

TokenOCR: An Attention Based Foundational Model for Intelligent Optical Character Recognition

Charith Gunasekara^a, Zachary Hamel, Feng Du and Connor Baillie
Department of National Defence, Government of Canada, Ottawa, ON, Canada

Keywords: Natural Language Processing, Optical Character Recognition, Transformer Architecture, Curriculum Learning.

Abstract: Optical Character Recognition (OCR) plays a pivotal role in digitizing and analyzing text from physical documents. Despite advancements in OCR technologies, challenges persist in handling diverse text layouts, poor-quality images, and complex fonts. In this paper, we present TokenOCR, an attention-based foundational model designed to overcome these limitations by integrating convolutional neural networks and transformer-based architectures. Unlike traditional OCR models that predict individual characters, TokenOCR predicts tokens, significantly enhancing recognition accuracy and efficiency. The model employs a ResNet50 feature extractor, an encoder with adaptive 2D positional embeddings, and a decoder utilizing multi-headed attention mechanisms for robust text recognition. To train TokenOCR, we used a dataset of six million images incorporating various real-world applications. Our training strategy integrates curriculum learning and adaptive learning rate scheduling to ensure efficient model convergence and generalization. Comprehensive evaluations using Word Error Rate (WER) and Character Error Rate (CER) demonstrate that TokenOCR consistently outperforms state-of-the-art models, including Tesseract and TrOCR, in both clean and degraded image conditions. These findings underscore TokenOCR's potential to set new standards in OCR technology, offering a scalable, efficient, and highly accurate solution for diverse text recognition applications.

1 INTRODUCTION

Optical Character Recognition (OCR) is a fundamental technology used for digitizing text from physical documents, including printed, handwritten, or scanned text. The primary goal of OCR systems is to convert images of text into machine-encoded text, facilitating tasks such as document indexing, data entry, and digital archiving. Traditional OCR systems typically consist of two main components: text detection and text recognition. Text detection identifies the areas of the image containing text, while text recognition converts these areas into editable text.

The most well-known OCR model, Tesseract (Smith, 2007), developed by HP and later released as open-source by Google, has become one of the most prominent frameworks in this field. Tesseract uses a combination of image processing and machine learning techniques to convert text images into text outputs. It operates in multiple stages: image binarization, line finding, word recognition, and finally, character recognition using adaptive classifi-

cation. Pytesseract, a Python wrapper for Google's Tesseract-OCR Engine, provides a simple interface for integrating Tesseract's powerful OCR capabilities into Python applications. Pytesseract allows users to extract text from images and scanned documents with ease, making it a popular choice for developers working on OCR-related tasks. Despite its capabilities, Pytesseract has limitations. Its performance is heavily dependent on the quality of input images and pre-processing steps. It also struggles with handwritten text, complex fonts, and images with significant noise or distortions. Additionally, Pytesseract's reliance on traditional machine learning techniques means it may not be as adaptable to diverse text styles as modern deep learning approaches.

Alongside Pytesseract, there are many popular OCR models. EasyOCR (Liao et al., 2022), a widely used Python library, leverages deep learning for OCR and supports over 80 languages. It is known for its simplicity and accuracy, making it highly accessible. However, it can be resource-intensive and may require fine-tuning for optimal performance on specific document types. Google Cloud Vision OCR (Cloud,

^a  <https://orcid.org/0000-0002-7213-883X>

2024) offers a cloud-based solution with high accuracy and support for multiple languages. It also provides additional features such as image classification and face detection. While it delivers excellent results, the requirement for an internet connection and potential data privacy concerns with cloud-based processing can be drawbacks. Amazon Textract (Services, 2024), another cloud-based OCR service provided by AWS, is designed to extract text and data from scanned documents. It excels at handling complex documents with tables and forms and integrates well with other AWS services. However, similar to Google Cloud Vision OCR, it requires an internet connection and may incur costs, with data privacy being a potential concern. PyTorch-based libraries offer high customization and flexibility for OCR models, such as the deep-text-recognition-benchmark (Baek et al., 2019a), a comprehensive benchmark for text recognition tasks. This benchmark includes several state-of-the-art models and provides pre-trained weights, enabling developers to experiment with different architectures. Character Region Awareness for Text detection (CRAFT) (Baek et al., 2019b) is another OCR framework that detects individual characters and links them into words, providing robust text detection in various environments.

