampered Text Det.	
	PSNR↑	MSSIM↑	MSE↓	FID↓	fgIoU↑	F↑	mIoU↑	mF↑
×	36.99	**97.83**	0.0433	12.86	88.69	94.01	71.42	83.10
✓	**37.14**	97.62	**0.0428**	**10.47**	**88.76**	**94.04**	**71.71**	**83.53**

performance. The *SegMIM* and *ViTEraser* pretrained weights may effectively learn the essential capacity to grasp text shapes and textures as well as the distinctive features between texts and backgrounds. Therefore, the UPOCR can be rapidly adapted to various pixel-level OCR tasks with superior performance. Although the *SegMIM* is also an effective pre-training approach for UPOCR, we opt for the *ViTEraser* pretrained weights due to the better performance on two of the three tasks.

E. Ablation Study on Feature Loss

As specified in Sec. 3.3, the training strategy of UPOCR is unified as minimizing the distance between predicted and GT images at pixel and feature spaces despite the heterogeneity within various pixel-level OCR tasks. In Tab. 12, we further investigate the effect of feature loss on the versatile capacity of UPOCR. It can be seen that removing feature loss leads to a decline in the performance of all three tasks. Especially for the FID metric of text removal which measures the similarity between predicted and GT images, the performance drops a lot from 10.47 to 12.86. For the text removal task, the feature loss diminishes the disparity between predicted and GT images from a human-like perceptual perspective, thereby enhancing the verisimilitude of text-erased outputs. Moreover, the high-level semantic constraints acquired by feature alignment can facilitate the perception of text patterns and fine-grained textures, which empowers the model to precisely distinguish texts from backgrounds and capture inconspicuous differences between real and tampered texts.

F. Ablation Study on Task Prompt

The learnable task prompts are indispensable to the generalist capability of UPOCR. Specifically, they are injected into the ViT-based encoder-decoder to push the general representations extracted by the encoder towards task-specific regions, enabling the decoder to generate output images for the target task. In Tab. 13, we present the performance of UPOCR with and without task prompts. For the experiment without task prompts, we remove task prompts from UPOCR and re-train the model with the same experimental settings as described in Sec. A.4. The experimental results demonstrated that the UPOCR with prompts substantially outperforms the counterpart without prompts. Moreover, it is interesting that the performance of the model without prompts is not extremely low. However, this phenomenon does not indicate the model can actually be aware of the target task. Because the three datasets adopted in our experiments are not *i.i.d.*, the model can implicitly recognize which dataset the samples belong to and perform the corresponding task, which is known as the dataset bias problem (Liu & He, 2024). Nevertheless, the samples of Tampered-IC13 are sourced from general text spotting datasets and severely overlap with the other two datasets in terms of styles and scenarios; therefore, the model cannot accurately classify these samples into correct datasets and perform correct tasks, resulting in much lower performance. The visualizations in Fig. 8 also qualitatively showcase that the model exhibits severe confusion about the target task due to the absence of task prompts. Moreover, we cannot perform multiple tasks on one image as shown in Fig. 5 if without task prompts, because the output is deterministic for the same input in this case. In contrast, the proposed task prompts can prevent the model from implicit dataset classification and effectively guide the model to perform diverse tasks.

Table 13. Ablation study on task prompt.

Task Prompt	Text Removal				Text Segmentation		Tampered Text Det.	
	PSNR↑	MSSIM↑	MSE↓	FID↓	fgIoU↑	F↑	mIoU↑	mF↑
×	36.33	96.34	0.641	13.22	83.59	91.06	43.39	60.37
✓	**37.14**	**97.62**	**0.0428**	**10.47**	**88.76**	**94.04**	**71.71**	**83.53**

