TFIC: End-to-End Text-Focused Image Compression for Coding for Machines

Stefano Della Fiore* Alessandro Gnutti* Marco Dalai*
Pierangelo Migliorati* Riccardo Leonardi*†

stefano.dellafiore@unibs.it alessandro.gnutti@unibs.it marco.dalai@unibs.it
pierangelo.migliorati@unibs.it riccardo.leonardi@unibs.it

*Department of Information Engineering, University of Brescia, Italy
†Department of Electronics Engineering, University of Roma Tor Vergata, Italy

***Abstract*—Traditional image compression methods aim to faithfully reconstruct images for human perception. In contrast, *Coding for Machines* focuses on compressing images to preserve information relevant to a specific machine task. In this paper, we present an image compression system designed to retain text-specific features for subsequent Optical Character Recognition (OCR). Our encoding process requires half the time needed by the OCR module, making it especially suitable for devices with limited computational capacity. In scenarios where on-device OCR is computationally prohibitive, images are compressed and later processed to recover the text content. Experimental results demonstrate that our method achieves significant improvements in text extraction accuracy at low bitrates, even improving over the accuracy of OCR performed on uncompressed images, thus acting as a local pre-processing step.**

***Index Terms*—Image Compression, Coding for Machines, End-to-End Optimization.**

I. Introduction

For decades, image compression has been a crucial research area, driven by the increasing demand for efficient data storage and transmission across diverse digital applications. The advent of deep learning has transformed this field, driving significant interest in end-to-end learned compression frameworks, with variational autoencoder (VAE)-based methods playing a key role [1], [2]. Unlike traditional techniques, these approaches jointly optimize the encoding and decoding stages within a unified framework. Recent works [3], [4], [5], [6], [7], [8] have demonstrated that learned compression techniques can outperform VVC [9], the state-of-the-art standard for image and video coding, in terms of key quality metrics such as Peak Signal-to-Noise Ratio (PSNR) and Multi-Scale Structural Similarity (MS-SSIM). These findings underscore the transformative potential of learned approaches in shaping the future of image coding [10].

Although most learned image compression methods are tailored for human perception, the growing prominence of machine vision applications has driven a shift toward task-specific optimization. Recently, image coding for machine perception has gained significant attention, fueled by the increasing need to transmit visual data efficiently for high-level recognition tasks across devices. Referred to as Image Coding for Machines (ICM), this strategy aims to refine the compression process to generate images better suited for specific downstream tasks. Some examples of tasks in this domain include denoising [11], computer vision tasks [12], face detection in the compressed domain [13], and, more recently, Multimodal Large Language Models (MLLMs) [14].

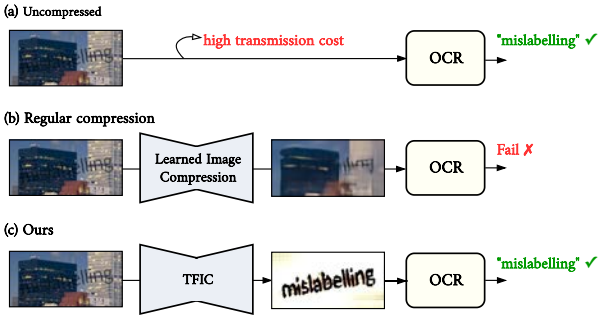


Fig. 1: Comparison of high-level frameworks: no compression, conventional compression, and our proposed TFIC.

Among these recognition applications, Optical Character Recognition (OCR) is a technology that enables the automatic extraction of text from images, scanned documents, and other visual data sources. OCR systems can convert printed or handwritten text into machine-readable formats. This capability is widely used across various applications, including document digitization, automated data entry, and assistive technologies for the visually impaired. Modern OCR systems often employ deep neural networks to enhance accuracy, enabling them to handle diverse fonts, handwriting styles, and challenging conditions such as poor lighting or distortions.