Despite the success of these traditional OCR models, several limitations persist. These models often struggle with complex document layouts, poor-quality images, and diverse fonts and languages. The reliance on CNNs and RNNs increases computational demands and limits scalability. Moreover, adapting to various text styles and noise levels remains a challenge (Raj and Kos, 2022). These gaps highlight the need for a more robust and flexible OCR solution.

Recent advancements in Transformer architectures (Vaswani et al., 2023) have shown promise in overcoming these challenges. Transformers, originally designed for natural language processing (NLP) tasks, have demonstrated impressive performance in image processing. These models excel at handling sequential data and can capture long-range dependencies within text, making them highly effective for OCR tasks. The attention mechanism in transformers allows the model to focus on different parts of the input image, enabling better recognition of complex text patterns. The introduction of Vision Transformers (Dosovitskiy et al., 2021) and their variants has paved the way for applying transformer-based architectures to OCR tasks. Transformer-based Optical Character Recognition (TrOCR) (Li et al., 2022) with Pre-trained Models introduces an innovative approach to OCR that leverages the Transformer architecture for both image understanding and text generation.

Despite the advancements brought by TrOCR in leveraging transformer architectures for OCR tasks, several challenges remain in handling variability in text appearances, poor-quality images, and diverse fonts. The expensive hardware requirements for pre-training fine-tuning, along with the need for millions of images for pre-training, present significant hurdles.

In this paper we present TokenOCR foundational model to improve OCR performance in document layouts where current models often struggle while reducing the model weight for faster training and inferring tasks. Despite advancements made by models like TrOCR, significant challenges remain in accurately recognizing text within complex layouts, dealing with varying image qualities, and accommodating diverse font styles. These limitations are critical in scenarios where precision and reliability are essential. TokenOCR aims to bridge these gaps by developing a model that excels in understanding contextual and spatial relationships within the text; it improves the handling of varying text layouts and enhances the model's ability to generalize across poor-quality input document types compared to the current alternatives.

2 MODEL ARCHITECTURE

TokenOCR combines convolutional neural networks and transformer-based architectures. This design enhances the model's ability to capture and process both visual and textual information efficiently, ensuring accurate text recognition even in complex scenarios.

Figure 1 illustrates the overall architecture of the TokenOCR model. Image input is processed through a ResNet (Pascanu et al., 2013) feature extractor, which transforms the image into a feature map by capturing the visual features. This feature map is then fed into the encoder, where it undergoes vectorization and positional encoding to retain spatial information. The encoder utilizes self-attention (Vaswani et al., 2023) mechanisms to focus on different parts of the image, global attention (Liu et al., 2021) to capture contextual information, and cross attention (Gheini et al., 2021) to align image sections with text sections. Additionally, the encoder incorporates 2D learnable positions (Yu et al., 2024) to encode spatial relationships and text position attention (Shaw et al., 2018) to emphasize specific text areas within the image. The output of the encoder is an encoded representation that combines visual and positional data. Next, the encoded representation is segmented into tokenized representations using a SentencePiece tokenizer, preparing the data for text generation. The tokenized data is then processed by the decoder, which employs self-

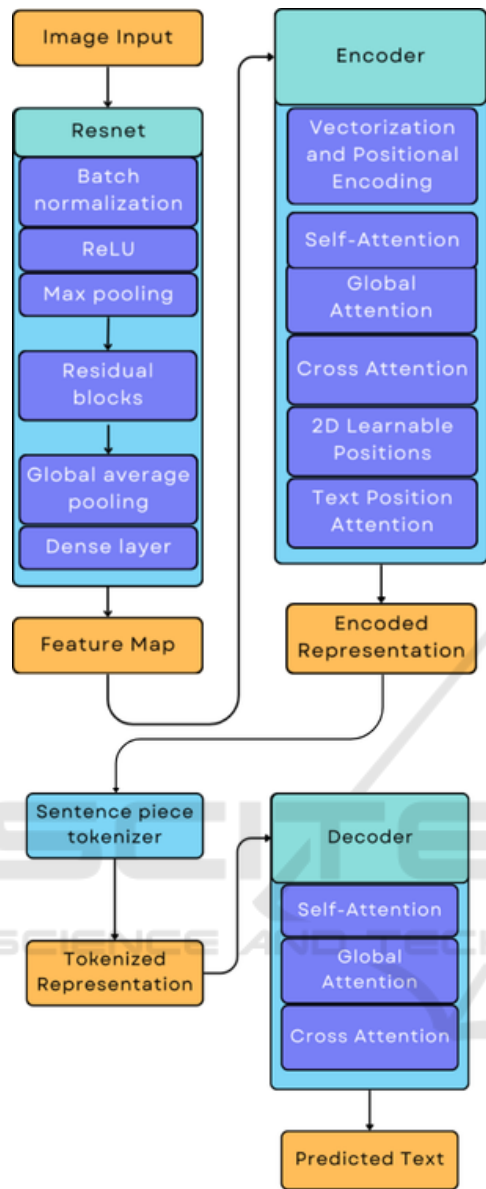


Figure 1: Overall Architecture of TokenOCR.

