```
# Install necessary libraries
# We use !apt-get for system-level dependencies like poppler-utils
!apt-get update
!apt-get install -y poppler-utils

# We use !pip for Python libraries
!pip install opencv-python pytesseract pdf2image
```

```
Hit:1 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:2 http://archive.ubuntu.com/ubuntu jammy InRelease
Hit:3 https://cloud.r-project.org/bin/linux/ubuntu jammy-cran40/ InRelease
Hit:4 http://archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:5 http://archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:6 https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2204/x86_64  InRelease
Hit:7 https://r2u.stat.illinois.edu/ubuntu jammy InRelease
Hit:8 https://ppa.launchpadcontent.net/deadsnakes/ppa/ubuntu jammy InRelease
Hit:9 https://ppa.launchpadcontent.net/graphics-drivers/ppa/ubuntu jammy InRelease
Hit:10 https://ppa.launchpadcontent.net/ubuntugis/ppa/ubuntu jammy InRelease
Reading package lists... Done
W: Skipping acquire of configured file 'main/source/Sources' as repository 'https://r2u.stat.illinois.edu/ubuntu jammy InRelease' does r
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
poppler-utils is already the newest version (22.02.0-2ubuntu0.8).
0 upgraded, 0 newly installed, 0 to remove and 35 not upgraded.
Requirement already satisfied: opencv-python in /usr/local/lib/python3.11/dist-packages (4.11.0.86)
Requirement already satisfied: pytesseract in /usr/local/lib/python3.11/dist-packages (0.3.13)
Requirement already satisfied: pdf2image in /usr/local/lib/python3.11/dist-packages (1.17.0)
Requirement already satisfied: numpy>=1.21.2 in /usr/local/lib/python3.11/dist-packages (from opencv-python) (2.0.2)
Requirement already satisfied: packaging>=21.3 in /usr/local/lib/python3.11/dist-packages (from pytesseract) (24.2)
Requirement already satisfied: Pillow>=8.0.0 in /usr/local/lib/python3.11/dist-packages (from pytesseract) (11.2.1)
```

Lab 09 Task 01

```
# Step 1: Install required libraries
!apt-get install -y poppler-utils tesseract-ocr
!pip install pytesseract pdf2image opencv-python Pillow

# Step 2: Import libraries
import cv2
import pytesseract
import numpy as np
from pdf2image import convert_from_path
from PIL import Image
import os

# Step 3: Upload a PDF file
from google.colab import files
uploaded = files.upload()

# Get the filename (only one file is expected)
pdf_file = next(iter(uploaded))

# Step 4: Convert PDF pages to images
pages = convert_from_path(pdf_file, dpi=300)

# Step 5: Function to preprocess image
def preprocess_image(image):
    gray = cv2.cvtColor(image, cv2.COLOR_RGB2GRAY)
    blurred = cv2.GaussianBlur(gray, (5, 5), 0)
    thresh = cv2.adaptiveThreshold(
        blurred, 255,
        cv2.ADAPTIVE_THRESH_GAUSSIAN_C,
        cv2.THRESH_BINARY,
        11, 2
    )
    return thresh

# Step 6: Run OCR on each page
print("=== Extracted Text from PDF ===\n")
for i, page in enumerate(pages):
    # Convert PIL image to OpenCV format
    open_cv_image = cv2.cvtColor(np.array(page), cv2.COLOR_RGB2BGR)
```

```
    # Preprocess image
    preprocessed = preprocess_image(open_cv_image)

    # OCR using Tesseract
    text = pytesseract.image_to_string(preprocessed)

    print(f"\n--- Page {i + 1} ---\n")
    print(text)

print("\n=== OCR Complete ===")
```

```
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
tesseract-ocr is already the newest version (4.1.1-2.1build1).
poppler-utils is already the newest version (22.02.0-2ubuntu0.8).
0 upgraded, 0 newly installed, 0 to remove and 35 not upgraded.
Requirement already satisfied: pytesseract in /usr/local/lib/python3.11/dist-packages (0.3.13)
Requirement already satisfied: pdf2image in /usr/local/lib/python3.11/dist-packages (1.17.0)
Requirement already satisfied: opencv-python in /usr/local/lib/python3.11/dist-packages (4.11.0.86)
Requirement already satisfied: Pillow in /usr/local/lib/python3.11/dist-packages (11.2.1)
Requirement already satisfied: packaging>=21.3 in /usr/local/lib/python3.11/dist-packages (from pytesseract) (24.2)
Requirement already satisfied: numpy>=1.21.2 in /usr/local/lib/python3.11/dist-packages (from opencv-python) (2.0.2)
Choose Files No file chosen    Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.
Saving image pdf.pdf to image pdf (1).pdf
=== Extracted Text from PDF ===


--- Page 1 ---

608 Computer Vision: Algorithms and Applications (August 28, 2021 draft)



Figure 10.1. Computational photography: (a) merging multiple exposures to create high
dynamic range images (Debevec and Malik 1997) © 1997 ACM; (b) merging flash and non-
flash photographs; (Petschnigg, Agrawala et al. 2004) © 2004 ACM; (c) image matting and
compositing; (Chuang, Curless et al. 2001) © 2001 IEEE; (d) hole filling with inpainting
(Criminisi, Pérez, and Toyama 2004) © 2004 IEEE.


=== OCR Complete ===
```

