```python
import os
import numpy as np
import cv2
import torch
from PIL import Image
from tqdm.notebook import tqdm
from transformers import TrOCRProcessor, VisionEncoderDecoderModel
import jiwer
```

```python
current_dir = os.getcwd()
dataset_image_path = os.path.join(current_dir, "Data", "image")  # Images on which OCR is to be performer
dataset_label_path = os.path.join(current_dir, "Data", "label")  # Labels of the images to evaluate the output
CRAFT_path = os.path.join(current_dir, "CRAFT-pytorch")  # Path to CRAFT-pytorch
CRAFT_text_file_path = os.path.join(CRAFT_path, "test.py")  # Path to CRAFT-pytorch's text.py for coordinates generation
CRAFT_weights_path = os.path.join(CRAFT_path, "craft_mlt_25k.pth")  # Weights of CRAFT-pytorch
bboxes_path = os.path.join(CRAFT_path, "result")  # Dir where CRAFT-pytorch saves the output

print("Dataset Image Path:", dataset_image_path)
print("Dataset Label Path:", dataset_label_path)
print("CRAFT Path:", CRAFT_path)
print("CRAFT Script Path:", CRAFT_text_file_path)
print("CRAFT Weights Path:", CRAFT_weights_path)
print("BBoxes Path:", bboxes_path)

device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
print(f"Using device: {device}")
```

```
Dataset Image Path: C:\Users\shobh\Python_Stuff\OCR\Data\image
Dataset Label Path: C:\Users\shobh\Python_Stuff\OCR\Data\label
CRAFT Path: C:\Users\shobh\Python_Stuff\OCR\CRAFT-pytorch
CRAFT Script Path: C:\Users\shobh\Python_Stuff\OCR\CRAFT-pytorch\test.py
CRAFT Weights Path: C:\Users\shobh\Python_Stuff\OCR\CRAFT-pytorch\craft_mlt_25k.pth
BBoxes Path: C:\Users\shobh\Python_Stuff\OCR\CRAFT-pytorch\result
Using device: cuda
```

```python
os.chdir(CRAFT_path)
print(os.getcwd())
```

```
C:\Users\shobh\Python_Stuff\OCR\CRAFT-pytorch
```

```python
!python "{CRAFT_text_file_path}" --trained_model="{CRAFT_weights_path}" --test_folder="{dataset_image_path}"
```

```
Loading weights from checkpoint (C:\Users\shobh\Python_Stuff\OCR\CRAFT-pytorch\craft_mlt_25k.pth)
Test image 1/18: C:\Users\shobh\Python_Stuff\OCR\Data\image\Buendia_1.png
Test image 2/18: C:\Users\shobh\Python_Stuff\OCR\Data\image\Buendia_2.png
Test image 3/18: C:\Users\shobh\Python_Stuff\OCR\Data\image\Buendia_3.png
Test image 4/18: C:\Users\shobh\Python_Stuff\OCR\Data\image\Constituciones_sinodales_calahorra_1.png
Test image 5/18: C:\Users\shobh\Python_Stuff\OCR\Data\image\Constituciones_sinodales_calahorra_2.png
Test image 6/18: C:\Users\shobh\Python_Stuff\OCR\Data\image\Constituciones_sinodales_calahorra_3.png
Test image 7/18: C:\Users\shobh\Python_Stuff\OCR\Data\image\Ezcaray_Vozes_1.png
Test image 8/18: C:\Users\shobh\Python_Stuff\OCR\Data\image\Ezcaray_Vozes_2.png
Test image 9/18: C:\Users\shobh\Python_Stuff\OCR\Data\image\Ezcaray_Vozes_3.png
Test image 10/18: C:\Users\shobh\Python_Stuff\OCR\Data\image\Mendo_Principe_perfecto_1.png
Test image 11/18: C:\Users\shobh\Python_Stuff\OCR\Data\image\Mendo_Principe_perfecto_2.png
Test image 12/18: C:\Users\shobh\Python_Stuff\OCR\Data\image\Mendo_Principe_perfecto_3.png
Test image 13/18: C:\Users\shobh\Python_Stuff\OCR\Data\image\Paredes_Reglas_Generales_1.png
Test image 14/18: C:\Users\shobh\Python_Stuff\OCR\Data\image\Paredes_Reglas_Generales_2.png
Test image 15/18: C:\Users\shobh\Python_Stuff\OCR\Data\image\Paredes_Reglas_Generales_3.png
Test image 16/18: C:\Users\shobh\Python_Stuff\OCR\Data\image\Porcones_1.png
Test image 17/18: C:\Users\shobh\Python_Stuff\OCR\Data\image\Porcones_2.png
Test image 18/18: C:\Users\shobh\Python_Stuff\OCR\Data\image\Porcones_3.png
elapsed time : 11.483039140701294s
E:\Anaconda\Lib\site-packages\torchvision\models\_utils.py:208: UserWarning: The parameter 'pretrained' is deprecated since 0.13 and may
