```
!pip install --upgrade pip
!pip install torch torchvision torchaudio
!pip install fsspec==2024.6.1
!pip install datasets==3.0.0
!pip install gcsfs==2024.6.0
!pip install jiwer
!pip install evaluate
```

```
Requirement already satisfied: pip in /usr/local/lib/python3.10/dist-packages (24.1.2)
Collecting pip
  Downloading pip-24.3.1-py3-none-any.whl.metadata (3.7 kB)
Downloading pip-24.3.1-py3-none-any.whl (1.8 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.8/1.8 MB 20.1 MB/s eta 0:00:00
Installing collected packages: pip
  Attempting uninstall: pip
    Found existing installation: pip 24.1.2
    Uninstalling pip-24.1.2:
      Successfully uninstalled pip-24.1.2
Successfully installed pip-24.3.1
Requirement already satisfied: torch in /usr/local/lib/python3.10/dist-packages (2.5.1+cu121)
Requirement already satisfied: torchvision in /usr/local/lib/python3.10/dist-packages (0.20.1+cu121)
Requirement already satisfied: torchaudio in /usr/local/lib/python3.10/dist-packages (2.5.1+cu121)
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from torch) (3.16.1)
Requirement already satisfied: typing-extensions>=4.8.0 in /usr/local/lib/python3.10/dist-packages (from torch) (4.12.2)
Requirement already satisfied: networkx in /usr/local/lib/python3.10/dist-packages (from torch) (3.4.2)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.10/dist-packages (from torch) (3.1.4)
Requirement already satisfied: fsspec in /usr/local/lib/python3.10/dist-packages (from torch) (2024.10.0)
Requirement already satisfied: sympy==1.13.1 in /usr/local/lib/python3.10/dist-packages (from torch) (1.13.1)
Requirement already satisfied: mpmath<1.4,>=1.1.0 in /usr/local/lib/python3.10/dist-packages (from sympy==1.13.1->torch) (1.3.0)
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (from torchvision) (1.26.4)
Requirement already satisfied: pillow!=8.3.*,>=5.3.0 in /usr/local/lib/python3.10/dist-packages (from torchvision) (11.0.0)
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.10/dist-packages (from jinja2->torch) (3.0.2)
Collecting fsspec==2024.6.1
  Downloading fsspec-2024.6.1-py3-none-any.whl.metadata (11 kB)
Downloading fsspec-2024.6.1-py3-none-any.whl (177 kB)
Installing collected packages: fsspec
  Attempting uninstall: fsspec
    Found existing installation: fsspec 2024.10.0
    Uninstalling fsspec-2024.10.0:
      Successfully uninstalled fsspec-2024.10.0
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is the sourc
gcsfs 2024.10.0 requires fsspec==2024.10.0, but you have fsspec 2024.6.1 which is incompatible.
Successfully installed fsspec-2024.6.1
Collecting datasets==3.0.0
  Downloading datasets-3.0.0-py3-none-any.whl.metadata (19 kB)
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from datasets==3.0.0) (3.16.1)
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.10/dist-packages (from datasets==3.0.0) (1.26.4)
Requirement already satisfied: pyarrow>=15.0.0 in /usr/local/lib/python3.10/dist-packages (from datasets==3.0.0) (17.0.0)
Collecting dill<0.3.9,>=0.3.0 (from datasets==3.0.0)
  Downloading dill-0.3.8-py3-none-any.whl.metadata (10 kB)
Requirement already satisfied: pandas in /usr/local/lib/python3.10/dist-packages (from datasets==3.0.0) (2.2.2)
Requirement already satisfied: requests>=2.32.2 in /usr/local/lib/python3.10/dist-packages (from datasets==3.0.0) (2.32.3)
Requirement already satisfied: tqdm>=4.66.3 in /usr/local/lib/python3.10/dist-packages (from datasets==3.0.0) (4.66.6)
Collecting xxhash (from datasets==3.0.0)
  Downloading xxhash-3.5.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (12 kB)
Collecting multiprocess (from datasets==3.0.0)
  Downloading multiprocess-0.70.17-py310-none-any.whl.metadata (7.2 kB)
