Build a training data set for a custom model

06/19/2019 • 3 minutes to read •

In this article

When you use the Form Recognizer custom model, you provide your own training data so the model can train to your industry-specific forms.

If you're training without manual labels, you can use five filled-in forms, or an empty form (you must include the word "empty" in the file name) plus two filled-in forms. Even if you have enough filled-in forms, adding an empty form to your training data set can improve the accuracy of the model.

If you want to use manually labeled training data, you must start with at least five filled-in forms of the same type. You can still use unlabeled forms and an empty form in addition to the required data set.

Training data tips

It's important to use a data set that's optimized for training. Use the following tips to ensure you get the best results from the Train Custom Model operation:

- If possible, use text-based PDF documents instead of image-based documents. Scanned PDFs are handled as images.
- For filled-in forms, use examples that have all of their fields filled in.
- Use forms with different values in each field.
- If your form images are of lower quality, use a larger data set (10-15 images, for example).
- The total size of the training data set can be up to 500 pages.

General input requirements

Make sure your training data set also follows the input requirements for all Form

Recognizer content.

Form Recognizer works on input documents that meet these requirements:

- Format must be JPG, PNG, PDF (text or scanned), or TIFF. Text-embedded PDFs are best because there's no possibility of error in character extraction and location.
- If your PDFs are password-locked, you must remove the lock before submitting them.
- PDF and TIFF documents must be 200 pages or less, and the total size of the training data set must be 500 pages or less.
- For images, dimensions must be between 600 x 100 pixels and 4200 x 4200 pixels.
- If scanned from paper documents, forms should be high-quality scans.
- Text must use the Latin alphabet (English characters).
- For unsupervised learning (without labeled data), data must contain keys and values.
- For unsupervised learning (without labeled data), keys must appear above or to the left of the values; they can't appear below or to the right.

Form Recognizer doesn't currently support these types of input data:

- Complex tables (nested tables, merged headers or cells, and so on).
- Checkboxes or radio buttons.

Upload your training data

When you've put together the set of form documents that you'll use for training, you need to upload it to an Azure blob storage container. If you don't know how to create an Azure storage account with a container, following the Azure Storage quickstart for Azure portal.

If you want to use manually labeled data, you'll also have to upload the *.labels.json* and *.ocr.json* files that correspond to your training documents. You can use the Sample labeling tool (or your own UI) to generate these files.

Organize your data in subfolders (optional)

By default, the Train Custom Model API will only use form documents that are located at the root of your storage container. However, you can train with data in subfolders if you specify it in the API call. Normally, the body of the Train Custom Model call has the following format, where `<SAS URL>` is the Shared access signature URL of your container:

JSON Copy

```
{
  "source":"<SAS URL>"
}
```

If you add the following content to the request body, the API will train with documents located in subfolders. The `"prefix"` field is optional and will limit the training data set to files whose paths begin with the given string. So a value of `"Test"`, for example, will cause the API to look at only the files or folders that begin with the word "Test".

JSON Copy

```
{
  "source": "<SAS URL>",
  "sourceFilter": {
    "prefix": "<prefix string>",
    "includeSubFolders": true
  },
  "useLabelFile": false
}
```

Next steps

Now that you've learned how to build a training data set, follow a quickstart to train a custom Form Recognizer model and start using it on your forms.

- Train a model and extract form data using cURL
- Train a model and extract form data using the REST API and Python
- Train with labels using the sample labeling tool
- Train with labels using the REST API and Python

Is this page helpful?

Yes No