However, a major challenge arises when applying image compression to OCR-relevant images. Compression techniques, especially at low bitrates, can introduce significant distortions, such as blurring, blocking artifacts, and loss of fine details, which can severely impact text readability. Since OCR

This work was supported in part by the European Union under the Italian National Recovery and Resilience Plan (NRRP) of NextGenerationEU, Partnership on "Telecommunications of the Future," Program "RESTART" under Grant PE00000001, "Netwin" Project (CUP E83C22004640001) and "FRAME" Project (CUP C89J24000250004).

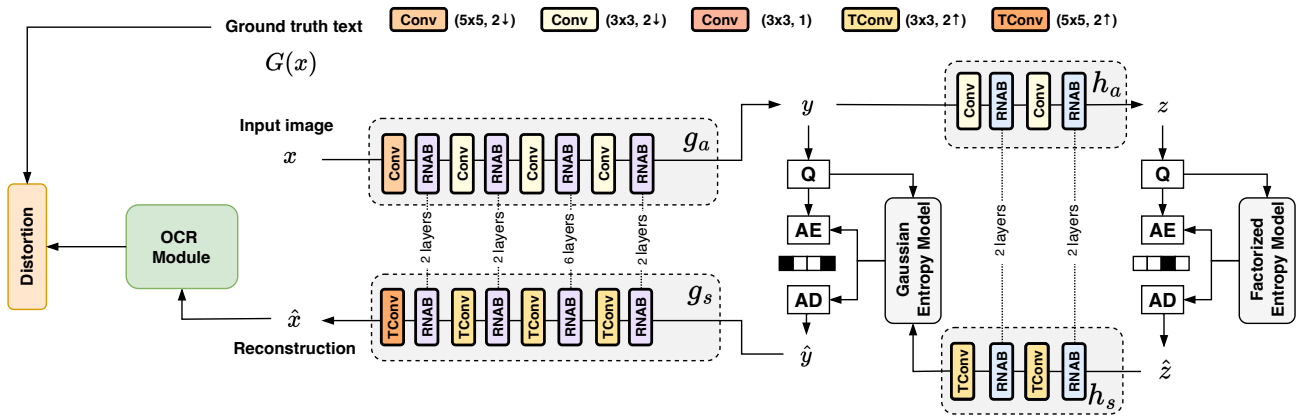


Fig. 2: High-level architectural framework of TFIC. The main image codec is composed by an encoder g_a that compresses the input image into a latent representation, which is quantized and transmitted. The decoder g_s reconstructs the image that is subsequently processed by the OCR module for text extraction.

performance heavily depends on the clarity and integrity of textual structures, excessive compression may lead to character misinterpretation or complete recognition failure.

In this paper, we introduce an image compression system designed specifically for OCR-based vision tasks. Our learnt Text-Focused Image Compression (TFIC) model is built on a standard architecture commonly used for neural image compression. However, we incorporate an OCR model at the system's output and optimize the compression process by minimizing the loss specific to OCR performance. This approach allows us to achieve a balance between efficient compression and high-quality text recognition, even at extremely lower compression rates where traditional methods strongly degrade the text to the point of being unreadable.

The main contributions of this paper are the following:

- We propose the first neural image compression system tailored specifically for OCR-based vision applications.
- Our results demonstrate the ability to compress images at extremely low rates while enabling the OCR system to accurately extract the text.
- The proposed approach offloads the entire computational complexity of the OCR system to the decoder side while preserving performance.

The remainder of the paper is organized as follows. Section II provides an overview of neural image compression and OCR systems. Section III outlines the architecture of our method and its end-to-end training approach. Section IV presents experimental results and comparisons with baseline methods. Finally, Section V concludes with a summary of our findings and potential directions for future research.