attention to concentrate on various parts of the tokenized sequence, global attention to maintain context, and cross-attention to align the encoded image representation with the text tokens accurately. Finally, the decoder generates the predicted text from the tokenized input, completing the OCR process.

2.1 ResNet

TokenOCR is built upon two foundational base models: ResNet50 and the EfficientNet family. These base models are renowned for their effectiveness in feature extraction tasks, particularly in image recognition.

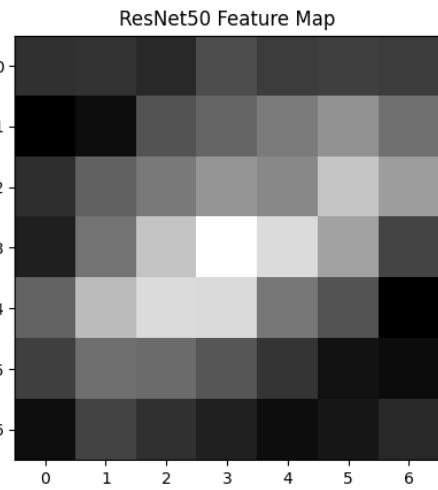


Figure 2: An example of a feature map generated by the sentence “Pariguana (meaning “near Iguana” in Greek) is an extinct genus of iguanid lizard from the Late Cretaceous of western North America” shown in the first row of figure 3.

ResNet50 introduced residual connections, known as skip connections, that facilitate training deep neural networks by mitigating the vanishing gradient problem, where the gradients of the loss function with respect to the weights in earlier layers become extremely small and insignificant for the learning process as training progresses. Leveraging ResNet50, TokenOCR capitalizes on its feature extraction capabilities across various computational constraints. Within the ResNet layer shown in the ResNet block of Figure 1, an input image is first processed by an initial convolutional block containing batch normalization, a ReLU activation function, and max pooling to capture the basic features. This is followed by the core of ResNet, which consists of multiple residual blocks. Each residual block applies a series of transformations on the feature map by adding new features in succession. Lastly, a series of operations involving global average pooling and a fully connected dense layer produces a final feature map as an output (Figure 2).

2.2 Encoder

The initial stage of the encoder takes the feature map and divides it into a set of fixed-sized patches. Each patch is subsequently flattened into a one-dimensional vector. The vectors then undergo a linear transformation, projecting the flattened patches onto a higher-dimensional embedding space. These resultant patch embeddings serve as the direct inputs to the encoder. Due to the lack of inherent sequence processing within the transformer architecture, positional en-

codings are computed and added to each patch embedding prior to their input into the encoder. The position encoding provides the transformer with information about the patch embedding location within the original image.

The encoder component (Encoder block in Figure 1) consists of two layers: a multi-headed attention layer and a feed-forward layer (Vaswani et al., 2023). The multi-headed attention layer incorporates various attention mechanisms, including self-attention, cross-attention, global attention, and text position attention. Additionally, the encoder utilizes adaptive 2D positional embeddings, which are learnable embeddings that encode the spatial arrangement of text elements within the image. Self-attention allows the model to weigh the importance of each word/token relative to others, capturing dependencies within the sequence irrespective of distance. Cross-attention enables the model to attend to information from one sequence (image features) while processing another sequence (text). Global attention provides the model with a broader contextual understanding of the entire input, aiding in text extraction. Text position attention specifically focuses on the positional information of text elements within the image, attending to the spatial arrangement of text. Lastly, adaptive 2D positional embeddings are updated during training, allowing the model to dynamically learn spatial relationships between text elements. These embeddings enhance the model's ability to understand the spatial layout of text within the image, facilitating accurate text extraction. The feed-forward layer processes each position in the sequence individually. It first expands the representation of each position to a higher-dimensional space through a linear transformation, applies a ReLU activation function, and then reduces the dimensionality back to the original size through another linear transformation, producing the encoded representation of the original input.