Lab 09 Task 02

```
# STEP 2: Upload your PDF file
from google.colab import files
uploaded = files.upload()
pdf_file = next(iter(uploaded))  # Get the uploaded filename

# STEP 3: Convert PDF to images
from pdf2image import convert_from_path
pages = convert_from_path(pdf_file, dpi=300)

# STEP 4: Import libraries
import cv2
import numpy as np
import easyocr
import matplotlib.pyplot as plt

# STEP 5: Initialize EasyOCR reader
reader = easyocr.Reader(['en'])  # Add 'ur' or 'ar' if needed

# STEP 6: Process each page
for i, page in enumerate(pages):
    print(f"\n--- Processing Page {i + 1} ---")

    # Convert PIL Image to OpenCV BGR format
    image = np.array(page)
    image = cv2.cvtColor(image, cv2.COLOR_RGB2BGR)

    # OCR with bounding box details
    results = reader.readtext(image)

    # Collect all recognized text for this page
```

```
page_text = "\n".join([text for _, text, _ in results])

# Save the extracted text to a separate file for each page
output_filename = f"page_{i+1}_text.txt"
with open(output_filename, "w", encoding="utf-8") as f:
    f.write(page_text)
print(f"Text for page {i+1} saved to {output_filename}")

# Draw bounding boxes and text annotations on image
for (bbox, text, confidence) in results:
    top_left = tuple(map(int, bbox[0]))
    top_right = tuple(map(int, bbox[1]))
    bottom_right = tuple(map(int, bbox[2]))
    bottom_left = tuple(map(int, bbox[3]))

    # Draw rectangle
    cv2.line(image, top_left, top_right, (0, 255, 0), 2)
    cv2.line(image, top_right, bottom_right, (0, 255, 0), 2)
    cv2.line(image, bottom_right, bottom_left, (0, 255, 0), 2)
    cv2.line(image, bottom_left, top_left, (0, 255, 0), 2)

    # Draw corner circles
    for point in [top_left, top_right, bottom_right, bottom_left]:
        cv2.circle(image, point, 5, (0, 0, 255), -1)

    # Put recognized text near the bounding box
    cv2.putText(image, text, (top_left[0], top_left[1] - 10),
                cv2.FONT_HERSHEY_SIMPLEX, 0.8, (255, 0, 0), 2)

# Convert image back to RGB for display
image_rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

# Show image with bounding boxes
plt.figure(figsize=(12, 12))
plt.imshow(image_rgb)
plt.title(f"OCR with Text Bounding Boxes - Page {i + 1}")
plt.axis("off")
plt.show()
```

Choose Files No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
WARNING:easyocr.easyocr:Neither CUDA nor MPS are available - defaulting to CPU. Note: This module is much faster with a GPU.
Saving image direction board.pdf to image direction board (2).pdf

--- Processing Page 1 ---
Text for page 1 saved to page_1_text.txt
```

OCR with Text Bounding Boxes - Page 1