II. Background

A. Neural image compression

An end-to-end learned image compression system generally consists of two key components: the main autoencoder and the hyperprior autoencoder. The main autoencoder is composed of an analysis transform g_a and a synthesis transform g_s. The analysis transform g_a encodes an RGB image $x \in \mathbb{R}^{H \times W \times 3}$, with height H and width W, into a latent representation $y \in \mathbb{R}^{\frac{H}{16} \times \frac{W}{16} \times C_y}$ using an encoding distribution $q_{g_a}(y|x)$. This latent y is then uniformly quantized as $\hat{y}$ and entropy encoded into a bitstream via a learned prior distribution $p(\hat{y})$. On the decoder side, $\hat{y}$ is entropy decoded and reconstructed as $\hat{x} \in \mathbb{R}^{H \times W \times 3}$ using a decoding distribution $q_{g_s}(\hat{x}|\hat{y})$, implemented by the synthesis transform g_s. Throughout this process, the prior distribution $p(\hat{y})$ plays a critical role in determining the number of bits needed to represent the quantized latent $\hat{y}$. To mitigate this, a hyperprior autoencoder [2] is introduced to model the prior distribution in a content-adaptive manner. The hyperprior autoencoder consists of a hyperprior analysis transform h_a and a hyperprior synthesis transform h_s. The transform h_a converts the image latent y into side information $z \in \mathbb{R}^{\frac{H}{64} \times \frac{W}{64} \times C_z}$, which typically represents a small portion of the compressed bitstream. This quantized version of z is then decoded from the bitstream through h_s, resulting in the learned prior distribution $p(\hat{y})$.

B. OCR systems

An OCR system is designed to extract text from images captured in unconstrained environments. The system generally consists of four integrated modules. First, a *detection* module identifies and localizes regions containing text within an image, often using object detection frameworks to output bounding boxes around potential text areas [15], [16], [17]. Next, the *transformation* stage normalizes these detected regions to correct for distortions such as skew, rotation, or perspective changes. For instance, a Spatial Transformer Network (STN) with a Thin-Plate Spline (TPS) [18] transformation may be employed to obtain a rectified image, denoted as $\tilde{x}$.

Subsequently, a *feature extraction* component, typically implemented with a convolutional neural network (CNN) like ResNet [19], converts the (transformed) image x or $\tilde{x}$ into a

rich feature map $V = \{v_i\}_{i=1}^{I}$, where each v_i represents features corresponding to a specific region of the image and I is the dimension of the feature map. Given the sequential nature of text, these features are then reorganized into a sequence and processed by a *sequence modeling* module—commonly a bidirectional LSTM (BiLSTM) (see [20], [21], [22]) or Transformer—that captures contextual dependencies between characters, yielding a contextualized representation. Finally, the *prediction* stage decodes this representation into the final text output using methods such as Connectionist Temporal Classification (CTC) [23] or attention-based decoding [21], [22]. The entire pipeline is trained end-to-end by minimizing a loss function that measures the discrepancy between the predicted sequence and the ground truth, ensuring robust performance in complex and real-world scenarios.

III. Proposed Text-Focused Image Compression

A. Architecture Overview

Figure 2 illustrates our proposed framework. The image codec architecture follows the key modules outlined in II-A. As a reference, we adopt the image compression model from [24], which utilizes Transformer-based backbones to implement both the main and hyperprior autoencoders.

An OCR system with frozen parameters is integrated downstream of the decoder alongside the autoencoder. Its architecture follows the key modules outlined in II-B, with the Scene Text Recognition model from [25] serving as a reference. During training, the OCR extracts text from $\hat{x}$, denoted as $T(\hat{x})$, which is compared to the ground truth text $G(x)$. The resulting loss signal is then backpropagated through the entire network. This process helps guide both the encoder and decoder to preserve text-relevant information.