2.3 Tokenizer

The encoded representation from the encoder is then fed into a tokenizer to produce a tokenized representation for the decoder. TokenOCR employs SentencePiece (Kudo and Richardson, 2018) tokenization to segment the text into variable-length subword units based on their frequency of occurrence. The choice of tokenization method significantly impacts the model's performance and the granularity of information it can capture. For instance, character-level tokenization preserves individual characters' information but may struggle with out-of-vocabulary words, whereas byte-pair encoding (BPE) (Sennrich et al., 2016) tokeniza-

tion can handle a larger vocabulary size and capture morphological information more effectively. BPE, chosen for TokenOCR, balances between character-level and word-level tokenization, providing a good compromise for many NLP tasks. For TokenOCR, a vocabulary size of 10,000 was chosen. The tokenizer was trained on approximately 30GB of text data from the No Language Left Behind (NLLB) en-fr corpus (Team et al., 2022), ensuring comprehensive coverage of language patterns and structures. Additionally, special tokens were introduced to the tokenized sequences: padding tokens (set to 0) to standardize sequence lengths, start-of-sequence tokens (set to 1) to mark the beginning of each sequence, end-of-sequence tokens (set to 2) to indicate the end of each sequence, and unknown tokens (set to 3) to represent out-of-vocabulary words. These specifications facilitate effective tokenization and model training while maintaining compatibility with the Transformer architecture.

2.4 Decoder

The decoder component (Decoder block in Figure 1) consists of a multi-headed attention layer and a feed-forward layer. Similar to the encoder, these attention mechanisms aid in generating accurate textual outputs based on the encoded image representations. The decoder also employs sinusoidal embeddings, which provide a continuous representation of textual tokens, enhancing the model's understanding of textual information. The decoder uses the same attention mechanisms as the encoder: self-attention, cross-attention, and global attention. These mechanisms enable the decoder to attend to relevant information within the encoded image representations while generating textual outputs. Self-attention allows the decoder to focus on relevant parts of the generated sequence, while cross-attention aligns the encoded image features with the generated text. Sinusoidal embeddings, inspired by positional encodings in transformers, encode both position and frequency information of tokens, enabling the model to capture sequential relationships effectively. The decoder starts with a start-of-sequence token (<s>). Using the encoded representation and the initial token, the decoder predicts the next token in the sequence, continuing this process iteratively. Lastly, beam search (Freitag and Al-Onaizan, 2017), a heuristic algorithm, explores the result set by expanding the most promising results within the set. The width of this beam, typically referred to as the "beam size," determines the number of nodes expanded at each level and balances exploration and exploitation. The resultant best se-

quence of tokens is converted into text, representing TokenOCR’s final output.

3 MODEL TRAINING DATA GENERATION

Data generation involves creating a comprehensive dataset of synthetic text images that TokenOCR can use for training. This approach ensures that the model can accurately recognize and interpret text across a wide range of real-world scenarios. Synthetic data generation offers several key advantages over real-world data:

- **Control Over Data Quality and Variety:** Tailors the dataset to include diverse text scenarios, ensuring consistency and diversity.
- **Handling Specific Use Cases:** Simulates specific distortions like skewing and blurring to enhance robustness.
- **Scalability and Efficiency:** Produces large volumes of data quickly, accelerating the training process.
- **Mitigating Data Scarcity:** Addresses the lack of high-quality labelled data in a niche field like OCR.
- **Ethical and Legal Considerations:** Avoids legal and ethical issues related to data usage.
- **Customizable Complexity:** Introduces systematic variations to mimic real-world imperfections.

3.1 Text Recognition Data Generator

To construct a robust training dataset for TokenOCR, we employed the Text Recognition Data Generator (TRDG) (Krishnan and Jawahar, 2016), an open-source tool designed for generating synthetic text images.

The content for the 6 million generated images (Figure 3) was sourced from Wikipedia articles using the `GeneratorFromWikipedia` function of TRDG. Wikipedia’s vast and diverse range of content, from detailed scientific entries to broad historical summaries, ensures that TokenOCR is exposed to various text types and complexities. This diversity enables the model to handle real-world textual scenarios requiring semantic understanding and contextual interpretation (Rusiñol et al., 2021).

