humanai-final-proposal-notebook

April 2, 2024

For my test, I developed three models based on encoder-decoder architecture. In the first model, I trained a simple encoder-decoder model from scratch on a small dataset and observed good performance. In the second model, I use pre-trained vision and language transformer models, employing the HuggingFace VisionEncoderDecoderModel class to fine-tune the models on a small test dataset. Lastly, to meet the requirement of 80% accuracy, I fine-tuned a pre-trained version of TrOCR.

-

1 Sections

1.1 Section 1: Prepare Dataset

1.2 Section 2: Train

1.2.1 Section 2.1 Training Encoder Decoder Model From Sctrach

1.2.2 Section 2.2 Create Transformers Based Model From Pretrained Encoder Decoder Models

1.2.3 Section 2.3: Using Pre Trained TrOCR

1.3 Section 3: Evaluation and Result

2 What Value I Can Add to Project

2.0.1 Create LoRA modules for each font

In the paper titled "**Combining OCR Models for Reading Early Modern Printed Books" by Seuret, Mathias et al.**, it is noted that **OCR performance is significantly affected by font style**. The authors found that **selecting fine-tuned models with font group recognition greatly improves the results**.

LoRA (Low-Rank Adaptation of Large Language Models) is a PEFT (Parameter-Efficient Fine-Tuning) technique that allows for the fine-tuning of modules on transformer models without changing the base model parameters. This technique can also be applied to vision transformers. **By using LoRA, modules can fine-tune for each font while utilizing base model adaptation with minimal computational power**. As part of the team responsible for creating the vision transformers course for **Hugging Face**, I have developed a course on using LoRA to fine-tune vision transformer models. While it is not officially released, you can access it here: johko/notebooks/Unit 3 - Vision Transformers/LoRA-Image-Classification.

2.0.2 Postprocessing with NLP techniques

NLP techniques can use for postprocessing. After applying OCR to a document, NLP models like BERT can be used to correct errors in the OCR output **by predicting the correct words within the context of the surrounding text**. BERT could use at understanding the context of a word in a sentence. It can provide replacements for words that might have been misrecognized by the OCR system 1 2. I have taken the CENG 3526 Natural Language Processing course and can develop a system that utilizes such techniques.

2.1 Set-up environment

[ ]:
```
! pip install -U -q accelerate
! pip install -U -q transformers
! pip install -q datasets jiwer
```

```
                                290.1/290.1
kB 6.3 MB/s eta 0:00:00
                                23.7/23.7 MB
33.4 MB/s eta 0:00:00
                                823.6/823.6
kB 48.1 MB/s eta 0:00:00
                                14.1/14.1 MB
53.2 MB/s eta 0:00:00
                                731.7/731.7
MB 705.8 kB/s eta 0:00:00
                                410.6/410.6
MB 2.9 MB/s eta 0:00:00
                                121.6/121.6
MB 9.6 MB/s eta 0:00:00
                                56.5/56.5 MB
10.2 MB/s eta 0:00:00
                                124.2/124.2
MB 3.4 MB/s eta 0:00:00
                                196.0/196.0
MB 3.0 MB/s eta 0:00:00
                                166.0/166.0
MB 4.8 MB/s eta 0:00:00
                                99.1/99.1 kB
11.3 MB/s eta 0:00:00
                                21.1/21.1 MB
69.4 MB/s eta 0:00:00
                                8.8/8.8 MB
22.1 MB/s eta 0:00:00
```

```
                                    510.5/510.5
kB 7.6 MB/s eta 0:00:00
                                    116.3/116.3
kB 9.8 MB/s eta 0:00:00
                                    194.1/194.1
kB 9.8 MB/s eta 0:00:00
                                    134.8/134.8
kB 8.1 MB/s eta 0:00:00
                                    3.4/3.4 MB
18.9 MB/s eta 0:00:00
```

Import necessery librarys

[2]:
```
import os
import pandas as pd
from sklearn.model_selection import train_test_split
import torch
from torch.utils.data import Dataset
from PIL import Image
from transformers import TrOCRProcessor
from transformers import VisionEncoderDecoderModel
from transformers import Seq2SeqTrainer, Seq2SeqTrainingArguments

os.environ["KERAS_BACKEND"] = "tensorflow"
import re
import numpy as np
import matplotlib.pyplot as plt

import tensorflow as tf
import keras
from keras import layers
from keras.applications import efficientnet
from keras.layers import TextVectorization

keras.utils.set_random_seed(111)

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

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call
drive.mount("/content/drive", force_remount=True).
```

3 Section 1 Prepare Dataset

To prepare the data for training, I utilize a synthetic data generator designed for printed documants. The library i use for this task is TextRecognitionDataGenerator which is the same library used to pretrain trocr on synthetic text. I search for public dataset. One of the dataset i found is Spanish Redonda (Round Script) 16th-17th Century. Although this dataset isn't publicly available, I contacted the author and they agreed to share if it is for research purposes, not for commercial use.

3.0.1 Font

Given the limited computational power, it is necessary to fine-tune TROCR using fonts that are similar. I use online font identification websites such as myfonts and whatfontisthis to find the most similar fonts. Once identified, I download these fonts from OnlineWebFonts.

Also disturtion, random_skew and random_blur added for preparing model to diffent edge cases

Example image to identify fonts

[ ]:
```
!pip install -q trdg
```

```
                                98.6/98.6 MB
7.7 MB/s eta 0:00:00
  Preparing metadata (setup.py) … done
  Preparing metadata (setup.py) … done
WARNING: The candidate selected for download or install is a yanked
version: 'arabic-reshaper' candidate (version 2.1.3 at https://files.pythonhoste
d.org/packages/47/27/7b9b824f5342d8ee180027333f2e15842ea36f5bc2d3d24a4e6bb31fb59
6/arabic_reshaper-2.1.3-py3-none-any.whl (from https://pypi.org/simple/arabic-
reshaper/))
Reason for being yanked: Doesn't work with Python 2
  Building wheel for diffimg (setup.py) … done
  Building wheel for wikipedia (setup.py) … done
```

[ ]:
```
!trdg -c 9000 -d 3 -w 1 -f 64 -l es --random_skew --random_blur -fd /content/
↪drive/MyDrive/ocr/fonts -dt /content/drive/MyDrive/ocr/dictionary/ep_es.txt␣
↪--output_dir /content/drive/MyDrive/ocr/13k_es_mixed_lib

!trdg -c 4000 -d 3 -w 1 -f 64 -l es --random_skew --random_blur -fd /content/
↪drive/MyDrive/ocr/fonts -dt /content/drive/MyDrive/ocr/dictionary/ep_es.txt␣
↪--output_dir /content/drive/MyDrive/ocr/13k_es_mixed_lib
```

```
2024-03-24 12:59:38.963521: E
external/local_xla/xla/stream_executor/cuda/cuda_dnn.cc:9261] Unable to register
```

```
cuDNN factory: Attempting to register factory for plugin cuDNN when one has
already been registered
2024-03-24 12:59:38.963577: E
external/local_xla/xla/stream_executor/cuda/cuda_fft.cc:607] Unable to register
cuFFT factory: Attempting to register factory for plugin cuFFT when one has
already been registered
2024-03-24 12:59:38.965277: E
external/local_xla/xla/stream_executor/cuda/cuda_blas.cc:1515] Unable to
register cuBLAS factory: Attempting to register factory for plugin cuBLAS when
one has already been registered
2024-03-24 12:59:40.104396: W
tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38] TF-TRT Warning: Could not
find TensorRT
100% 9000/9000 [03:35<00:00, 41.67it/s]
2024-03-24 13:03:21.902078: E
external/local_xla/xla/stream_executor/cuda/cuda_dnn.cc:9261] Unable to register
cuDNN factory: Attempting to register factory for plugin cuDNN when one has
already been registered
2024-03-24 13:03:21.902141: E
external/local_xla/xla/stream_executor/cuda/cuda_fft.cc:607] Unable to register
cuFFT factory: Attempting to register factory for plugin cuFFT when one has
already been registered
2024-03-24 13:03:21.904802: E
external/local_xla/xla/stream_executor/cuda/cuda_blas.cc:1515] Unable to
register cuBLAS factory: Attempting to register factory for plugin cuBLAS when
one has already been registered
2024-03-24 13:03:23.490270: W
tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38] TF-TRT Warning: Could not
find TensorRT
100% 4000/4000 [01:27<00:00, 45.52it/s]
```

[ ]:
```
dataset_directory ="/content/drive/MyDrive/ocr/13k_es_mixed_lib"
model_save_folder = "/content/drive/MyDrive/trocr_es13k_finetune"
len(os.listdir("/content/drive/MyDrive/ocr/13k_es_mixed_lib"))
```

[ ]:
```
12999
```

3.1 Prepare data for fine tuning

[ ]:
```
file_names = []
texts = []

for file_name in os.listdir(dataset_directory):
    text = file_name.split('_')[0].replace('-', ' ')
    with open(os.path.join(dataset_directory, file_name), 'r') as file:
        file_names.append(file_name)
        texts.append(file_name.split('_')[0])
```

```
# Create a DataFrame from the lists
df = pd.DataFrame({'file_name': file_names, 'text': texts})

df
```

[ ]:

	file_name	text
0	parecido_3000.jpg	parecido
1	christiana_3001.jpg	christiana
2	alguna_3002.jpg	alguna
3	aviendose_3003.jpg	aviendose
4	buscando_3004.jpg	buscando
…	…	…
12994	mismas_994.jpg	mismas
12995	van_995.jpg	van
12996	mismas_996.jpg	mismas
12997	possible_997.jpg	possible
12998	ocultó_998.jpg	ocultó

[12999 rows x 2 columns]

[ ]:
```
sub_df = df.iloc[:500]
train_df, test_df = train_test_split(sub_df, test_size=0.2)
train_df.reset_index(drop=True, inplace=True)
test_df.reset_index(drop=True, inplace=True)
```

3.1.1 Preparing data for training model from scratch

[14]:
```
def train_val_split(ocr_data, train_size=0.8, shuffle=True):
    all_images = list(ocr_data.keys())

    if shuffle:
        np.random.shuffle(all_images)

    train_size = int(len(ocr_data) * train_size)
    training_data = {img_name: ocr_data[img_name] for img_name in all_images[:
 ↪train_size]}
    validation_data = {img_name: ocr_data[img_name] for img_name in␣
 ↪all_images[train_size:]}

    return training_data, validation_data
```

[17]:
```
path = "/content/drive/MyDrive/ocr/ocr_clean_test_10k"
ocr_map = {}
vocab = []
vocab_set = set()
```

```
for i, dir in enumerate(os.listdir(path)[:2500]):
    if not dir.endswith('.jpg'):
        continue

    cap = "<start> " + dir.split("_")[0] + " <end>"
    dir_path = os.path.join(path, dir)
    ocr_map[dir_path] = cap
    vocab.append(cap)
    vocab_set.add(dir.split("_")[0])

train_data, valid_data = train_val_split(ocr_map)
VOCAB_SIZE = len(vocab_set)
text_data = list(vocab_set)
```

4 Section 2 Training Models

4.0.1 Section 2.1 Training Encoder Decoder Model From Scratch

Parameters Setup

[13]:
```
IMAGE_SIZE = (60, 120)
VOCAB_SIZE = 0
SEQ_LENGTH = 3
EMBED_DIM = 512
FF_DIM = 512
BATCH_SIZE = 1024
EPOCHS = 140
AUTOTUNE = tf.data.AUTOTUNE
```

Vectorizing the text data

Use the TextVectorization layer to vectorize the text data to turn the original strings into integer sequences where each integer represents the index of a word in a vocabulary.

[18]:
```
# Vectorizing the text data
def custom_standardization(input_string):
    lowercase = tf.strings.lower(input_string)
    return tf.strings.regex_replace(lowercase, "[%s]" % re.escape(strip_chars),␣
 ↪"")


strip_chars = "!\"#$%&'()*+,-./:;<=>?@[\]^_`{|}~"
strip_chars = strip_chars.replace("<", "")
strip_chars = strip_chars.replace(">", "")

vectorization = TextVectorization(
    max_tokens=VOCAB_SIZE,
```

```
    output_mode="int",
    output_sequence_length=SEQ_LENGTH,
    standardize=custom_standardization,
)
vectorization.adapt(text_data)
```

Building a Dataset pipeline for training

[19]:
```
def decode_and_resize(img_path):
    img = tf.io.read_file(img_path)
    img = tf.image.decode_jpeg(img, channels=3)
    img = tf.image.resize(img, IMAGE_SIZE)
    img = tf.image.convert_image_dtype(img, tf.float32)
    return img


def process_input(img_path, ocrs):
    return decode_and_resize(img_path), vectorization(ocrs)


def make_dataset(images, ocrs):
    dataset = tf.data.Dataset.from_tensor_slices((images, ocrs))
    dataset = dataset.shuffle(BATCH_SIZE * 8)
    dataset = dataset.map(process_input, num_parallel_calls=AUTOTUNE)
    dataset = dataset.batch(BATCH_SIZE).prefetch(AUTOTUNE)

    return dataset

train_dataset = make_dataset(list(train_data.keys()), list(train_data.values()))
valid_dataset = make_dataset(list(valid_data.keys()), list(valid_data.values()))
```

Building the model The model consist of three part:

CNN : Used to extract the image features

TransformerEncoder: The extracted image features are then passed to a Transformer based encoder that generates a new representation of the inputs

TransformerDecoder: This model takes the encoder output and the text data as inputs and tries to learn to generate the caption.

[20]:
```
def get_cnn_model():
    base_model = efficientnet.EfficientNetB0(
        input_shape=(*IMAGE_SIZE, 3),
        include_top=False,
        weights="imagenet",
    )
    base_model.trainable = False
    base_model_out = base_model.output
```

```
    base_model_out = layers.Reshape((-1, base_model_out.
 ↪shape[-1]))(base_model_out)
    cnn_model = keras.models.Model(base_model.input, base_model_out)
    return cnn_model


class TransformerEncoderBlock(layers.Layer):
    def __init__(self, embed_dim, dense_dim, num_heads, **kwargs):
        super().__init__(**kwargs)
        self.embed_dim = embed_dim
        self.dense_dim = dense_dim
        self.num_heads = num_heads
        self.attention_1 = layers.MultiHeadAttention(
            num_heads=num_heads, key_dim=embed_dim, dropout=0.0
        )
        self.layernorm_1 = layers.LayerNormalization()
        self.layernorm_2 = layers.LayerNormalization()
        self.dense_1 = layers.Dense(embed_dim, activation="relu")

    def call(self, inputs, training, mask=None):
        inputs = self.layernorm_1(inputs)
        inputs = self.dense_1(inputs)

        attention_output_1 = self.attention_1(
            query=inputs,
            value=inputs,
            key=inputs,
            attention_mask=None,
            training=training,
        )
        out_1 = self.layernorm_2(inputs + attention_output_1)
        return out_1


class PositionalEmbedding(layers.Layer):
    def __init__(self, sequence_length, vocab_size, embed_dim, **kwargs):
        super().__init__(**kwargs)
        self.token_embeddings = layers.Embedding(
            input_dim=vocab_size, output_dim=embed_dim
        )
        self.position_embeddings = layers.Embedding(
            input_dim=sequence_length, output_dim=embed_dim
        )
        self.sequence_length = sequence_length
        self.vocab_size = vocab_size
        self.embed_dim = embed_dim
        self.embed_scale = tf.math.sqrt(tf.cast(embed_dim, tf.float32))
```

```
    def call(self, inputs):
        length = tf.shape(inputs)[-1]
        positions = tf.range(start=0, limit=length, delta=1)
        embedded_tokens = self.token_embeddings(inputs)
        embedded_tokens = embedded_tokens * self.embed_scale
        embedded_positions = self.position_embeddings(positions)
        return embedded_tokens + embedded_positions

    def compute_mask(self, inputs, mask=None):
        return tf.math.not_equal(inputs, 0)


class TransformerDecoderBlock(layers.Layer):
    def __init__(self, embed_dim, ff_dim, num_heads, **kwargs):
        super().__init__(**kwargs)
        self.embed_dim = embed_dim
        self.ff_dim = ff_dim
        self.num_heads = num_heads
        self.attention_1 = layers.MultiHeadAttention(
            num_heads=num_heads, key_dim=embed_dim, dropout=0.1
        )
        self.attention_2 = layers.MultiHeadAttention(
            num_heads=num_heads, key_dim=embed_dim, dropout=0.1
        )
        self.ffn_layer_1 = layers.Dense(ff_dim, activation="relu")
        self.ffn_layer_2 = layers.Dense(embed_dim)

        self.layernorm_1 = layers.LayerNormalization()
        self.layernorm_2 = layers.LayerNormalization()
        self.layernorm_3 = layers.LayerNormalization()

        self.embedding = PositionalEmbedding(
            embed_dim=EMBED_DIM,
            sequence_length=SEQ_LENGTH,
            vocab_size=VOCAB_SIZE,
        )
        self.out = layers.Dense(VOCAB_SIZE, activation="softmax")

        self.dropout_1 = layers.Dropout(0.3)
        self.dropout_2 = layers.Dropout(0.5)
        self.supports_masking = True

    def call(self, inputs, encoder_outputs, training, mask=None):
        inputs = self.embedding(inputs)
        causal_mask = self.get_causal_attention_mask(inputs)
```

```
        if mask is not None:
            padding_mask = tf.cast(mask[:, :, tf.newaxis], dtype=tf.int32)
            combined_mask = tf.cast(mask[:, tf.newaxis, :], dtype=tf.int32)
            combined_mask = tf.minimum(combined_mask, causal_mask)

        attention_output_1 = self.attention_1(
            query=inputs,
            value=inputs,
            key=inputs,
            attention_mask=combined_mask,
            training=training,
        )
        out_1 = self.layernorm_1(inputs + attention_output_1)

        attention_output_2 = self.attention_2(
            query=out_1,
            value=encoder_outputs,
            key=encoder_outputs,
            attention_mask=padding_mask,
            training=training,
        )
        out_2 = self.layernorm_2(out_1 + attention_output_2)

        ffn_out = self.ffn_layer_1(out_2)
        ffn_out = self.dropout_1(ffn_out, training=training)
        ffn_out = self.ffn_layer_2(ffn_out)

        ffn_out = self.layernorm_3(ffn_out + out_2, training=training)
        ffn_out = self.dropout_2(ffn_out, training=training)
        preds = self.out(ffn_out)
        return preds

    def get_causal_attention_mask(self, inputs):
        input_shape = tf.shape(inputs)
        batch_size, sequence_length = input_shape[0], input_shape[1]
        i = tf.range(sequence_length)[:, tf.newaxis]
        j = tf.range(sequence_length)
        mask = tf.cast(i >= j, dtype="int32")
        mask = tf.reshape(mask, (1, input_shape[1], input_shape[1]))
        mult = tf.concat(
            [
                tf.expand_dims(batch_size, -1),
                tf.constant([1, 1], dtype=tf.int32),
            ],
            axis=0,
        )
        return tf.tile(mask, mult)
```

```python
class OCRModel(keras.Model):
    def __init__(
        self,
        cnn_model,
        encoder,
        decoder,
        image_aug=None,
    ):
        super().__init__()
        self.cnn_model = cnn_model
        self.encoder = encoder
        self.decoder = decoder
        self.loss_tracker = keras.metrics.Mean(name="loss")
        self.acc_tracker = keras.metrics.Mean(name="accuracy")
        self.image_aug = image_aug

    def calculate_loss(self, y_true, y_pred, mask):
        loss = self.loss(y_true, y_pred)
        mask = tf.cast(mask, dtype=loss.dtype)
        loss *= mask
        return tf.reduce_sum(loss) / tf.reduce_sum(mask)

    def calculate_accuracy(self, y_true, y_pred, mask):
        accuracy = tf.equal(y_true, tf.argmax(y_pred, axis=2))
        accuracy = tf.math.logical_and(mask, accuracy)
        accuracy = tf.cast(accuracy, dtype=tf.float32)
        mask = tf.cast(mask, dtype=tf.float32)
        return tf.reduce_sum(accuracy) / tf.reduce_sum(mask)

    def _compute_loss_and_acc(self, img_embed, batch_seq, training=True):
        encoder_out = self.encoder(img_embed, training=training)
        batch_seq_inp = batch_seq[:, :-1]
        batch_seq_true = batch_seq[:, 1:]
        mask = tf.math.not_equal(batch_seq_true, 0)
        batch_seq_pred = self.decoder(
            batch_seq_inp, encoder_out, training=training, mask=mask
        )
        loss = self.calculate_loss(batch_seq_true, batch_seq_pred, mask)
        acc = self.calculate_accuracy(batch_seq_true, batch_seq_pred, mask)
        return loss, acc

    def train_step(self, batch_data):
        batch_img, batch_seq = batch_data
        batch_loss = 0
        batch_acc = 0
```

```
        if self.image_aug:
            batch_img = self.image_aug(batch_img)
        img_embed = self.cnn_model(batch_img)
        with tf.GradientTape() as tape:
            loss, acc = self._compute_loss_and_acc(
                img_embed, batch_seq, training=True
            )
            batch_loss += loss
            batch_acc += acc
        train_vars = (
            self.encoder.trainable_variables + self.decoder.trainable_variables
        )
        grads = tape.gradient(loss, train_vars)
        self.optimizer.apply_gradients(zip(grads, train_vars))
        self.loss_tracker.update_state(batch_loss)
        self.acc_tracker.update_state(batch_acc)
        return {
            "loss": self.loss_tracker.result(),
            "acc": self.acc_tracker.result(),
        }

    def test_step(self, batch_data):
        batch_img, batch_seq = batch_data
        batch_loss = 0
        batch_acc = 0
        img_embed = self.cnn_model(batch_img)
        loss, acc = self._compute_loss_and_acc(
            img_embed, batch_seq, training=False
        )
        batch_loss += loss
        batch_acc += acc
        return {
            "loss": batch_loss,
            "acc": batch_acc,
        }

    @property
    def metrics(self):
        return [self.loss_tracker, self.acc_tracker]


cnn_model = get_cnn_model()
encoder = TransformerEncoderBlock(embed_dim=EMBED_DIM, dense_dim=FF_DIM,␣
 ↪num_heads=1)
decoder = TransformerDecoderBlock(embed_dim=EMBED_DIM, ff_dim=FF_DIM,␣
 ↪num_heads=2)
ocr_model = OCRModel(
```

```
    cnn_model=cnn_model,
    encoder=encoder,
    decoder=decoder,
)
```