  warnings.warn(
E:\Anaconda\Lib\site-packages\torchvision\models\_utils.py:223: UserWarning: Arguments other than a weight enum or `None` for 'weights'
  warnings.warn(msg)
C:\Users\shobh\Python_Stuff\OCR\CRAFT-pytorch\test.py:128: FutureWarning: You are using `torch.load` with `weights_only=False` (the curr
  net.load_state_dict(copyStateDict(torch.load(args.trained_model)))
```

```python
processor = TrOCRProcessor.from_pretrained("qantev/trocr-large-spanish")
model = VisionEncoderDecoderModel.from_pretrained("qantev/trocr-large-spanish").to(device)
model.eval()
```

```
preprocessor_config.json:  100%                                        364/364  [00:00<?,  ?B/s]
E:\Anaconda\Lib\site-packages\huggingface_hub\file_download.py:142: UserWarning: `huggingface_hub` cache-system uses symlinks by default
To support symlinks on Windows, you either need to activate Developer Mode or to run Python as an administrator. In order to activate de
  warnings.warn(message)
Using a slow image processor as `use_fast` is unset and a slow processor was saved with this model. `use_fast=True` will be the default
tokenizer_config.json:  100%                                        1.38k/1.38k  [00:00<00:00,  119kB/s]
vocab.json:  100%                                        798k/798k  [00:00<00:00,  1.27MB/s]
merges.txt:  100%                                        456k/456k  [00:00<00:00,  28.8MB/s]
tokenizer.json:  100%                                        2.11M/2.11M  [00:00<00:00,  8.93MB/s]
special_tokens_map.json:  100%                                        957/957  [00:00<00:00,  239kB/s]
config.json:  100%                                        4.97k/4.97k  [00:00<00:00,  314kB/s]
pytorch_model.bin:  100%                                        2.44G/2.44G  [01:38<00:00,  25.6MB/s]
model.safetensors:  100%                                        2.44G/2.44G  [01:41<00:00,  23.7MB/s]
Config of the encoder: <class 'transformers.models.vit.modeling_vit.ViTModel'> is overwritten by shared encoder config: ViTConfig {
  "attention_probs_dropout_prob": 0.0,
  "encoder_stride": 16,
  "hidden_act": "gelu",
  "hidden_dropout_prob": 0.0,
  "hidden_size": 1024,
  "image_size": 384,
  "initializer_range": 0.02,
  "intermediate_size": 4096,
  "layer_norm_eps": 1e-12,
  "model_type": "vit",
  "num_attention_heads": 16,
  "num_channels": 3,
  "num_hidden_layers": 24,
  "patch_size": 16,
  "qkv_bias": false,
  "torch_dtype": "float32",
  "transformers_version": "4.49.0"
}

Config of the decoder: <class 'transformers.models.trocr.modeling_trocr.TrOCRForCausalLM'> is overwritten by shared decoder config: TrOC
  "activation_dropout": 0.0,
  "activation_function": "relu",
  "add_cross_attention": true,
  "attention_dropout": 0.0,
  "bos_token_id": 0,
  "classifier_dropout": 0.0,
  "d_model": 1024,
  "decoder_attention_heads": 16,
  "decoder_ffn_dim": 4096,
  "decoder_layerdrop": 0.0,
  "decoder_layers": 12,
  "decoder_start_token_id": 2,
  "dropout": 0.1,
  "encoder_hidden_size": 1024,
  "eos_token_id": 2,
  "init_std": 0.02,
  "is_decoder": true,
  "layernorm_embedding": false,
  "max_position_embeddings": 1024,
  "model_type": "trocr",
  "pad_token_id": 1,
  "scale_embedding": true,
  "tie_word_embeddings": false,
  "torch_dtype": "float32",
  "transformers_version": "4.49.0",
  "use_cache": false,
  "use_learned_position_embeddings": false,
  "vocab_size": 50265
}

generation_config.json:  100%                                        420/420  [00:00<00:00,  39.6kB/s]
VisionEncoderDecoderModel(
  (encoder): ViTModel(