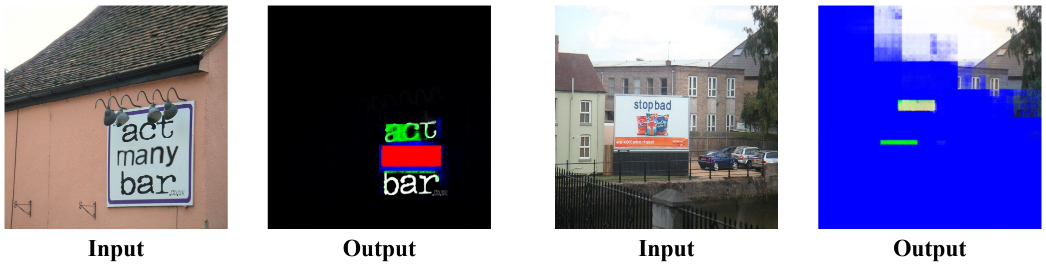


Figure 8. Visualization results of UPOCR on Tampered-IC13 without task prompts. The output images may comprise a mixture of predicted pixels for divergent tasks, indicating the model is unaware of the target task.

G. Speed

The inference speed of UPOCR is 17fps using an NVIDIA RTX 3090 GPU and 2.5fps using an Intel Xeon Platinum 8375C CPU. Both the speeds on GPU and CPU are tested directly with PyTorch implementation and a batch size of 1.

H. Comparison with ViTEraser

Although the encoder-decoder architecture of UPOCR is implemented following ViTEraser (Peng et al., 2024), the proposed UPOCR is a brand-new approach that significantly differs from ViTEraser in the following aspects.

- The UPOCR aims to build a generalist model for the unified pixel-level OCR interface. To achieve this, UPOCR unifies the paradigm, architecture, and training strategy of diverse pixel-level OCR tasks. Comprehensive experiments demonstrate the state-of-the-art performance of UPOCR in simultaneously handling text removal, text segmentation, and tampered text detection tasks with a single unified model. In contrast, ViTEraser specializes in the text removal task without generalist capacities. Moreover, it relies on dedicated modules including a discriminator and an auxiliary mask branch and is trained following a complicated GAN-based strategy. Although ViTEraser can be extended to the tampered text detection task, its paradigm and training strategy are adapted from image generation to segmentation and its parameters are re-trained using the tampered text detection dataset.

- To obtain the multi-task processing ability, UPOCR introduces learnable task prompts into ViT-based encoder-decoder architecture in a simple-yet-effective fashion. The task prompt pushes the general representation extracted by the encoder towards task-specific regions, empowering the decoder to generate output images for individual tasks. *Although we borrow the encoder-decoder architecture of ViTEraser to instantiate UPOCR, it is irrelevant to the novel part of our study.* Additionally, the U-shaped ViT-based architecture of ViTEraser itself is a general framework which has been broadly investigated arcoss multiple domains (Cao et al., 2022; Wang et al., 2022d).

- In this paper, extensive experiments and analyses are conducted to provide deep insights into the construction of unified pixel-level OCR interfaces, which we believe could spark more future research on generalist OCR models.

I. Limitation

Despite the effectiveness of the proposed UPOCR as a unified pixel-level OCR interface, the limitation lies in the generalization ability to unseen tasks. Although the learnable task prompts empower the UPOCR to simultaneously handle multiple tasks, they are fixed once the model training is finished, which means re-training is necessary if the pre-set range of tasks is expanded. Since task prompts function as feature-level offsets that push general representations to task-specific spaces, the