Model Training

[21]:
```
class PrintEpochCallback(keras.callbacks.Callback):
    def __init__(self, print_frequency=10):
        self.print_frequency = print_frequency
        self.epoch_counter = 0

    def on_epoch_end(self, epoch, logs=None):
        self.epoch_counter += 1
        if self.epoch_counter % self.print_frequency == 0:
            print(f"Epoch {epoch+1}/{EPOCHS}")
            print(f"Loss: {logs['loss']:.4f}, Accuracy: {logs['acc']:.4f}")
tion
cross_entropy = keras.losses.SparseCategoricalCrossentropy(
    from_logits=False,
    reduction="sum_over_batch_size",
)


# Learning Rate Scheduler for the optimizer
class LRSchedule(keras.optimizers.schedules.LearningRateSchedule):
    def __init__(self, post_warmup_learning_rate, warmup_steps):
        super().__init__()
        self.post_warmup_learning_rate = post_warmup_learning_rate
        self.warmup_steps = warmup_steps

    def __call__(self, step):
        global_step = tf.cast(step, tf.float32)
        warmup_steps = tf.cast(self.warmup_steps, tf.float32)
        warmup_progress = global_step / warmup_steps
        warmup_learning_rate = self.post_warmup_learning_rate * warmup_progress
        return tf.cond(
            global_step < warmup_steps,
            lambda: warmup_learning_rate,
            lambda: self.post_warmup_learning_rate,
        )


# Create a learning rate schedule
num_train_steps = len(train_dataset) * EPOCHS
num_warmup_steps = num_train_steps // 15
```

```
lr_schedule = LRSchedule(post_warmup_learning_rate=1e-3,␣
 ↪warmup_steps=num_warmup_steps)

# Compile the model
ocr_model.compile(optimizer=keras.optimizers.Adam(lr_schedule),␣
 ↪loss=cross_entropy)

print_epoch_callback = PrintEpochCallback(print_frequency=20)

ocr_model.fit(
    train_dataset,
    epochs=EPOCHS,
    validation_data=valid_dataset,
    callbacks=[print_epoch_callback],
    verbose=0
)
```

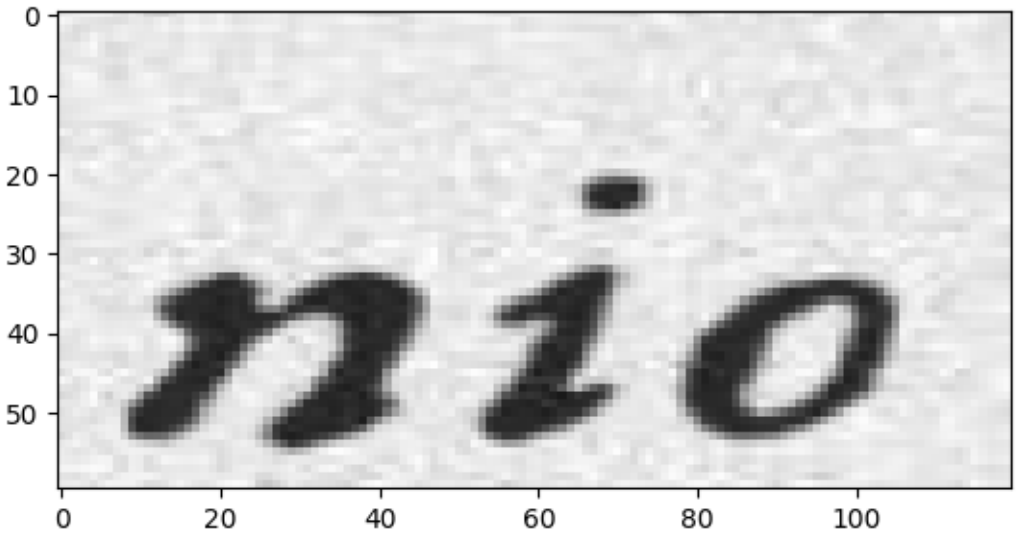
```
Epoch 20/140
Loss: 3.5360, Accuracy: 0.5013
Epoch 40/140
Loss: 3.0531, Accuracy: 0.5053
Epoch 60/140
Loss: 2.2473, Accuracy: 0.5580
Epoch 80/140
Loss: 1.3632, Accuracy: 0.6866
Epoch 100/140
Loss: 0.5658, Accuracy: 0.8928
Epoch 120/140
Loss: 0.1762, Accuracy: 0.9838
Epoch 140/140
Loss: 0.0635, Accuracy: 0.9985
```

[21]:
```
<keras.src.callbacks.History at 0x7cb9ccd69150>
```

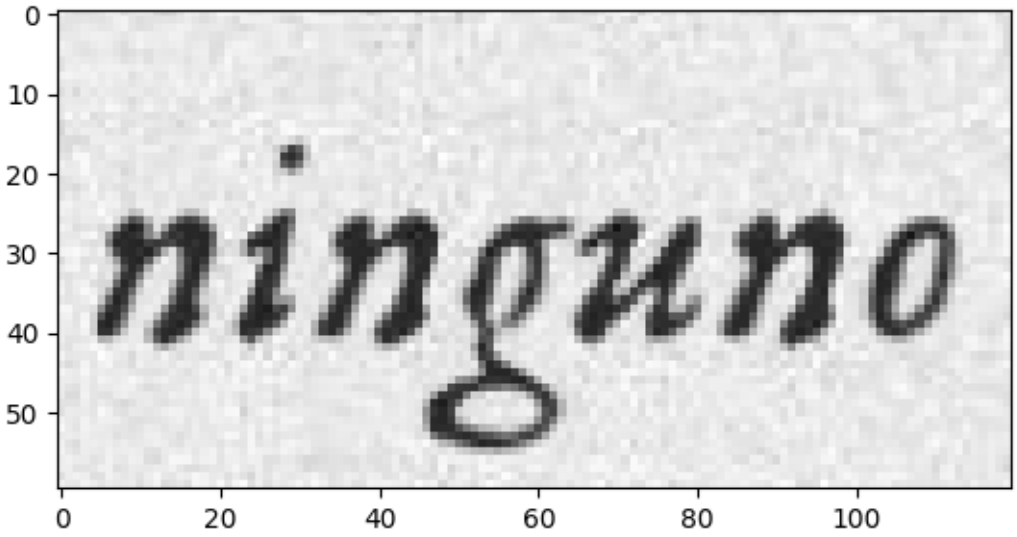
Check sample predictions

[26]:
```
vocab = vectorization.get_vocabulary()
index_lookup = dict(zip(range(len(vocab)), vocab))
max_decoded_sentence_length = SEQ_LENGTH - 1
valid_images = list(valid_data.keys())


def predict_ocr():
    sample_img = np.random.choice(valid_images)

    sample_img = decode_and_resize(sample_img)
    img = sample_img.numpy().clip(0, 255).astype(np.uint8)
```

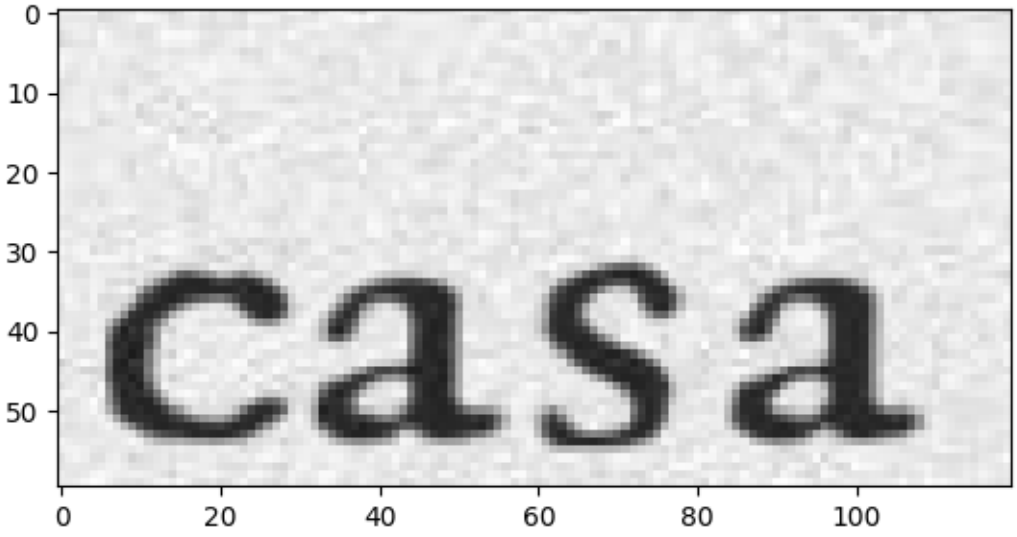
```
    plt.imshow(img)
    plt.show()

    img = tf.expand_dims(sample_img, 0)
    img = ocr_model.cnn_model(img)

    encoded_img = ocr_model.encoder(img, training=False)

    decoded_ocr = "<start> "
    for i in range(max_decoded_sentence_length):
        tokenized_result = vectorization([decoded_ocr])[:, :-1]
        mask = tf.math.not_equal(tokenized_result, 0)
        predictions = ocr_model.decoder(
            tokenized_result, encoded_img, training=False, mask=mask
        )
        sampled_token_index = np.argmax(predictions[0, i, :])
        sampled_token = index_lookup[sampled_token_index]
        if sampled_token == "[UNK]":
            break
        decoded_ocr += " " + sampled_token

    print("Predicted Caption: ", decoded_ocr)


predict_ocr()
predict_ocr()
predict_ocr()
```




```
Predicted Caption:  tito
```




```
Predicted Caption:  ninguno
```




```
Predicted Caption:  casa
```

Please note that this is an example of an encoder-decoder architecture and not the final version of OCR.

4.0.2 Section 2.2 Create Transformers Based Model From Pretrained Encoder Decoder Models

I will follow the same architecture with Trocr which is stated "**To effectively train the TrOCR model, the encoder can be initialized with pre-trained ViT-style models (Dosovitskiy et al. 2021; Touvron et al. 2021; Bao, Dong, and Wei 2021) while the decoder can be initialized with pre-trained BERT-style models (Devlin et al. 2019; Liu et al. 2019; Dong et al. 2019; Wang et al. 2020b), respectively**."

I'll use vit-base-patch16-224-in21k for processing images and bert-base-uncased for breaking down text into tokens. Transformers are more precise than CNN-based OCR. The top 5 OCRs in the IAM dataset utilize transformers. However, transformers typically require more training data. TrOCR, for instance, was trained on a dataset with 684 million lines of text. Since I don't have the computational power for that, I've developed a Proof of Concept model trained and evaluated on a smaller dataset.

[ ]:
```
from transformers import ViTImageProcessor, BertTokenizer,␣
 ↪VisionEncoderDecoderModel
from datasets import load_dataset

from transformers import BertConfig, ViTConfig, VisionEncoderDecoderConfig,␣
 ↪VisionEncoderDecoderModel

config_encoder = ViTConfig()
config_decoder = BertConfig()
config = VisionEncoderDecoderConfig.
 ↪from_encoder_decoder_configs(config_encoder, config_decoder)

model = VisionEncoderDecoderModel(config=config)
image_processor = ViTImageProcessor.from_pretrained("google/
 ↪vit-base-patch16-224-in21k")
tokenizer = BertTokenizer.from_pretrained("bert-base-uncased")

# set special tokens used for creating the decoder_input_ids from the labels
model.config.decoder_start_token_id = tokenizer.cls_token_id
model.config.pad_token_id = tokenizer.pad_token_id
# make sure vocab size is set correctly
model.config.vocab_size = model.config.decoder.vocab_size

# set beam search parameters
model.config.eos_token_id = 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 = 2
```

[ ]:
```
class test_sp(Dataset):
    def __init__(self, root_dir, df,processor, image_processor,␣
 ↪max_target_length=128):
        self.root_dir = root_dir
        self.df = df
        self.processor = processor
        self.image_processor = image_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.image_processor(image, return_tensors="pt").
 ↪pixel_values
        # add labels (input_ids) by encoding the text
        labels = self.processor(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.pad_token_id else -100 for␣
 ↪label in labels]

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

[ ]:
```
train_dataset = test_sp(root_dir=dataset_directory + "/",
                        df=train_df,
                        processor=tokenizer, image_processor=image_processor)
eval_dataset = test_sp(root_dir=dataset_directory + "/",
                        df=test_df,
                        processor=tokenizer, image_processor=image_processor)
```

[ ]:
```
from transformers import Trainer, TrainingArguments
device = torch.device('cuda') if torch.cuda.is_available() else torch.
 ↪device('cpu')

model=model.to(device)
model.train()
```

```
training_args = TrainingArguments(
    output_dir='./results',
    num_train_epochs=60,
    per_device_train_batch_size=20,
    per_device_eval_batch_size=32,
    # warmup_steps=50,
    # weight_decay=0.01,
    logging_dir='./logs',
    logging_steps=150,
    # eval_steps=25
)


trainer = Trainer(
    model=model,
    args=training_args,
    train_dataset=train_dataset,
    eval_dataset=eval_dataset
)
trainer.train()

save_folder = "/content/drive/MyDrive/generated_vision_encoder_decoder_20-5"
trainer.save_model(save_folder)
```

```
/usr/local/lib/python3.10/dist-packages/accelerate/accelerator.py:432:
FutureWarning: Passing the following arguments to `Accelerator` is deprecated
and will be removed in version 1.0 of Accelerate: dict_keys(['dispatch_batches',
'split_batches', 'even_batches', 'use_seedable_sampler']). Please pass an
`accelerate.DataLoaderConfiguration` instead:
dataloader_config = DataLoaderConfiguration(dispatch_batches=None,
split_batches=False, even_batches=True, use_seedable_sampler=True)
  warnings.warn(
We strongly recommend passing in an `attention_mask` since your input_ids may be
padded. See https://huggingface.co/docs/transformers/troubleshooting#incorrect-
output-when-padding-tokens-arent-masked.

<IPython.core.display.HTML object>

Some non-default generation parameters are set in the model config. These should
go into a GenerationConfig file
(https://huggingface.co/docs/transformers/generation_strategies#save-a-custom-
decoding-strategy-with-your-model) instead. This warning will be raised to an
exception in v4.41.
Non-default generation parameters: {'max_length': 64, 'early_stopping': True,
'num_beams': 2}
Your generation config was originally created from the model config, but the
model config has changed since then. Unless you pass the `generation_config`
argument to this model's `generate` calls, they will revert to the legacy
```

```
behavior where the base `generate` parameterization is loaded from the model
config instead. To avoid this behavior and this warning, we recommend you to
overwrite the generation config model attribute before calling the model's
`save_pretrained`, preferably also removing any generation kwargs from the model
config. This warning will be raised to an exception in v4.41.
Removed shared tensor {'decoder.cls.predictions.decoder.weight',
'decoder.cls.predictions.decoder.bias'} while saving. This should be OK, but
check by verifying that you don't receive any warning while reloading
Some non-default generation parameters are set in the model config. These should
go into a GenerationConfig file
(https://huggingface.co/docs/transformers/generation_strategies#save-a-custom-
decoding-strategy-with-your-model) instead. This warning will be raised to an
exception in v4.41.
Non-default generation parameters: {'max_length': 64, 'early_stopping': True,
'num_beams': 2}
Your generation config was originally created from the model config, but the
model config has changed since then. Unless you pass the `generation_config`
argument to this model's `generate` calls, they will revert to the legacy
behavior where the base `generate` parameterization is loaded from the model
config instead. To avoid this behavior and this warning, we recommend you to
overwrite the generation config model attribute before calling the model's
`save_pretrained`, preferably also removing any generation kwargs from the model
config. This warning will be raised to an exception in v4.41.
Some non-default generation parameters are set in the model config. These should
go into a GenerationConfig file
(https://huggingface.co/docs/transformers/generation_strategies#save-a-custom-
decoding-strategy-with-your-model) instead. This warning will be raised to an
exception in v4.41.
Non-default generation parameters: {'max_length': 64, 'early_stopping': True,
'num_beams': 2}
Your generation config was originally created from the model config, but the
model config has changed since then. Unless you pass the `generation_config`
argument to this model's `generate` calls, they will revert to the legacy
behavior where the base `generate` parameterization is loaded from the model
config instead. To avoid this behavior and this warning, we recommend you to
overwrite the generation config model attribute before calling the model's
`save_pretrained`, preferably also removing any generation kwargs from the model
config. This warning will be raised to an exception in v4.41.
```

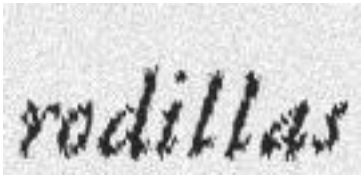
[ ]:
```
model = VisionEncoderDecoderModel.from_pretrained(save_folder)
```

[ ]:
```
def test_model(directory):
    test_image = dataset_directory + "/" + directory
    image = Image.open(test_image).convert("RGB")
    display(image)
    pixel_values = image_processor(image, return_tensors="pt").pixel_values
    generated_ids = model.generate(pixel_values)
```