To replicate real-world conditions as closely as possible, we incorporated the following image variation techniques during data generation:

- **Skewing:** Images were tilted at random angles between 0 to 2 degrees to simulate misalignment commonly seen in scanned documents. This helps the model recognize text in less-than-ideal alignments.
- **Blurring:** Randomized blur effects with values between 0 to 3 were applied to mimic varying levels of camera focus and motion blur. This prepares the model to handle practical situations with imperfect image clarity.
- **Size Variability:** Images were generated with varying widths between 780 and 1280 pixels, reflecting typical size variations in digital documents and enhancing adaptability to diverse dimensions.

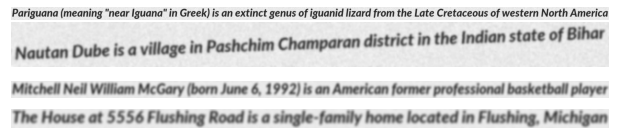


Figure 3: Examples of images generated by TRDG with various blurs and skews.

4 TRAINING

TokenOCR’s training strategy employs adaptive learning rate scheduling (Xu et al., 2019) and curriculum learning (Soviany et al., 2022). Techniques like early stopping and dropout regularization prevent overfitting, ensuring the model generalizes well to unseen data.

4.1 Curriculum Learning

Curriculum learning divides the training process into stages, introducing progressively complex data. In TokenOCR, this approach enhances generalization across various textual complexities and conditions. The curriculum structure is as follows:

- **Random Letters:** Initial training with images of random letters teaches the model basic character recognition without word structure complexity.
- **Nonsensical Words:** The next stage introduces images of nonsensical words, enabling the model to recognize letter combinations and spacing nuances.
- **Random Words:** Images of random words help the model understand common letter groupings and word structures.
- **Real Semantically Correct Sentences:** The final stage involves semantically correct sentences,

Table 4: Sample Blurry Images from the Test.

1	Image	
	Ground Truth	question treatment can develop
	TokenOCR	question treatment can develop
	Tesseract	(no answer)
	TrOCR
2	Image	
	Ground Truth	clearly according clearly she toward
	TokenOCR	clearly according clearly she toward
	Tesseract	clearly according Clearly she toward
	TrOCR
3	Image	
	Ground Truth	process road piece force interesting look move sound
	TokenOCR	process road piece force interesting look move sound
	Tesseract	process road parce force mteresting 008 Mowe soured
	TrOCR
4	Image	
	Ground Truth	special rise family fast and travel send
	TokenOCR	special rise family fast and travel send
	Tesseract	Spec rae fore fast ond trove send
	TrOCR
5	Image	
	Ground Truth	your campaign language remember model remain large
	TokenOCR	your campaign language remember model remain large
	Tesseract	your Compargn language remember mode! remon large
	TrOCR

The test results are given in Tables 1, 2, 3 and 4.

$$WER = \frac{\# \text{ Substitutions} + \# \text{ Deletions} + \# \text{ Insertions}}{\# \text{ Words in the reading frame}} \quad (1)$$

$$CER = \frac{\# \text{ Substitutions} + \# \text{ Deletions} + \# \text{ Insertions}}{\# \text{ Characters in the reading frame}} \quad (2)$$

TokenOCR was benchmarked against Pytesseract and TrOCR. Pytesseract, a well-known OCR engine, provides a strong baseline, while TrOCR’s transformer-based architecture offers a modern comparison.

TokenOCR consistently outperformed benchmarks in both clean and challenging conditions, demonstrating its reliability and versatility across various use cases. These results position TokenOCR as a leading tool in OCR applications requiring precision and adaptability.

6 CONCLUSION

This paper introduces TokenOCR, a novel attention-based OCR model that addresses persistent challenges in text recognition across complex and real-world scenarios. By integrating convolutional neural networks with transformer architectures, TokenOCR effectively captures both visual and textual information. A key innovation of TokenOCR lies in its prediction of tokens instead of individual characters, enabling the model to leverage contextual information for significantly improved performance in text recognition tasks. This approach reduces errors caused by out-of-context character predictions and enhances the model’s ability to generalize across diverse text layouts and fonts.

TokenOCR’s architecture features ResNet50-based feature extraction, adaptive 2D positional embeddings, and multi-headed attention mechanisms, all contributing to its robust performance. Training on a large-scale synthetic dataset generated with TRDG allowed the model to become resilient to real-world imperfections, such as skewing and blurring. Additionally, a curriculum learning-based training strategy progressively exposed the model to increasing complexities, ensuring strong generalization across various document types and conditions.