Requirement already satisfied: fsspec<=2024.6.1,>=2023.1.0 in /usr/local/lib/python3.10/dist-packages (from fsspec[http]<=2024.6.1,>=2
Requirement already satisfied: aiohttp in /usr/local/lib/python3.10/dist-packages (from datasets==3.0.0) (3.11.9)
Requirement already satisfied: huggingface-hub>=0.22.0 in /usr/local/lib/python3.10/dist-packages (from datasets==3.0.0) (0.26.3)
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-packages (from datasets==3.0.0) (24.2)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.10/dist-packages (from datasets==3.0.0) (6.0.2)
Requirement already satisfied: aiohappyeyeballs>=2.3.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets==3.0.0) (2.4
Requirement already satisfied: aiosignal>=1.1.2 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets==3.0.0) (1.3.1)
Requirement already satisfied: async-timeout<6.0,>=4.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets==3.0.0) (4.0
Requirement already satisfied: attrs>=17.3.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets==3.0.0) (24.2.0)
```

```
from google.colab import drive
import os, sys, itertools
os.environ['TOKENIZERS_PARALLELISM']='false'

import pandas as pd
from sklearn.model_selection import train_test_split

from PIL import Image

import torch
from torch.utils.data import Dataset

import datasets
from datasets import load_dataset

import transformers
from transformers import Seq2SeqTrainingArguments, Seq2SeqTrainer
from transformers import VisionEncoderDecoderModel, TrOCRProcessor, default_data_collator

import evaluate
```

```
print("Python:".rjust(15), sys.version[0:6])
print("Pandas:".rjust(15), pd.__version__)
print("Datasets:".rjust(15), datasets.__version__)
print("Transformers:".rjust(15), transformers.__version__)
print("Torch:".rjust(15), torch.__version__)
```

```
            Python: 3.10.1
            Pandas: 2.2.2
          Datasets: 3.0.0
      Transformers: 4.46.3
             Torch: 2.5.1+cu121
```

```
# Mount Google Drive
drive.mount('/content/drive', force_remount=True)
# Path to your dataset directory
path = '/content/drive/My Drive/CMPE 252 Project/whiteplate_normal/'

# Create the DataFrame
file_names = []
texts = []

# Loop through the directory and extract file names and labels
for file in os.listdir(path):
    if file.endswith(('.jpg', '.png')):  # Adjust extensions as per your dataset
        file_names.append(file)
        # Extract license plate from the file name (assuming the file name is the plate)
        texts.append(os.path.splitext(file)[0])  # Remove file extension

# Create a DataFrame
dataset = pd.DataFrame({'file_name': file_names, 'text': texts})

# Train/test split
train_dataset, test_dataset = train_test_split(dataset, train_size=0.80, random_state=42)

# Reset indices
train_dataset.reset_index(drop=True, inplace=True)
test_dataset.reset_index(drop=True, inplace=True)

# Print dataset information
print("Train Dataset Info:")
print(train_dataset.info())
print("\nTest Dataset Info:")
print(test_dataset.info())
```

```
Mounted at /content/drive
Train Dataset Info:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9624 entries, 0 to 9623
Data columns (total 2 columns):
 #   Column     Non-Null Count  Dtype 
---  ------     --------------  ----- 
 0   file_name  9624 non-null   object
 1   text       9624 non-null   object
dtypes: object(2)
memory usage: 150.5+ KB
None

Test Dataset Info:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2406 entries, 0 to 2405
Data columns (total 2 columns):
 #   Column     Non-Null Count  Dtype 
---  ------     --------------  ----- 
 0   file_name  2406 non-null   object
 1   text       2406 non-null   object
dtypes: object(2)
memory usage: 37.7+ KB
None
```

```
train_dataset.head(12)
```

