```
!pip install pytesseract pillow pyttsx3
```

```
Requirement already satisfied: pytesseract in /usr/local/lib/python3.11/dist-packages (0.3.13)
Requirement already satisfied: pillow in /usr/local/lib/python3.11/dist-packages (11.1.0)
Requirement already satisfied: pyttsx3 in /usr/local/lib/python3.11/dist-packages (2.98)
Requirement already satisfied: packaging>=21.3 in /usr/local/lib/python3.11/dist-packages (from pytesseract) (24.2)
```

```
!apt-get update
!apt-get install -y espeak tesseract-ocr
!pip install pytesseract pillow pyttsx3
```

```
Preparing to unpack .../2-espeak-data_1.48.15+dfsg-3_amd64.deb ...
Unpacking espeak-data:amd64 (1.48.15+dfsg-3) ...
Selecting previously unselected package libespeak1:amd64.
Preparing to unpack .../3-libespeak1_1.48.15+dfsg-3_amd64.deb ...
Unpacking libespeak1:amd64 (1.48.15+dfsg-3) ...
Selecting previously unselected package espeak.
Preparing to unpack .../4-espeak_1.48.15+dfsg-3_amd64.deb ...
Unpacking espeak (1.48.15+dfsg-3) ...
Selecting previously unselected package tesseract-ocr-eng.
Preparing to unpack .../5-tesseract-ocr-eng_1%3a4.00~git30-7274cfa-1.1_all.deb ...
Unpacking tesseract-ocr-eng (1:4.00~git30-7274cfa-1.1) ...
Selecting previously unselected package tesseract-ocr-osd.
Preparing to unpack .../6-tesseract-ocr-osd_1%3a4.00~git30-7274cfa-1.1_all.deb ...
Unpacking tesseract-ocr-osd (1:4.00~git30-7274cfa-1.1) ...
Selecting previously unselected package tesseract-ocr.
Preparing to unpack .../7-tesseract-ocr_4.1.1-2.1build1_amd64.deb ...
Unpacking tesseract-ocr (4.1.1-2.1build1) ...
Setting up libportaudio2:amd64 (19.6.0-1.1) ...
Setting up tesseract-ocr-eng (1:4.00~git30-7274cfa-1.1) ...
Setting up libsonic0:amd64 (0.2.0-11build1) ...
Setting up tesseract-ocr-osd (1:4.00~git30-7274cfa-1.1) ...
Setting up espeak-data:amd64 (1.48.15+dfsg-3) ...
Setting up libespeak1:amd64 (1.48.15+dfsg-3) ...
Setting up espeak (1.48.15+dfsg-3) ...
Setting up tesseract-ocr (4.1.1-2.1build1) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for libc-bin (2.35-0ubuntu3.8) ...
/sbin/ldconfig.real: /usr/local/lib/libumf.so.0 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtcm.so.1 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libur_adapter_opencl.so.0 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbbbind_2_0.so.3 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbbbind_2_5.so.3 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbbmalloc.so.2 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbbbind.so.3 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libur_adapter_level_zero.so.0 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtcm_debug.so.1 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libhwloc.so.15 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbb.so.12 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbbmalloc_proxy.so.2 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libur_loader.so.0 is not a symbolic link

Requirement already satisfied: pytesseract in /usr/local/lib/python3.11/dist-packages (0.3.13)
Requirement already satisfied: pillow in /usr/local/lib/python3.11/dist-packages (11.1.0)
Requirement already satisfied: pyttsx3 in /usr/local/lib/python3.11/dist-packages (2.98)
Requirement already satisfied: packaging>=21.3 in /usr/local/lib/python3.11/dist-packages (from pytesseract) (24.2)
```

```
import pytesseract
from PIL import Image
import pyttsx3
import re
from pathlib import Path
import logging
import platform
import subprocess
import io
import base64
from IPython.display import display, Audio
import numpy as np
from PIL import Image, ImageDraw, ImageFont

class SampleDataGenerator:
    @staticmethod
```

```
        def create_sample_newspaper_image():
            """Create a sample newspaper image with text"""
            # Create a new image with a white background
            width = 800
            height = 600
            image = Image.new('RGB', (width, height), 'white')
            draw = ImageDraw.Draw(image)

            # Add sample text
            sample_text = """
            DAILY NEWS

            CITY PLANS NEW PARK
            Local government announces plans for
            a new community park in downtown area.