    (embeddings): ViTEmbeddings(
      (patch_embeddings): ViTPatchEmbeddings(
        (projection): Conv2d(3, 1024, kernel_size=(16, 16), stride=(16, 16))
      )
      (dropout): Dropout(p=0.0, inplace=False)
    )
    (encoder): ViTEncoder(
      (layer): ModuleList(
        (0-23): 24 x ViTLayer(
          (attention): ViTSdpaAttention(
            (attention): ViTSdpaSelfAttention(
              (query): Linear(in_features=1024, out_features=1024, bias=False)
              (key): Linear(in_features=1024, out_features=1024, bias=False)
              (value): Linear(in_features=1024, out_features=1024, bias=False)
              (dropout): Dropout(p=0.0, inplace=False)
            )
            (output): ViTSelfOutput(
              (dense): Linear(in_features=1024, out_features=1024, bias=True)
              (dropout): Dropout(p=0.0, inplace=False)
            )
          )
          (intermediate): ViTIntermediate(
            (dense): Linear(in_features=1024, out_features=4096, bias=True)
            (intermediate_act_fn): GELUActivation()
          )
          (output): ViTOutput(
            (dense): Linear(in_features=4096, out_features=1024, bias=True)
            (dropout): Dropout(p=0.0, inplace=False)
          )
          (layernorm_before): LayerNorm((1024,), eps=1e-12, elementwise_affine=True)
          (layernorm_after): LayerNorm((1024,), eps=1e-12, elementwise_affine=True)
        )
      )
    )
    (layernorm): LayerNorm((1024,), eps=1e-12, elementwise_affine=True)
    (pooler): ViTPooler(
      (dense): Linear(in_features=1024, out_features=1024, bias=True)
      (activation): Tanh()
    )
  )
  (decoder): TrOCRForCausalLM(
    (model): TrOCRDecoderWrapper(
      (decoder): TrOCRDecoder(
        (embed_tokens): TrOCRScaledWordEmbedding(50265, 1024, padding_idx=1)
        (embed_positions): TrOCRSinusoidalPositionalEmbedding()
        (layers): ModuleList(
          (0-11): 12 x TrOCRDecoderLayer(
            (self_attn): TrOCRAttention(
              (k_proj): Linear(in_features=1024, out_features=1024, bias=True)
              (v_proj): Linear(in_features=1024, out_features=1024, bias=True)
              (q_proj): Linear(in_features=1024, out_features=1024, bias=True)
              (out_proj): Linear(in_features=1024, out_features=1024, bias=True)
            )
            (activation_fn): ReLU()
            (self_attn_layer_norm): LayerNorm((1024,), eps=1e-05, elementwise_affine=True)
            (encoder_attn): TrOCRAttention(
              (k_proj): Linear(in_features=1024, out_features=1024, bias=True)
              (v_proj): Linear(in_features=1024, out_features=1024, bias=True)
              (q_proj): Linear(in_features=1024, out_features=1024, bias=True)
              (out_proj): Linear(in_features=1024, out_features=1024, bias=True)
            )
            (encoder_attn_layer_norm): LayerNorm((1024,), eps=1e-05, elementwise_affine=True)
            (fc1): Linear(in_features=1024, out_features=4096, bias=True)
            (fc2): Linear(in_features=4096, out_features=1024, bias=True)
            (final_layer_norm): LayerNorm((1024,), eps=1e-05, elementwise_affine=True)
          )
        )
      )
    )
    (output_projection): Linear(in_features=1024, out_features=50265, bias=False)
  )
)
```

```python
def get_bboxes(image_name):
    name = image_name.split(".")[0]
    bboxes_file_name = "res_" + name + ".txt"
    bboxes_file_path = os.path.join(bboxes_path, bboxes_file_name)

    bboxes_raw = open(bboxes_file_path, 'r', encoding="utf-8").read()
    bboxes_strings = bboxes_raw.split("\n\n")

    if bboxes_strings[-1] =="":
        bboxes_strings.pop()

