OCR-Based Manual Summarizer

Yerin Lee(20240381),
KENTECH, School of energy Engineering

1. Title & Authors

OCR-Based Manual Summarizer
Author: YerinLee(20240381)
KENTECH, School of Energy Engineering

2. Motivation & Problem Definition

Many user manuals have small, dense text, making them difficult and tiring to read. Some manuals are also written in English, creating a language barrier for Korean users.

Goal of this project:
To develop a system that extracts the text, summarizes the important parts, translates English text into Korean, and converts the final result into audio.

Why it matters:

-Helps users understand long manuals more quickly
-Improves accessibility with audio output
-Removes language barriers through automatic translation
-Integrates OCR, summarization, translation, and TTS into one pipeline

3. Method

1. Idea & Motivation

User manuals often have small, dense text, and manuals written in English, making them hard to read.

Goal: build a system that extracts, summarizes, translates, and generates audio from manual text.

2. Model Selection

We used Tesseract for OCR, DeepSeek Chat for summarization, and gTTS for Korean TTS to keep the pipeline simple and efficient

-Teserract OCR
-DeepSeek Chat
-GTTS

OCR → summarization → TTS

3. Environment

Used Anaconda virtual environment.
Installed required Python packages, installed Tesseract OCR local, and configured the DeepSeek API key.

Model Analysis:
verified that Tesseract accurately reads small manual text.

4. Development

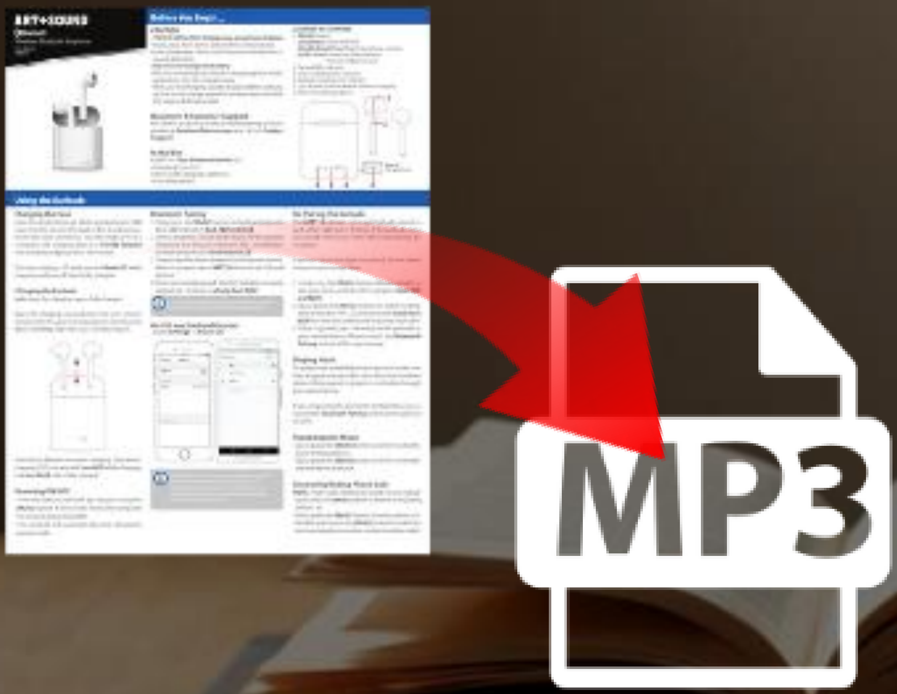
Created separate files
▪ ocr.py: extracts text using Tesseract
▪ summarizer.py: generates Korean summaries via DeepSeek API
▪ tts.py: converts summaries to audio using gTTS

Integrated all functions into a pipeline in main.py.

5. Distribution

Full source code available on GitHub.
Includes modular OCR-Summarization-TTS pipeline.
Provides a ready-to-run framework for document processing automation.

4. Results



- The OCR module accurately recognized small English text in manual images, while showing lower recognition performance for Korean, Chinese, and Japanese.
- Blurry or low-quality text also reduced OCR accuracy.
- The summarization module (DeepSeek API) generated concise Korean summaries that captured essential instructions and cautions.
- The TTS module (gTTS) functioned reliably, though the generated Korean speech sounded slightly unnatural.

The full pipeline (OCR → Summarization → TTS) ran consistently through main.py, confirming stable integration across all modules.

5. Discussion & Future work

Key Takeaways

-OCR module reads small English text accurately
-Summarization using the DeepSeek API consistently produces concise and relevant Korean summaries.
-The integrated pipeline (OCR → Summary → TTS) runs smoothly and demonstrates a functional automation workflow.

Future Improvements

-Improve multilingual OCR performance by adding language-specific models.
-Enhance TTS quality using a more natural neural TTS engine.
-Add support for multiple output languages (summary + audio).

6. Reference & Acknowledgement

- Tesseract OCR Documentation
- DeepSeek API Documentation
- gTTS (Google Text-to-Speech) Library
- Python Pillow (PIL) Documentation