



## Tell us what your idea is.

---

Imagine taking a picture of your bookshelf, and instantly being presented with a list of books to add to your mobile library, without ever having to pick up a single book. Just swipe left to discard a book, and swipe right to add a book to your library. Shelf Shuffle's custom machine learning models will use the **Firestore Vision API** and **Tensorflow Lite** title *object detection* models to scoop up whole shelves of books at a time, and store them on your device with just a few swipes. You can then quickly browse your book list, edit details, or make notes completely *on-device*.

Shelf Shuffle does offer basic book scanning options like barcode scanning, or manually entering titles/ISBNs. But, Shelf Shuffle also allows people with physical book collections to quickly, easily, and privately catalog all their books on their mobile devices--without having to enter every book on your shelf individually. Utilizing *on-device machine learning models* and a simple clean UI, users will be able to quickly scan even large libraries of books and then view, manage, save, grow, and share their library with others.

Shelf Shuffle aims to give users the simplicity, speed, and privacy of an entirely *on-device book tracking solution*. The app does integrate with **Google Books API** and **Goodreads APIs** which are used to anonymously query detailed information about newly added books, but users' book lists and camera images will never leave their mobile device. Using *on-device machine learning*, the app is able to detect and identify your book titles using only your local hardware and camera--providing a *fast, secure, simple* solution.

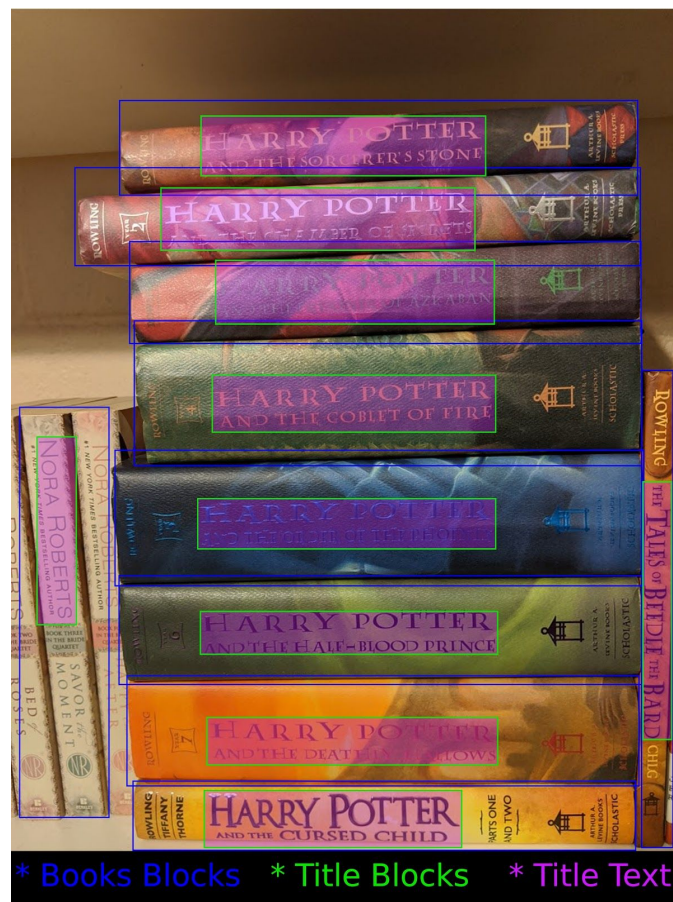


## Tell us how you plan on bringing it to life.

The app is in the proof-of-concept stage at this point, but contains a working prototype written using Dart and Flutter. Most of the remaining work is just debugging, polishing the UI, adding convenience features, and improving the machine learning model. For the debugging, UI, and feature work, I'll defer to my [GitHub issue page](#). But for the machine learning work, I'll need the help of experienced machine learning developers to get to the next stage.

Currently, I'm [using the Firebase Vision API's text recognition tooling to find book titles in an image](#). This provides a [list of text blocks](#) that I can use to [query for detailed book information](#). However, in practice the text returned by MLKit includes a lot of cruft text besides the title that tends to yield incorrect or empty query results.

## Planned Machine Learning Implementation:





To minimize this cruft text, I'd like help and mentorship from the Google Machine Learning team to implement the following machine learning models using TensorFlow Lite (see accompanying diagram on previous page):

1. (Blue) A model to detect books and their boundaries in an image. In particular, this model needs to work well from a side-view for scanning books lined up on a shelf. This will filter out any text that can not be a book title. This can likely be implemented using existing object detection models like **MobileNet**.
2. (Green) A model to select all text blocks that look like titles and are within a book boundary defined by model #1. This is the hardest part of the project and where I expect to need the most help. My rough plan is to feed each cropped book image into a custom **Tensorflow Lite** object recognition model to produce bounding boxes around all detected titles.
3. (Purple) From here, I can simply use the on-device **Firestore Vision API** tools to convert the detected title sub-images into the title of a given book.

One nice feature of this project is that there are already many large, open databases of book information to pull examples from for training our models. A few such APIs I find especially useful are listed below.

- The Google Books API
- The Goodreads API
- OpenLibrary API

Additional ways Google could help include:

1. Help choosing which models to use as starting points for training.
2. Help deciding how much training data I should plan to collect for training and how to properly label training data.
3. Help optimizing Tensorflow Lite models.
4. Help figuring out how to chain/combine the relevant models together efficiently on-device.

You can find the code for this project as well as screen captures of the app in action at <https://github.com/dannydabbles/ShelfShuffle>.



## Projected Development Timeline:

- **December:**
  - Improve project documentation.
- **January:**
  - Work through the existing [backlog of known application issues](#).
- **February-March:**
  - Work on improving machine learning tooling with the help of Google's Machine Learning team.
  - Collect and tag any additional training data needed.
  - Write machine learning models for book title object detection.
- **April-May:**
  - Continue to optimize the machine learning models.
  - Polish the UI/UX for release.
  - Conduct final round of user testing.
  - Finalize app submission to the Play Store.
- **Beyond:**
  - Since Dart/Flutter support cross-platform development, I'd like to port my app to iOS as well. I think this could be a great showcase for Google machine learning tooling running on iOS devices.



## Tell us about you.

---

I'm a Software Engineer living in the Sacramento area of California with my wife and son. In my free time, I enjoy playing with my son, tinkering with new software/hardware projects, and gardening in my backyard. In college, I majored in Computer Game Design at UC Santa Cruz. More recently, I've been working on Augmented Reality development, as well as mobile development.

In over a decade working as a Software Engineer, I've built/tested/fixed/released/containerized software for hardware products ranging from physical server boxes in data centers, to satellites orbiting the sun, to Augmented Reality headsets. I've also worked extensively building web tools, data pipelines and visualization tools to make sense of large quantities of data.

For years my wife has asked me for an app to help track her physical book collection. She's an avid reader who never cared much for ebooks. So, over the years, she has tried most book tracking tools on the market. However, they were all either too difficult, too buggy, or too slow to use and each was abandoned over time in turn.

In particular, her main feature requests were:

1. A clean, simple, and intuitive user interface.
2. The ability to quickly scan a shelf of books without having to pick up each book to expose the front cover.
3. The ability to import/export saved library data to a simple file.
4. No requirement to log in to anything or otherwise create credentials.

...These are now the core ideas behind the Shelf Shuffle app. While I welcome feedback from any and all users of the app, it's been very helpful to have such an amazing and engaged "in-house" alpha tester in my wonderful wife.

For more information about me, check out [my website](#) and [my projects](#), including a [review of Google's machine learning crash course](#), and an explainer for [how to run machine learning models on Nvidia GPUs using docker](#).