B. Loss Function and End-to-End Training

The overall training objective $\mathcal{L}_{\text{total}}$ of our method is a combination of three loss components:

$$\mathcal{L}_{\text{total}} = \lambda \cdot \mathcal{L}_{\text{dist}}(x, \hat{x}) + \mathcal{L}_{\text{rate}}(\hat{y}, \hat{z}) + \gamma \cdot \mathcal{L}_{\text{OCR}}(G(x), T(\hat{x})),$$

where:

- $\mathcal{L}_{\text{dist}}(x, \hat{x})$ is the distortion loss, measured by the mean squared error (MSE) between the original image x and its reconstruction $\hat{x}$.
- $\mathcal{L}_{\text{rate}}(\hat{y}, \hat{z}) = -\log p(\hat{z}) - \log p(\hat{y}|\hat{z})$ represents the rate loss, which estimates the bitstream length needed to encode the quantized latent representation $\hat{y}$ and side information $\hat{z}$.
- $\mathcal{L}_{\text{OCR}}(G(x), T(\hat{x}))$ is the OCR loss that quantifies the difference between the ground truth text $G(x)$ and the text recognized from the reconstructed image $T(\hat{x})$. This is measured using the cross-entropy loss between the text-indexes of the ground truth text and the reconstructed one.

The parameters λ and γ balance the trade-offs between pixel-level fidelity, compression rate, and text recognition accuracy. Our training procedure consists of two stages. In the first stage, we set $\gamma = 0$, allowing the model to be pre-trained solely with the MSE for a given target rate, which is defined by the value of λ. In the second stage, we set $\lambda = 0$, enabling the model to minimize only the OCR loss. During this stage, γ is set to 0.1 for all rate points.

Training is conducted end-to-end, enabling the encoder, decoder, and OCR module to co-adapt such that the final compressed representation retains essential textual information while achieving high compression efficiency. We recall that only the image codec is re-trained, while the OCR system remains unchanged.

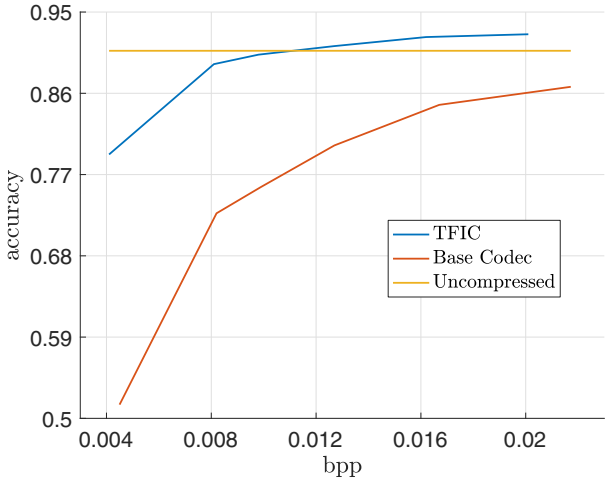


Fig. 3: Rate vs. OCR performance curves for images that are uncompressed, decoded with the pre-trained MSE-based codec, and decoded using the proposed TFIC.

IV. Experiments

A. Dataset and Experimental Setup

We use a synthetic data generator for text recognition [26] to create training and test sets of approximately 20k and 600 images, respectively. Each image is resized to a fixed resolution of 256×512 pixels. The dataset covers a diverse range of fonts, layouts, and background complexities, ensuring a comprehensive evaluation of text extraction under various conditions.

For evaluation, we compare the proposed method with the base codec optimized exclusively for MSE on the same training set, thus without including the OCR loss component. Both models are evaluated based on bitrate (bits-per-pixel, bpp) and OCR performance, which is measured by the Levenshtein edit-distance $\text{lev}(G(x), T(\hat{x}))$ (see [27]) that represents the minimum number of single-character edits (insertions, deletions or substitutions) required to change the reconstructed text $T(\hat{x})$ into the ground truth one $G(x)$. The accuracy is then defined as

$$1 - \frac{\text{lev}(G(x), T(\hat{x}))}{\max\{|G(x)|, |T(\hat{x})|\}},$$

where $|G(x)|$ and $|T(\hat{x})|$ are the number of alphanumeric characters contained in the ground truth text and in the reconstructed one, respectively.