	file_name	text
0	OX08CBB.png	OX08CBB
1	SN40FFE.png	SN40FFE
2	RH66ZDD.png	RH66ZDD
3	TT37EVC.png	TT37EVC
4	FX07EUZ.png	FX07EUZ
5	NO34ABY.png	NO34ABY
6	ZV30IMV.png	ZV30IMV

7	JP78IEZ.png	JP78IEZ
8	YX68TFU.png	YX68TFU
9	EV90FEX.png	EV90FEX
10	JF24RRQ.png	JF24RRQ
11	NQ61HUW.png	NQ61HUW

Next steps: Generate code with `train_dataset` | View recommended plots | New interactive sheet

```
class License_Plates_OCR_Dataset(Dataset):

    def __init__(self, root_dir, df, processor, max_target_length=128):
        self.root_dir = root_dir
        self.df = df
        self.processor = processor
        self.max_target_length = max_target_length

    def __len__(self):
        return len(self.df)

    def __getitem__(self, idx):
        # get file name + text
        file_name = self.df['file_name'][idx]
        text = self.df['text'][idx]
        # prepare image (i.e. resize + normalize)
        image = Image.open(self.root_dir + file_name).convert("RGB")
        pixel_values = self.processor(image, return_tensors="pt").pixel_values
        # add labels (input_ids) by encoding the text
        labels = self.processor.tokenizer(text, padding="max_length", max_length=self.max_target_length).input_ids
        # important: make sure that PAD tokens are ignored by the loss function
        labels = [label if label != self.processor.tokenizer.pad_token_id
                  else -100 for label in labels]

        encoding = {"pixel_values" : pixel_values.squeeze(), "labels" : torch.tensor(labels)}
        return encoding
```

```
MODEL_CKPT = "microsoft/trocr-base-printed"
MODEL_NAME =  MODEL_CKPT.split("/")[-1] + "_license_plates_ocr"
NUM_OF_EPOCHS = 2
```

```
# Initialize the processor
processor = TrOCRProcessor.from_pretrained(MODEL_CKPT)

# Define the License_Plates_OCR_Dataset class (assuming it's implemented elsewhere)
# root_dir is now set to the dataset_path, and df is passed for train and test datasets

train_ds = License_Plates_OCR_Dataset(
    root_dir=path,
    df=train_dataset,
    processor=processor
)

test_ds = License_Plates_OCR_Dataset(
    root_dir=path,
    df=test_dataset,
    processor=processor
)
```

```
/usr/local/lib/python3.10/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 sec
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(
preprocessor_config.json: 100% 224/224 [00:00<00:00, 18.8kB/s]
tokenizer_config.json: 100% 1.12k/1.12k [00:00<00:00, 96.4kB/s]
vocab.json: 100% 899k/899k [00:00<00:00, 12.7MB/s]
merges.txt: 100% 456k/456k [00:00<00:00, 35.2MB/s]
special_tokens_map.json: 100% 772/772 [00:00<00:00, 70.3kB/s]
```

```
print(f"The training dataset has {len(train_ds)} samples in it.")
print(f"The testing dataset has {len(test_ds)} samples in it.")
```

```
The training dataset has 9624 samples in it.
The testing dataset has 2406 samples in it.
```

```
encoding = train_ds[0]

for k,v in encoding.items():
    print(k, " : ", v.shape)
```

```
pixel_values  :  torch.Size([3, 384, 384])
labels  :  torch.Size([128])
```

```
image = Image.open(train_ds.root_dir + train_dataset['file_name'][0]).convert("RGB")

image
```


```
labels = encoding['labels']
labels[labels == -100] = processor.tokenizer.pad_token_id
label_str = processor.decode(labels, skip_special_tokens=True)
print(label_str)
```

```
OX08CBB
```

```
model = VisionEncoderDecoderModel.from_pretrained(MODEL_CKPT)
```

```
config.json: 100% 4.13k/4.13k [00:00<00:00, 359kB/s]
model.safetensors: 100% 1.33G/1.33G [00:06<00:00, 223MB/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": 768,
  "image_size": 384,
  "initializer_range": 0.02,
  "intermediate_size": 3072,
  "layer_norm_eps": 1e-12,
  "model_type": "vit",
  "num_attention_heads": 12,
  "num_channels": 3,
  "num_hidden_layers": 12,
  "patch_size": 16,
  "qkv_bias": false,
  "transformers_version": "4.46.3"
}

Config of the decoder: <class 'transformers.models.trocr.modeling_trocr.TrOCRForCausalLM'> is overwritten by shared decoder config: Tr
  "activation_dropout": 0.0,
  "activation_function": "gelu",
  "add_cross_attention": true,
  "attention_dropout": 0.0,
  "bos_token_id": 0,
  "classifier_dropout": 0.0,
  "cross_attention_hidden_size": 768,
  "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,
  "eos_token_id": 2,
  "init_std": 0.02,
  "is_decoder": true,
  "layernorm_embedding": true,
  "max_position_embeddings": 512,
  "model_type": "trocr",
  "pad_token_id": 1,
  "scale_embedding": false,
  "transformers_version": "4.46.3",
  "use_cache": false,
  "use_learned_position_embeddings": true,
  "vocab_size": 50265
}

Some weights of VisionEncoderDecoderModel were not initialized from the model checkpoint at microsoft/trocr-base-printed and are newly
You should probably TRAIN this model on a down-stream task to be able to use it for predictions and inference.
```