Evaluation results highlight TokenOCR’s superiority, consistently outperforming established benchmarks such as Tesseract and TrOCR in Word Error Rate (WER) and Character Error Rate (CER) metrics across clean and degraded datasets. This performance demonstrates the practical impact of token-based prediction, which aligns well with real-world OCR challenges.

The implications of this work are substantial for industries requiring precise and reliable text recognition, including legal documentation, healthcare, and digital archiving. Future research directions include

optimizing computational efficiency, expanding multilingual capabilities, and exploring applications to more complex document structures. With its demonstrated efficacy and innovative approach, TokenOCR represents a significant advancement in OCR technology, setting a new standard for text digitization and analysis.

REFERENCES

- Baek, J., Kim, G., Lee, J., Park, S., Han, D., Yun, S., Oh, S. J., and Lee, H. (2019a). What is wrong with scene text recognition model comparisons? dataset and model analysis.
- Baek, Y., Lee, B., Han, D., Yun, S., and Lee, H. (2019b). Character region awareness for text detection.
- Cloud, G. (2024). Cloud vision documentation. <https://cloud.google.com/vision/docs>. Accessed: 2024-08-28.
- Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. (2021). An image is worth 16x16 words: Transformers for image recognition at scale.
- Freitag, M. and Al-Onaizan, Y. (2017). Beam search strategies for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*. Association for Computational Linguistics.
- Gheini, M., Ren, X., and May, J. (2021). Cross-attention is all you need: Adapting pretrained transformers for machine translation.
- Krishnan, P. and Jawahar, C. V. (2016). Generating synthetic data for text recognition.
- Kudo, T. and Richardson, J. (2018). Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing.
- Li, M., Lv, T., Chen, J., Cui, L., Lu, Y., Florencio, D., Zhang, C., Li, Z., and Wei, F. (2022). Trocr: Transformer-based optical character recognition with pre-trained models.
- Liao, M., Zou, Z., Wan, Z., Yao, C., and Bai, X. (2022). Real-time scene text detection with differentiable binarization and adaptive scale fusion.
- Liu, Y., Shao, Z., and Hoffmann, N. (2021). Global attention mechanism: Retain information to enhance channel-spatial interactions.
- Pascanu, R., Mikolov, T., and Bengio, Y. (2013). On the difficulty of training recurrent neural networks.
- Raj, R. and Kos, A. (2022). A comprehensive study of optical character recognition. In *2022 29th International Conference on Mixed Design of Integrated Circuits and System (MIXDES)*, pages 151–154.
- Rusiñol, M., Sanchez, J.-M., and Karatzas, D. (2021). Document image quality assessment via explicit blur and text size estimation. In *International Conference on Document Analysis and Recognition*, pages 308–321. Springer.
- Senrich, R., Haddow, B., and Birch, A. (2016). Neural machine translation of rare words with subword units.
- Services, A. W. (2024). Amazon textract documentation. <https://aws.amazon.com/textract/>. Accessed: 2024-08-28.
- Shaw, P., Uszkoreit, J., and Vaswani, A. (2018). Self-attention with relative position representations.
- Smith, R. (2007). An overview of the tesseract ocr engine. *Google Inc.* Available at <https://code.google.com/p/tesseract-ocr/>.
- Soviany, P., Ionescu, R. T., Rota, P., and Sebe, N. (2022). Curriculum learning: A survey.
- Team, N., Costa-jussà, M. R., Cross, J., Çelebi, O., Elbayad, M., Heffernan, K., Kalbassi, E., Lam, J., Licht, D., Maillard, J., Sun, A., Wang, S., Wenzek, G., Youngblood, A., Akula, B., Barrault, L., Gonzalez, G. M., Hansanti, P., Hoffman, J., Jarrett, S., Sadagopan, K. R., Rowe, D., Spruit, S., Tran, C., Andrews, P., Ayan, N. F., Bhosale, S., Edunov, S., Fan, A., Gao, C., Goswami, V., Guzmán, F., Koehn, P., Mourachko, A., Ropers, C., Saleem, S., Schwenk, H., and Wang, J. (2022). No language left behind: Scaling human-centered machine translation.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2023). Attention is all you need.
- Xu, Z., Dai, A. M., Kemp, J., and Metz, L. (2019). Learning an adaptive learning rate schedule.
- Yu, R., Wang, Z., Wang, Y., Li, K., Liu, C., Duan, H., Ji, X., and Chen, J. (2024). Lape: Layer-adaptive position embedding for vision transformers with independent layer normalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5886–5896. IEEE.