(a) Original Image
Bpp: 0.36 - Accuracy: 48%

(b) Reconstructed with the base codec
Bpp: 0.016 - Accuracy: 19%

(c) Reconstructed with the base codec
Bpp: 0.0082 - Accuracy: 3%

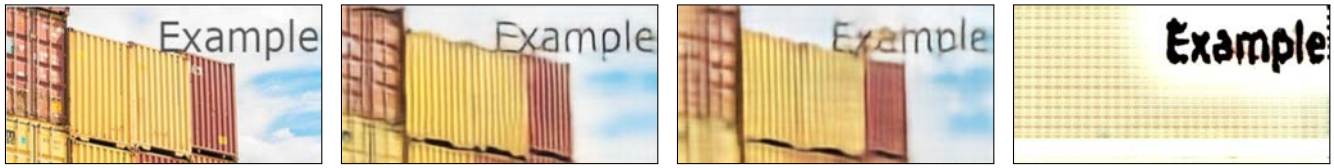
(d) Reconstructed with TFIC
Bpp: 0.0080 - Accuracy: 100%

(e) Original Image
Bpp: 0.45 - Accuracy: 70%

(f) Reconstructed with the base codec
Bpp: 0.016 - Accuracy: 42%

(g) Reconstructed with the base codec
Bpp: 0.0082 - Accuracy: 2%

(h) Reconstructed with TFIC
Bpp: 0.0080 - Accuracy: 100%



(i) Original Image
Bpp: 0.37 - Accuracy: 100%

(j) Reconstructed with the base codec
Bpp: 0.016 - Accuracy: 100%

(k) Reconstructed with the base codec
Bpp: 0.0082 - Accuracy: 80%

(l) Reconstructed with TFIC
Bpp: 0.0080 - Accuracy: 100%

Fig. 4: Visual comparison of reconstructed images. Although the base codec preserves more global details, our method retains critical textual regions that yield superior OCR performance.

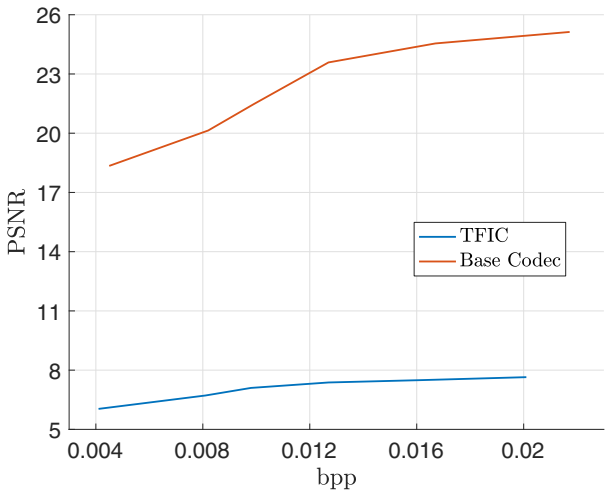


Fig. 5: Rate vs. PSNR performance curves for the base codec and the proposed TFIC.

B. Results

Figure 3 shows the rate vs. OCR performance curves for both the base codec and the proposed TFIC model. The base codec exhibits a significant drop in OCR accuracy, with lower accuracy values at comparable bitrates. In contrast, our method consistently achieves higher accuracy, effectively preserving text information even at lower bitrates. Notably, TFIC even surpasses the OCR performance on uncompressed images, suggesting that it acts as a beneficial pre-processing step.

	Encoding	OCR module
Computational times	12.9 ± 1.8	24.1 ± 3.3

TABLE I: Average and standard deviation of encoding and OCR times per image, measured in milliseconds.