```
    generated_text = tokenizer.batch_decode(generated_ids,␣
↪skip_special_tokens=True)[0]
    print("actual text: " + directory.split("_")[0] + ", predicted text: " +␣
↪generated_text)

test_model(sub_df.iloc[1]["file_name"])
```

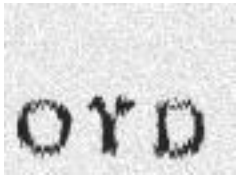

```
/usr/local/lib/python3.10/dist-packages/transformers/generation/utils.py:1197:
UserWarning: You have modified the pretrained model configuration to control
generation. This is a deprecated strategy to control generation and will be
removed soon, in a future version. Please use and modify the model generation
configuration (see
https://huggingface.co/docs/transformers/generation_strategies#default-text-
generation-configuration )
  warnings.warn(
actual text: sus, predicted text: sus
```

[ ]:
```
test_model(sub_df.iloc[0]["file_name"])
```




```
actual text: rodillas, predicted text: rodillas
```

[ ]:
```
test_model(sub_df.iloc[2]["file_name"])
```




```
actual text: oyd, predicted text: civil
```

5 Section 2.3 Fine Tune Transformers base TrOCR Model

[ ]:
```
class SpanishPrintedDataset(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
```

[ ]:
```
train_df, test_df = train_test_split(df, test_size=0.2)
train_df.reset_index(drop=True, inplace=True)
test_df.reset_index(drop=True, inplace=True)
```

Initializing the training and evaluation datasets

[ ]:
```
processor = TrOCRProcessor.from_pretrained("microsoft/trocr-base-printed")
#processor = TrOCRProcessor.from_pretrained('microsoft/trocr-large-str')
train_dataset = SpanishPrintedDataset(root_dir=dataset_directory + "/",
                            df=train_df,
                            processor=processor)
eval_dataset = SpanishPrintedDataset(root_dir=dataset_directory + "/",
                            df=test_df,
                            processor=processor)
```

```
print("Number of training examples:", len(train_dataset))
print("Number of validation examples:", len(eval_dataset))

encoding = train_dataset[0]
for k,v in encoding.items():
  print(k, v.shape)

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

```
/usr/local/lib/python3.10/dist-packages/huggingface_hub/utils/_token.py:88:
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 secret in your Google Colab
and restart your session.
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:   0%|          | 0.00/228 [00:00<?, ?B/s]
```

```
Could not find image processor class in the image processor config or the model
config. Loading based on pattern matching with the model's feature extractor
configuration. Please open a PR/issue to update `preprocessor_config.json` to
use `image_processor_type` instead of `feature_extractor_type`. This warning
will be removed in v4.40.
```

```
tokenizer_config.json:   0%|          | 0.00/1.12k [00:00<?, ?B/s]
```

```
vocab.json:   0%|          | 0.00/899k [00:00<?, ?B/s]
```

```
merges.txt:   0%|          | 0.00/456k [00:00<?, ?B/s]
```

```
special_tokens_map.json:   0%|          | 0.00/772 [00:00<?, ?B/s]
```

```
Number of training examples: 10399
Number of validation examples: 2600
pixel_values torch.Size([3, 384, 384])
labels torch.Size([128])
```

[ ]:
```
model = VisionEncoderDecoderModel.from_pretrained("microsoft/
  ↪trocr-base-printed") #VisionEncoderDecoderModel.from_pretrained("microsoft/
  ↪trocr-base-stage1") #/content/drive/MyDrive/trocr_model
```

```
config.json:   0%|          | 0.00/4.13k [00:00<?, ?B/s]
```

```
pytorch_model.bin:   0%|          | 0.00/1.33G [00:00<?, ?B/s]
```

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

generation_config.json:   0%|          | 0.00/190 [00:00<?, ?B/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 = 1.0
model.config.num_beams = 2
```

[ ]:
```
training_args = Seq2SeqTrainingArguments(
    predict_with_generate=True,
    evaluation_strategy="steps",
    per_device_train_batch_size=12,
    per_device_eval_batch_size=12,
    fp16=True,
    output_dir="./content/res",
    logging_steps=2,
    save_steps=1000,
    eval_steps=250,
    num_train_epochs=3


)
```

I will evaluate the model on the Character Error Rate (CER).

[ ]:
```
from datasets import load_metric

cer_metric = load_metric("cer")
```

```
<ipython-input-12-c81d87c6f9c2>:3: FutureWarning: load_metric is deprecated and
will be removed in the next major version of datasets. Use 'evaluate.load'
instead, from the new library  Evaluate: https://huggingface.co/docs/evaluate
  cer_metric = load_metric("cer")
/usr/local/lib/python3.10/dist-packages/datasets/load.py:756: FutureWarning: The
repository for cer contains custom code which must be executed to correctly load
the metric. You can inspect the repository content at
https://raw.githubusercontent.com/huggingface/datasets/2.18.0/metrics/cer/cer.py
```

```
You can avoid this message in future by passing the argument
`trust_remote_code=True`.
Passing `trust_remote_code=True` will be mandatory to load this metric from the
next major release of `datasets`.
  warnings.warn(

Downloading builder script:   0%|          | 0.00/2.16k [00:00<?, ?B/s]
```

[ ]:
```python
def compute_metrics(pred):
    labels_ids = pred.label_ids
    pred_ids = pred.predictions

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

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

    return {"cer": cer}
```

[ ]:
```python
from transformers import default_data_collator

# instantiate trainer
trainer = Seq2SeqTrainer(
    model=model,
    tokenizer=processor.feature_extractor,
    args=training_args,
    compute_metrics=compute_metrics,
    train_dataset=train_dataset,
    eval_dataset=eval_dataset,
    data_collator=default_data_collator,

)
trainer.train()
trainer.save_model("/content/drive/MyDrive/trocr_es14k_finetune_mixed")
```

```
/usr/local/lib/python3.10/dist-
packages/transformers/models/trocr/processing_trocr.py:136: FutureWarning:
`feature_extractor` is deprecated and will be removed in v5. Use
`image_processor` instead.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/accelerate/accelerator.py:432:
FutureWarning: Passing the following arguments to `Accelerator` is deprecated
and will be removed in version 1.0 of Accelerate: dict_keys(['dispatch_batches',
'split_batches', 'even_batches', 'use_seedable_sampler']). Please pass an
`accelerate.DataLoaderConfiguration` instead:
dataloader_config = DataLoaderConfiguration(dispatch_batches=None,
split_batches=False, even_batches=True, use_seedable_sampler=True)
```

```
  warnings.warn(
```

```
<IPython.core.display.HTML object>
```

```
/usr/local/lib/python3.10/dist-packages/transformers/generation/utils.py:1197:
UserWarning: You have modified the pretrained model configuration to control
generation. This is a deprecated strategy to control generation and will be
removed soon, in a future version. Please use and modify the model generation
configuration (see
https://huggingface.co/docs/transformers/generation_strategies#default-text-
generation-configuration )
  warnings.warn(
Some non-default generation parameters are set in the model config. These should
go into a GenerationConfig file
(https://huggingface.co/docs/transformers/generation_strategies#save-a-custom-
decoding-strategy-with-your-model) instead. This warning will be raised to an
exception in v4.41.
Non-default generation parameters: {'max_length': 64, 'early_stopping': True,
'num_beams': 2, 'no_repeat_ngram_size': 3}
Removed shared tensor {'decoder.output_projection.weight'} while saving. This
should be OK, but check by verifying that you don't receive any warning while
reloading
Some non-default generation parameters are set in the model config. These should
go into a GenerationConfig file
(https://huggingface.co/docs/transformers/generation_strategies#save-a-custom-
decoding-strategy-with-your-model) instead. This warning will be raised to an
exception in v4.41.
Non-default generation parameters: {'max_length': 64, 'early_stopping': True,
'num_beams': 2, 'no_repeat_ngram_size': 3}
Some non-default generation parameters are set in the model config. These should
go into a GenerationConfig file
(https://huggingface.co/docs/transformers/generation_strategies#save-a-custom-
decoding-strategy-with-your-model) instead. This warning will be raised to an
exception in v4.41.
Non-default generation parameters: {'max_length': 64, 'early_stopping': True,
'num_beams': 2, 'no_repeat_ngram_size': 3}
```

Load fine tuned model

[ ]:
```
model_save_folder = "/content/drive/MyDrive/trocr_es14k_finetune_mixed"
processor = TrOCRProcessor.from_pretrained("microsoft/trocr-base-printed")
model = VisionEncoderDecoderModel.from_pretrained(model_save_folder)
```

```
/usr/local/lib/python3.10/dist-packages/huggingface_hub/utils/_token.py:88:
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 secret in your Google Colab
and restart your session.
```

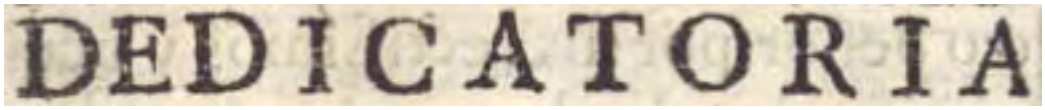
```
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access
public models or datasets.
  warnings.warn(

preprocessor_config.json:   0%|          | 0.00/228 [00:00<?, ?B/s]

Could not find image processor class in the image processor config or the model
config. Loading based on pattern matching with the model's feature extractor
configuration. Please open a PR/issue to update `preprocessor_config.json` to
use `image_processor_type` instead of `feature_extractor_type`. This warning
will be removed in v4.40.

tokenizer_config.json:   0%|          | 0.00/1.12k [00:00<?, ?B/s]

vocab.json:   0%|          | 0.00/899k [00:00<?, ?B/s]

merges.txt:   0%|          | 0.00/456k [00:00<?, ?B/s]

special_tokens_map.json:   0%|          | 0.00/772 [00:00<?, ?B/s]
```

5.1 Section 3: Evaluation and Result

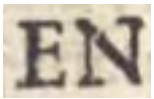
[ ]:
```python
import re

def process_image_folders(directory_path, model, processor):
    def get_numeric_parts(folder_name):
        """Extracts all numeric parts from a folder name."""
        return [int(part) for part in re.findall(r'\d+', folder_name)]

    def sort_folders(folder_names):
        """Sorts folder names based on embedded numeric sequences."""
        return sorted(folder_names, key=get_numeric_parts)

    folder_names = os.listdir(directory_path)
    sorted_folder_names = sort_folders(folder_names)
    sorted_folder_names
    print(sorted_folder_names)
    generated_text_result = ""

    for folder_name in sorted_folder_names:
        folder_path = os.path.join(directory_path, folder_name)
        image = Image.open(folder_path).convert("RGB")
        display(image)
        pixel_values = processor(image, return_tensors="pt").pixel_values
        generated_ids = model.generate(pixel_values)
        generated_text = processor.batch_decode(generated_ids,␣
 ↪skip_special_tokens=True)[0]

        if generated_text_result == "":
```

```
            generated_text_result += generated_text
        else:
            generated_text_result = generated_text_result + " " + generated_text

        print(generated_text)

    return generated_text_result

directory_path = '/content/drive/MyDrive/ocr/extracted_images/page_1'
result = process_image_folders(directory_path, model, processor)
```

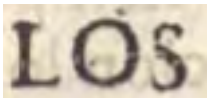
```
['text_0.png', 'text_1.png', 'text_2.png', 'text_3.png', 'text_4.png',
'text_5.png', 'text_6.png', 'text_7.png', 'text_8.png', 'text_9.png',
'text_10.png', 'text_11.png', 'text_12.png', 'text_13.png', 'text_14.png',
'text_15.png', 'text_16.png', 'text_17.png', 'text_18.png', 'text_19.png',
'text_20.png', 'text_21.png', 'text_22.png', 'text_23.png', 'text_24.png',
'text_25.png', 'text_29.png', 'text_30.png', 'text_31.png', 'text_32.png',
'text_33.png', 'text_34.png', 'text_35.png', 'text_36.png', 'text_37.png',
'text_38.png', 'text_39.png', 'text_40.png', 'text_41.png', 'text_42.png',
'text_43.png', 'text_44.png', 'text_45.png', 'text_46.png', 'text_47.png',
'text_48.png', 'text_49.png', 'text_50.png', 'text_51.png', 'text_53.png',
'text_54.png', 'text_55.png', 'text_56.png', 'text_57.png', 'text_58.png',
'text_59.png', 'text_60.png', 'text_61.png', 'text_62.png', 'text_63-1.png',
'text_63-2.png', 'text_64.png', 'text_65.png', 'text_66.png', 'text_67-1.png',
'text_67-2.png', 'text_68.png', 'text_69.png', 'text_70.png', 'text_71.png',
'text_74.png', 'text_75.png', 'text_76.png', 'text_77.png', 'text_78-1.png',
'text_78-2.png', 'text_79.png', 'text_80.png']
```



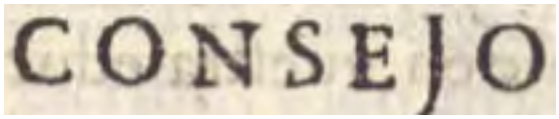

```
dedicatoria
```



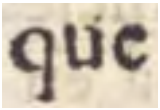

```
en
```



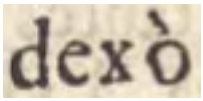
los



consejo

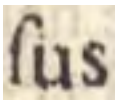


que

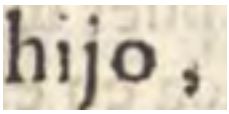


dexó

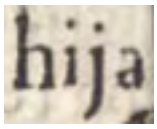
á



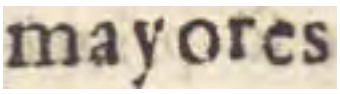
fus



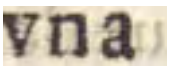
hijo



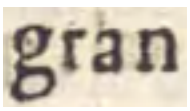
hija



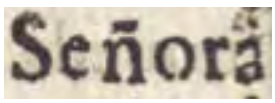
mayores



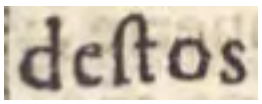
vna



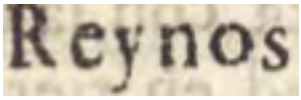
gran



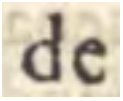
seña



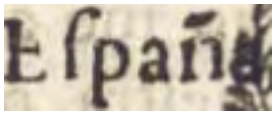
defos



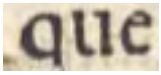
reynos



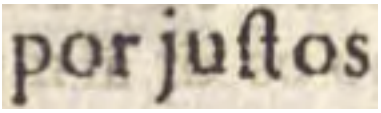
de



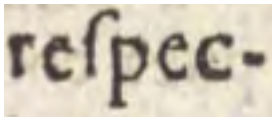
fipańri



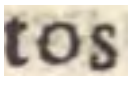
que



porjutos

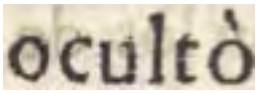


relpec



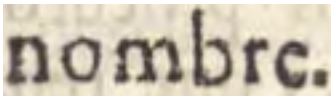
tos

fe

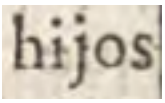


oculto

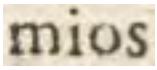
flu



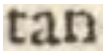
nombre



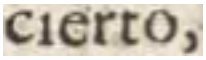
hijos



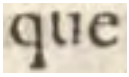
mios



tan



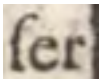
cierto



que

el

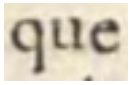
virtulóo



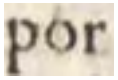
fer

es

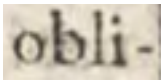
el



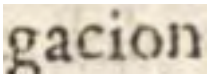
que



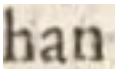
por



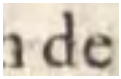
obli



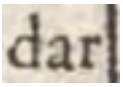
gacion



han

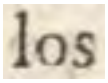


ide

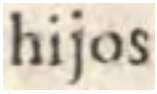


dar

a

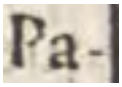


los

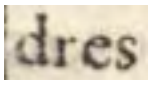


hijos

fus

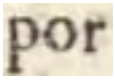


pa



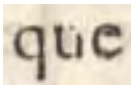
dres

y

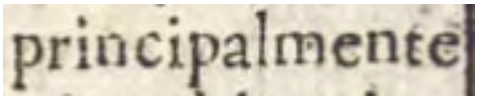


por

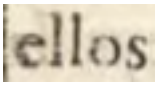
el



que

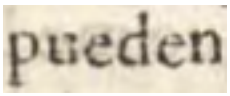


principalmente

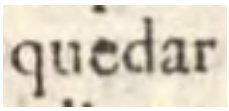


ellos

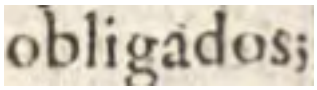
les



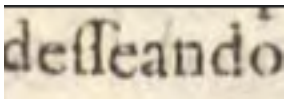
pueden



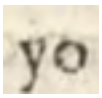
quedar



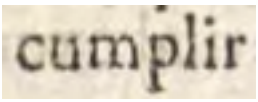
obligados



defeandó

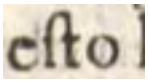


yo

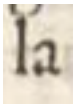


cumplir

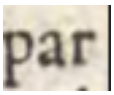
en



cito

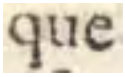


la



par

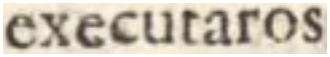
te



que

me

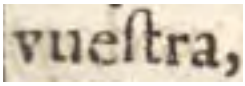
toca



executaros

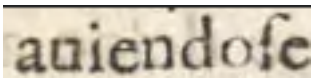
en

la

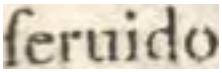


vuetra

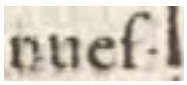
no



aviendole



```
ferido
```



```
nuefl
```

[ ]:
```
result
```

[ ]:
```
'dedicatoria en los consejo que dexó á fus hijo hija mayores vna gran seña
defos reynos de fipañri que porjutos relpec tos fe oculto flu nombre hijos mios
tan cierto que el virtulóo fer es el que por obli gacion han ide dar a los
hijos fus pa dres y por el que principalmente ellos les pueden quedar obligados
defeandó yo cumplir en cito la par te que me toca executaros en la vuetra no
aviendole ferido nuefl'
```

[ ]:
```
true_text = "DEDICATORIA EN LOS CONSEJOS que dexo a sus hijo, e hija mayores␣
↪una gran Señora destos Reynos de España que por justos respec- tos se ocultó␣
↪su nombre. SIENDO (hijos mios) tan cierto, que el virtuoso ser es el que por␣
↪obli- gacion han de dar a los hijos sus Pa- dres, y por el que␣
↪principalmente ellos les pueden quedar obligados; desseando yo cumplir en␣
↪esto la par te que me toca, y executaros en la vuestra, no aviendose servido␣
↪nues"
```

6 Metrices

I believe that the most appropriate method to measure the similarity between two texts in this case is by using the Levenshtein distance. The Levenshtein distance is a metric designed to determine the similarity between two strings. It is calculated as the smallest number of single-character edits (insertions, deletions, or substitutions) needed to transform one string into the other. This method is valuable because when the Levenshtein distance is high, it **can significantly improve the accuracy of OCR predictions by using a dictionary to identify the most similar word to a falsely predicted one**.

[ ]:
```
def levenshtein(s1, s2):
    if len(s1) < len(s2):
        return levenshtein(s2, s1)

    if len(s2) == 0:
        return len(s1)

    previous_row = range(len(s2) + 1)
```

```
    for i, c1 in enumerate(s1):
        current_row = [i + 1]
        for j, c2 in enumerate(s2):
            insertions = previous_row[j + 1] + 1
            deletions = current_row[j] + 1
            substitutions = previous_row[j] + (c1 != c2)
            current_row.append(min(insertions, deletions, substitutions))
        previous_row = current_row

    return previous_row[-1]
```

[ ]:
```
def accuracy(predicted_text, actual_text):
    predicted_text = predicted_text.replace("\n", "")
    actual_text = actual_text.replace("\n", "")
    lev_distance = levenshtein(predicted_text, actual_text)
    return (1 - lev_distance / max(len(predicted_text), len(actual_text))) * 100

print(accuracy(result.lower(), true_text.lower()))
```

```
85.77981651376146
```

Another useful metric is cosine similarity, which create vector of texts then measures how similar two vectors are. It's commonly used in natural language processing to mesure text similarity. The accuracy of cosine similarity depends on the number of similar and different words in the given text, making it highly suitable on OCR result.

[ ]:
```
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.metrics.pairwise import cosine_similarity

def calculate_cosine_similarity(text1, text2):
    vectorizer = CountVectorizer().fit_transform([text1, text2])
    vectors = vectorizer.toarray()
    csim = cosine_similarity(vectors)
    return csim[0][1]
calculate_cosine_similarity(result, true_text)
```

[ ]:
```
0.813976578514622
```

6.1 Predicting Test Pages

6.1.1 Text Detection

[ ]:
```
!pip install -q "python-doctr[tf]"
```

