```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
!sudo apt install tesseract-ocr
!pip install pytesseract pillow transformers tqdm
```

```
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  tesseract-ocr-eng tesseract-ocr-osd
The following NEW packages will be installed:
  tesseract-ocr tesseract-ocr-eng tesseract-ocr-osd
0 upgraded, 3 newly installed, 0 to remove and 29 not upgraded.
Need to get 4,816 kB of archives.
After this operation, 15.6 MB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu jammy/universe amd64 tesseract-ocr-eng all 1:4.00~git30-7274cfa-1.1 [1,591 kB]
Get:2 http://archive.ubuntu.com/ubuntu jammy/universe amd64 tesseract-ocr-osd all 1:4.00~git30-7274cfa-1.1 [2,990 kB]
Get:3 http://archive.ubuntu.com/ubuntu jammy/universe amd64 tesseract-ocr amd64 4.1.1-2.1build1 [236 kB]
Fetched 4,816 kB in 1s (3,971 kB/s)
debconf: unable to initialize frontend: Dialog
debconf: (No usable dialog-like program is installed, so the dialog based frontend cannot be used. at /usr/share/perl5/Debconf/FrontE
debconf: falling back to frontend: Readline
debconf: unable to initialize frontend: Readline
debconf: (This frontend requires a controlling tty.)
debconf: falling back to frontend: Teletype
dpkg-preconfigure: unable to re-open stdin:
Selecting previously unselected package tesseract-ocr-eng.
(Reading database ... 124947 files and directories currently installed.)
Preparing to unpack .../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 .../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 .../tesseract-ocr_4.1.1-2.1build1_amd64.deb ...
Unpacking tesseract-ocr (4.1.1-2.1build1) ...
Setting up tesseract-ocr-eng (1:4.00~git30-7274cfa-1.1) ...
Setting up tesseract-ocr-osd (1:4.00~git30-7274cfa-1.1) ...
Setting up tesseract-ocr (4.1.1-2.1build1) ...
Processing triggers for man-db (2.10.2-1) ...
Collecting pytesseract
  Downloading pytesseract-0.3.13-py3-none-any.whl.metadata (11 kB)
Requirement already satisfied: pillow in /usr/local/lib/python3.11/dist-packages (11.1.0)
Requirement already satisfied: transformers in /usr/local/lib/python3.11/dist-packages (4.48.3)
Requirement already satisfied: tqdm in /usr/local/lib/python3.11/dist-packages (4.67.1)
Requirement already satisfied: packaging>=21.3 in /usr/local/lib/python3.11/dist-packages (from pytesseract) (24.2)
Requirement already satisfied: filelock in /usr/local/lib/python3.11/dist-packages (from transformers) (3.17.0)
Requirement already satisfied: huggingface-hub<1.0,>=0.24.0 in /usr/local/lib/python3.11/dist-packages (from transformers) (0.28.1)
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.11/dist-packages (from transformers) (1.26.4)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.11/dist-packages (from transformers) (6.0.2)
Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib/python3.11/dist-packages (from transformers) (2024.11.6)
Requirement already satisfied: requests in /usr/local/lib/python3.11/dist-packages (from transformers) (2.32.3)
Requirement already satisfied: tokenizers<0.22,>=0.21 in /usr/local/lib/python3.11/dist-packages (from transformers) (0.21.0)
Requirement already satisfied: safetensors>=0.4.1 in /usr/local/lib/python3.11/dist-packages (from transformers) (0.5.3)
Requirement already satisfied: fsspec>=2023.5.0 in /usr/local/lib/python3.11/dist-packages (from huggingface-hub<1.0,>=0.24.0->transf
Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/python3.11/dist-packages (from huggingface-hub<1.0,>=0.24
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.11/dist-packages (from requests->transformers) (3.4
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.11/dist-packages (from requests->transformers) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.11/dist-packages (from requests->transformers) (2.3.0)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.11/dist-packages (from requests->transformers) (2025.1.31
Downloading pytesseract-0.3.13-py3-none-any.whl (14 kB)
Installing collected packages: pytesseract
```

```
import os
import pytesseract
from PIL import Image
from transformers import pipeline
from tqdm import tqdm
import re
```

```
image_folder = "/content/drive/My Drive/Delta"



def extract_text_from_images(folder_path):
    extracted_data = {}

    image_files = [f for f in os.listdir(folder_path) if f.lower().endswith(('.png', '.jpg', '.jpeg'))]

    print(f"🔍 Processing {len(image_files)} images...\n")

    for filename in tqdm(image_files, desc="Extracting Text"):
        image_path = os.path.join(folder_path, filename)

        try:
            image = Image.open(image_path)
            text = pytesseract.image_to_string(image)


            clean_text = re.sub(r'\n+', '\n', text).strip()
            extracted_data[filename] = clean_text
        except Exception as e:
            print(f"❌ Error processing {filename}: {e}")

    return extracted_data

extracted_texts = extract_text_from_images(image_folder)
```

```
🔍 Processing 300 images...

Extracting Text: 100%|██████████| 300/300 [26:43<00:00,  5.35s/it]
```

```
%env CUDA_LAUNCH_BLOCKING=1
```

```
env: CUDA_LAUNCH_BLOCKING=1
```

```
import random

sample_texts = random.sample(list(extracted_texts.values()), 5)
for i, text in enumerate(sample_texts):
    print(f"\n🔍 Sample {i+1}:\n{text}\n{'='*50}")
```

```

🔍 Sample 1:
POSTS HIGHEST EVER REVENUE FORQ4
Foxconn sales beat
view on Al demand
REUTERS
Taipei, January 5
TAIWAN'S FOXCONN, THE
‘world’s largest contract elec-
tronics maker, beat expecta-
tions to post its highest-ever
revenue forthe fourth quar-
ter on continued strong
demand for artifical intelli
gence (Al) servers.
Revenue for Apple'sbiggest
iphone assembler jumped
15.2% to T$2.13 trillion
($64.72 billion), Foxconn said
inastatementon Sunday.
Itwasalso ahead of aT $2.1
tuillion LSEG SmartEstimate,
which gives greater weight to
forecasts from analysts who
aremoreconsistentlyaccurate.
Robust Al server demand
led to strong revenue growth
forits cloud and networking
products division, said Fox:
conn, whose customers
include Al chip firm Nvidia,
For smart consumer elec-
tronics, which includes
iphones, there was “roughly
```

```
flattish” year-on-year growth,
itsaid,
‘Totalrevenuein December
alone reached T$ 654.8 billion,
uup42.3%yearonyearandthe
second-highest ever level for

STRONG SHOW
mRevenve for Apple's ere
iPhoneassemblerjumped
‘For smart
15.2% to $64.72 billion — Ryn aes

‘the month,
“In the first quarter of
2025,overall operations have
gradually entered the tradi
tional off-season,” Foxconn
said of its outlook for the cur-
rent quarter.
“Even with record high rev-
‘enue in the fourth quarter of
2024, the sequential perfor-
```

```
from transformers import pipeline


summarizer = pipeline("summarization", model="facebook/bart-large-cnn", device=-1)
```

```
/usr/local/lib/python3.11/dist-packages/huggingface_hub/utils/_auth.py:94: UserWarning:
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab (https://huggingface.co/settings/tokens), set it as secre
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access public models or datasets.
  warnings.warn(
```