    bboxes = []
    for bboxes_string in bboxes_strings:
        bboxes_string_split = bboxes_string.split(",")
        bboxes.append((int(bboxes_string_split[0]), int(bboxes_string_split[1]), int(bboxes_string_split[4]), int(bboxes_string_split[5

    return bboxes


def merge_bounding_boxes(bboxes, x_thresh=150, y_thresh=10):
    if not bboxes:
        return []

    bboxes = np.array(bboxes)
    bboxes = bboxes[bboxes[:, 1].argsort()]
    merged = []
    visited = np.zeros(len(bboxes), dtype=bool)  # Track visited boxes

    def find_cluster(idx):
        cluster = [idx]
        x1, y1, x2, y2 = bboxes[idx]
        x_center, y_center = (x1 + x2) / 2, (y1 + y2) / 2

        for j in range(len(bboxes)):
            if j != idx and not visited[j]:
                x1_j, y1_j, x2_j, y2_j = bboxes[j]
                x_center_j, y_center_j = (x1_j + x2_j) / 2, (y1_j + y2_j) / 2

                dx = abs(x_center - x_center_j)
                dy = abs(y_center - y_center_j)

                if dx <= x_thresh and dy <= y_thresh:
                    visited[j] = True
                    cluster.extend(find_cluster(j))

        return cluster

    for i in range(len(bboxes)):
        if not visited[i]:
            visited[i] = True
            cluster = find_cluster(i)
            merged_x1 = np.min(bboxes[cluster, 0])
            merged_y1 = np.min(bboxes[cluster, 1])
            merged_x2 = np.max(bboxes[cluster, 2])
            merged_y2 = np.max(bboxes[cluster, 3])
            merged.append((merged_x1, merged_y1, merged_x2, merged_y2))

    return merged



def ordered_bounding_boxes(bboxes, x_thresh=200, y_thresh=100):
    if not bboxes:
        return []

    bboxes = np.array(bboxes)
    ordered_bboxes = []
    visited = np.zeros(len(bboxes), dtype=bool)

    def find_cluster(idx):
        cluster = [idx]
        x1, y1, x2, y2 = bboxes[idx]
        x_center, y_center = (x1 + x2) / 2, (y1 + y2) / 2

        for j in range(len(bboxes)):
            if j != idx and not visited[j]:
                x1_j, y1_j, x2_j, y2_j = bboxes[j]
                x_center_j, y_center_j = (x1_j + x2_j) / 2, (y1_j + y2_j) / 2

                dx = abs(x_center - x_center_j)
                dy = abs(y_center - y_center_j)

                if dx <= x_thresh and dy <= y_thresh:
                    visited[j] = True
                    cluster.extend(find_cluster(j))

        return cluster

    side_1 = []
    side_2 = []
    for i in range(len(bboxes)):
        if not visited[i]:
            visited[i] = True
            cluster = find_cluster(i)
            if len(side_1) == 0:
                for c in cluster:
                    x1, y1, x2, y2 = bboxes[c]
                    side_1.append((x1, y1, x2, y2))
            else:
                for c in cluster:
                    x1, y1, x2, y2 = bboxes[c]
                    side_2.append((x1, y1, x2, y2))

    if len(side_2) != 0 and side_1[0][0] < side_2[0][0]:
        ordered_bboxes.extend(side_1)
        ordered_bboxes.extend(side_2)

    else:
        ordered_bboxes.extend(side_2)
        ordered_bboxes.extend(side_1)

    return ordered_bboxes
```

```
for x in os.walk(dataset_image_path):
    image_names = x[2]

print(image_names)
```

```
['Buendia_1.png', 'Buendia_2.png', 'Buendia_3.png', 'Constituciones_sinodales_calahorra_1.png', 'Constituciones_sinodales_calahorra_2.pn
```

```
predicted_text = []
ground_truth = []
for image_name in tqdm(image_names, desc="Images", unit="images"):

    bboxes = get_bboxes(image_name=image_name)
    merged_bboxes = merge_bounding_boxes(bboxes)
    ordered_bboxes = ordered_bounding_boxes(merged_bboxes)