This observation is supported by Figure 4, which provides a visual comparison of reconstructed images from the base codec and TFIC. While the base codec achieves higher overall PSNR (see Figure 5), text regions often appear blurred or distorted, resulting in poor OCR accuracy. In contrast, our method preserves sharp and legible text, even if non-essential areas of the image undergo destructive compression.

A comparison of the results shown in the second, third and fourth columns of Figure 4, shows that our method can also be useful when used *in addition* to a standard compression scheme, whenever transmission of a low rate visual content of the background is also required. Indeed, the total rate for sending *both* images in the third and fourth columns (as two separate layers) equals the rate for the image in the second column. The computation time would in this case be doubled, but would still remain lower than the time needed for the OCR on the original image alone, while also doing compression.

C. Runtime Analysis

One motivation behind our method is a difference in processing time between the encoding and OCR stages. Empirical measurements indicate that the encoding process of our

method requires only about one half of the time needed to perform OCR on the same image. This efficiency is crucial for resource-limited devices, enabling rapid on-device compression with deferred, more intensive OCR processing on external hardware. Table I shows the average and standard deviation of encoding and OCR times per image computed over the entire test set, revealing that the encoding stage takes roughly half the time required by the OCR stage.

V. Discussion and Conclusion

In this paper, we proposed TFIC, an end-to-end text-focused image compression system designed for image coding for machines applications. By integrating an OCR module into the compression training pipeline and optimizing the autoencoder with an OCR-specific loss function, our approach targets improved performance for text-related tasks. This design prioritizes the preservation of textual information over complete visual fidelity. Experimental results demonstrate that our method effectively balances the trade-off between compression efficiency and OCR accuracy. While conventional reconstruction-based methods optimize for overall image fidelity, our approach leverages an OCR-specific loss to guide the image codec in preserving critical text features.

The reduction in encoding time further validates our approach, making it particularly well-suited for devices with limited computational resources. Indeed, the encoding time constitutes only one half of the overall OCR processing time; hence, when on-device OCR is not feasible, images can be compressed and later decompressed to recover text with high accuracy.

Nevertheless, our method has certain limitations. Its performance depends on the specific OCR module used, and future improvements in OCR technology may necessitate adaptations in our compression strategy. Additionally, the balance between the distortion, rate, and OCR losses, which is controlled by the hyperparameters λ and γ, requires careful tuning to suit different applications and datasets. Future work could focus on integrating more advanced OCR architectures and extending our framework to additional task-specific scenarios.

References

[1] Johannes Ballé, Valero Laparra, and Eero P Simoncelli, "End-to-end optimized image compression," *arXiv preprint arXiv:1611.01704*, 2016.

[2] Johannes Ballé, David Minnen, Saurabh Singh, Sung Jin Hwang, and Nick Johnston, "Variational image compression with a scale hyperprior," *arXiv preprint arXiv:1802.01436*, 2018.

[3] Renjie Zou, Chunfeng Song, and Zhaoxiang Zhang, "The devil is in the details: Window-based attention for image compression," *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022.

[4] Yinhao Zhu, Yang Yang, and Taco Cohen, "Transformer-based transform coding," in *International Conference on Learning Representations*, 2022.

[5] Fangdong Chen, Yumeng Xu, and Li Wang, "Two-stage octave residual network for end-to-end image compression," in *Proceedings of AAAI conference on artificial intelligence*, 2022.

[6] Dailan He, Ziming Yang, Weikun Peng, Rui Ma, Hongwei Qin, and Yan Wang, "Elic: Efficient learned image compression with unevenly grouped space-channel contextual adaptive coding," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[7] Dezhao Wang, Wenhan Yang, Yueyu Hu, and Jiaying Liu, "Neural data-dependent transform for learned image compression," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 17379–17388.

[8] Jinming Liu, Heming Sun, and Jiro Katto, "Learned image compression with mixed transformer-cnn architectures," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023, pp. 14388–14397.

