PADDLE CODE OCR

from paddleocr import PaddleOCR

# Initialize PaddleOCR with Dutch language (nl)

ocr = PaddleOCR(use\_angle\_cls=True, lang="nl") # Change language if needed

def split\_double\_page(image):

"""Splits a double-page image into left and right halves."""

height, width, \_ = image.shape

mid\_x = width // 2

left\_page = image[:, :mid\_x] # Left half

right\_page = image[:, mid\_x:] # Right half

return left\_page, right\_page

def process\_image(image):

"""Extract text from a single image."""

results = ocr.ocr(image, cls=True)

extracted\_text = []

for res in results:

if res is not None:

for line in res:

text = line[1][0] # Extract detected text

extracted\_text.append(text)

return "\n".join(extracted\_text)

def process\_pdf(pdf\_path, output\_folder):

"""Convert PDF pages to images, split double pages, and extract text from each half."""

os.makedirs(output\_folder, exist\_ok=True)

images = convert\_from\_path(pdf\_path)

extracted\_text = []

for i, image in enumerate(images):

image\_path = os.path.join(output\_folder, f"page\_{i+1}.jpg")

image.save(image\_path, "JPEG")

# Load image with OpenCV

img = cv2.imread(image\_path)

if img is None:

print(f"Error loading image: {image\_path}")

continue

# Split the image into two pages

left\_page, right\_page = split\_double\_page(img)

print(f"Processing page {i+1}: Left side...")

text\_left = process\_image(left\_page)

print(f"Processing page {i+1}: Right side...")

text\_right = process\_image(right\_page)

# Append in reading order (left first, then right)

extracted\_text.append(f"\n=== Page {i+1} (Left) ===\n{text\_left}")

extracted\_text.append(f"\n=== Page {i+1} (Right) ===\n{text\_right}")

return "\n".join(extracted\_text)

def main(input\_path):

"""Extract text from PDF with double-page spreads."""

output\_text\_file = f"{os.path.splitext(input\_path)[0]}.txt"

if input\_path.lower().endswith('.pdf'):

text = process\_pdf(input\_path, "pdf\_pages")

else:

print("Unsupported file format. Please use PDF files.")

return

with open(output\_text\_file, "w", encoding="utf-8") as f:

f.write(text)

print(f"Extracted text saved to: {output\_text\_file}")

# Place your PDF file path in the variable below

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

input\_path = "name.pdf" # <-- Replace with your PDF file path

main(input\_path)