**AIM:** Final Mini Project: Integrate CV and NLP (e.g., Read text from signboards or documents and translate/summarize it).

**DESCRIPTION:**

For integrating **Computer Vision (CV) and Natural Language Processing (NLP)** for tasks like reading text from images (signboards, documents) and then translating or summarizing it:

The goal of this project is to create a system that can **automatically read text from images** using computer vision techniques and then **process the extracted text** using NLP methods for translation, summarization, or sentiment analysis. This combines the fields of CV and NLP in a practical application.

**Components**

**1. Computer Vision (CV)**

* **Objective:** Detect and extract text from images.
* **Techniques & Tools:**
  + **Optical Character Recognition (OCR):** Converts images of text into machine-readable text.
    - **Libraries:** Tesseract OCR (pytesseract in Python)
  + **Image Preprocessing:** Improves OCR accuracy.
    - Convert images to grayscale.
    - Apply thresholding or filters to reduce noise.
    - Adjust contrast and brightness if needed.
  + **Optional:** Detecting text regions using **OpenCV** with contour detection or EAST text detector.

**2. Natural Language Processing (NLP)**

* **Objective:** Process extracted text for understanding, translation, or summarization.
* **Tasks:**
  + **Language Detection:** Identify the language of the text.
    - Libraries: googletrans or langdetect.
  + **Translation:** Convert text from source language to target language.
    - Libraries: googletrans or other translation APIs.
  + **Summarization:** Reduce text to key points.
    - Techniques:
      * **Extractive Summarization:** Select key sentences (e.g., sumy, gensim).
      * **Abstractive Summarization:** Generate concise paraphrased summaries (using models like BART, T5).
  + **Optional NLP Tasks:** Sentiment analysis, keyword extraction, or named entity recognition (NER).

**3. Text-to-Speech (Optional)**

* **Objective:** Read the summarized or translated text aloud.
* **Tools:** pyttsx3 or gTTS.

**Workflow**

1. **Image Input:** Capture an image from a camera, upload, or a dataset of signboards/documents.
2. **Preprocessing:** Enhance image quality for OCR.
3. **Text Extraction:** Apply OCR to extract textual information.
4. **Text Cleaning:** Remove noise, unwanted characters, and normalize text.
5. **Language Detection:** Identify the language.
6. **Translation (if needed):** Translate text to a target language.
7. **Summarization:** Generate a concise summary of the text.
8. **Text-to-Speech:** Optionally read the summary aloud.
9. **Output:** Display or save the translated/summarized text.

**Applications**

* Reading and translating road signs for tourists.
* Summarizing scanned lab manuals or study materials.
* Assisting visually impaired individuals by reading text aloud.
* Automating document translation and summarization workflows.

**Tools & Libraries**

| **Task** | **Libraries/Tools** |
| --- | --- |
| OCR | Tesseract OCR (pytesseract) |
| Image Preprocessing | PIL, OpenCV |
| Language Detection | googletrans, langdetect |
| Translation | googletrans, Microsoft Translator API |
| Summarization | sumy, gensim, Hugging Face Transformers |
| Text-to-Speech | pyttsx3, gTTS |

**Challenges**

* OCR accuracy can be affected by poor image quality, handwriting, or unusual fonts.
* Multi-language text requires robust detection and translation.
* Summarization may lose important context if the text is very short or very long.
* Real-time processing requires optimization for speed.

**Summary**

This project demonstrates a **practical integration of CV and NLP**:

* Computer Vision handles **text detection and extraction**.
* NLP handles **understanding, translation, and summarization**.
* Additional features like **text-to-speech** can enhance accessibility and usability.

**PROGRAM:**

import subprocess

import sys

import os

from PIL import Image, ImageEnhance, ImageFilter

# 🔹 Auto-install missing packages

def install(package):

subprocess.check\_call([sys.executable, "-m", "pip", "install", package])

# Install & import required libraries

try:

from googletrans import Translator

except ImportError:

install("googletrans==4.0.0-rc1")

from googletrans import Translator

try:

import pytesseract

except ImportError:

install("pytesseract")

import pytesseract

try:

import pyttsx3

except ImportError:

install("pyttsx3")

import pyttsx3

# Optional: set tesseract path manually on Windows

pytesseract.pytesseract.tesseract\_cmd = r"C:\Program Files\Tesseract-OCR\tesseract.exe"

