

Data Collection and Preprocessing Phase

Date	15 March 2024
Team ID	739890
Project Title	Bookmate: A Chatbot Companion For Book Recommendations
Maximum Marks	6 Marks

Preprocessing Template

The images will be preprocessed by resizing, normalizing, augmenting, denoising, adjusting contrast, detecting edges, converting color space, cropping, batch normalizing, and whitening data. These steps will enhance data quality, promote model generalization, and improve convergence during neural network training, ensuring robust and efficient performance across various computer vision tasks.

Section	Description
Data Overview	<p>Goodreads API: A rich source of book metadata, ratings, and reviews.</p> <p>Open Library API: A free and open-source library of books.</p> <p>Google Books API: Access to Google Books metadata, summaries, and author information.</p>
Resizing	While text-based data doesn't require resizing, images, such as book covers, might need to be resized for efficient processing and display.
Normalization	While Bookmate is primarily a text-based recommendation system, if you're working with image data (e.g., book covers), normalization is a crucial step in preprocessing.
Data Augmentation	It's important to clarify that data augmentation techniques like flipping, rotation, etc. are typically used for image, audio, or text datasets in machine learning applications. They wouldn't directly apply to a chatbot companion for book recommendations.
Denoising	While denoising is primarily used for image and audio data, it can be indirectly applied to text data to improve its quality.

Edge Detection	The project's core function is recommending books, not image analysis. While images might be present in book covers or user profiles, edge detection wouldn't contribute significantly to the recommendation process.
Color Space Conversion	Bookmate, as a recommendation chatbot, doesn't have built-in functionality to convert image color spaces. It's designed to focus on recommending books and wouldn't typically need to manipulate images.
Image Cropping	Download the image from Bookmate (if allowed) and edit it using a free or paid image editing app like GIMP , Snapseed , or Photos (on iPhone/iPad). These apps allow you to easily crop the image and focus on the region containing the object of interest (e.g., the book cover)
Batch Normalization	<p>Batch normalization is a technique used to normalize the inputs of each layer in a neural network. It helps in:</p> <ul style="list-style-type: none"> • Accelerated Training: By stabilizing the learning process, it allows for higher learning rates. • Improved Generalization: By reducing internal covariate shift, it helps the network generalize better to unseen data.
Data Preprocessing Code Screenshots	
Loading Data	<p>CSV (Comma-Separated Values): Tabular data with rows and columns.</p> <p>XML (Extensible Markup Language): Hierarchical data format.</p> <pre>import pandas as pd # Load the CSV file df = pd.read_csv("book_data.csv") # Display the first 5 rows print(df.head())</pre>
Resizing	<pre>Python from PIL import Image def resize_image(image_path, new_size=(224, 224)): img = Image.open(image_path) img = img.resize(new_size) return img</pre>

Normalization	<p>Python</p> <pre>from sklearn.preprocessing import MinMaxScaler scaler = MinMaxScaler() scaled_data = scaler.fit_transform(data)</pre>
Data Augmentation	<p>Python</p> <pre>from keras.preprocessing.image import ImageDataGenerator datagen = ImageDataGenerator(rotation_range=40, width_shift_range=0.2, height_shift_range=0.2, shear_range=0.2, zoom_range=0.2, horizontal_flip=True, fill_mode='nearest') </pre>
Denoising	<p>Python</p> <pre>import cv2 def denoise_image(image): denoised_image = cv2.fastNlMeansDenoisingColored(image, None, 10, 10 return denoised_image</pre>
Edge Detection	<p>Python</p> <pre>import cv2 def edge_detection(image): edges = cv2.Canny(image, 100, 200) return edges</pre>
Color Space Conversion	<p>Python</p> <pre>import cv2 def convert_color_space(image, color_space='HSV'): converted_image = cv2.cvtColor(image, cv2.COLOR_BGR2HSV) # Example return converted_image</pre>

Image Cropping	<pre>Python def crop_image(image, x, y, width, height): cropped_image = image[y:y+height, x:x+width] return cropped_image</pre>
Batch Normalization	<pre>Python from keras.layers import BatchNormalization model.add(BatchNormalization())</pre>