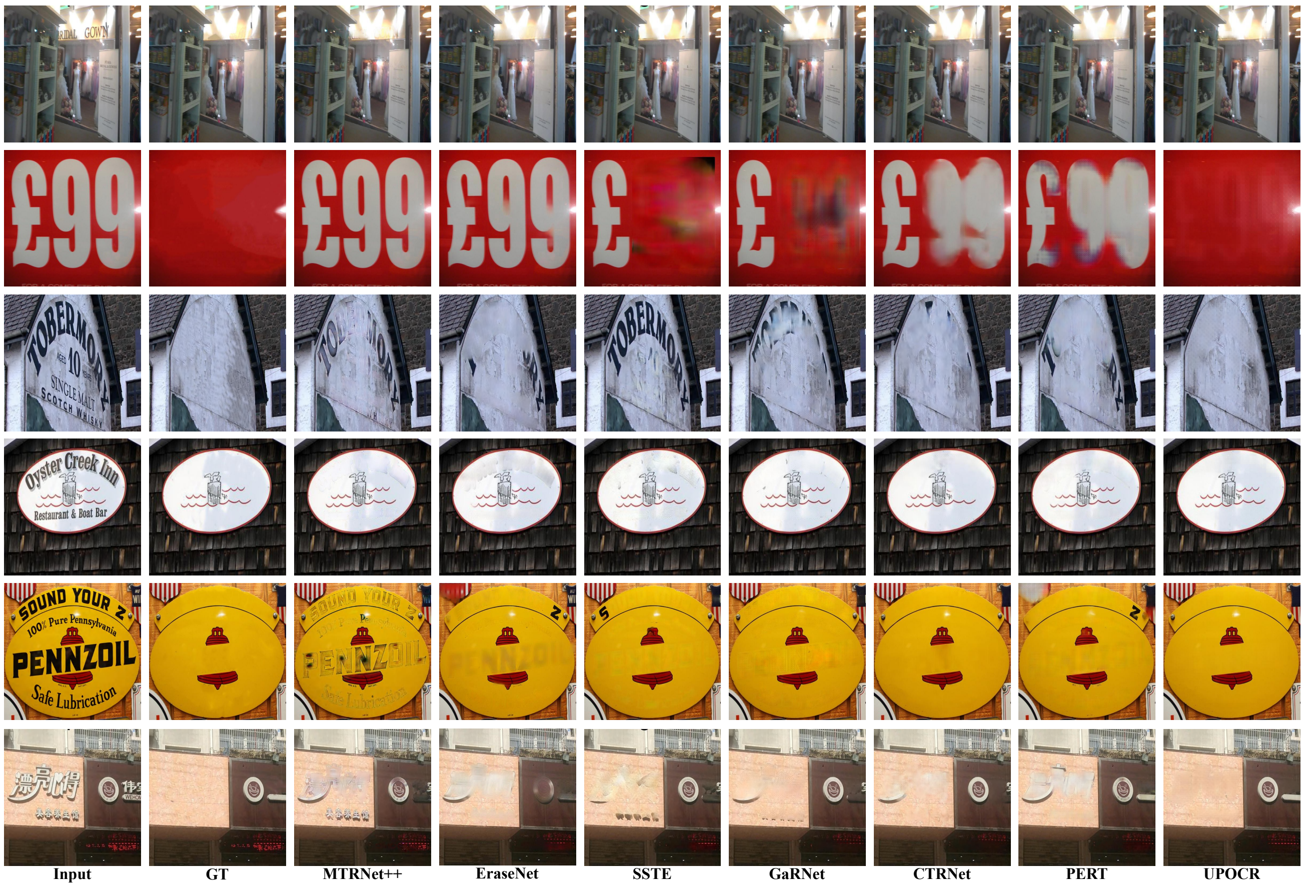


Figure 9. More visualizations on text removal, where the inference results are obtained by MTRNet++ (Tursun et al., 2020), EraseNet (Liu et al., 2020), SSTE (Tang et al., 2021), GaRNet (Lee & Choi, 2022), CTRNet (Liu et al., 2022a), PERT (Wang et al., 2023c), and UPOCR (ours). Zoom in for a better view.

automatic learning of prompts from example input-output pairs of new tasks is critical to a flexible generalist pixel-level OCR model, which is worth studying in future work.

J. More Visualization

In this section, we provide more qualitative results on text removal, text segmentation, and tampered text detection tasks in Figs. 9, 10, and 11, respectively. It can be seen that the proposed UPOCR simultaneously excels in multiple tasks using a unified single model without task- or benchmark-specific finetuning. Especially in the first row in Fig. 9, our method demonstrates superiority in tackling tiny and densely distributed text.

Moreover, in addition to Fig. 5 of the main paper, we supplement more visualizations of three-task outputs using the test set of Tampered-IC13 (Wang et al., 2022b) in Fig. 12. Specifically, all tasks are conducted with a single UPOCR model using corresponding task prompts. As demonstrated by the visualizations, UPOCR can effectively perform diverse tasks on the same input image, indicating the proposed learnable task prompts adequately guide the generation process of the decoder.