[9] Joint Video Experts Team, "Vvc official test model vtm.," 2021.

[10] Elena Alshina, João Ascenso, and Touradj Ebrahimi, "Jpeg ai: The first international standard for image coding based on an end-to-end learning-based approach," *IEEE MultiMedia*, vol. 31, no. 4, pp. 60–69, 2024.

[11] Yi-Hsin Chen, Kuan-Wei Ho, Shiau-Rung Tsai, Guan-Hsun Lin, Alessandro Gnutti, Wen-Hsiao Peng, and Riccardo Leonardi, "Transformer-based learned image compression for joint decoding and denoising," in *2024 Picture Coding Symposium (PCS)*, 2024, pp. 1–5.

[12] Yi-Hsin Chen, Ying-Chieh Weng, Chia-Hao Kao, Cheng Chien, Wei-Chen Chiu, and Wen-Hsiao Peng, "Transtic: Transferring transformer-based image compression from human perception to machine perception," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2023, pp. 23297–23307.

[13] Ayman Alkhateeb, Alessandro Gnutti, Fabrizio Guerrini, Riccardo Leonardi, João Ascenso, and Fernando Pereira, "Jpeg ai compressed domain face detection," in *2024 IEEE 26th International Workshop on Multimedia Signal Processing (MMSP)*, 2024, pp. 1–6.

[14] Chia-Hao Kao, Cheng Chien, Yu-Jen Tseng, Yi-Hsin Chen, Alessandro Gnutti, Shao-Yuan Lo, Wen-Hsiao Peng, and Riccardo Leonardi, "Bridging compressed image latents and multimodal large language models," 2025.

[15] Youngmin Baek, Bado Lee, Dongyoon Han, Sangdoo Yun, and Hwalsuk Lee, "Character region awareness for text detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 9365–9374.

[16] Pan He, Weilin Huang, Tong He, Qile Zhu, Yu Qiao, and Xiaolin Li, "Single shot text detector with regional attention," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 3047–3055.

[17] Xinyu Zhou, Cong Yao, He Wen, Yuzhi Wang, Shuchang Zhou, Weiran He, and Jiajun Liang, "East: an efficient and accurate scene text detector," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2017, pp. 5551–5560.

[18] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al., "Spatial transformer networks," in *NIPS*, 2015, pp. 2017–2025.

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778.

[20] Baoguang Shi, Xiang Bai, and Cong Yao, "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition," in *TPAMI*. 2017, vol. 39, pp. 2298–2304, IEEE.

[21] Baoguang Shi, Xinggang Wang, Pengyuan Lyu, Cong Yao, and Xiang Bai, "Robust scene text recognition with automatic rectification," in *CVPR*, 2016, pp. 4168–4176.

[22] Zhanzhan Cheng, Fan Bai, Yunlu Xu, Gang Zheng, Shiliang Pu, and Shuigeng Zhou, "Focusing attention: Towards accurate text recognition in natural images," in *ICCV*, 2017, pp. 5086–5094.

[23] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *ICML*, 2006, pp. 369–376.

[24] Ming Lu, Fangdong Chen, Shiliang Pu, and Zhan Ma, "High-efficiency lossy image coding through adaptive neighborhood information aggregation," *arXiv preprint arXiv:2204.11448*, 2022.

[25] Jeonghun Baek, Geewook Kim, Junyeop Lee, Sungrae Park, Dongyoon Han, Sangdoo Yun, Seong Joon Oh, and Hwalsuk Lee, "What is wrong with scene text recognition model comparisons? dataset and model analysis," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 4715–4723.

[26] *https://github.com/Belval/TextRecognitionDataGenerator*.

[27] Vladimir I Levenshtein et al., "Binary codes capable of correcting deletions, insertions, and reversals," in *Soviet physics doklady*. Soviet Union, 1966, vol. 10, pp. 707–710.