```
                                        295.0/295.0
kB 4.4 MB/s eta 0:00:00
                                        2.8/2.8 MB
35.4 MB/s eta 0:00:00
```

```
                              908.3/908.3
kB 52.0 MB/s eta 0:00:00
                              981.5/981.5
kB 45.5 MB/s eta 0:00:00
  Preparing metadata (setup.py) … done
                              271.5/271.5
kB 17.9 MB/s eta 0:00:00
                              88.8/88.8 kB
8.2 MB/s eta 0:00:00
  Installing build dependencies … done
  Getting requirements to build wheel … done
  Installing backend dependencies … done
  Preparing metadata (pyproject.toml) … done
                              235.5/235.5
kB 24.8 MB/s eta 0:00:00
                              455.8/455.8
kB 38.8 MB/s eta 0:00:00
                              11.6/11.6 MB
67.9 MB/s eta 0:00:00
                              15.7/15.7 MB
33.4 MB/s eta 0:00:00
                              2.0/2.0 MB
46.9 MB/s eta 0:00:00
                              848.9/848.9
kB 42.8 MB/s eta 0:00:00
                              3.0/3.0 MB
52.5 MB/s eta 0:00:00
  Building wheel for langdetect (setup.py) … done
  Building wheel for mplcursors (pyproject.toml) … done
```

[ ]:
```python
from doctr.io import DocumentFile
from doctr.models import ocr_predictor
import math
import os
from PIL import Image
from PIL import ImageDraw
import matplotlib.pyplot as plt
import PIL

detection_model = ocr_predictor(det_arch = "db_resnet50", pretrained = True,␣
 ↪assume_straight_pages=True,straighten_pages=True)
detection_model.det_predictor.model.postprocessor.bin_thresh = 0.35

def convert_coordinates(geometry, page_dim):
```

```
    len_x = page_dim[1]
    len_y = page_dim[0]
    (x_min, y_min) = geometry[0]
    (x_max, y_max) = geometry[1]
    x_min = math.floor(x_min * len_x)
    x_max = math.ceil(x_max * len_x)
    y_min = math.floor(y_min * len_y)
    y_max = math.ceil(y_max * len_y)
    return [x_min, x_max, y_min, y_max]

def get_coordinates(output):
    page_dim = output['pages'][0]["dimensions"]
    text_coordinates = []
    for obj1 in output['pages'][0]["blocks"]:
        for obj2 in obj1["lines"]:
            for obj3 in obj2["words"]:
                converted_coordinates = convert_coordinates(
                                        obj3["geometry"],page_dim
                                       )
                text_coordinates.append(converted_coordinates)
    return text_coordinates

#Save
def save_bounded_texts(image, bounds, output_dir):
    for i, b in enumerate(bounds):
        p0, p1, p2, p3 = [b[0],b[2]], [b[1],b[2]], [b[1],b[3]], [b[0],b[3]]
        min_x = min(p0[0], p1[0], p2[0], p3[0])
        min_y = min(p0[1], p1[1], p2[1], p3[1])
        max_x = max(p0[0], p1[0], p2[0], p3[0])
        max_y = max(p0[1], p1[1], p2[1], p3[1])

        # Crop the image to the bounding box dimensions
        cropped_image = image.crop((min_x, min_y, max_x, max_y))

        # Save the cropped image
        cropped_image.save(os.path.join(output_dir, f'text_{i}.png'))

#Show
def draw_bounds(image, bound):
    draw = ImageDraw.Draw(image)
    for b in bound:
        p0, p1, p2, p3 = [b[0],b[2]], [b[1],b[2]], \
                         [b[1],b[3]], [b[0],b[3]]
        draw.line([*p0,*p1,*p2,*p3,*p0], fill='blue', width=2)
    return image

def process_image(image_path, model, save_path):
```

```
    img = DocumentFile.from_images(image_path)
    result = model(img)
    output = result.export()

    graphical_coordinates = get_coordinates(output)

    image = PIL.Image.open(image_path)
    save_bounded_texts(image, graphical_coordinates, save_path)

    image = PIL.Image.open(image_path)
    result_image = draw_bounds(image, graphical_coordinates)

    plt.figure(figsize=(15, 15))
    plt.imshow(result_image)
    plt.show()
```

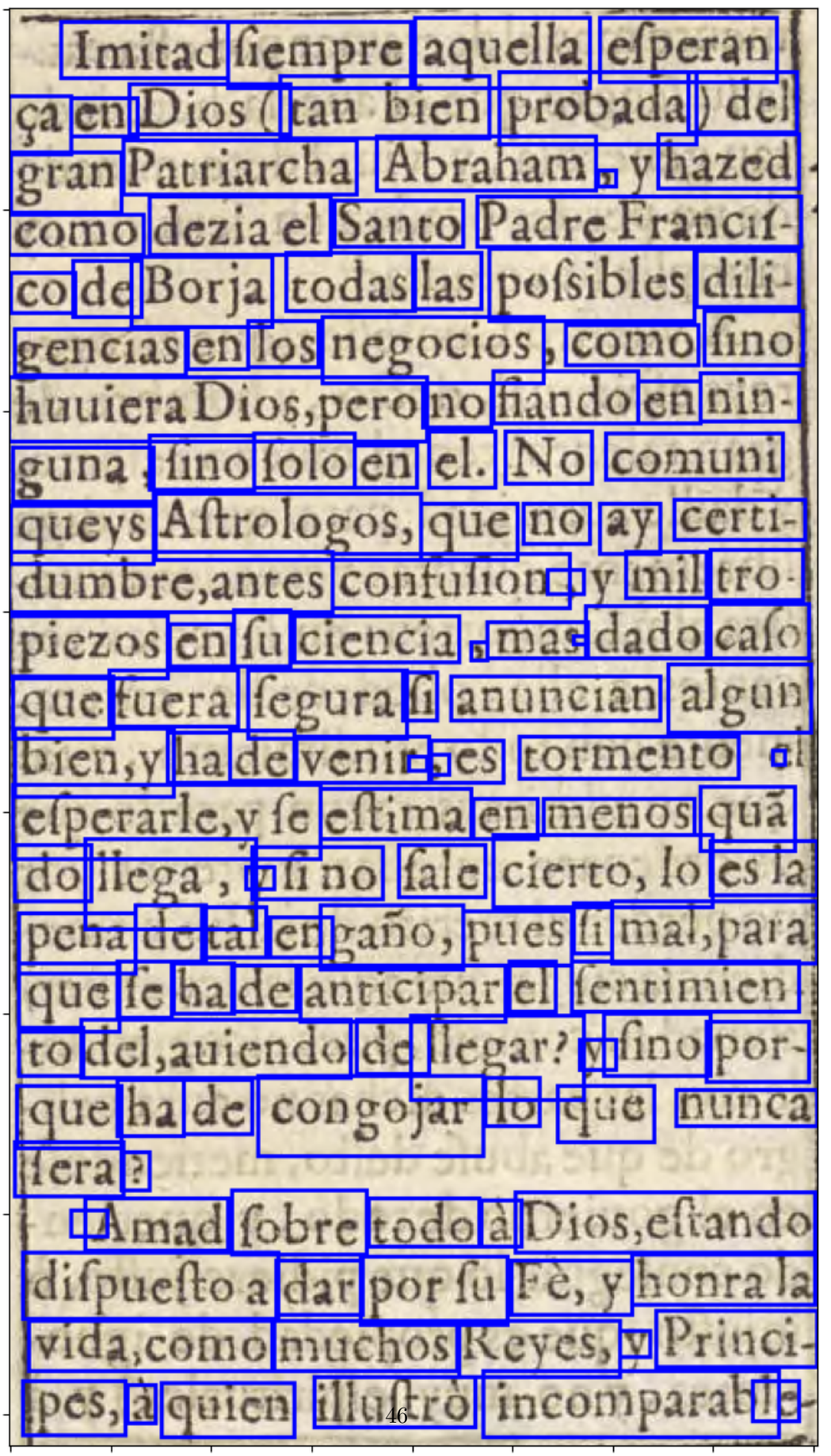
[ ]:
```
def create_folder_if_not_exists(folder_path):
    if not os.path.exists(folder_path):
        os.makedirs(folder_path)
        print(f"Folder '{folder_path}' created successfully.")
    else:
        print(f"Folder '{folder_path}' already exists.")
```

[ ]:
```
image_path = "/content/drive/MyDrive/ocr/test/page14/page14_1.png"
save_path = "/content/drive/MyDrive/ocr/extracted_text_image/page14_1"
create_folder_if_not_exists(save_path)
process_image(image_path, detection_model, save_path)
```

```
Folder '/content/drive/MyDrive/ocr/extracted_text_image/page14_1' created
successfully.
```

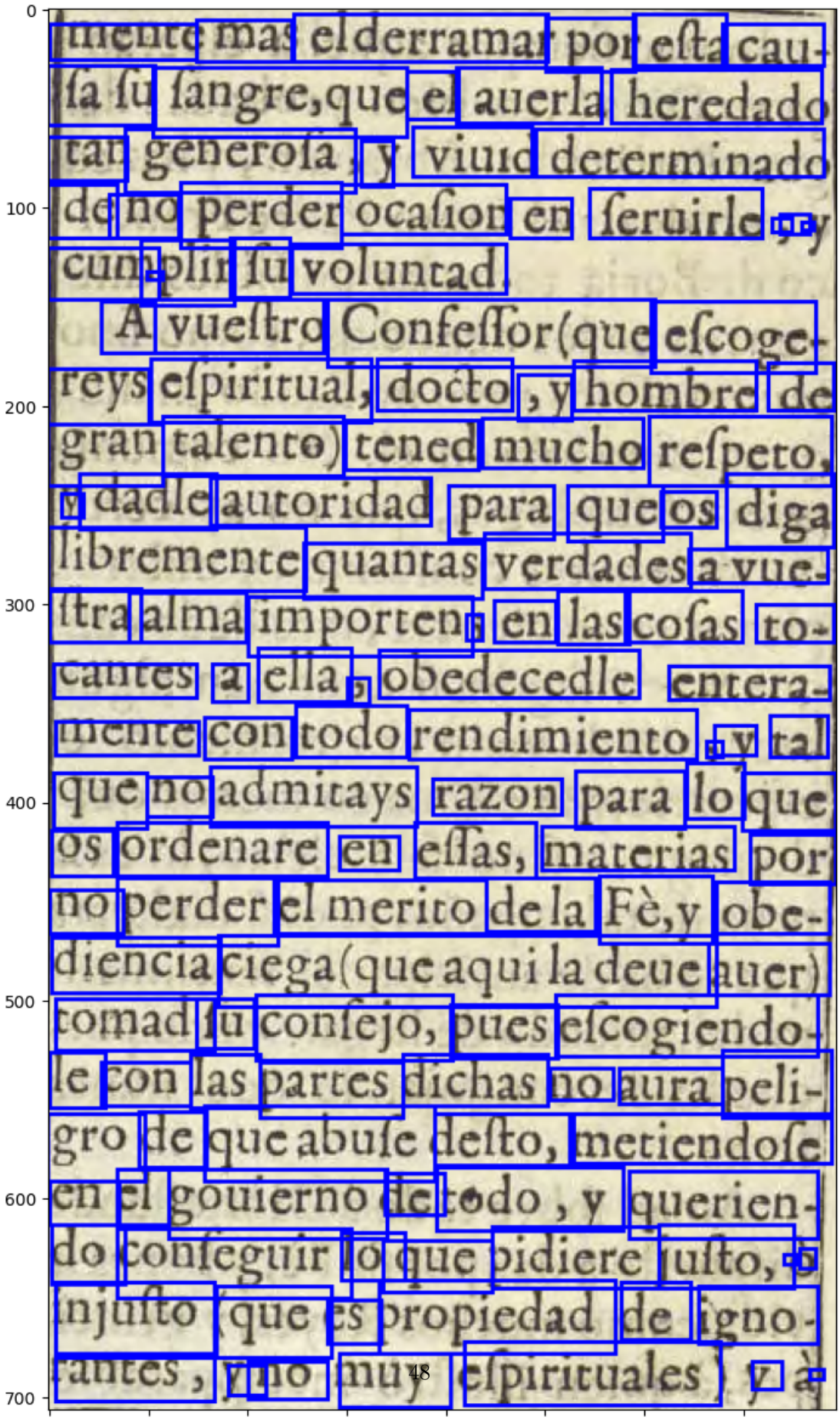
[ ]:
```
image_path = "/content/drive/MyDrive/ocr/test/page14/page14_2.png"
save_path = "/content/drive/MyDrive/ocr/extracted_text_image/page15_1"
create_folder_if_not_exists(save_path)
process_image(image_path, detection_model, save_path)
```

```
Folder '/content/drive/MyDrive/ocr/extracted_text_image/page15_1' created
successfully.
```


Imitad ſiempre aquella eſperan
ça en Dios (tan bien probada) del
gran Patriarcha Abraham, y hazed
como dezia el Santo Padre Francif-
co de Borja todas las poſsibles dili-
gencias en los negocios, como ſino
huuiera Dios, pero no fiando en nin-
guna, ſino ſolo en el. No comuni
queys Aſtrologos, que no ay certi-
dumbre, antes confuſion, y mil tro-
piezos en ſu ciencia, mas dado caſo
que fuera ſegura ſi anuncian algun
bien, y ha de venir, es tormento el
eſperarle, y ſe eſtima en menos quã
do llega, y ſi no ſale cierto, lo es la
pena de tal engaño, pues ſi mal, para
que ſe ha de anticipar el ſentimien
to del, auiendo de llegar? y ſino por-
que ha de congojar lo que nunca
ſera?
Amad ſobre todo à Dios, eſtando
diſpueſto a dar por ſu Fè, y honra la
vida, como muchos Reyes, y Princi-
pes, à quien illuſtrò incomparable-
0
100
200
300
400
500
600
700
0
50
100
150
200
250
300
350
400

[ ]:
```
image_path = "/content/drive/MyDrive/ocr/test/page15/page15_1.png"
save_path = "/content/drive/MyDrive/ocr/extracted_text_image/pdf15_1"
create_folder_if_not_exists(save_path)
process_image(image_path, detection_model, save_path)
```

```
Folder '/content/drive/MyDrive/ocr/extracted_text_image/pdf15_1' created
successfully.
```

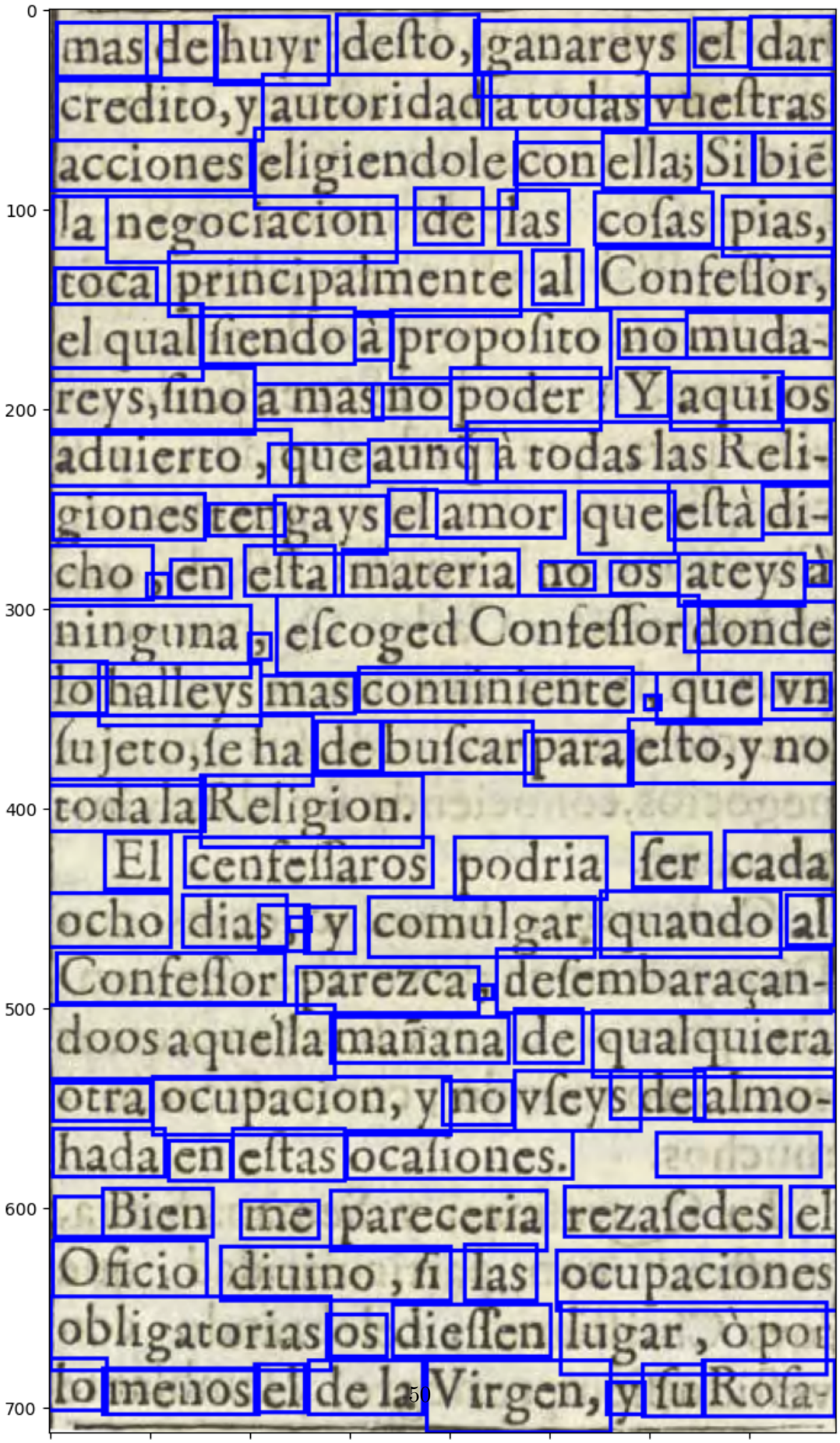


mente mas el derramar por esta cau-
sa su sangre, que el auerla heredado
tan generosa; y viuid determinado
de no perder ocasion en seruirle, y
cumplir su voluntad.

A vuestro Confessor (que escoge-
reys espiritual, docto, y hombre de
gran talento) tened mucho respeto,
y dadle autoridad para que os diga
libremente quantas verdades a vue-
stra alma importen en las cosas to-
cantes a ella, obedecedle entera-
mente con todo rendimiento, y tal
que no admitays razon para lo que
os ordenare en essas materias por
no perder el merito de la Fè, y obe-
diencia ciega (que aqui la deue auer)
tomad su consejo, pues escogiendo-
le con las partes dichas no aura peli-
gro de que abuse desto, metiendose
en el gouierno de todo, y querien-
do conseguir lo que pidiere justo, o
injusto (que es propiedad de igno-
rantes, y no muy espirituales) y à

[ ]:
```
image_path = "/content/drive/MyDrive/ocr/test/page15/page15_2.png"
save_path = "/content/drive/MyDrive/ocr/extracted_text_image/pdf15_2"
create_folder_if_not_exists(save_path)
process_image(image_path, detection_model, save_path)
```

