Lab 04 Reflective Journal

In this lab on deep learning data preprocessing, I gained a deeper understanding of the importance of preprocessing. Although deep learning models have the ability to perform automatic feature extraction, the lab made me realize that the quality of raw data still largely determines the final performance of the model. By learning preprocessing methods for different data types (images, text, and time series), I came to understand that proper preprocessing not only improves training stability but also reduces the risk of overfitting and accelerates convergence.

During the lab, I encountered several challenges. The most significant challenge occurred in Step 4 of the “video data simplified” section, where the function *cv2.destroyAllWindows()* could not be executed in Google Colab. This issue arose because Google Colab is a hosted notebook service rather than a local desktop environment. After consideration, I concluded that since the grayscale output had already been printed, this function could safely be ignored. Therefore, I commented out *cv2.destroyAllWindows()* and included a note explaining that the function is typically used to close all HighGUI windows, but due to the limitations of Google Colab, it had to be removed.

Some concepts covered in the lab were already familiar to me. For example, in previous courses I had learned basic text data cleaning techniques, including the use of NLTK and SpaCy. This prior knowledge helped me quickly understand the concepts of stop words and lemmatization in the lab. However, the lab further expanded my understanding by demonstrating how these steps could be implemented more concisely, giving me a clearer perspective on preprocessing methods for text data.

What surprised me most was the extent to which data preprocessing impacts model performance. Initially, I assumed that the powerful feature extraction capabilities of deep learning would compensate for issues in raw data. However, the results showed that if data is poorly distributed, excessively noisy, or not properly normalized, the model not only trains more slowly but also suffers a significant drop in prediction accuracy. This experience made me realize that preprocessing is not optional but rather an essential step in the deep learning workflow.

The preprocessing techniques learned in this lab have numerous real-world applications. For instance, in medical imaging analysis, normalization and data augmentation can improve performance when working with small datasets. In natural language processing tasks such as sentiment analysis, tokenization, noise removal, and word embeddings are indispensable. In financial time series forecasting, standardization can help models better capture patterns in price fluctuations. These examples highlight the crucial role of preprocessing in practical applications.

Looking ahead, I hope to further explore automated approaches to data preprocessing. While manually designed preprocessing workflows are important, automation and intelligent preprocessing strategies may be more efficient in complex applications. I am also interested in studying preprocessing techniques for audio and video data in greater depth, as these are critical for tasks such as speech recognition and video analysis.

Overall, this lab has shown me that data preprocessing is not only a fundamental skill but also a key component of the deep learning workflow. The insights and experience I gained through this lab will be highly valuable for my future learning and research.