# Keras-OCR

Keras-OCR represents an open-source machine learning library designed to offer developers a comprehensive toolset for constructing and deploying robust optical character recognition (OCR) pipelines (faustomorales s, 2019).

The architecture of Keras-OCR revolves around two primary components: a text detector and a text recognizer. The text detector, which integrates the CRAFT (Character Region Awareness for Text Detection) algorithm (Baek et al., 2019), functions to identify text regions within input images. Following text detection, the text recognizer module, which can utilize various models like STAR-Net (Xing et al., 2019) or CRNN (Shi, Bai, & Yao, 2015), proceeds to recognize the text contained within the detected regions.

Moreover, Keras-OCR incorporates an attention mechanism inspired by the principles outlined in "Attention is All You Need" (Vaswani et al., 2017), aiming to enhance recognition accuracy. This architecture ensures that Keras-OCR offers a versatile and effective solution for text detection and recognition tasks in diverse application scenarios.

Keras-OCR distinguishes itself from traditional OCR systems in several key aspects. Unlike conventional OCR approaches that often rely on handcrafted features and complex preprocessing pipelines, Keras-OCR adopts a deep learning-based paradigm, leveraging neural networks to automatically learn hierarchical representations directly from raw image data. This shift towards deep learning enables Keras-OCR to achieve superior performance across a wide range of text detection and recognition tasks, especially in handling complex and irregular text layouts.

Keras-OCR offers a modular and flexible architecture, empowering developers to easily customize and fine-tune OCR models according to specific application requirements and datasets. In contrast, traditional OCR systems tend to have rigid architectures and limited adaptability, making them less versatile and scalable in real-world scenarios.

Additionally, Keras-OCR benefits from the rich ecosystem of the Keras framework, providing seamless integration with other deep learning tools and libraries, thereby streamlining the development and deployment of OCR solutions.

# EasyOCR

EasyOCR is an open source OCR engine based on deep learning that can be used for text detection and recognition in multiple languages.

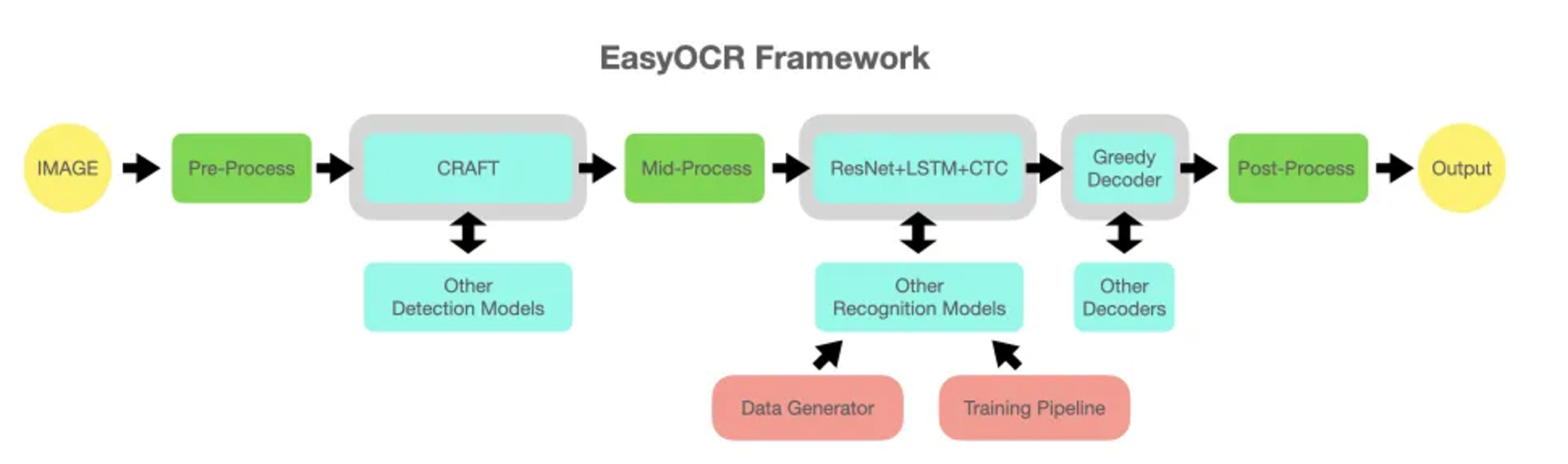


Figure 1 The overall framework structure of EasyOCR (GitHub, n.d.)

