

Bookstore and Image to ISBN Problem

December 22, 2023



- Author: J.H. Martel*

1 Bookstore Problem.

“People don’t buy what they don’t see”. Or equivalently “People only buy what they see”.

We walk into Book Bazaar on Bank Street, and we see shelves on shelves filled with books. And great titles. *But who is buying?* This is the same problem as the vintage clothing shops. The owners are still running the old fashioned brick and mortar model where the owner waits until:

- clients walk into the store;
- clients browse the shelves;
- clients maybe identify an item they want;
- clients maybe buy the item.

We ask the owner of BB: “Do you have a searchable index of these books?” I.e. do you even know what your inventory *is*? The owner says “No.” In many cases the owner indeed knows alot of titles in his/her head. But the problem is the owner remains the *only person* with a virtual approximate index in their head, and who can search it?

So what’s the problem, and what’s the solution? We need a basic function to take images of bookshelves, and output lists of books (i.e. authors, titles, etc..). This is like *image_to_ISBN_list* routine.

- Input: digital images of book stacked in bookshelves.
- Output: Searchable index of ISBNs and titles.

We see the algorithm factored into three steps: 1) individualization, 2) image-to-text, 3) text-to-isbn.

- Step 1: (Individualization) Isolate rectangles of the individual books in the image.

Concretely this means partitioning the digital image into rectangles where each rectangle contains the “spine” of the book and the relevant individual text. The text is unstructured and fragmented.

Remark. If the books are randomly distributed on the book shelf, then with large probability the books are different colours and different shapes and there is alot of contrast between adjacent books. Therefore the individual rectangles/squares/individualization should perform best on random bookshelves.

By contrast the BB has a shelf of “penguin classics” and they are paperbacks which look identical in size and colour, and the only difference becomes the small faint text. However we don’t think this setting is relevant to the BB example, since these books are already “worthless”.

- Step 2: (Image-to-text) We compose the individualized rectangles into a basic `image_to_text` function. This provides some unordered text data. I.e., just words or letters, etc, and whatever symbols are available on the spine image.

There is limited text on the spine therefore we need extract all the words, as much as possible. But the text is unstructured, i.e. not consisting of sentences and sometimes images, i.e. publisher’s logo.

- Step 3: (Text to ISBN) Finally we compose the text output from Step 2 with a `text_to_isbn` function.

Again the text extracted from Step 2 is partly unstructured since the spine contains limited publisher information. In Step 3 we take a “best guess” of the book title (ISBN) using the text extracted from Step 2.

Claim: The composition of steps $3 \circ 2 \circ 1$ gives an `image_to_isbn` mapping. This is the basic tool which we offer to the owners of the bookstores. For practical purposes, the ISBN assignment would be printed as excel table. Therefore the owners would be able to apply basic search functions.

Remark. For practical application to the book store, another key problem is updating the index based on daily or weekly in-flows and out-flows of books. Most stores keep a written list of sales outgoing which can be photographed at end of week and then broadcast/merged into the index.