# ========== STEP 0: Load local image ==========

IMAGE\_PATH = r'C:\surekha\Lab Manual\R23\Tesser.jpeg'

# ========== STEP 1: Preprocess Image ==========

def preprocess\_image(image\_path):

img = Image.open(image\_path)

img = img.convert('L') # Grayscale

img = img.filter(ImageFilter.MedianFilter())

enhancer = ImageEnhance.Contrast(img)

img = enhancer.enhance(2)

preprocessed\_path = "preprocessed\_image.png"

img.save(preprocessed\_path)

return preprocessed\_path

# ========== STEP 2: OCR - Extract text ==========

def extract\_text(image\_path):

print("\n🔍 Extracting text from image...")

text = pytesseract.image\_to\_string(Image.open(image\_path))

print("\n📝 Extracted Text:\n", text)

return text.strip()

# ========== STEP 3: Detect language ==========

def detect\_language(text):

print("\n🧐 Detecting language...")

translator = Translator()

detected = translator.detect(text)

confidence = detected.confidence if detected.confidence is not None else 1.0

print(f"\n✅ Detected Language: {detected.lang} (confidence: {confidence:.2f})")

return detected.lang

# ========== STEP 4: Translate into multiple languages ==========

def translate\_text\_multi(text, target\_languages=['en', 'es', 'fr', 'de']):

translator = Translator()

translations = {}

print("\n🌍 Translating text into multiple languages...")

for lang in target\_languages:

translated = translator.translate(text, dest=lang)

translations[lang] = translated.text

print(f"\n✅ {lang} translation:\n{translated.text}")

return translations

# ========== STEP 5: Simple Text Summarization ==========

def summarize\_text(text, num\_sentences=2):

sentences = text.split('.')

summary = '. '.join(sentences[:num\_sentences]).strip()

if not summary.endswith('.'):

summary += '.'

return summary

# ========== STEP 6: Text-to-Speech ==========

def speak\_text(text, lang='en'):

print("\n🔊 Reading summary aloud...")

engine = pyttsx3.init()

engine.say(text)

engine.runAndWait()

# ========== STEP 7: Save Summary ==========

def save\_summary(summary, lang\_code):

filename = rf"C:\surekha\Lab Manual\R23\summary\_{lang\_code}.txt"

with open(filename, "w", encoding="utf-8") as f:

f.write(summary)

print(f"💾 Summary for '{lang\_code}' saved to '{filename}'")

# ========== MAIN ==========

if \_\_name\_\_ == "\_\_main\_\_":

print("🚀 OCR + Multi-Language Translation + Summarization + TTS + Save Summary Starting...\n")

try:

if not os.path.exists(IMAGE\_PATH):

raise FileNotFoundError(f"❌ Image file not found at path: {IMAGE\_PATH}")

# Preprocess image

preprocessed\_path = preprocess\_image(IMAGE\_PATH)

# OCR

extracted\_text = extract\_text(preprocessed\_path)

if extracted\_text:

# Detect original language

detected\_lang = detect\_language(extracted\_text)

# Define target languages (ISO codes)

target\_languages = ['en', 'es', 'fr', 'de'] # English, Spanish, French, German

# Translate into multiple languages

translations = translate\_text\_multi(extracted\_text, target\_languages)

# Summarize, speak, and save for each language

for lang, translated\_text in translations.items():

print(f"\n✂️ Summarizing text in {lang}...")

summary = summarize\_text(translated\_text)

print(f"\n🧠 Summary ({lang}):\n{summary}")

speak\_text(summary, lang)

save\_summary(summary, lang)