EasyOCR presents a comprehensive and modular framework for optical character recognition (OCR) tasks. The pipeline commences with a pre-processing stage, where input images undergo essential transformations, such as resizing and enhancement. Subsequently, the text detection module, primarily employing the CRAFT model, which will localizes and delineates the text regions within the image (Baek et al., 2019). However, the framework maintains flexibility by allowing the integration of alternative detection models as per specific requirements. For example, the TextSnake text detector can detect curves and text instances in any direction very well (Long et al., 2018), and the DB model proposed by PaddleOCR can directly predict the binary segmentation map of text instances from the input image (Liao et al., 2020).

Once the text regions are identified, a mid-processing stage ensues, wherein the detected text regions undergo further refinements, including cropping and rotation correction. The text recognition module then takes over, leveraging a combination of ResNet, LSTM, and CTC architectures to transcribe the cropped text regions into their corresponding character sequences. Akin to the detection module, the recognition stage offers the option to incorporate alternative models seamlessly.

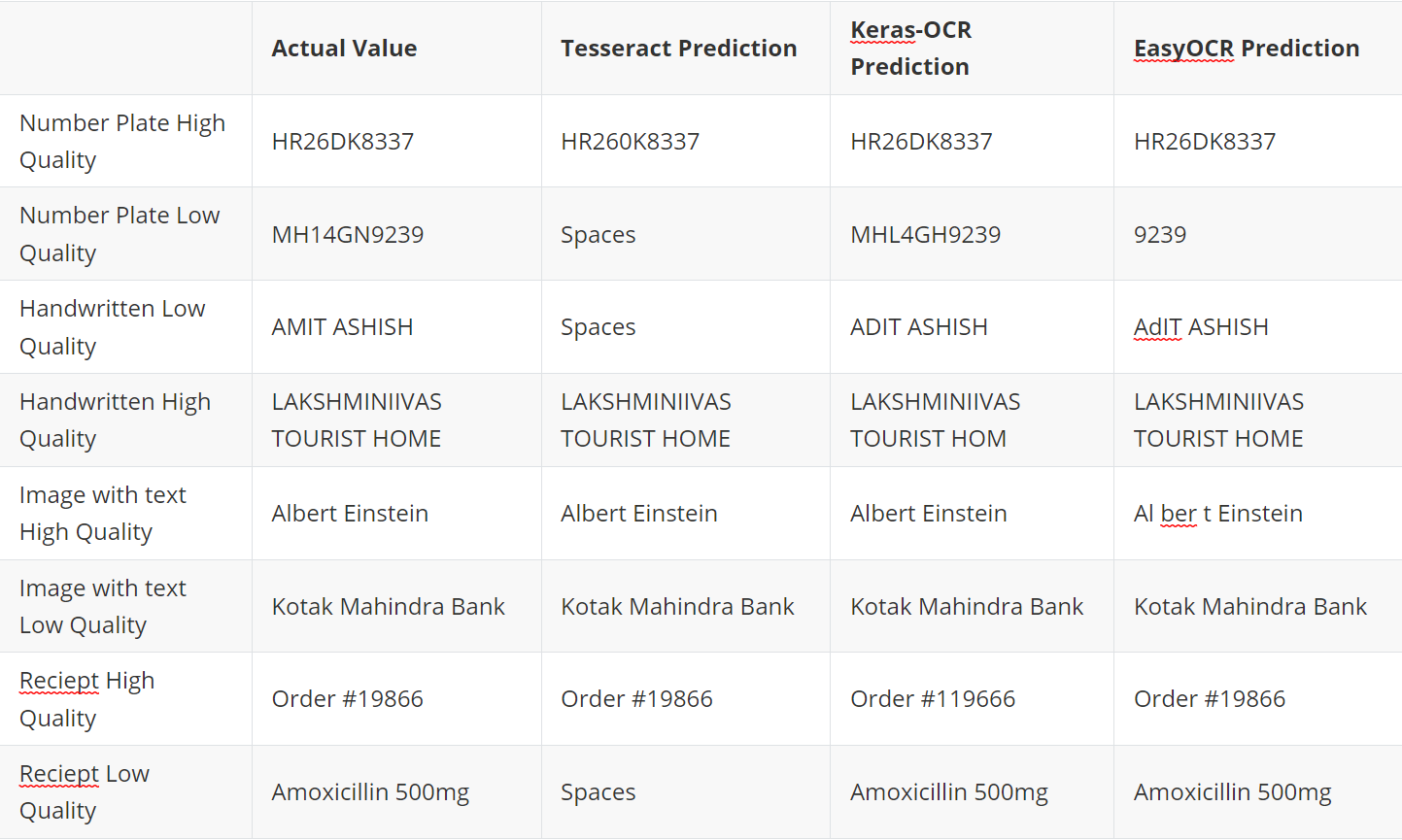
The recognized character sequences are subsequently decoded by the Greedy Decoder module, which generates the final textual output. Again, the framework permits the utilization of alternative decoders if desired. Finally, a post-processing stage handles any remaining formatting or refinement operations on the output text.