```
Folder '/content/drive/MyDrive/ocr/extracted_text_image/pdf15_2' created
successfully.
```



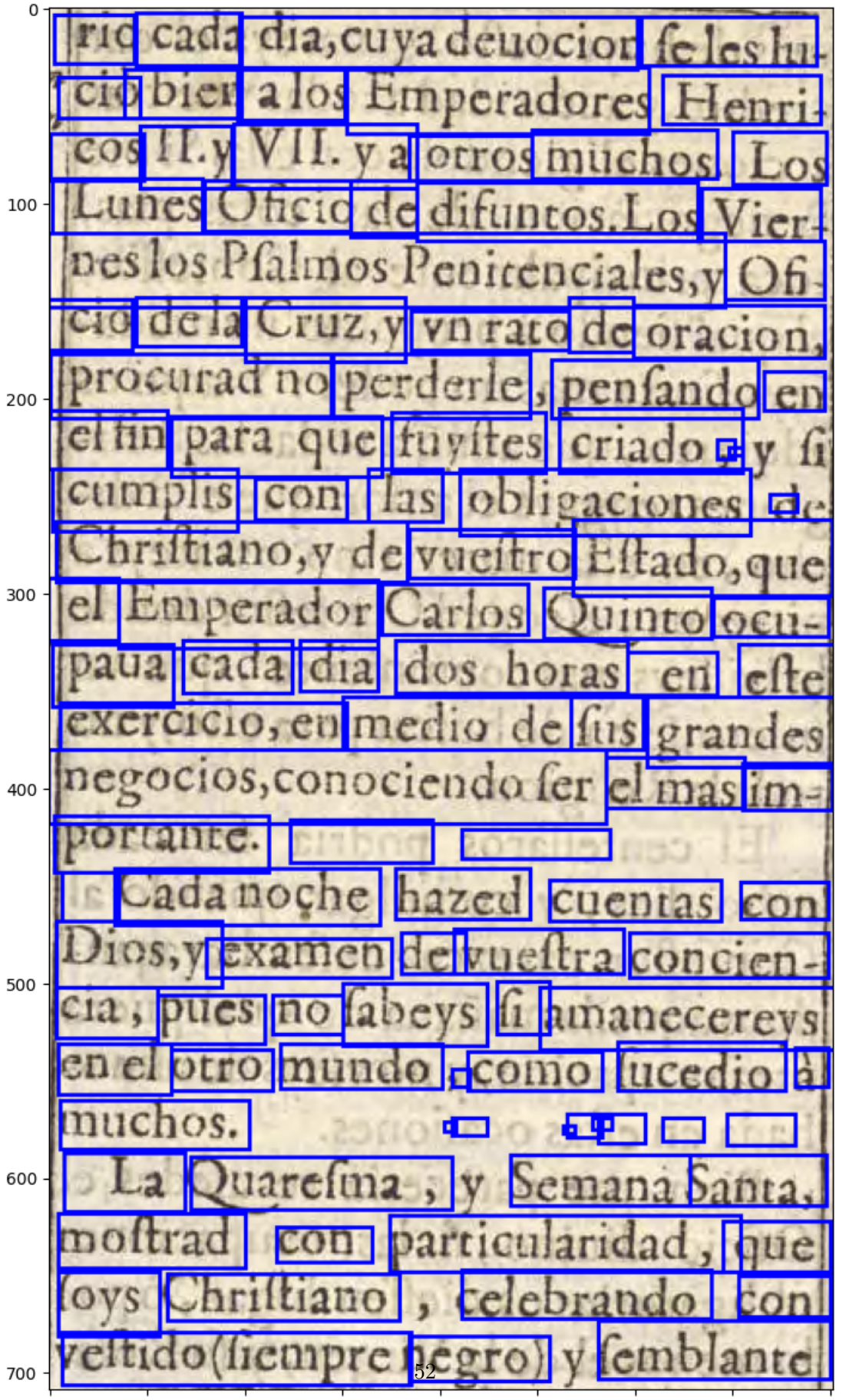
mas de huyr deſto, ganareys el dar
credito, y autoridad a todas vueſtras
acciones eligiendole con ella; Si biẽ
la negociacion de las coſas pias,
toca principalmente al Confeſſor,
el qual ſiendo à propoſito no muda-
reys, ſino a mas no poder. Y aqui os
aduierto, que aunq; à todas las Reli-
giones tengays el amor que eſtà di-
cho, en eſta materia no os ateys à
ninguna, eſcoged Confeſſor donde
lo halleys mas conuiniente, que vn
ſujeto, ſe ha de buſcar para eſto, y no
toda la Religion.

El confeſſaros podria ſer cada
ocho dias, y comulgar quando al
Confeſſor parezca, deſembaraçan-
doos aquella mañana de qualquiera
otra ocupacion, y no vſeys de almo-
hada en eſtas ocaſiones.

Bien me pareceria rezaſedes el
Oficio diuino, ſi las ocupaciones
obligatorias os dieſſen lugar, ò por
lo menos el de la Virgen, y ſu Roſa-

[ ]:
```
image_path = "/content/drive/MyDrive/ocr/test/page16/page16_1.png"
save_path = "/content/drive/MyDrive/ocr/extracted_text_image/pdf16_1"
create_folder_if_not_exists(save_path)
process_image(image_path, detection_model, save_path)
```

```
Folder '/content/drive/MyDrive/ocr/extracted_text_image/pdf16_1' created
successfully.
```



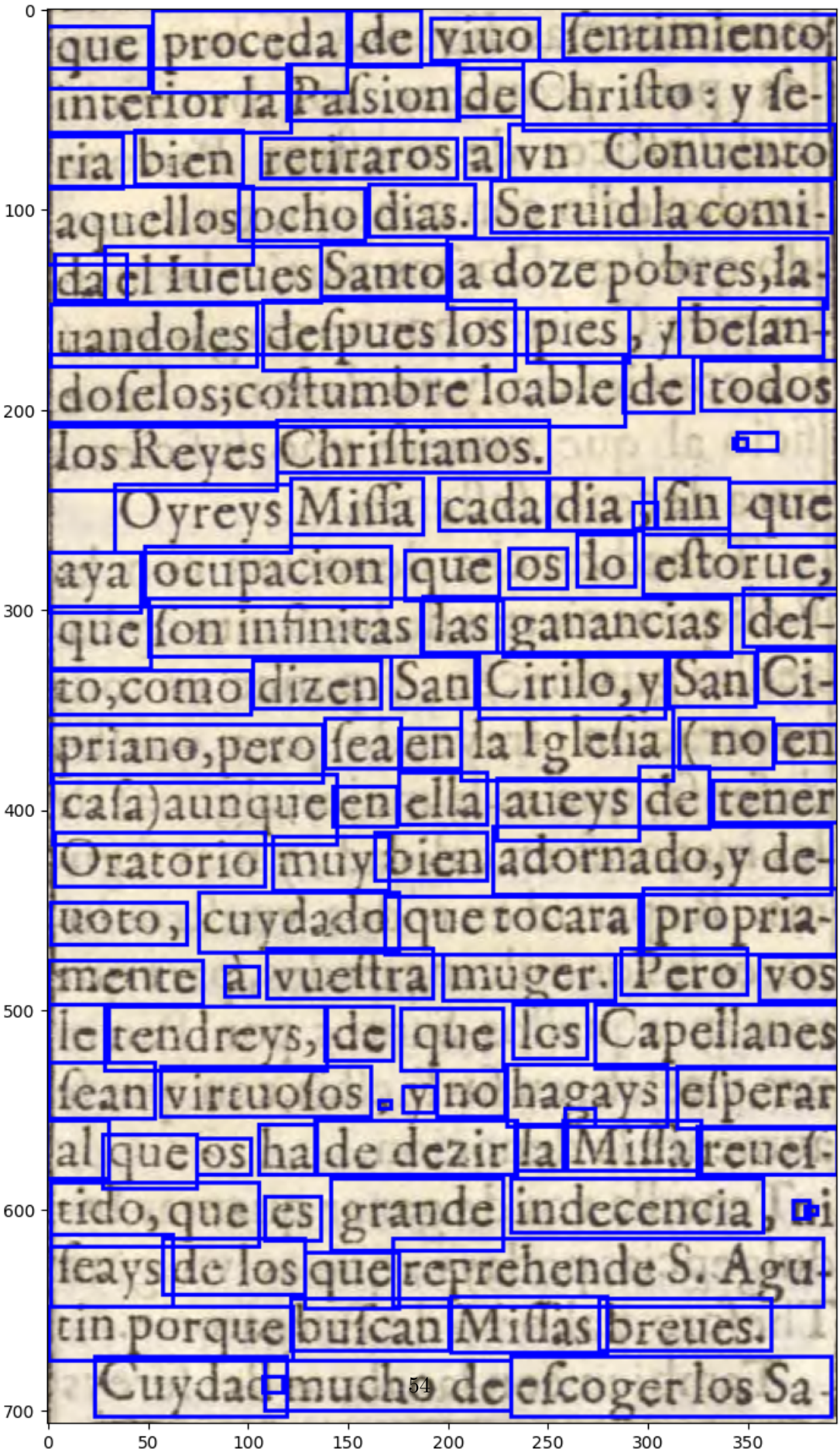
ric cada dia, cuya deuocion fe les hu-
cio bien a los Emperadores Henri-
cos II. y VII. y a otros muchos. Los
Lunes Oficio de difuntos. Los Vier-
nes los Pſalmos Penitenciales, y Ofi-
cio de la Cruz, y vn rato de oracion,
procurad no perderle, penſando en
el fin para que fuyſtes criado, y ſi
cumplis con las obligaciones de
Chriſtiano, y de vueſtro Eſtado, que
el Emperador Carlos Quinto ocu-
paua cada dia dos horas en eſte
exercicio, en medio de ſus grandes
negocios, conociendo ſer el mas im-
portante.

Cada noche hazed cuentas con
Dios, y examen de vueſtra concien-
cia, pues no ſabeys ſi amanecereys
en el otro mundo, como ſucedio a
muchos.

La Quareſma, y Semana Santa,
moſtrad con particularidad, que
ſoys Chriſtiano, celebrando con
veſtido (ſiempre negro) y ſemblante

[ ]:
```
image_path = "/content/drive/MyDrive/ocr/test/page16/page16_2.png"
save_path = "/content/drive/MyDrive/ocr/extracted_text_image/pdf16_2"
create_folder_if_not_exists(save_path)
process_image(image_path, detection_model, save_path)
```

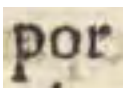
```
Folder '/content/drive/MyDrive/ocr/extracted_text_image/pdf16_2' created
successfully.
```


que proceda de viuo ſentimiento
interior la Paſsion de Chriſto: y ſe-
ria bien retiraros a vn Conuento
aquellos ocho dias. Seruid la comi-
da el Iueues Santo a doze pobres, la-
uandoles deſpues los pies, y belan-
doſelos; coſtumbre loable de todos
los Reyes Chriſtianos.
Oyreys Miſſa cada dia, ſin que
aya ocupacion que os lo eſtoruе,
que ſon infinitas las ganancias deſ-
to, como dizen San Cirilo, y San Ci-
priano, pero ſea en la Igleſia (no en
caſa) aunque en ella aueys de tener
Oratorio muy bien adornado, y de-
uoto, cuydado que tocara propria-
mente a vueſtra muger. Pero vos
le tendreys, de que los Capellanes
ſean virtuoſos. y no hagays eſperar
al que os ha de dezir la Miſſa reueſ-
tido, que es grande indecencia, ni
ſeays de los que reprehende S. Agu-
tin porque buſcan Miſſas breues.
Cuydad mucho de eſcoger los Sa-
0
50
100
150
200
250
300
350
0
100
200
300
400
500
600
700

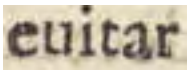
[ ]:
```
directory_path = '/content/drive/MyDrive/ocr/extracted_text_image/pdf14_1'
result14_1 = process_image_folders(directory_path, model, processor)
result14_1
```

```
['text_0.png', 'text_1.png', 'text_2.png', 'text_3.png', 'text_4.png',
'text_5.png', 'text_6.png', 'text_7.png', 'text_8.png', 'text_9.png',
'text_10.png', 'text_11.png', 'text_12.png', 'text_13.png', 'text_14.png',
'text_15.png', 'text_16.png', 'text_17.png', 'text_18.png', 'text_19.png',
'text_20.png', 'text_21.png', 'text_22.png', 'text_23.png', 'text_25_1.png',
'text_26.png', 'text_27.png', 'text_28.png', 'text_29.png', 'text_30.png',
'text_31.png', 'text_32.png', 'text_33.png', 'text_34.png', 'text_35.png',
'text_36.png', 'text_37.png', 'text_38-1.png', 'text_38-2.png', 'text_39.png',
'text_40.png', 'text_45.png', 'text_46.png', 'text_48.png', 'text_49.png',
'text_50.png', 'text_51.png', 'text_52.png', 'text_53.png', 'text_54.png',
'text_56.png', 'text_57-1.png', 'text_57-2.png', 'text_59.png', 'text_60.png',
'text_61.png', 'text_62.png', 'text_65.png', 'text_66.png', 'text_67-1.png',
'text_67-2.png', 'text_68-1.png', 'text_68-2.png', 'text_69.png', 'text_70.png',
'text_71.png', 'text_72.png', 'text_73.png', 'text_74.png', 'text_75.png',
'text_76.png', 'text_77.png', 'text_78.png', 'text_79.png', 'text_81.png',
'text_82.png', 'text_83.png', 'text_84.png', 'text_85.png', 'text_86.png',
'text_87-1.png', 'text_87-2.png', 'text_88-1.png', 'text_88-2.png',
'text_88-3.png', 'text_89.png', 'text_90.png', 'text_91.png', 'text_92-1.png',
'text_92-2.png', 'text_93.png', 'text_95.png', 'text_96.png', 'text_97.png',
'text_99.png', 'text_100.png', 'text_101.png', 'text_102.png', 'text_103.png',
'text_104.png', 'text_105.png', 'text_106.png', 'text_107.png', 'text_108.png',
'text_109.png', 'text_110.png', 'text_111.png', 'text_112-1.png',
'text_112-2.png', 'text_113.png', 'text_114.png', 'text_115.png',
'text_116.png', 'text_117.png', 'text_118.png', 'text_119.png', 'text_120.png',
'text_121.png', 'text_122.png', 'text_123.png', 'text_124.png', 'text_125.png',
'text_126.png', 'text_127.png', 'text_128.png', 'text_129.png', 'text_130.png',
'text_131.png', 'text_132.png', 'text_133.png', 'text_134.png', 'text_135.png',
'text_136.png', 'text_137.png', 'text_138.png', 'text_139.png', 'text_140.png',
'text_141.png', 'text_142.png', 'text_143.png', 'text_144.png']
```