config.json: 100%	1.58k/1.58k [00:00<00:00, 128kB/s]
model.safetensors: 100%	1.63G/1.63G [00:11<00:00, 204MB/s]
generation_config.json: 100%	363/363 [00:00<00:00, 41.2kB/s]
vocab.json: 100%	899k/899k [00:00<00:00, 3.95MB/s]
merges.txt: 100%	456k/456k [00:00<00:00, 1.95MB/s]
tokenizer.json: 100%	1.36M/1.36M [00:00<00:00, 1.58MB/s]

```
Device set to use cpu
```

```
import torch
torch.cuda.empty_cache()
torch.cuda.reset_peak_memory_stats()
```

```
from transformers import pipeline
from tqdm import tqdm
import torch

torch.cuda.empty_cache()

summarizer = pipeline("summarization", model="facebook/bart-large-cnn", device=-1)

def summarize_extracted_texts(text_dict, batch_size=5):
    summarized_data = {}

    print("\n📖 Summarizing extracted text in batches of", batch_size, "...\n")

    filenames = list(text_dict.keys())
    texts = list(text_dict.values())

    for i in tqdm(range(0, len(texts), batch_size), desc="Summarizing Batches"):
        batch_filenames = filenames[i:i + batch_size]
        batch_texts = [text[:1500] for text in texts[i:i + batch_size]]

        try:
            summaries = summarizer(batch_texts, max_length=80, min_length=30, do_sample=False)
            for j, filename in enumerate(batch_filenames):
                summarized_data[filename] = summaries[j]['summary_text']
```

```
        except Exception as e:
            print(f"⚠️ Error summarizing batch {i}-{i + batch_size}: {e}")
            for filename in batch_filenames:
                summarized_data[filename] = "Summarization failed."

    return summarized_data

summarized_texts = summarize_extracted_texts(extracted_texts)

for filename, summary in list(summarized_texts.items())[:5]:
    print(f"\n📰 {filename}: {summary}\n")
```

```
Device set to use cpu

📖 Summarizing extracted text in batches of 5 ...

Summarizing Batches:   3%|▌          | 2/60 [01:48<51:58, 53.77s/it]Your max_length is set to 80, but your input_length is only 79. Sinc
Summarizing Batches:  15%|▌          | 9/60 [08:35<50:17, 59.17s/it]Your max_length is set to 80, but your input_length is only 57. Since
Summarizing Batches: 100%|▌          | 60/60 [59:50<00:00, 59.84s/it]
📰 Screenshot 2025-02-10 154322.png: Pralay is India’s first quasi-ballistic missile. It will be showcased at the Republic Day parade i


📰 Screenshot 2025-02-10 154042.png: The Indian Navy's INS Sar-                vekshak completed the fi-                nal phase of th


📰 Screenshot 2025-02-10 153903.png: BSF to hold talks with BGB in Delhi from February 16 to 20. Fencing of border meticulouslyareas, t


📰 Screenshot 2025-02-10 154510.png: Israel announced on Friday that the withdrawal ofits forces from southern Lebanon would continue b


📰 Screenshot 2025-02-10 153653.png: The United Nations on Sunday told its staff not to go to the airport and to stay in place. The int
```

```
summarized_file = "/content/summarized_texts.txt"

with open(summarized_file, "w", encoding="utf-8") as f:
    for filename, summary in summarized_texts.items():
        f.write(f"{filename}:\n{summary}\n\n")

from google.colab import files
files.download(summarized_file)
```