            The project will begin next month.

            COMMUNITY EVENTS
            Senior center hosts weekly reading club
            every Tuesday at 2 PM. All are welcome
            to join and participate.

            WEATHER REPORT
            Sunny skies expected this weekend with
            temperatures reaching 75 degrees.
            Perfect weather for outdoor activities.
            """

            # Draw text on image
            draw.text((50, 50), sample_text, fill='black')

            # Save image to bytes
            img_byte_arr = io.BytesIO()
            image.save(img_byte_arr, format='PNG')
            img_byte_arr.seek(0)

            return img_byte_arr

    class NewspaperReader:
        def _init_(self):
            self.engine = None
            self.setup_tts_engine()

        def setup_tts_engine(self):
            """Initialize TTS engine with fallback options"""
            try:
                self.engine = 'espeak'  # Use espeak directly in Colab
            except:
                raise RuntimeError("Could not initialize TTS engine")

        def process_image(self, image_bytes):
            """Extract text from newspaper image using OCR"""
            try:
                image = Image.open(image_bytes)
                text = pytesseract.image_to_string(image)
                return self.clean_text(text)
            except Exception as e:
                return f"Error processing image: {str(e)}"

        def clean_text(self, text):
            """Clean and format extracted text"""
            text = re.sub(r'\s+', ' ', text)
            text = re.sub(r'[^\w\s.,!?-]', '', text)
            text = re.sub(r'([A-Z]{2,})\s', r'\1. ', text)
            return text.strip()

        def text_to_speech(self, text, output_path=None):
            """Convert text to speech using espeak"""
            try:
                if output_path:
                    subprocess.run(['espeak', '-w', output_path, text])
                    return "Success"
                else:
                    subprocess.run(['espeak', text])
                    return "Success"
            except Exception as e:
                return f"Error converting text to speech: {str(e)}"

        def process_newspaper(self, image_bytes, output_path=None):
            """Main function to process newspaper and convert to speech"""
            if output_path:
                Path(output_path).parent.mkdir(parents=True, exist_ok=True)
```

```
        text = self.process_image(image_bytes)
        if text.startswith("Error"):
            return text, None

        result = self.text_to_speech(text, output_path)
        return result, text

def main():
    try:
        # Generate sample data
        print("Generating sample newspaper image...")
        sample_generator = SampleDataGenerator()
        image_bytes = sample_generator.create_sample_newspaper_image()

        # Create output directory
        output_dir = Path("output")
        output_dir.mkdir(exist_ok=True)

        # Process sample image
        print("\nProcessing sample newspaper...")
        reader = NewspaperReader()
        result, extracted_text = reader.process_newspaper(
            image_bytes,
            "output/sample_article.wav"
        )

        print("\nExtracted Text:")
        print("-" * 50)
        print(extracted_text)
        print("-" * 50)

        print("\nTTS Result:", result)

        # Display audio if successful
        if result == "Success":
            print("\nPlaying audio... (If you're in Colab, you can play the audio file below)")
            display(Audio("output/sample_article.wav"))

    except Exception as e:
        print(f"Error: {str(e)}")
        print("Please ensure all required dependencies are installed:")
        print("1. For OCR: !apt-get install tesseract-ocr")
        print("2. For TTS: !apt-get install espeak")
        print("3. Python packages: !pip install pytesseract pillow pyttsx3")

if __name__ == "__main__": # Corrected the variable name to '__name__'
   main()
```

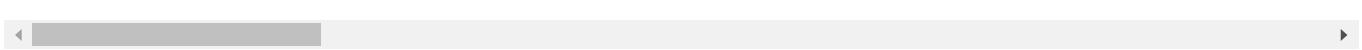
```
Generating sample newspaper image...

Processing sample newspaper...

Extracted Text:
--------------------------------------------------
DAILY. NEWS. CITY. PLANS. NEW. PARK. Local gavernmentannounces plans for a new community park in downtown area, The project begin ne
--------------------------------------------------

TTS Result: Success

Playing audio... (If you're in Colab, you can play the audio file below)
```



Start coding or generate with AI.