    image_path = os.path.join(dataset_image_path, image_name)
    image = cv2.imread(image_path)

    label_name = image_name.split('.')[0] + ".txt"
    label_path = os.path.join(dataset_label_path, label_name)
    label = open(label_path, 'r', encoding="utf-8").read()
    label = label.split('\n')
    ground_truth.append(' '.join(label).lower())

    ocr_text = []

    for i, (x1, y1, x2, y2) in tqdm(enumerate(ordered_bboxes), unit="bboxes", desc="bboxes", total=len(ordered_bboxes)):
        cropped = image[y1:y2, x1:x2]

        cropped_pil = Image.fromarray(cv2.cvtColor(cropped, cv2.COLOR_BGR2RGB))

        pixel_values = processor(images=cropped_pil, return_tensors="pt").pixel_values.to(device)

        with torch.no_grad():
            generated_ids = model.generate(pixel_values)

        recognized_text = processor.batch_decode(generated_ids, skip_special_tokens=True)[0]

        ocr_text.append(recognized_text)

    predicted_text.append(" ".join(ocr_text).lower())
```

```
Images: 100%                                    18/18 [05:09<00:00, 20.77s/images]
bboxes: 100%                                    27/27 [00:11<00:00,  2.34bboxes/s]
E:\Anaconda\Lib\site-packages\transformers\generation\utils.py:1532: UserWarning: You have modified the pretrained model configuration t
  warnings.warn(
bboxes: 100%                                    60/60 [00:25<00:00,  2.85bboxes/s]
bboxes: 100%                                    34/34 [00:15<00:00,  2.32bboxes/s]
bboxes: 100%                                    39/39 [00:19<00:00,  2.22bboxes/s]
bboxes: 100%                                    36/36 [00:17<00:00,  1.93bboxes/s]
bboxes: 100%                                    42/42 [00:20<00:00,  2.32bboxes/s]
bboxes: 100%                                    26/26 [00:09<00:00,  3.70bboxes/s]
bboxes: 100%                                    25/25 [00:10<00:00,  2.91bboxes/s]
bboxes: 100%                                    29/29 [00:11<00:00,  2.95bboxes/s]
bboxes: 100%                                    27/27 [00:11<00:00,  1.90bboxes/s]
bboxes: 100%                                    29/29 [00:13<00:00,  2.56bboxes/s]
bboxes: 100%                                    35/35 [00:18<00:00,  2.10bboxes/s]
bboxes: 100%                                    33/33 [00:12<00:00,  2.09bboxes/s]
bboxes: 100%                                    40/40 [00:19<00:00,  2.36bboxes/s]
bboxes: 100%                                    43/43 [00:22<00:00,  2.72bboxes/s]
bboxes: 100%                                    56/56 [00:26<00:00,  1.84bboxes/s]
```

```
predicted_text[1]
```

```
bboxes: 100%                                    40/40 [00:20<00:00,  2.68bboxes/s]
'guro diffeño de fu edad :  la reli- gion para con dios en la devora alsiftécia à los templos;la piedad con los padres en la
obediencia más rendida;  la modetta, y de feo  de  faber», con los mayores, guitando más  de  oir, y pregun 24%. ibid.    car, que de
definir, y refolver. que efto en vueftra infinita sabi- duria fue foberana dignacion,  y en la natural  ignorancia de los niños es
indifpenfable necesi dad. ni tienen folamente en vos el daffeño, la luz, y el exemplo, fino también  el amor, y protec- pjal-114-6,
cion.   vos, º 118-13°    tro dé los niños, les daís ent mattb.19.   dimiento, y comunicais la labi- 14.    «duría. vós les promereis
el reyn marci, 10.  de los cielos", y os  indignais con 14°    quien les aparta de vos, y les matt. 18.  proponeis por norma  del can-
2.ºc.    « dór», inocencia», mildad. vueftro amor parece que no pudo explicarfe más tierno, y liberal con los niños,  pues no contento