```
si
```

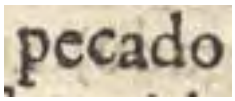


por



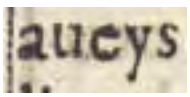
quitar

un



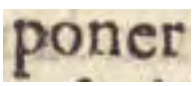
pecado

morta

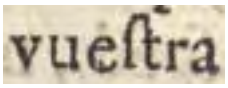


aueys

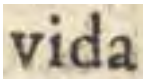
de



poner



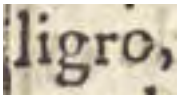
vuetra



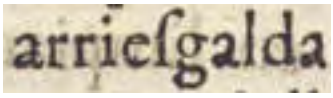
vida

en

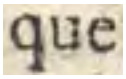
pe



ligro



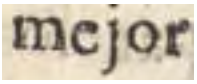
arrigala



que

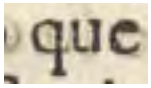
es

el

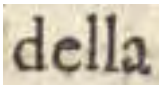


mejor

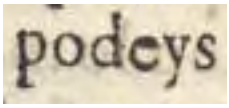
emplo



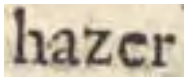
que



della

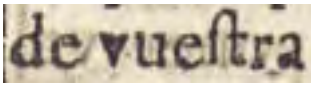


podeys



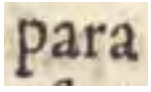
hazer

y

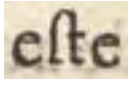


devuéra

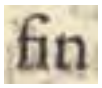
hazienda



para

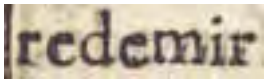


efte

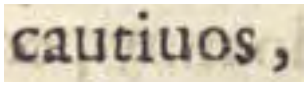


fin

en

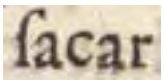


redemir

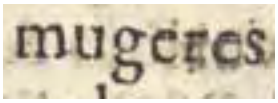


cautuos

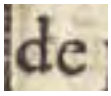
y



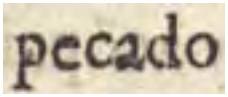
facar



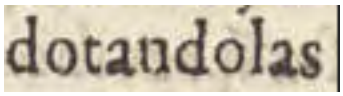
mugeres



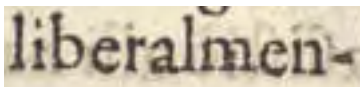
de



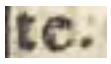
pecado



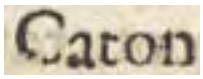
dotandolas



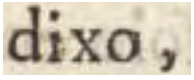
liberalmen



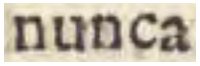
le



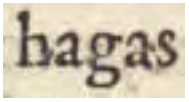
caton



dixó

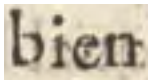


nunca

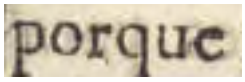


hagas

el

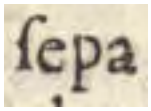


bien

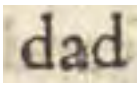


porque

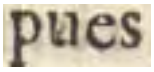
fe



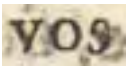
fepa



dad

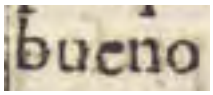


pues

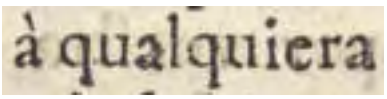


vos

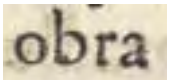
hin



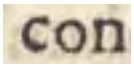
bueno



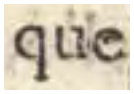
aqualquiera



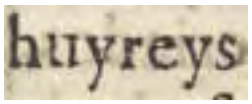
obra



con



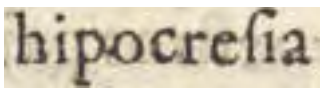
que



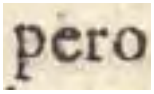
huyreys

de

ha

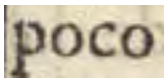


hiporéa

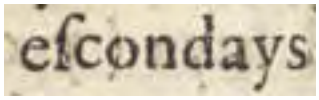


pero

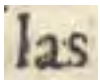
tami



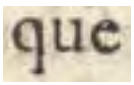
poco



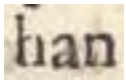
efcondays



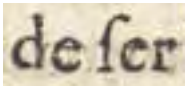
las



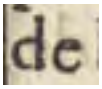
que



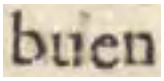
han



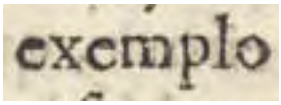
defer



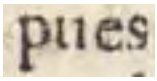
de



buien

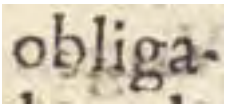


exemplo

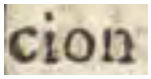


pues

es

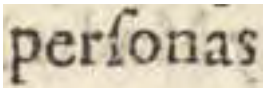


obliga

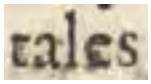


cion

de

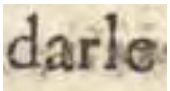


perfoñas



tales

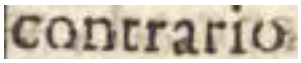
el



darlo

y

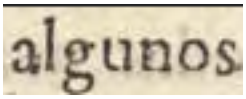
lo



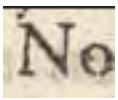
contranio

tentacion

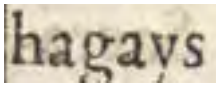
en



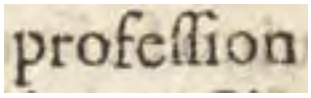
algunos



nó

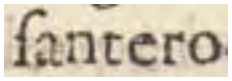


hagays



profellon

de



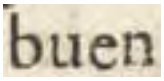
fantero

pe

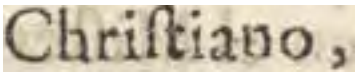
ro

íi

de

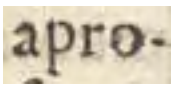


buen

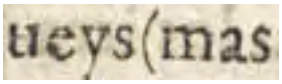


christiano

no

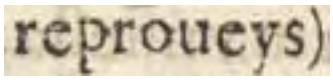


apro

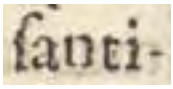


ueyśmas

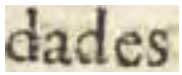
tampoco



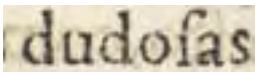
reproueys



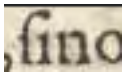
fauti



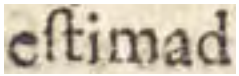
dades



mudofas

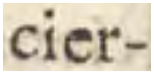


fino

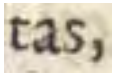


efimad

las

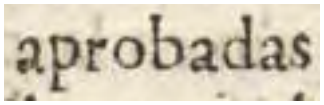


cier



tas

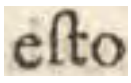
y



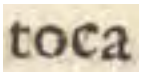
aprobadas

y

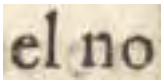
a



effo

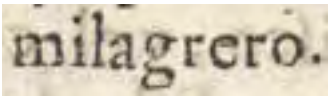


toca



elno

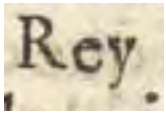
fer



mibalrero

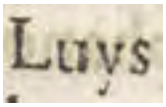
acordaos

del

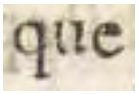


rey

s

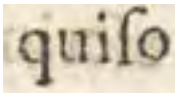


luys



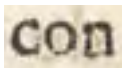
que

no



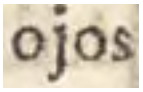
quillo

ver



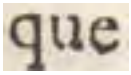
con

los

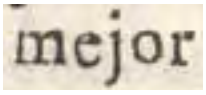


ojos

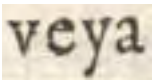
lo



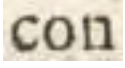
que



mejor

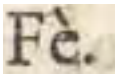


veya



con

la



fe

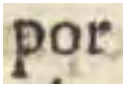
[ ]:
```
'si por quitar un pecado morta aueys de poner vuetra vida en pe ligro arrigala
que es el mejor emplo que della podeys hazer y devuéra hazienda para efte fin
en redemir cautuos y facar mugeres de pecado dotandolas liberalmen le caton
dixó nunca hagas el bien porque fe fepa dad pues vos hin bueno aqualquiera obra
con que huyreys de ha hiporéa pero tami poco efcondays las que han defer de
buien exemplo pues es obliga cion de perfoñas tales el darlo y lo contranio
tentacion en algunos nó hagays profellon de fantero pe ro íi de buen
christiano no apro ueyśmas tampoco reproueys fauti dades mudofas fino efimad
las cier tas y aprobadas y a effo toca elno fer mibalrero acordaos del rey s
luys que no quillo ver con los ojos lo que mejor veya con la fe'
```

[ ]:
```
directory_path = '/content/drive/MyDrive/ocr/extracted_text_image/pdf14_2'
result14_2 = process_image_folders(directory_path, model, processor)
result14_2
```

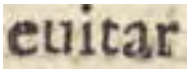
```
['text_0.png', 'text_1.png', 'text_2.png', 'text_3.png', 'text_4.png',
'text_5.png', 'text_6.png', 'text_7.png', 'text_8.png', 'text_9.png',
'text_10.png', 'text_11.png', 'text_12.png', 'text_13.png', 'text_14.png',
'text_15.png', 'text_16.png', 'text_17.png', 'text_18.png', 'text_19.png',
'text_20.png', 'text_21.png', 'text_22.png', 'text_23.png', 'text_25_1.png',
'text_26.png', 'text_27.png', 'text_28.png', 'text_29.png', 'text_30.png',
'text_31.png', 'text_32.png', 'text_33.png', 'text_34.png', 'text_35.png',
'text_36.png', 'text_37.png', 'text_38-1.png', 'text_38-2.png', 'text_39.png',
'text_40.png', 'text_45.png', 'text_46.png', 'text_48.png', 'text_49.png',
'text_50.png', 'text_51.png', 'text_52.png', 'text_53.png', 'text_54.png',
'text_56.png', 'text_57-1.png', 'text_57-2.png', 'text_59.png', 'text_60.png',
'text_61.png', 'text_62.png', 'text_65.png', 'text_66.png', 'text_67-1.png',
'text_67-2.png', 'text_68-1.png', 'text_68-2.png', 'text_69.png', 'text_70.png',
'text_71.png', 'text_72.png', 'text_73.png', 'text_74.png', 'text_75.png',
'text_76.png', 'text_77.png', 'text_78.png', 'text_79.png', 'text_81.png',
'text_82.png', 'text_83.png', 'text_84.png', 'text_85.png', 'text_86.png',
'text_87-1.png', 'text_87-2.png', 'text_88-1.png', 'text_88-2.png',
```

```
'text_88-3.png', 'text_89.png', 'text_90.png', 'text_91.png', 'text_92-1.png',
'text_92-2.png', 'text_93.png', 'text_95.png', 'text_96.png', 'text_97.png',
'text_99.png', 'text_100.png', 'text_101.png', 'text_102.png', 'text_103.png',
'text_104.png', 'text_105.png', 'text_106.png', 'text_107.png', 'text_108.png',
'text_109.png', 'text_110.png', 'text_111.png', 'text_112-1.png',
'text_112-2.png', 'text_113.png', 'text_114.png', 'text_115.png',
'text_116.png', 'text_117.png', 'text_118.png', 'text_119.png', 'text_120.png',
'text_121.png', 'text_122.png', 'text_123.png', 'text_124.png', 'text_125.png',
'text_126.png', 'text_127.png', 'text_128.png', 'text_129.png', 'text_130.png',
'text_131.png', 'text_132.png', 'text_133.png', 'text_134.png', 'text_135.png',
'text_136.png', 'text_137.png', 'text_138.png', 'text_139.png', 'text_140.png',
'text_141.png', 'text_142.png', 'text_143.png', 'text_144.png']
```

si

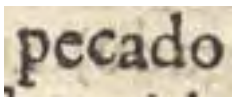


por



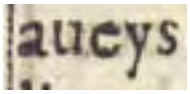
quitar

un



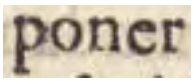
pecado

morta

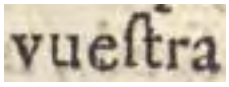


aueys

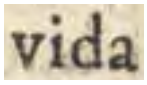
de



poner



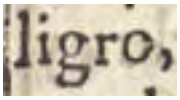
vuetra



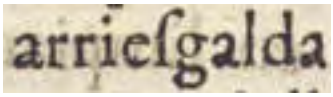
vida

en

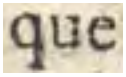
pe



ligro



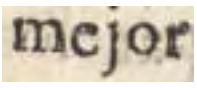
arrigala



que

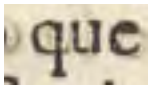
es

el

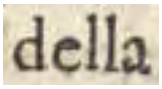


mejor

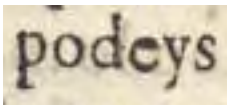
emplo



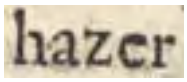
que



della

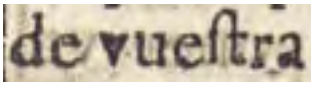


podeys



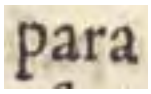
hazer

y

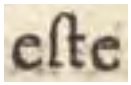


devuéra

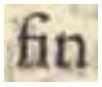
hazienda



para

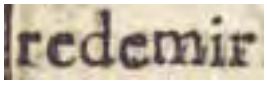


efte

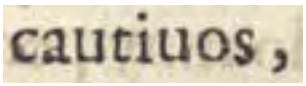


fin

en

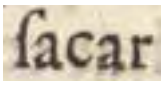


redemir

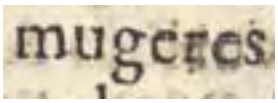


cautuos

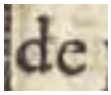
y



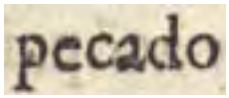
facar



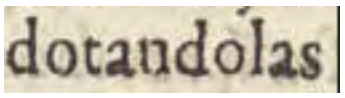
mugeres



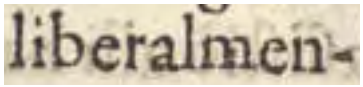
de



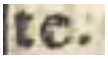
pecado



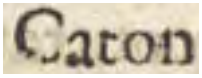
dotandolas



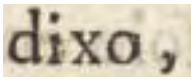
liberalmen



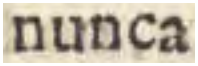
le



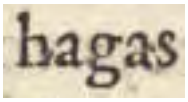
caton



dixó

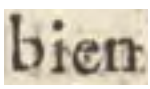


nunca

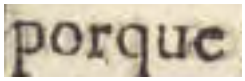


hagas

el

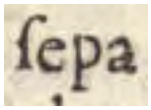


bien

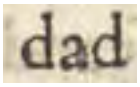


porque

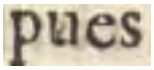
fe



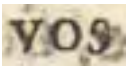
fepa



dad

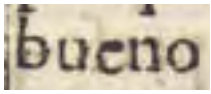


pues

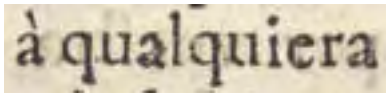


vos

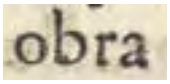
hin



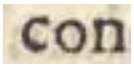
bueno



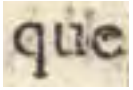
aqualquiera



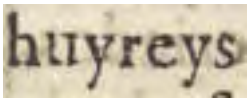
obra



con



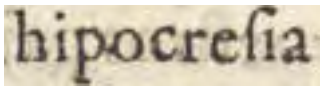
que



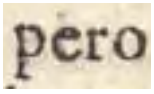
huyreys

de

ha

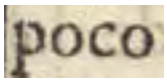


hiporéa

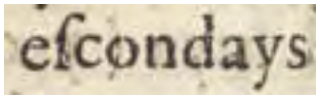


pero

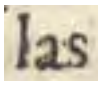
tami



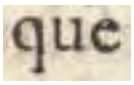
poco



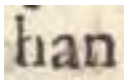
efcondays



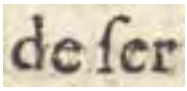
las



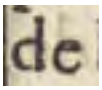
que



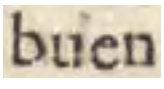
han



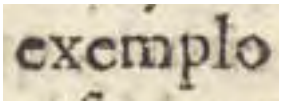
defer



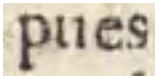
de



buien

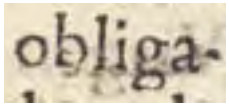


exemplo

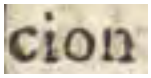


pues

es

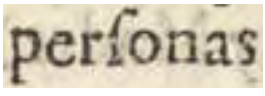


obliga

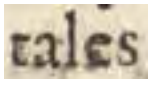


cion

de

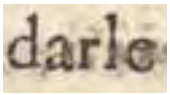


perfoñas



tales

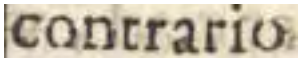
el



darlo

y

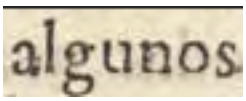
lo



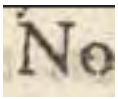
contranio

tentacion

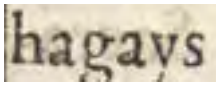
en



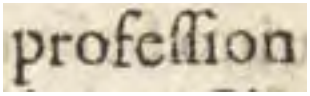
algunos



nó

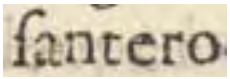


hagays



profellon

de



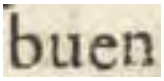
fantero

pe

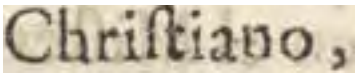
ro

íi

de

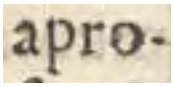


buen

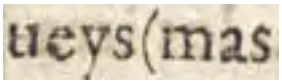


christiano

no

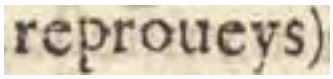


apro

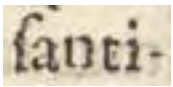


ueyśmas

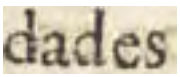
tampoco



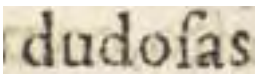
reproueys



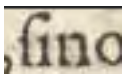
fauti



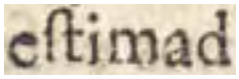
dades



mudofas

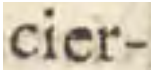


fino

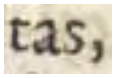


efimad

las

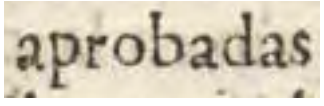


cier



tas

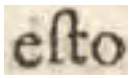
y



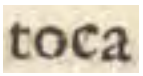
aprobadas

y

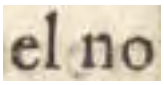
a



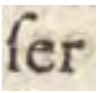
effo



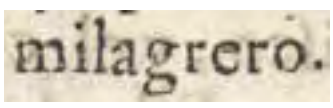
toca



elno



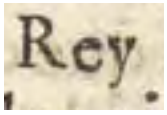
fer



mibalrero

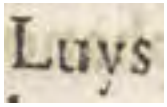
acordaos

del

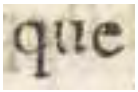


rey

s

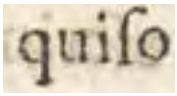


luys



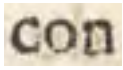
que

no



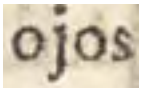
quillo

ver



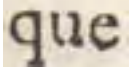
con

los

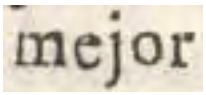


ojos

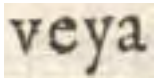
lo



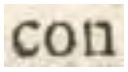
que



mejor

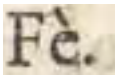


veya



con

la

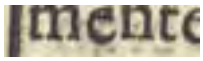


fe

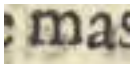
[ ]: 'si por quitar un pecado morta aueys de poner vuetra vida en pe ligro arrigala que es el mejor emplo que della podeys hazer y devuéra hazienda para efte fin en redemir cautuos y facar mugeres de pecado dotandolas liberalmen le caton dixó nunca hagas el bien porque fe fepa dad pues vos hin bueno aqualquiera obra con que huyreys de ha hiporéa pero tami poco efcondays las que han defer de buien exemplo pues es obliga cion de perfoñas tales el darlo y lo contranio tentacion en algunos nó hagays profellon de fantero pe ro íi de buen christiano no apro ueyśmas tampoco reproueys fauti dades mudofas fino efimad las cier tas y aprobadas y a effo toca elno fer mibalrero acordaos del rey s luys que no quillo ver con los ojos lo que mejor veya con la fe'

[ ]:
```
directory_path = '/content/drive/MyDrive/ocr/extracted_text_image/pdf15_1'
result15_1 = process_image_folders(directory_path, model, processor)
result15_1
```

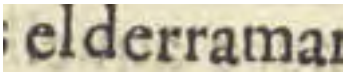
```
['text_0.png', 'text_1.png', 'text_2.png', 'text_3.png', 'text_4.png',
'text_5.png', 'text_6.png', 'text_7.png', 'text_8.png', 'text_9.png',
'text_10.png', 'text_11.png', 'text_12.png', 'text_13.png', 'text_14.png',
'text_15.png', 'text_16.png', 'text_17.png', 'text_18.png', 'text_19.png',
'text_20.png', 'text_21.png', 'text_25.png', 'text_28.png', 'text_29.png',
'text_30.png', 'text_31.png', 'text_32-1.png', 'text_32-2.png', 'text_33.png',
'text_34.png', 'text_35.png', 'text_36.png', 'text_37.png', 'text_38.png',
'text_39.png', 'text_40.png', 'text_41.png', 'text_42.png', 'text_43.png',
'text_44.png', 'text_45.png', 'text_46.png', 'text_47.png', 'text_48.png',
'text_49.png', 'text_50.png', 'text_51.png', 'text_52.png', 'text_53.png',
'text_54.png', 'text_55.png', 'text_56.png', 'text_57.png', 'text_58.png',
'text_60.png', 'text_61.png', 'text_62.png', 'text_63.png', 'text_64.png',
'text_65.png', 'text_66.png', 'text_68.png', 'text_69.png', 'text_70.png',
'text_71.png', 'text_72.png', 'text_73.png', 'text_75.png', 'text_76.png',
'text_77.png', 'text_78.png', 'text_79.png', 'text_80.png', 'text_81.png',
'text_82.png', 'text_83.png', 'text_84.png', 'text_85.png', 'text_86.png',
'text_87.png', 'text_88.png', 'text_89.png', 'text_90.png', 'text_91.png',
'text_92-1.png', 'text_92-2.png', 'text_93.png', 'text_94-1.png',
'text_94-2.png', 'text_95.png', 'text_96.png', 'text_97-1.png', 'text_97-2.png',
'text_97-3.png', 'text_97-4.png', 'text_97-5.png', 'text_98.png', 'text_99.png',
'text_100.png', 'text_101.png', 'text_102.png', 'text_103.png', 'text_104.png',
'text_105.png', 'text_106.png', 'text_107.png', 'text_108.png', 'text_109.png',
'text_110.png', 'text_111.png', 'text_112.png', 'text_113.png', 'text_114.png',
'text_115.png', 'text_116.png', 'text_117.png', 'text_118.png', 'text_119.png',
'text_120.png', 'text_121-1.png', 'text_121-2.png', 'text_122.png',
'text_123.png', 'text_124.png', 'text_125.png', 'text_126.png', 'text_127.png',
'text_128.png', 'text_131.png', 'text_132.png', 'text_133.png', 'text_134.png',
'text_135.png', 'text_136.png', 'text_137.png', 'text_139.png', 'text_140.png',
'text_141.png', 'text_142.png']
```



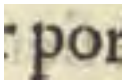
```
imente
```