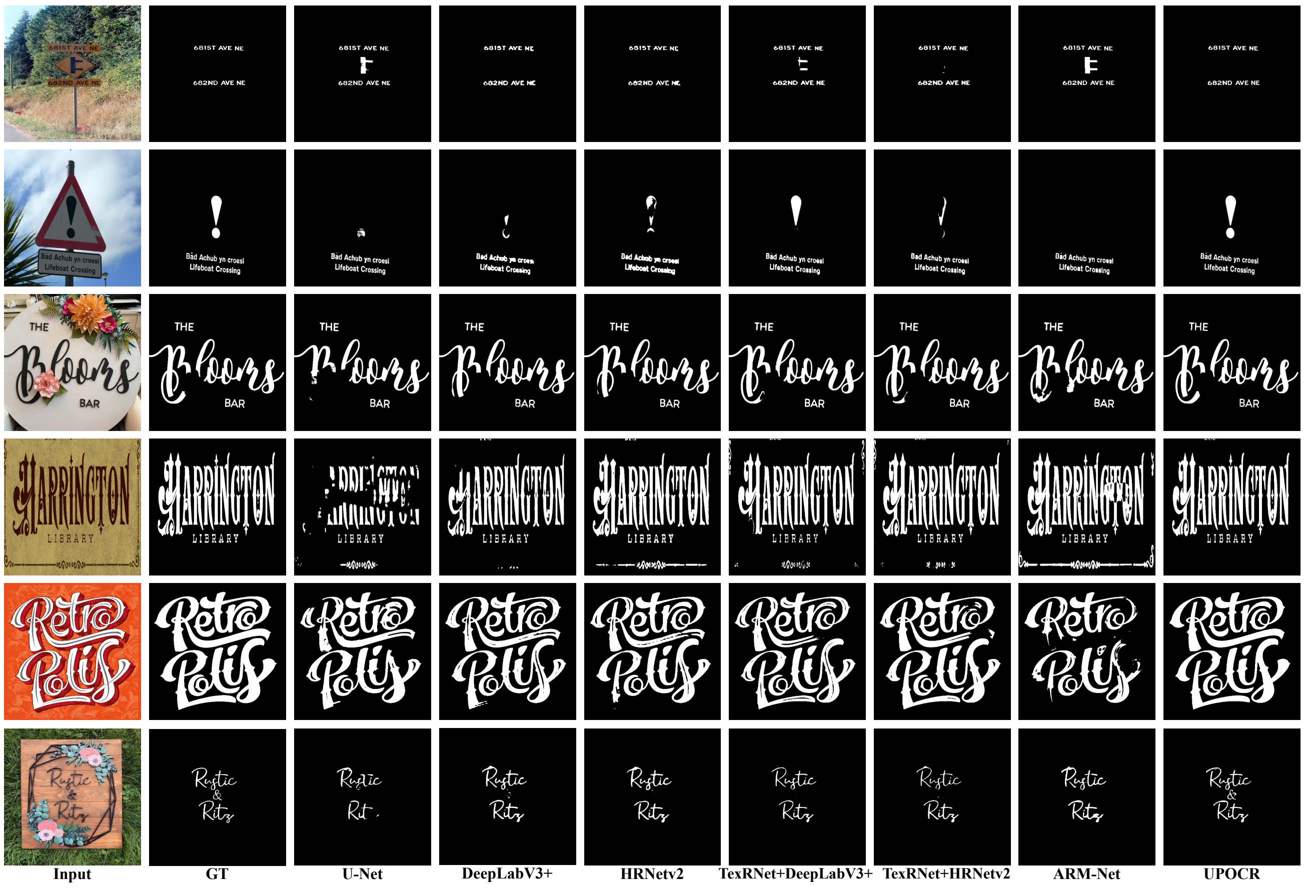


Figure 10. More visualizations on text segmentation, where the inference results are obtained by U-Net (Ronneberger et al., 2015), DeepLabV3+ (Chen et al., 2018), HRNetv2 (Wang et al., 2020a), TexRNet+DeepLabV3+ (Xu et al., 2021), TexRNet+HRNetv2 (Xu et al., 2021), ARM-Net (Ren et al., 2022), and UPOCR (ours). Zoom in for a better view.