generation_config.json: 100% 190/190 [00:00<00:00, 17.7kB/s]

```
model.config.decoder_start_token_id = processor.tokenizer.cls_token_id
model.config.pad_token_id = processor.tokenizer.pad_token_id

model.config.vocab_size = model.config.decoder.vocab_size

model.config.eos_token_id = processor.tokenizer.sep_token_id
model.config.max_length = 64
model.config.early_stopping = True
model.config.no_repeat_ngram_size = 3
model.config.length_penalty = 2.0
model.config.num_beams = 4
```

```
cer_metric = evaluate.load("cer")

def compute_metrics(pred):
    label_ids = pred.label_ids
    pred_ids = pred.predictions

    pred_str = processor.batch_decode(pred_ids, skip_special_tokens=True)
    label_ids[label_ids == -100] = processor.tokenizer.pad_token_id
    label_str = processor.batch_decode(label_ids, skip_special_tokens=True)

    cer = cer_metric.compute(predictions=pred_str, references=label_str)

    return {"cer" : cer}
```

Downloading builder script: 100% 5.60k/5.60k [00:00<00:00, 463kB/s]

```
args = Seq2SeqTrainingArguments(
    output_dir = MODEL_NAME,
    num_train_epochs=NUM_OF_EPOCHS,
    predict_with_generate=True,
    evaluation_strategy="epoch",
    save_strategy="epoch",
    per_device_train_batch_size=8,
    per_device_eval_batch_size=8,
    logging_first_step=True,
    hub_private_repo=True,
    push_to_hub=True
)
```

```
/usr/local/lib/python3.10/dist-packages/transformers/training_args.py:1568: FutureWarning: `evaluation_strategy` is deprecated and wil
  warnings.warn(
```

