HTRやってみたver3

データ作成コード

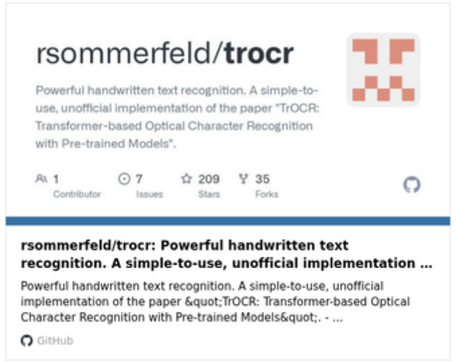
```python
import csv, math, pathlib, random
from typing import List
import numpy as np
from PIL import Image, ImageDraw, ImageFont, ImageOps

# 設定定数
INPUT_FILE = "datadummy/gutenberg/sentences.txt"
OUTPUT_ROOT = "datadummy/synth_gutenberg"
FONT_DIR = "./fonts"
FONT_SIZE = 48
IMG_H = 64
MARGIN = 10
MAX_ROT = 2.0
NOISE = 5.0
TRAIN_RATIO = 0.9

def list_fonts(font_dir: pathlib.Path, size: int) -> List[ImageFont.FreeTypeFont]:
    return [ImageFont.truetype(str(fp), size=size) for fp in font_dir.glob("*.ttf")]

def render(text: str, font: ImageFont.FreeTypeFont,
           img_h: int, margin: int) -> Image.Image:
    bbox = font.getbbox(text)
    w = bbox[2] - bbox[0]  # width
    canvas = Image.new("L", (w + margin * 2, img_h), "white")
    draw = ImageDraw.Draw(canvas)
    draw.text((margin, (img_h - font.size) // 2), text, font=font, fill="black")
    return canvas

def augment(img: Image.Image, max_rot: float, noise_std: float) -> Image.Image:
    # 回転
    angle = random.uniform(-max_rot, max_rot)
    img = img.rotate(angle, expand=1, fillcolor="white")
    # ガウシアンノイズ
    if noise_std > 0:
        arr = np.array(img, dtype=np.float32)
        arr += np.random.normal(scale=noise_std, size=arr.shape)
        img = Image.fromarray(np.clip(arr, 0, 255).astype(np.uint8))
    # トリミング（前後余白を詰める）
    img = ImageOps.crop(img, border=1)
    return img
```

```python
def main():
    root = pathlib.Path(OUTPUT_ROOT)
    train_dir = root / "train" / "img"
    val_dir   = root / "val"   / "img"
    for d in (train_dir, val_dir):
        d.mkdir(parents=True, exist_ok=True)

    # フォントロード
    fonts = list_fonts(pathlib.Path(FONT_DIR), FONT_SIZE)
    if not fonts:
        raise RuntimeError(f"No TTF fonts found in {FONT_DIR}")

    # コーパス読み込み & シャッフル
    sentences = pathlib.Path(INPUT_FILE).read_text(encoding="utf-8").splitlines()
    random.shuffle(sentences)
    n_train = int(len(sentences) * TRAIN_RATIO)

    def dump(split_name, img_dir, rows):
        label_path = img_dir.parent / "labels.tsv"
        with label_path.open("w", encoding="utf-8", newline="") as tsv:
            writer = csv.writer(tsv, delimiter="\t")
            for idx, text in enumerate(rows):
                font = random.choice(fonts)
                img  = render(text, font, IMG_H, MARGIN)
                img  = augment(img, MAX_ROT, NOISE)
                fname = f"{idx:06d}.png"
                img.save(img_dir / fname)
                writer.writerow([fname, text])

    dump("train", train_dir, sentences[:n_train])
    dump("val",   val_dir,   sentences[n_train:])

    print(f"Generated {n_train:,} train and {len(sentences)-n_train:,} val images → {root}")

if __name__ == "__main__":
    main()
```

モデル

本家の事前学習済みのモデルなかったっぽ.....すべてなし
そして、動かすことに失敗。。。諦め

そこで、HFの事前学習済みモデルを用いてすぐに動かせそうなcode発見！！とういことでこれを動かしてみた。



結果

I took the class from 10:00 to 12:40 , worked from 13:00 to 18:00, and then worked another job from 18:30 to 20:20 .

location

While I waited for her to finish the work , I read an English vocabulary book .Afterwards, we went to the station .