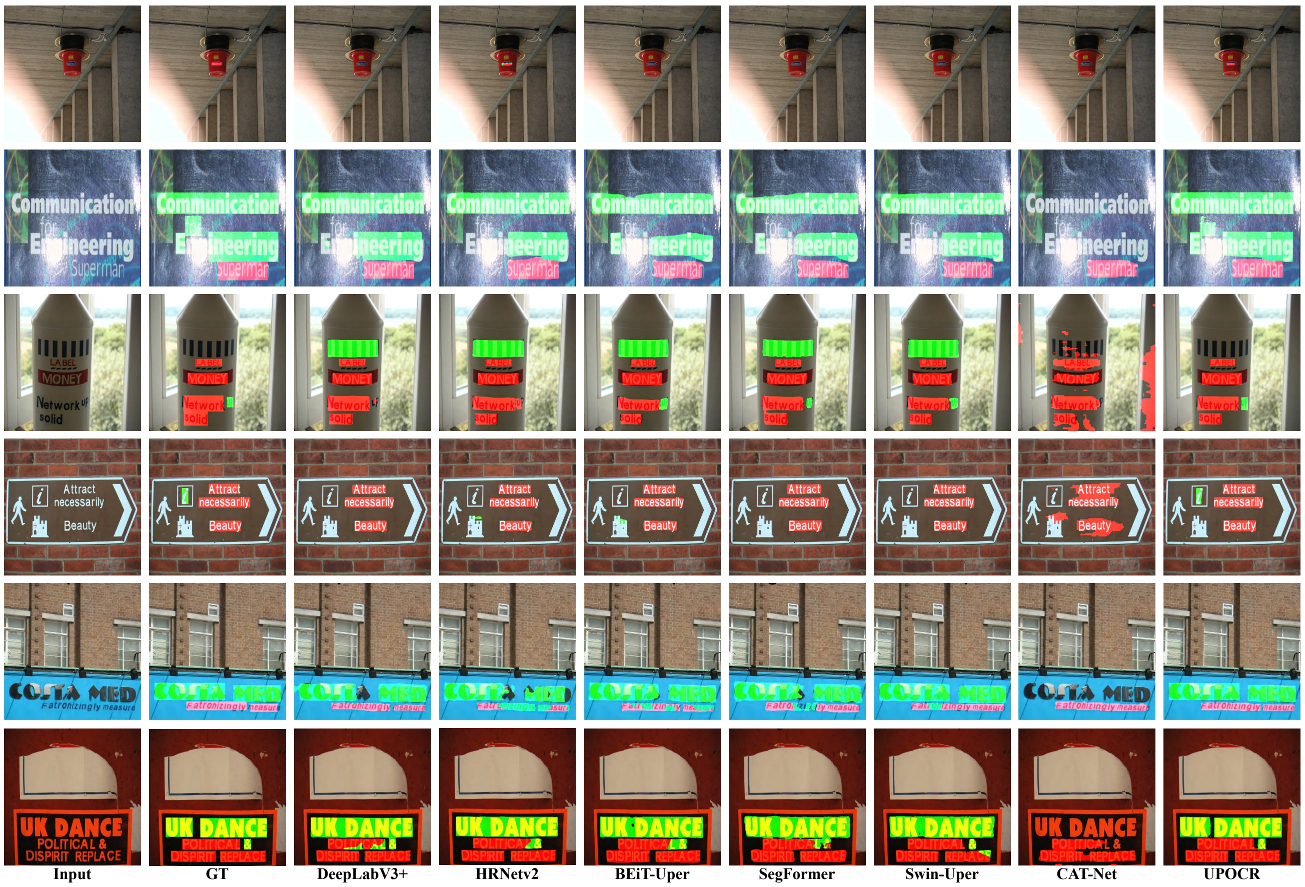


Figure 11. More visualizations on tampered text detection, where the inference results are obtained by DeepLabV3+ (Chen et al., 2018), HRNetv2 (Wang et al., 2020a), BEiT-Uper (Bao et al., 2022), SegFormer (Xie et al., 2021), Swin-Uper (Liu et al., 2021), CAT-Net (Kwon et al., 2022), and UPOCR (ours). Zoom in for a better view.