testingLicensePlate

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

```
!huggingface-cli login
```

```
    _|    _|  _|    _|    _|_|_|    _|_|_|  _|_|_|  _|      _|    _|_|_|      _|_|_|_|    _|_|      _|_|_|  _|_|_|_|
    _|    _|  _|    _|  _|        _|          _|    _|_|    _|  _|            _|        _|    _|  _|        _|
    _|_|_|_|  _|    _|  _|  _|_|  _|  _|_|    _|    _|  _|  _|  _|  _|_|      _|_|_|    _|_|_|_|  _|        _|_|_|
    _|    _|  _|    _|  _|    _|  _|    _|    _|    _|    _|_|  _|    _|      _|        _|    _|  _|        _|
    _|    _|    _|_|      _|_|_|    _|_|_|  _|_|_|  _|      _|    _|_|_|      _|        _|    _|    _|_|_|  _|_|_|_|

    To log in, `huggingface_hub` requires a token generated from https://huggingface.co/settings/tokens .
Enter your token (input will not be visible):
Add token as git credential? (Y/n) n
Token is valid (permission: fineGrained).
The token `testingLicensePlate` has been saved to /root/.cache/huggingface/stored_tokens
Your token has been saved to /root/.cache/huggingface/token
Login successful.
The current active token is: `testingLicensePlate`
```

```
from transformers import Seq2SeqTrainer, Seq2SeqTrainingArguments

# Update args for Seq2SeqTrainer
args = Seq2SeqTrainingArguments(
    output_dir="./results",
    eval_strategy="epoch",  # Changed from `evaluation_strategy` to `eval_strategy`
    learning_rate=5e-5,
    per_device_train_batch_size=16,
    per_device_eval_batch_size=16,
    weight_decay=0.01,
    save_total_limit=3
```

```
    save_total_limit=3,
    num_train_epochs=NUM_OF_EPOCHS,
    predict_with_generate=True,
    logging_dir="./logs",
    logging_strategy="epoch",
)

# Initialize the Trainer
trainer = Seq2SeqTrainer(
    model=model,
    processing_class=processor,  # Updated to use `processing_class`
    args=args,
    compute_metrics=compute_metrics,
    train_dataset=train_ds,
    eval_dataset=test_ds,
    data_collator=default_data_collator
)
```

```
train_results = trainer.train()
```

```
wandb: WARNING The `run_name` is currently set to the same value as `TrainingArguments.output_dir`. If this was not intended, please sp
wandb: Using wandb-core as the SDK backend.  Please refer to https://wandb.me/wandb-core for more information.
wandb: Logging into wandb.ai. (Learn how to deploy a W&B server locally: https://wandb.me/wandb-server)
wandb: You can find your API key in your browser here: https://wandb.ai/authorize
wandb: Paste an API key from your profile and hit enter, or press ctrl+c to quit: ··········
wandb: Appending key for api.wandb.ai to your netrc file: /root/.netrc
```

Tracking run with wandb version 0.18.7
Run data is saved locally in `/content/wandb/run-20241206_050641-2ghzllh0`
Syncing run **./results** to Weights & Biases (docs)
View project at https://wandb.ai/sofia-silva-san-jose-state-university/huggingface
View run at https://wandb.ai/sofia-silva-san-jose-state-university/huggingface/runs/2ghzllh0

[1204/1204 27:36, Epoch 2/2]