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
import os
import pytesseract
import re
from PIL import Image

image_folder = "/content/drive/My Drive/Delta"


cleaned_extracted_text_file_path = "/content/drive/My Drive/extracted_texts_indexed_cleaned.txt"

def clean_text(text):
    text = text.strip()
    text = re.sub(r'\n+', '\n', text)
    text = re.sub(r'\s+', ' ', text)
    text = re.sub(r'[^\w\s.,!?-]', '', text)
    return text


extracted_texts = {}
index = 1

for filename in sorted(os.listdir(image_folder)):
```

```
    if filename.lower().endswith(('.png', '.jpg', '.jpeg')):
        image_path = os.path.join(image_folder, filename)

        try:
            img = Image.open(image_path)
            raw_text = pytesseract.image_to_string(img)
            cleaned_text = clean_text(raw_text)
            extracted_texts[index] = (filename, cleaned_text)
            index += 1
        except Exception as e:
            print(f"⚠️ Error processing {filename}: {e}")

print("✅ Extracted Text Indexed & Cleaned Successfully!")
```

```
✅ Extracted Text Indexed & Cleaned Successfully!
```

```
with open(cleaned_extracted_text_file_path, "w", encoding="utf-8") as f:
    for idx, (filename, text) in extracted_texts.items():
        f.write(f"{idx}. Image: {filename}\n   Extracted Text: {text}\n\n")

print(f"✅ Cleaned extracted text file saved at: {cleaned_extracted_text_file_path}")
```

```
✅ Cleaned extracted text file saved at: /content/drive/My Drive/extracted_texts_indexed_cleaned.txt
```

```
from google.colab import files
files.download(cleaned_extracted_text_file_path)
```