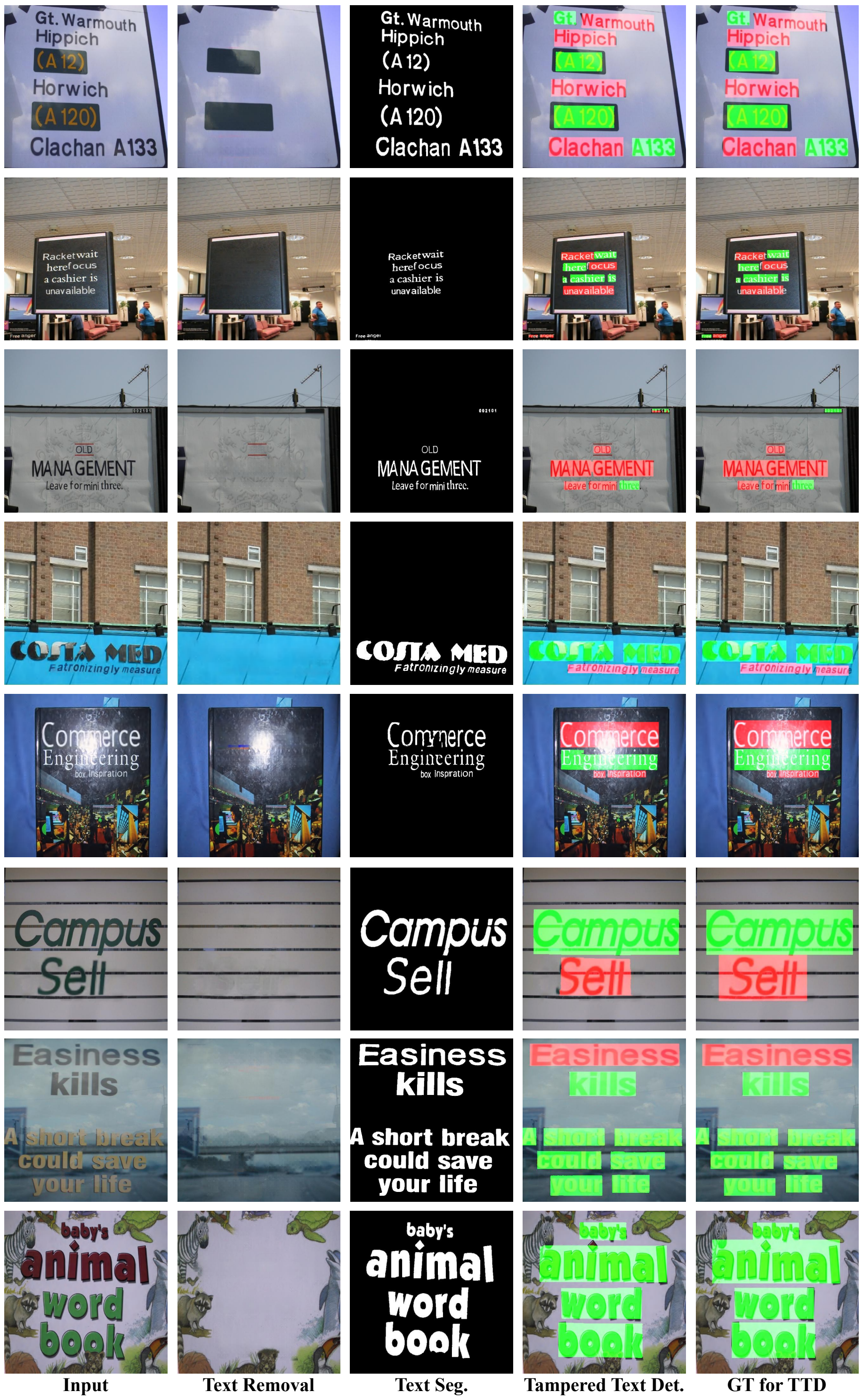


Figure 12. More visualizations of three-task outputs on Tampered-IC13 dataset. All tasks are performed with a single UPOCR model using corresponding task prompts. The GT of tampered text detection (TTD) provided by Tampered-IC13 is also presented for comparison. Zoom in for a better view. (Seg.: Segmentation, Det.: Detection)