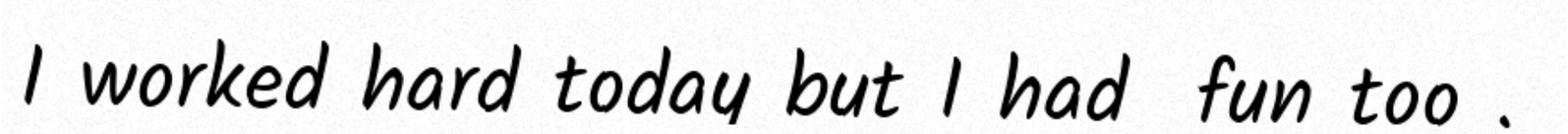
At the second part-time job ,my student didn't come, so I finished earlier than usual, but I was disappointed .

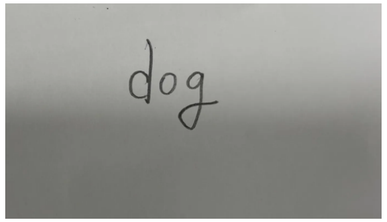
station

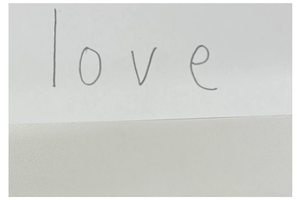
I worked two part-time jobs and took a class today .

	column 1	column 2	column 3
1	image_path	prediction	confidence
2	test\000000.png	I took the classification 1000 to 1240, worked from 1300 to 1800, and then worked	0.6758796572685242
3	test\000001.png	location	0.7301119565963745
4	test\000002.png	While I waited for her to finish the work, I read an English vocabularybook Afterwards,	0.994128942489624
5	test\000003.png	At the second part-time job my student didn't come, so I finished earlier than	0.3744376003742218
6	test\000004.png	station	0.3993540406227112
7	test\000005.png	I worked two part-time jobs and took a class today.	0.8442443013191223

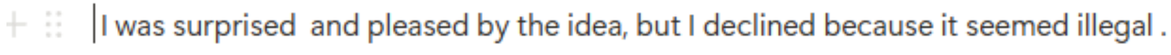
結果







I have not yet received any information about the intern from LINE Yahoo yet .I have been waiting for three weeks . This company is known for offering the most popular internship for aspiring engineers .



8	test\000006.png	love.	0.0556899793446064
9	test\000007.png	dog	0.02518794685602188
10	test\000008.png	I worked hard today but I had fun too.	0.04577196389436722
11	test\image.png	# the greatest relationship in its emotion to the time in the nation's UN Cabinet which have been	0.2551206946372986
12	test\image0.png	30thoge 2	4.2775418478413485e-06
13	test\image1.png	30 May	1.3765670701104682e-06
14	test\image2.png	love.	0.07344535738229752
15	test\imagee.png	# I was surprised and pleased by the idea but I declined because it seemed illegal...	0.27938058972358704

現状のコード理解状況

実行コマンド：python -m src train

```python
from .cli import cli

main = cli

if __name__ == "__main__":
    main()
```

__main__.py

```python
import typer
from pathlib import Path

from .main import TrocrPredictor, main_train, main_validate

cli = typer.Typer(name="TrOCR")


@cli.command()
def train(local_model: bool = False):
    main_train(local_model)


@cli.command()
def validate(local_model: bool = True):
    main_validate(local_model)


@cli.command()
def predict(image_paths: list[str], local_model: bool = True):
    # コマンドライン引数から画像パスを受け取り、予測を行う
    predictions = TrocrPredictor(local_model).predict_for_image_paths(image_paths)
    for path, (prediction, confidence) in zip(image_paths, predictions):
        print(f"Path:\t\t{path}\nPrediction:\t{prediction}\nConfidence:\t{confidence}\n")
```

cli.py

現状のコード理解状況

```python
def main_train(use_local_model: bool = False):
    processor = load_processor()
    train_dataset = HCRDataset(paths.train_dir, processor)
    train_dataloader = DataLoader(train_dataset, constants.batch_size, shuffle=True, num_workers=constants.num_workers)

    val_dataset = HCRDataset(paths.val_dir, processor)
    val_dataloader = DataLoader(val_dataset, constants.batch_size, num_workers=constants.num_workers)

    model = load_model(use_local_model)
    init_model_for_training(model, processor)

    context = Context(model, processor, train_dataset, train_dataloader, val_dataset, val_dataloader)
    train(context, constants.train_epochs)
    debug_print(f"Saving model to {paths.model_path}...")
    model.save_pretrained(paths.model_path)
```

main.py(一部抜粋)

今後の展望

もう少し、このコードの理解を進める
入力画像にが学習用データに近くなるようなデータ前処理を入れる？？