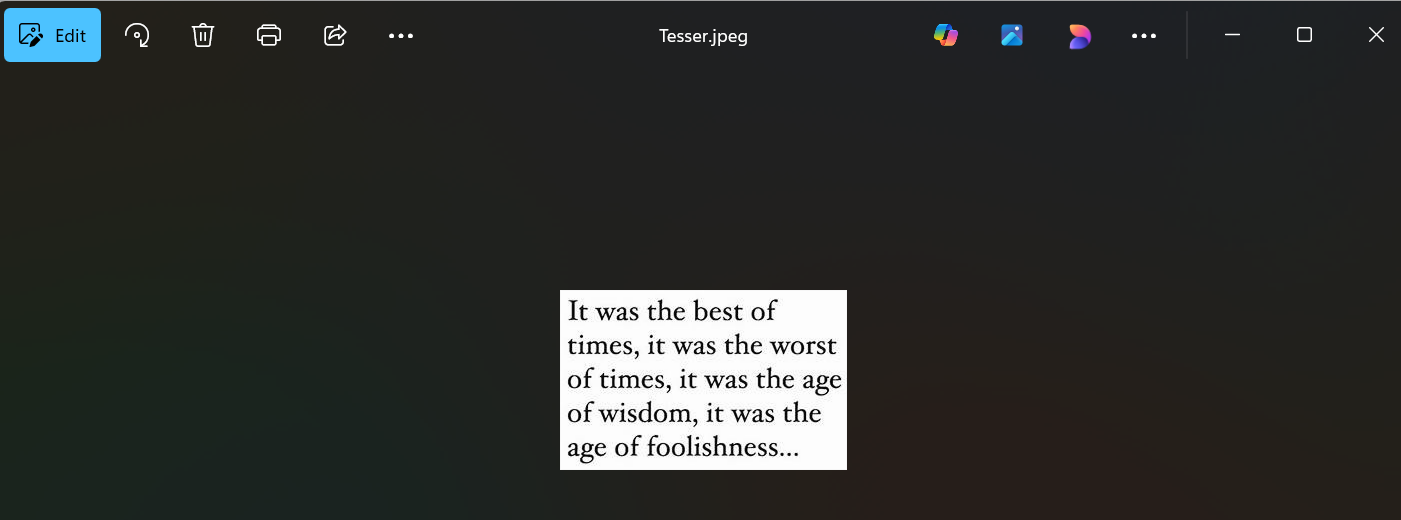
else:

print("⚠️ No text detected in the image.")

except Exception as e:

print(f"\n❌ Error: {e}")

**OUTPUT:**

****

"C:\Users\Surekha Swarna\PycharmProjects\NLP3RDYEARPROJ\.venv\Scripts\python.exe" "C:\Users\Surekha Swarna\PycharmProjects\NLP3RDYEARPROJ\app.py"

🚀 OCR + Multi-Language Translation + Summarization + TTS + Save Summary Starting...

🔍 Extracting text from image...

📝 Extracted Text:

It was the best of

times, it was the worst

of times, it was the age

of wisdom, it was the

age of foolishness...

🧐 Detecting language...

✅ Detected Language: en (confidence: 1.00)

🌍 Translating text into multiple languages...

✅ en translation:

It was the best of

times, it was the worst

of times, it was the age

of wisdom, it was the

age of foolishness...

✅ es translation:

fue lo mejor de

veces, fue lo peor

de veces, era la edad

de sabiduría, fue el

era de la tontería...

✅ fr translation:

C'était le meilleur de

des fois, c'était le pire

des fois, c'était l'âge

de sagesse, c'était le

l'âge de la bêtise...

✅ de translation:

Es war das Beste

Mal war es das Schlimmste

Manchmal war es das Alter

der Weisheit, es war das

Zeitalter der Dummheit ...

✂️ Summarizing text in en...

🧠 Summary (en):

It was the best of

times, it was the worst

of times, it was the age

of wisdom, it was the

age of foolishness.

🔊 Reading summary aloud...

💾 Summary for 'en' saved to 'C:\surekha\Lab Manual\R23\summary\_en.txt'

✂️ Summarizing text in es...

🧠 Summary (es):

fue lo mejor de

veces, fue lo peor

de veces, era la edad

de sabiduría, fue el

era de la tontería.

🔊 Reading summary aloud...

💾 Summary for 'es' saved to 'C:\surekha\Lab Manual\R23\summary\_es.txt'

✂️ Summarizing text in fr...

🧠 Summary (fr):

C'était le meilleur de

des fois, c'était le pire

des fois, c'était l'âge

de sagesse, c'était le

l'âge de la bêtise.

🔊 Reading summary aloud...

💾 Summary for 'fr' saved to 'C:\surekha\Lab Manual\R23\summary\_fr.txt'

✂️ Summarizing text in de...

🧠 Summary (de):

Es war das Beste

Mal war es das Schlimmste

Manchmal war es das Alter

der Weisheit, es war das

Zeitalter der Dummheit .

🔊 Reading summary aloud...

💾 Summary for 'de' saved to 'C:\surekha\Lab Manual\R23\summary\_de.txt'

Process finished with exit code 0

**RESULT:** Hence, Integrate CV and NLP (e.g., Read text from signboards or documents and translate/summarize it).