de echarles vueftras di divi- divinás bendiciones, les unifteis à vuestro fagrado pecho con fue vifsimos abrazos.  dichola edad,
marci. que os mereció tan regalados ca-    16º riño\'s y pues en la  celebial jeru- falén no ha mudado de condicion vueftra  benignidad
o  niño  tierno, y  dios eterno, profeguid, profeguid en bendecirles, y favo- recerles.  sean tan fervorolamen te devotos de vueftra
admirable   cant 8.1 madre, que le porten como fus hijos, y hermanos de leche  con vos.  serán fabios, fi fueren café    54). 1 tos ;
que no entra vueltra  sabi- duría, donde no ay mucha pure- za de  conciencia.   crezcan en vueftro lanto temór, y amor, co- como en los
años, y mucho  más. adelantente en la virtud,  como en las letras, y mucho más ; haf-  ad epbé ta que lleguen,  por vueltra imi-
tacion, à fer varones perfectos, y   confumados vueftros ojos, y provechofos à agradables    �  la república,  que libra café- da fu
felizidad en  la  acerrada crian-'
```

```
ground_truth[1]
```

```
'guro disseño de su edad: la reli- gion para con dios en la devota assistencia a los templos; la piedad con los padres en la obediencia
mas rendida; y la modestia, y de- seo de saber, con los mayores,  gustando mas de oir, y pregun- tar, que de definir, y resolver. bien
que esto en vuestra infinita sabi- duria fue soberana dignacion, y en la natural ignorancia de los niños es indispensable necessi- dad.
ni tienen solamente en vos  el disseño, la luz, y el exemplo,  sino tambien el amor, y protec- cion. vos, como singular maes- tro de
los niños, les dais enten- dimiento, y comunicais la sabi- duria. vos les prometeis el reyno de los cielos, y os indignais con quien
les aparta de vos, y les proponeis por norma del can- dor, inocencia, y christiana hu- mildad. vuestro amor parece que no pudo
explicarse mas tierno, y liberal con los niños, pues no contento de echarles vuestras di- vinas bendiciones, les unisteis a vuestro
sagrado pecho con sua- vissimos abrazos. dichosa edad,  que os merecio tan regalados cariños! y pues en la celestial jeru- salen no ha
mudado de condicion vuestra benignidad, proseguid,  o niño tierno, y dios eterno,  proseguid en bendecirles, y favo- recerles. sean tan
fervorosamen- te devotos de vuestra admirable madre, que se porten como sus hijos, y hermanos de leche con vos. seran sabios, si fueren
cas- tos; que no entra vuestra sabi- duria, donde no ay mucha pure- za de conciencia. crezcan en vuestro santo temor, y amor, co- como
en los años, y mucho mas.  adelantense en la virtud, como en las letras, y mucho mas; has- ta que lleguen, por vuesetra imi- tacion, a
ser varones perfectos,  y consumados, agradables a vuestros ojos, y provechosos a la republica, que libra casi to- da su felizidad en
la acertada '
```

```python
cer = jiwer.cer(ground_truth, predicted_text)
wer = jiwer.wer(ground_truth, predicted_text)
```

```python
books = set()
for image_name in image_names:
    img = image_name.split("_")
    books.add(img[0])

books = sorted(list(books))
```

```python
print(f"CER: {cer*100: .2f}% , WER: {wer*100: .2f}%")
```

```
CER:  36.76% , WER:  65.24%
```

```python
i = 0
for n in range(0, len(predicted_text), 3):
    cer = jiwer.cer(ground_truth[n:n+3], predicted_text[n:n+3])
    wer = jiwer.wer(ground_truth[n:n+3], predicted_text[n:n+3])
    print(f"Book: {books[i]}, CER: {cer*100: .2f}% , WER: {wer*100: .2f}%")
    i +=1
```

```
Book: Buendia, CER:  20.51% , WER:  42.20%
Book: Constituciones, CER:  40.37% , WER:  72.86%
Book: Ezcaray, CER:  19.03% , WER:  37.05%
Book: Mendo, CER:  38.86% , WER:  70.52%
Book: Paredes, CER:  36.90% , WER:  67.36%
```