

In this tutorial, you will learn how to train an Optical Character Recognition (OCR) model using Keras, TensorFlow, and Deep Learning. **This post is the first in a two-part series on OCR with Keras and TensorFlow:**

- **Part 1:** *Training an OCR model with Keras and TensorFlow* (today's post)
- **Part 2:** *Basic handwriting recognition with Keras and TensorFlow* (next week's post)

For now, we'll primarily be focusing on how to train a custom Keras/TensorFlow model to recognize alphanumeric characters (i.e., the digits 0-9 and the letters A-Z).

Building on today's post, next week we'll learn how we can use this model to correctly classify handwritten characters in custom input images.

The goal of this two-part series is to obtain a deeper understanding of how deep learning is applied to the classification of handwriting, and more specifically, our goal is to:

- Become familiar with some well-known, readily available handwriting datasets for both digits and letters
- Understand how to train deep learning model to recognize handwritten digits and letters
- Gain experience in applying our custom-trained model to some real-world sample data
- Understand some of the challenges with real-world noisy data and how we might want to augment our handwriting datasets to improve our model and results

We'll be starting with the fundamentals of using well-known handwriting datasets and training a ResNet deep learning model on these data.

To learn how to train an OCR model with Keras, TensorFlow, and deep learning, *just keep reading.*