```
mas
```

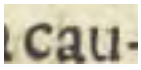


ederrama

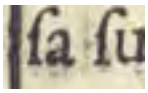


por

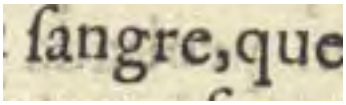
efa



cau

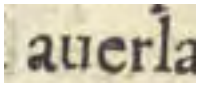


faflu

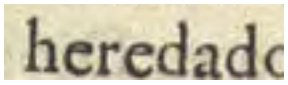


fangreyque

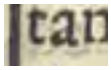
elel



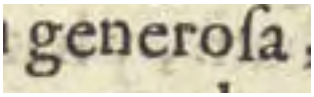
auerla



heredado

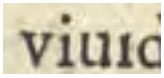


itan

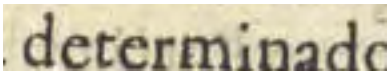


generolá

y



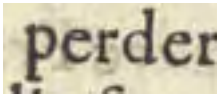
vivio



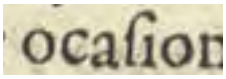
defeminado

ide

ino

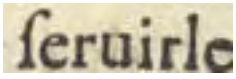


perder

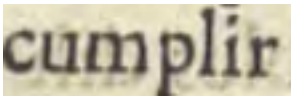


ocacion

en

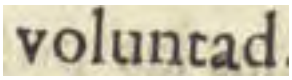


fervirle



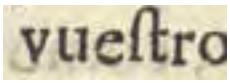
cumplir

flu

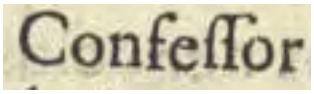


voluntad

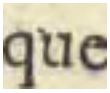
a



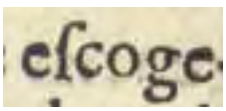
vuestro



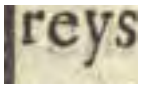
confer



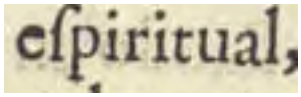
que



elege



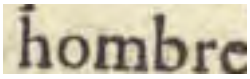
reys



eipiritual

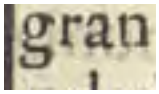
docto

yy

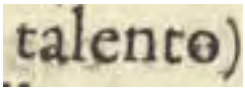


hombre

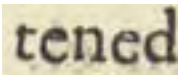
dé



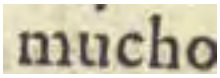
gran



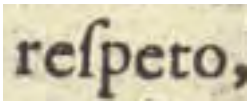
talento



tened

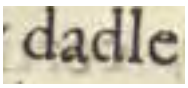


mucho

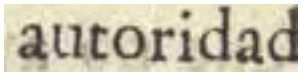


refeto

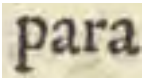
y



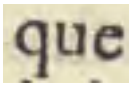
dadle



autoridad

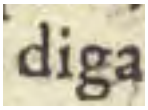


para

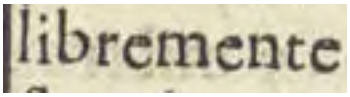


que

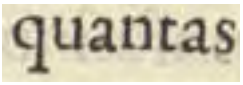
os



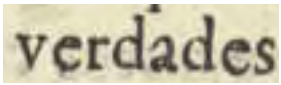
diga



libremente

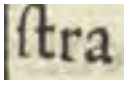


quantas

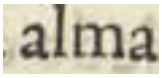


verdades

ave



frra

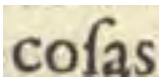


alma

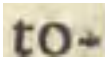
importanten

en

las



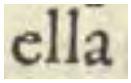
colas



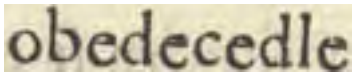
tos

cantes

a



ella

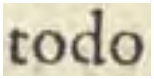


obedecedle

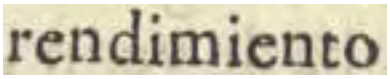
enterá

mente

con



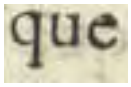
todo



rendimiento

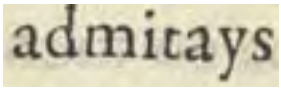
y

tal

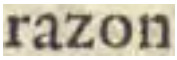


que

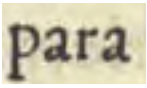
no



admitays

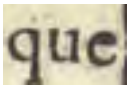


razon



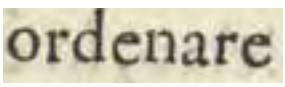
para

ló



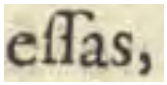
que

os

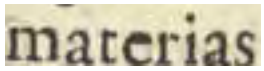


ordenerre

en

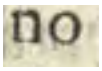


ellas

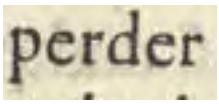


materias

por

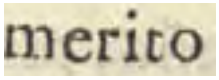


no

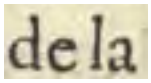


perder

el



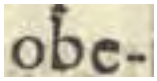
merito



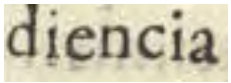
dela

fe

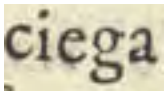
yy



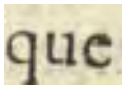
obe



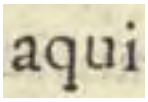
diencia



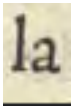
clega



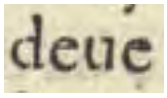
que



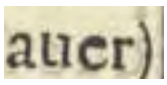
aqui



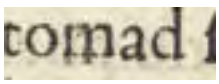
la



deue

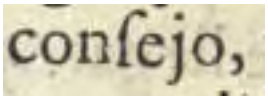


auer)

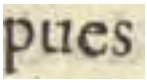


tomad

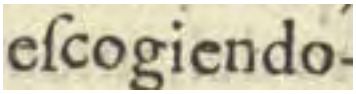
lu



confejo

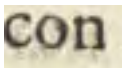


pues



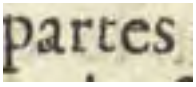
eologiendó

le

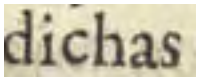


con

las

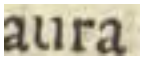


partes

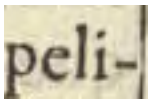


dichas

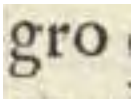
uo



aura

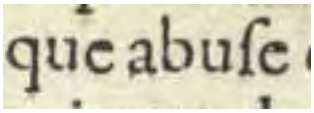


pelli

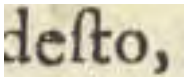


gro

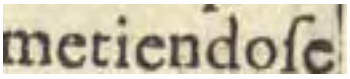
de



qúe



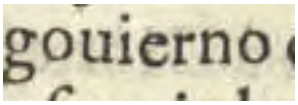
defo



metiendole

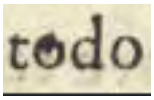
en

el



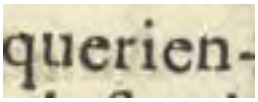
gobierno

de

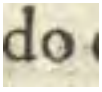


todo

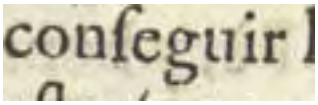
y



querien

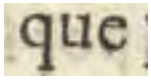


do

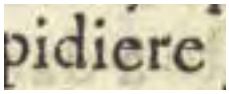


confequirl

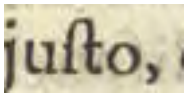
lo



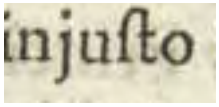
que



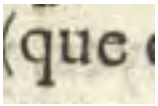
pidiéri



julto

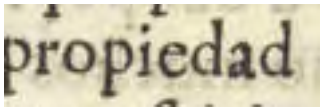


injulto



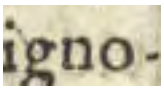
que

es

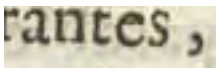


propiedad

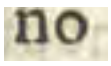
dé



igno

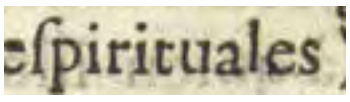


santes



no

muy



epiritales

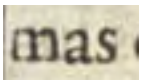
yy

[ ]: 'imente mas ederrama por efa cau faflu fangreyque elel auerla heredado itan generolá y vivio defeminado ide ino perder ocacion en fervirle cumplir flu voluntad a vuestro confer que elege reys eipiritual docto yy hombre dé gran talento tened mucho refeto y dadle autoridad para que os diga libremente quantas verdades ave frra alma importanten en las colas tos cantes a ella obedecedle enterá mente con todo rendimiento y tal que no admitays razon para ló que os ordenerre en ellas materias por no perder el merito dela fe yy obe diencia clega que aqui la deue auer) tomad lu confejo pues eologiendó le con las partes dichas uo aura pelli gro de qúe defo metiendole en el gobierno de todo y querien do confequirl lo que pidiéri julto injulto que es propiedad dé igno santes no muy epiritales yy'

[ ]:
```
directory_path = '/content/drive/MyDrive/ocr/extracted_text_image/pdf15_2'
result15_2 = process_image_folders(directory_path, model, processor)
result15_2
```

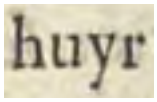
```
['text_0.png', 'text_1.png', 'text_2.png', 'text_3.png', 'text_4.png',
'text_5.png', 'text_6.png', 'text_7-1.png', 'text_7-2.png', 'text_8.png',
'text_9.png', 'text_10.png', 'text_11.png', 'text_12.png', 'text_13.png',
'text_14.png', 'text_15.png', 'text_16.png', 'text_17.png', 'text_18.png',
'text_19.png', 'text_20.png', 'text_21.png', 'text_22.png', 'text_23.png',
'text_24.png', 'text_25.png', 'text_26.png', 'text_27-1.png', 'text_27-2.png',
'text_28.png', 'text_29.png', 'text_30.png', 'text_31.png', 'text_32.png',
'text_33-1.png', 'text_33-2.png', 'text_34.png', 'text_35.png', 'text_36.png',
'text_37.png', 'text_38.png', 'text_39.png', 'text_40.png', 'text_41.png',
'text_42.png', 'text_43-1.png', 'text_43-2.png', 'text_43-3.png',
'text_43-4.png', 'text_44.png', 'text_45.png', 'text_46.png', 'text_47.png',
'text_48.png', 'text_49.png', 'text_50.png', 'text_51.png', 'text_52.png',
'text_54.png', 'text_55.png', 'text_56.png', 'text_57.png', 'text_58.png',
'text_59.png', 'text_60.png', 'text_61.png', 'text_63-1.png', 'text_63-2.png',
'text_64.png', 'text_65.png', 'text_66.png', 'text_67.png', 'text_68.png',
'text_70.png', 'text_71.png', 'text_72-1.png', 'text_72-2.png', 'text_72-3.png',
'text_73.png', 'text_74.png', 'text_75.png', 'text_76-1.png', 'text_76-2.png',
'text_76-3.png', 'text_77-1.png', 'text_77-2.png', 'text_78.png', 'text_79.png',
'text_80.png', 'text_81.png', 'text_82.png', 'text_83.png', 'text_84.png',
'text_85.png', 'text_88.png', 'text_89.png', 'text_90.png', 'text_91.png',
'text_92.png', 'text_93.png', 'text_95.png', 'text_96.png', 'text_97.png',
'text_98.png', 'text_99.png', 'text_100.png', 'text_101-1.png',
'text_101-2.png', 'text_102.png', 'text_103.png', 'text_104.png',
'text_105.png', 'text_106.png', 'text_107.png', 'text_108.png', 'text_109.png',
```

```
'text_112.png', 'text_113.png', 'text_114.png', 'text_115.png', 'text_116.png',
'text_117.png', 'text_118-1.png', 'text_118-2.png', 'text_119.png',
'text_120.png', 'text_121.png', 'text_122.png', 'text_123.png', 'text_124.png',
'text_125.png', 'text_126.png', 'text_127.png', 'text_128.png', 'text_129.png',
'text_130.png', 'text_131.png', 'text_132.png']
```

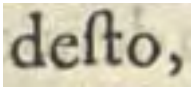



```
mas
```

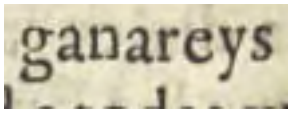

```
de
```




```
huyr
```

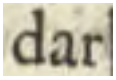



```
defó
```

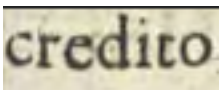



```
ganareys
```

el

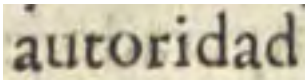


dar

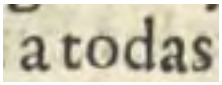


credito

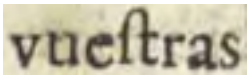
oy



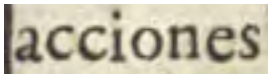
autoridad



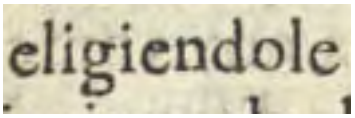
atodas



vuetras

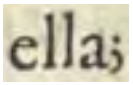


acciones



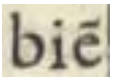
eigiendole

con



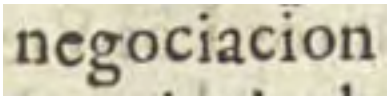
ellas

si



bie

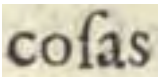
lla



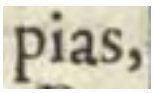
negocacion

de

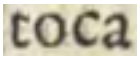
las



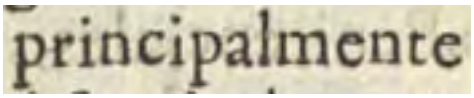
colas



pias

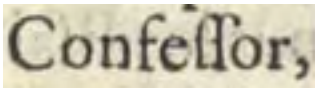


toca



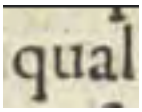
principalmente

al

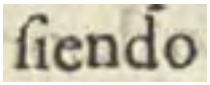


conferior

el

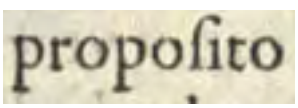


qual



tiendo

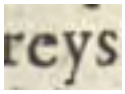
á



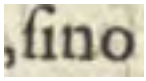
propõo

no

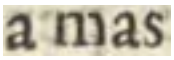
muda



reys

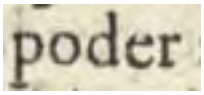


fluo



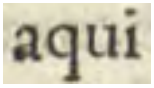
armas

no



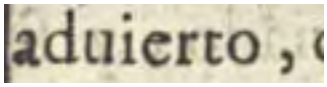
poder

y

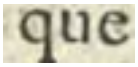


aqui

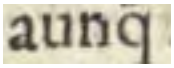
os



aduierto

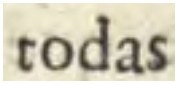


que



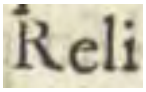
aunqu

a

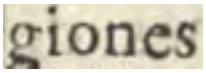


todas

las

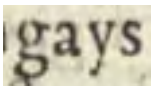


reli



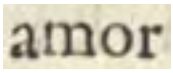
giones

ten

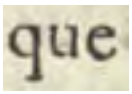


gays

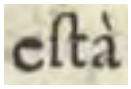
el



amor

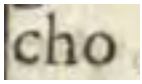


que



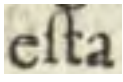
efa

dí

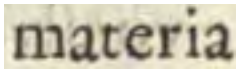


cho

en



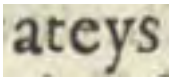
ẽa



materia

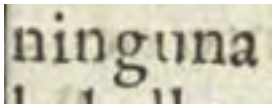
no

os

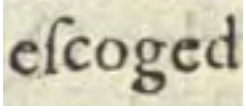


aleys

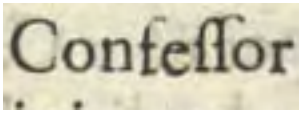
a



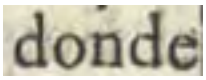
ninguna



eleged

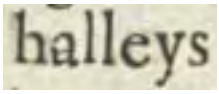


confer

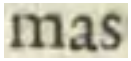


doñde

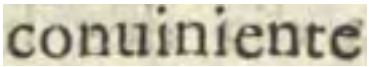
lo



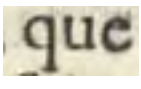
halleys



mas

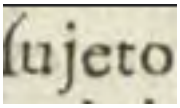


conveniente



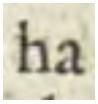
que

un



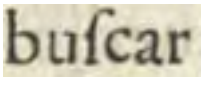
fuyeto

lée

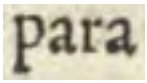


ha

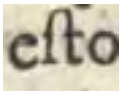
de



bufcar



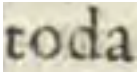
para



efõo

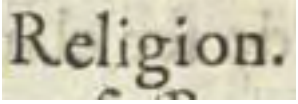
y

no



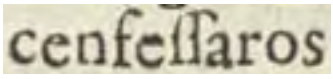
toda

la

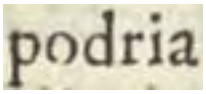


religion

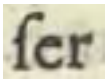
el



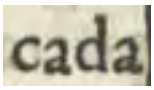
confẽaros



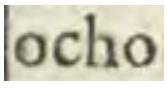
podria



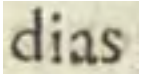
fer



cada

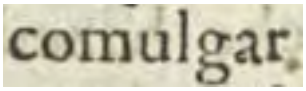


ocho

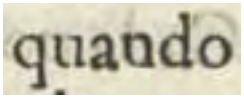


dias

y

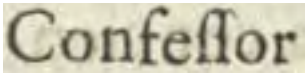


comulgar

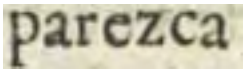


quado

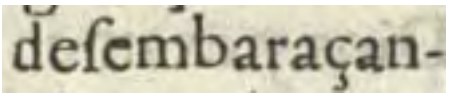
al



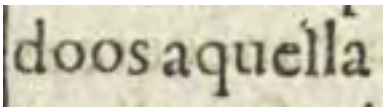
confer



parezca



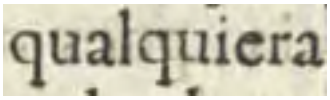
defeñan



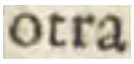
dosaellá

mañava

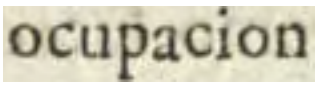
de



qualquiera



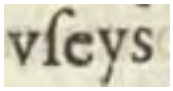
otra



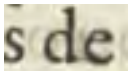
ocupacion

y

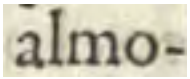
no



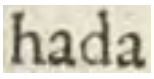
vieys



sie

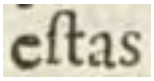


almo

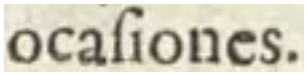


hada

en



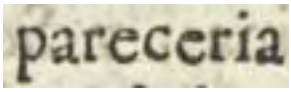
efas



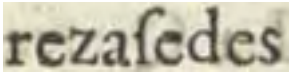
ocallones

bien

me

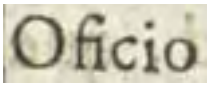


parecia



rezafes

el

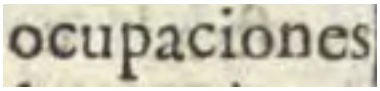


oficio

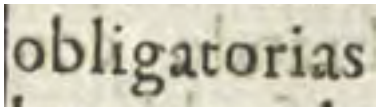
divino

ĩi

las

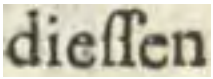


ocupaciones

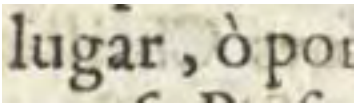


obligatorias

os



diedien

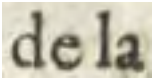


lugarpora

lo

meuos

el

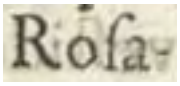


dela

virgen

y

flu



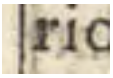
rĩar

[ ]:
```
'mas de huyr defó ganareys el dar credito oy autoridad atodas vuetras acciones
eigiendole con ellas si bie lla negocacion de las colas pias toca principalmente
al conferior el qual tiendo á propõo no muda reys fluo armas no poder y aqui
os aduierto que aunqu a todas las reli giones ten gays el amor que efa dí cho
en ẽa materia no os aleys a ninguna eleged confer doñde lo halleys mas
conveniente que un fuyeto lée ha de bufcar para efõo y no toda la religion el
confẽaros podria fer cada ocho dias y comulgar quado al confer parezca defeñan
dosaellá mañava de qualquiera otra ocupacion y no vieys sie almo hada en efas
ocallones bien me parecia rezafes el oficio divino ĩi las ocupaciones
obligatorias os diedien lugarpora lo meuos el dela virgen y flu rĩar'
```

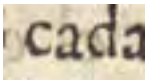
[ ]:
```
directory_path = '/content/drive/MyDrive/ocr/extracted_text_image/pdf16_1'
result16_1 = process_image_folders(directory_path, model, processor)
result16_1
```