Notably, EasyOCR incorporates auxiliary modules that facilitate data generation and model training. The Data Generator module generates training data to support the development and fine-tuning of detection and recognition models, while the Training Pipeline module manages and orchestrates the entire training process for these models.

# **Tesseract vs Keras-OCR vs EasyOCR**

Image Datasets:

1. <https://raw.githubusercontent.com/Thangasami/OCR-/main/number/1.jpg>
2. <https://raw.githubusercontent.com/Thangasami/OCR-/main/number/2.jpg>
3. <https://raw.githubusercontent.com/Thangasami/OCR-/main/number/3.jpg>
4. <https://raw.githubusercontent.com/Thangasami/OCR-/main/number/4.jpg>
5. <https://raw.githubusercontent.com/Thangasami/OCR-/main/number/5.jpg>
6. <https://raw.githubusercontent.com/Thangasami/OCR-/main/number/6.jpg>
7. <https://raw.githubusercontent.com/Thangasami/OCR-/main/number/7.jpg>
8. <https://raw.githubusercontent.com/Thangasami/OCR-/main/number/8.jpg>
9. <https://raw.githubusercontent.com/Thangasami/OCR-/main/number/9.jpg>
10. <https://raw.githubusercontent.com/Thangasami/OCR-/main/number/10.jpg>



OCR prediction accuracy is influenced by a multitude of factors beyond the choice of model itself. These factors encompass image clarity, grayscale levels, hyperparameter settings, and the specific weight initialization utilized during training. Notably, Tesseract excels in its performance when presented with high-resolution images. Certain morphological operations, such as dilation, erosion, and OTSU binarization, can contribute to enhancing the performance of pytesseract, a Python wrapper for the Tesseract engine.

EasyOCR, a lightweight model, exhibits robust performance in scenarios involving receipts or PDF conversion. It demonstrates superior accuracy when processing organized text found in documents such as receipts, bills, and PDFs. In contrast, Keras-OCR is tailored for image-specific OCR tasks. When the text is embedded within an image and lacks a consistent font or color scheme, Keras-OCR proves to be an effective tool, delivering reliable results. These observations underscore the importance of considering the specific characteristics of the input data and the intended application when selecting an appropriate OCR model to ensure optimal performance.

# References

faustomorales (2019). \*faustomorales/keras-ocr\*. [online] GitHub. Available at: <https://github.com/faustomorales/keras-ocr>.

Baek, Y., Lee, B., Han, D., Yun, S. and Lee, H. (2019). *Character Region Awareness for Text Detection*. [online] IEEE Xplore. doi:https://doi.org/10.1109/CVPR.2019.00959.

Xing, L., Tian, Z., Huang, W. and Scott, M.R. (2019). Convolutional Character Networks. arXiv (Cornell University). doi:https://doi.org/10.1109/iccv.2019.00922.

Shi, B., Bai, X. and Yao, C. (2015). An End-to-End Trainable Neural Network for Image-based Sequence Recognition and Its Application to Scene Text Recognition. arXiv:1507.05717 [cs]. [online] Available at: <https://arxiv.org/abs/1507.05717>.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L. and Polosukhin, I. (2017). Attention Is All You Need. [online] arXiv.org. doi:https://doi.org/10.48550/arXiv.1706.03762.

GitHub. (n.d.). GitHub - JaidedAI/EasyOCR: Ready-to-use OCR with 80+ supported languages and all popular writing scripts including Latin, Chinese, Arabic, Devanagari, Cyrillic and etc. [online] Available at: <https://github.com/JaidedAI/EasyOCR>.

Liao, M., Wan, Z., Yao, C., Chen, K. and Bai, X. (2020). Real-Time Scene Text Detection with Differentiable Binarization. Proceedings of the AAAI Conference on Artificial Intelligence, [online] 34(07), pp.11474–11481. doi:https://doi.org/10.1609/aaai.v34i07.6812.