Epoch	Training Loss	Validation Loss	Cer
1	0.179400	0.066785	0.004684
2	0.028600	0.029214	0.001779

```
/usr/local/lib/python3.10/dist-packages/transformers/modeling_utils.py:2817: UserWarning: Moving the following attributes in the confi
  warnings.warn(
Trainer.tokenizer is now deprecated. You should use Trainer.processing_class instead.
Trainer.tokenizer is now deprecated. You should use Trainer.processing_class instead.
Trainer.tokenizer is now deprecated. You should use Trainer.processing_class instead.
Trainer.tokenizer is now deprecated. You should use Trainer.processing_class instead.
Trainer.tokenizer is now deprecated. You should use Trainer.processing_class instead.
Trainer.tokenizer is now deprecated. You should use Trainer.processing_class instead.
Trainer.tokenizer is now deprecated. You should use Trainer.processing_class instead.
Trainer.tokenizer is now deprecated. You should use Trainer.processing_class instead.
Trainer.tokenizer is now deprecated. You should use Trainer.processing_class instead.
Trainer.tokenizer is now deprecated. You should use Trainer.processing_class instead.
Trainer.tokenizer is now deprecated. You should use Trainer.processing_class instead.
Trainer.tokenizer is now deprecated. You should use Trainer.processing_class instead.
Trainer.tokenizer is now deprecated. You should use Trainer.processing_class instead.
Trainer.tokenizer is now deprecated. You should use Trainer.processing_class instead.
Trainer.tokenizer is now deprecated. You should use Trainer.processing_class instead.
Trainer.tokenizer is now deprecated. You should use Trainer.processing_class instead.
Trainer.tokenizer is now deprecated. You should use Trainer.processing_class instead.
Trainer.tokenizer is now deprecated. You should use Trainer.processing_class instead.
Trainer.tokenizer is now deprecated. You should use Trainer.processing_class instead.
Trainer.tokenizer is now deprecated. You should use Trainer.processing_class instead.
Trainer.tokenizer is now deprecated. You should use Trainer.processing_class instead.
Trainer.tokenizer is now deprecated. You should use Trainer.processing_class instead.
Trainer.tokenizer is now deprecated. You should use Trainer.processing_class instead.
Trainer.tokenizer is now deprecated. You should use Trainer.processing_class instead.
Trainer.tokenizer is now deprecated. You should use Trainer.processing_class instead.
Trainer.tokenizer is now deprecated. You should use Trainer.processing_class instead.
Trainer.tokenizer is now deprecated. You should use Trainer.processing_class instead.
Trainer.tokenizer is now deprecated. You should use Trainer.processing_class instead.
Trainer.tokenizer is now deprecated. You should use Trainer.processing_class instead.
Trainer.tokenizer is now deprecated. You should use Trainer.processing_class instead.
Trainer.tokenizer is now deprecated. You should use Trainer.processing_class instead.
Trainer.tokenizer is now deprecated. You should use Trainer.processing_class instead.
Trainer.tokenizer is now deprecated. You should use Trainer.processing_class instead.
Trainer.tokenizer is now deprecated. You should use Trainer.processing_class instead.
Trainer.tokenizer is now deprecated. You should use Trainer.processing_class instead.
Trainer.tokenizer is now deprecated. You should use Trainer.processing_class instead.
Trainer.tokenizer is now deprecated. You should use Trainer.processing_class instead.
Trainer.tokenizer is now deprecated. You should use Trainer.processing_class instead.
Trainer.tokenizer is now deprecated. You should use Trainer.processing_class instead.
Trainer.tokenizer is now deprecated. You should use Trainer.processing_class instead.
Trainer.tokenizer is now deprecated. You should use Trainer.processing_class instead.
Trainer.tokenizer is now deprecated. You should use Trainer.processing_class instead.
Trainer.tokenizer is now deprecated. You should use Trainer.processing_class instead.
Trainer.tokenizer is now deprecated. You should use Trainer.processing_class instead.
Trainer.tokenizer is now deprecated. You should use Trainer.processing_class instead.
Trainer.tokenizer is now deprecated. You should use Trainer.processing_class instead.
Trainer.tokenizer is now deprecated. You should use Trainer.processing_class instead.
```

Trainer.tokenizer is now deprecated. You should use Trainer.processing_class instead.

```
trainer.save_model()
trainer.log_metrics("train", train_results.metrics)
trainer.save_metrics("train", train_results.metrics)
trainer.save_state()
```

```
     ***** train metrics *****
       epoch                    =           2.0
       total_flos               = 13413832854GF
       train_loss               =         0.104
       train_runtime            =    0:28:00.20
       train_samples_per_second =        11.456
       train_steps_per_second   =         0.717
```

```
metrics = trainer.evaluate()
trainer.log_metrics("eval", metrics)
trainer.save_metrics("eval", metrics)
```