```
['text_0.png', 'text_1.png', 'text_2-1.png', 'text_2-2.png', 'text_2-3.png',
'text_3-1.png', 'text_3-2.png', 'text_4.png', 'text_5.png', 'text_6.png',
'text_6-1.png', 'text_6-2.png', 'text_7.png', 'text_8.png', 'text_9.png',
'text_10.png', 'text_11.png', 'text_12.png', 'text_13.png', 'text_14.png',
'text_15.png', 'text_16.png', 'text_17.png', 'text_18-1.png', 'text_18-2.png',
```

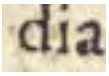
```
'text_19.png', 'text_20-1.png', 'text_20-2.png', 'text_20-3.png',
'text_20-4.png', 'text_20-5.png', 'text_21.png', 'text_22.png', 'text_23.png',
'text_24-1.png', 'text_24-2.png', 'text_25.png', 'text_26.png', 'text_27.png',
'text_28-1.png', 'text_28-2.png', 'text_29.png', 'text_30.png', 'text_31.png',
'text_32.png', 'text_33-1.png', 'text_33-2.png', 'text_34.png', 'text_35.png',
'text_38.png', 'text_39.png', 'text_40.png', 'text_41.png', 'text_43-1.png',
'text_43-2.png', 'text_43-3.png', 'text_44.png', 'text_45-1.png',
'text_45-2.png', 'text_46.png', 'text_47.png', 'text_48.png', 'text_49.png',
'text_50.png', 'text_51.png', 'text_52.png', 'text_53.png', 'text_54-1.png',
'text_54-2.png', 'text_55.png', 'text_56.png', 'text_57-1.png', 'text_57-2.png',
'text_58-1.png', 'text_58-2.png', 'text_59.png', 'text_60.png', 'text_61-1.png',
'text_61-2.png', 'text_61-3.png', 'text_62.png', 'text_63.png', 'text_64.png',
'text_68.png', 'text_69.png', 'text_70.png', 'text_71.png', 'text_72-1.png',
'text_72-2.png', 'text_73.png', 'text_74.png', 'text_75.png', 'text_76.png',
'text_77.png', 'text_78.png', 'text_79.png', 'text_80.png', 'text_81.png',
'text_82.png', 'text_83-1.png', 'text_83-2.png', 'text_84.png', 'text_85.png',
'text_87.png', 'text_88.png', 'text_89.png', 'text_90.png', 'text_99.png',
'text_100.png', 'text_101.png', 'text_102.png', 'text_103.png', 'text_104.png',
'text_105.png', 'text_106.png', 'text_107.png', 'text_108.png', 'text_109.png',
'text_110.png', 'text_111-1.png', 'text_111-2.png', 'text_112.png',
'text_113.png']
```



```
pro
```

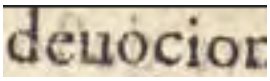


```
cada
```



```
dia
```

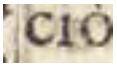
```
cuya
```



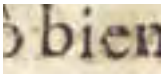
deuocior

feles

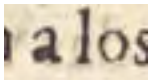
hu



cio

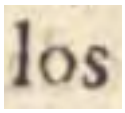


obien

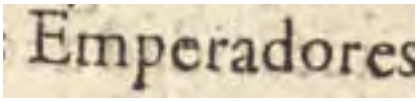


arlos

a

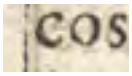


los

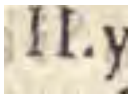


imperadores

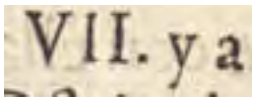
heri



icos

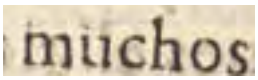


illy

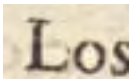


villya

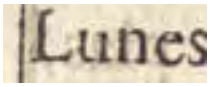
otros



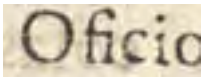
muchos



los

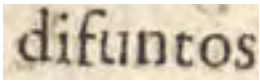


llunes

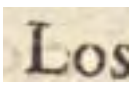


oficio

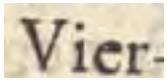
de



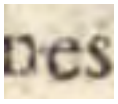
difuntos



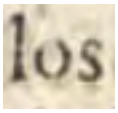
los



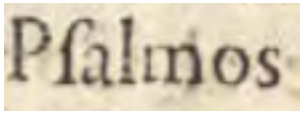
vier



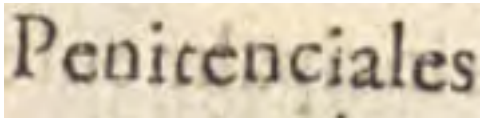
des



los



plallos

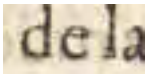


pontenciales

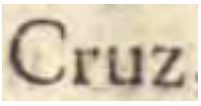
y

of

cio



dela

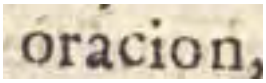


Cruz

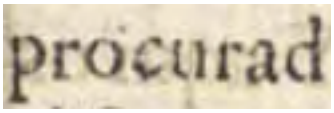
y

viralo

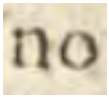
de



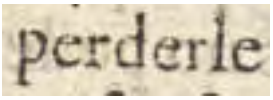
oracion



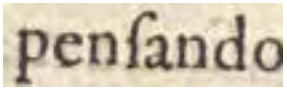
procurad



no

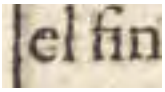


perderle

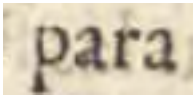


penfando

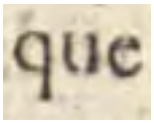
en



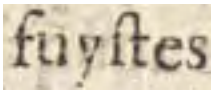
jelfin



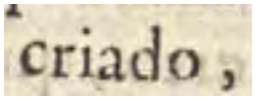
para



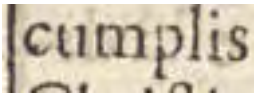
que



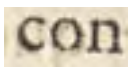
fuyfes



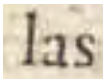
criado



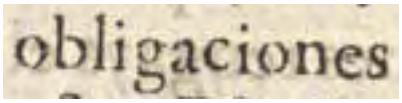
cumplis



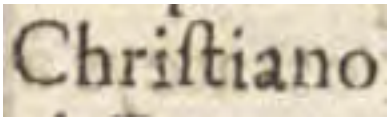
con



las

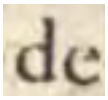


obligaciones

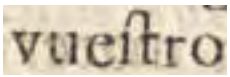


christiano

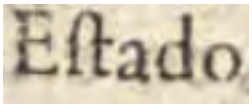
y



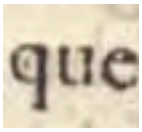
de



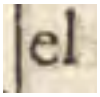
vuestro



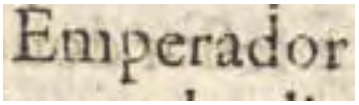
Eitado



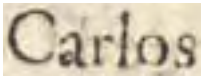
que



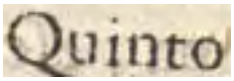
el



superador

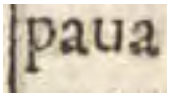


carlos

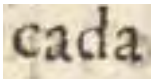


quinto

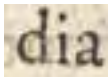
ocui



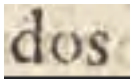
pava



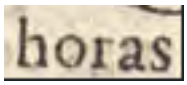
cada



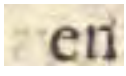
dia



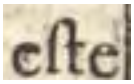
dos



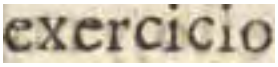
horas



ren

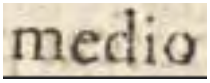


efte



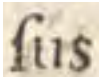
exercicio

en

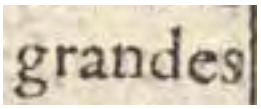


medio

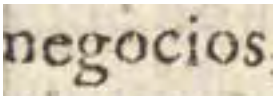
de



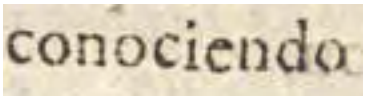
fus



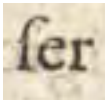
grandes



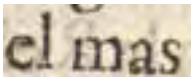
negocios



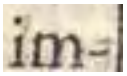
conociendo



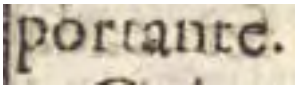
fer



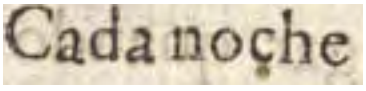
elmas



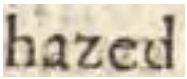
imem



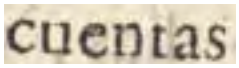
porante



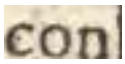
gadohecha



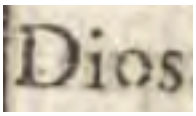
hazed



cuentas



con



dios

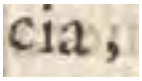
y

examen

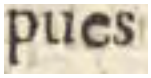
de

vuetra

concien

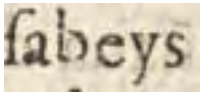


ciaji



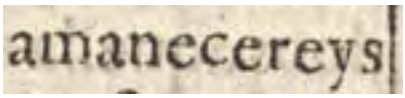
pues

no



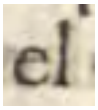
fabeys

flu

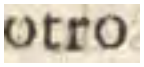


aunareceres

en



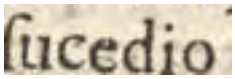
el



otro

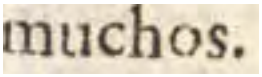
mundo

como

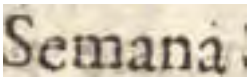


fuedio

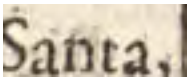
al



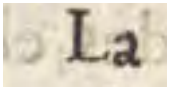
muchos



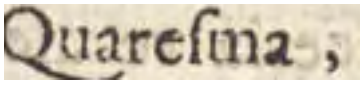
semana



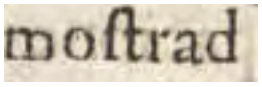
santa



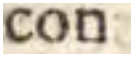
ola



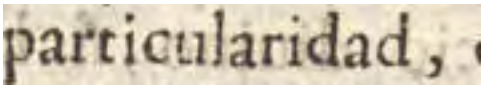
quarefmá



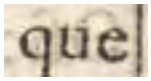
mofrad



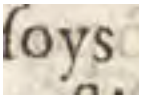
con



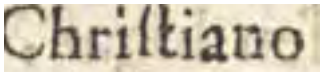
particularidad



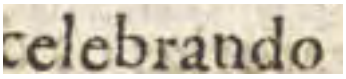
qúl



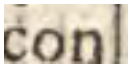
loys



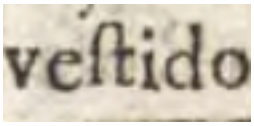
christiano



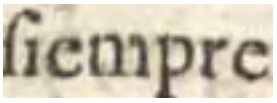
celebrando



conl

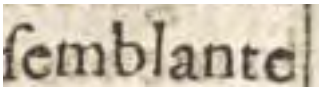


veftido



fiedpre

segro



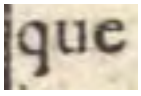
ferblante

[ ]:
```
'pro cada dia cuya deuocior feles hu cio obien arlos a los imperadores heri icos
illy villya otros muchos los llunes oficio de difuntos los vier des los plallos
pontenciales y of cio dela Cruz y viralo de oracion procurad no perderle
penfando en jelfin para que fuyfes criado cumplis con las obligaciones
christiano y de vuestro Eitado que el superador carlos quinto ocui pava cada dia
dos horas ren efte exercicio en medio de fus grandes negocios conociendo fer
elmas imem porante gadohecha hazed cuentas con dios y examen de vuetra concien
ciaji pues no fabeys flu aunareceres en el otro mundo como fuedio al muchos
semana santa ola quarefmá mofrad con particularidad qúl loys christiano
celebrando conl veftido fiedpre segro ferblante'
```

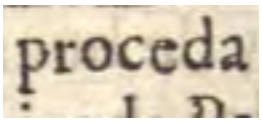
[ ]:
```
directory_path = '/content/drive/MyDrive/ocr/extracted_text_image/pdf16_2'
result16_2 = process_image_folders(directory_path, model, processor)
```

```
result16_2
```

```
['text_0.png', 'text_1.png', 'text_2.png', 'text_3.png', 'text_4.png',
'text_5-1.png', 'text_5-2.png', 'text_6.png', 'text_7.png', 'text_8-1.png',
'text_8-2.png', 'text_8-3.png', 'text_9.png', 'text_10.png', 'text_11.png',
'text_12.png', 'text_13-1.png', 'text_13-2.png', 'text_14.png', 'text_15.png',
'text_16.png', 'text_17-1.png', 'text_17-2.png', 'text_17-3.png', 'text_18.png',
'text_19-1.png', 'text_19-2.png', 'text_20.png', 'text_21-1.png',
'text_21-2.png', 'text_21-3.png', 'text_21-4.png', 'text_22.png',
'text_23-1.png', 'text_23-2.png', 'text_24.png', 'text_25.png', 'text_26-1.png',
'text_26-2.png', 'text_26-3.png', 'text_27.png', 'text_28.png', 'text_29-1.png',
'text_29-2.png', 'text_30.png', 'text_33.png', 'text_34.png', 'text_35.png',
'text_36.png', 'text_38.png', 'text_39.png', 'text_40.png', 'text_41.png',
'text_42.png', 'text_43.png', 'text_44.png', 'text_45.png', 'text_46.png',
'text_47-1.png', 'text_47-2.png', 'text_48.png', 'text_49.png', 'text_50.png',
'text_51-1.png', 'text_51-2.png', 'text_52.png', 'text_53.png', 'text_54-1.png',
'text_54-2.png', 'text_55.png', 'text_56.png', 'text_57-1.png', 'text_57-2.png',
'text_58.png', 'text_59.png', 'text_60-1.png', 'text_60-2.png', 'text_61.png',
'text_62.png', 'text_63-1.png', 'text_63-2.png', 'text_64.png', 'text_65.png',
'text_66.png', 'text_67.png', 'text_68.png', 'text_69.png', 'text_70.png',
'text_71.png', 'text_72-1.png', 'text_72-2.png', 'text_72-3.png', 'text_73.png',
'text_74.png', 'text_75.png', 'text_76.png', 'text_77.png', 'text_78.png',
'text_79.png', 'text_80.png', 'text_81.png', 'text_82.png', 'text_83.png',
'text_84.png', 'text_85.png', 'text_86.png', 'text_87.png', 'text_88.png',
'text_89.png', 'text_90.png', 'text_92.png', 'text_93.png', 'text_94.png',
'text_96.png', 'text_97.png', 'text_98.png', 'text_99.png', 'text_100.png',
'text_101-1.png', 'text_101-2.png', 'text_102.png', 'text_103.png',
'text_104.png', 'text_105-1.png', 'text_105-2.png', 'text_106.png',
'text_107.png', 'text_108.png', 'text_109.png', 'text_110-1.png',
'text_110-2.png', 'text_111.png', 'text_112-1.png', 'text_113-1.png',
'text_113-2.png', 'text_114.png', 'text_115.png', 'text_116.png',
'text_119.png', 'text_121-1.png', 'text_121-2.png', 'text_122-1.png',
'text_122-2.png', 'text_122-3.png']
```

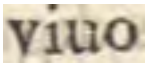


que

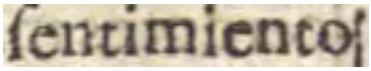


proceda

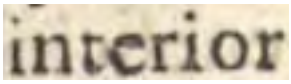
de



vuo

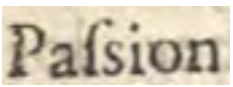


ferimientó



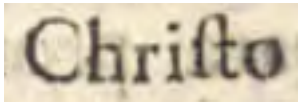
interior

la



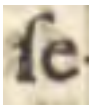
pafion

de



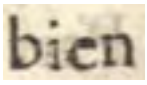
christo

y



le

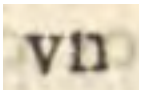
ria



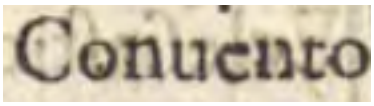
bien

retitaros

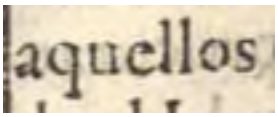
a



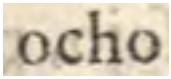
vu



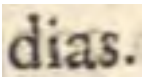
conuento



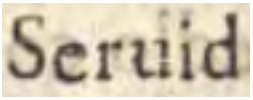
aquellos



ocino

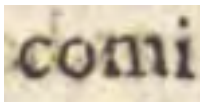


dias



serid

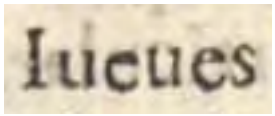
lla



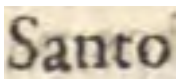
comi

do

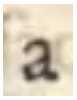
el



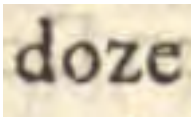
lueues



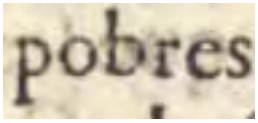
santo



a

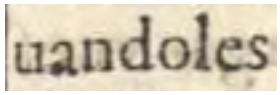


doze

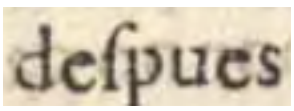


pobres

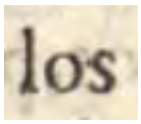
lla



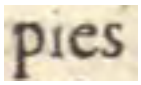
vandoles



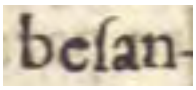
defpues



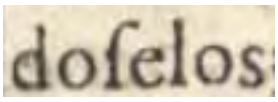
los



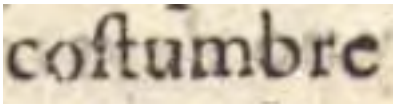
pies



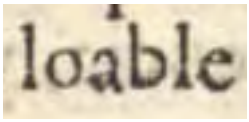
belan



dofelos

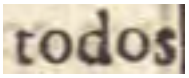


cofumbre

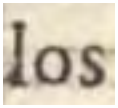


lóle

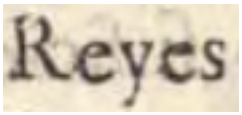
de



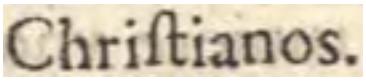
todos



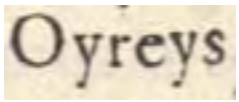
los



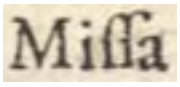
reyes



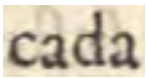
christianos



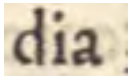
oyreys



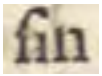
milla



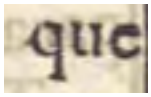
cada



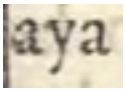
dia



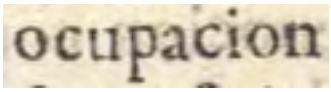
fin



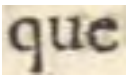
que



aya



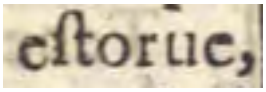
ocupacion



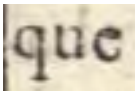
que

os

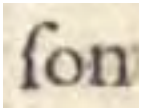
lo



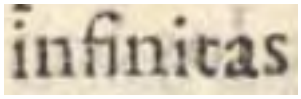
eforue



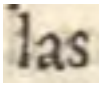
que



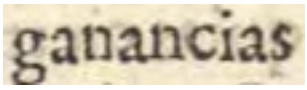
fon



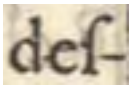
infinitas



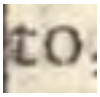
las



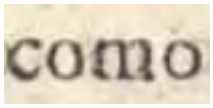
ganancias



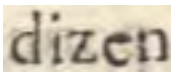
defi



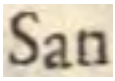
to



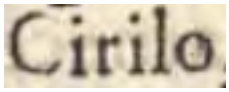
como



dizeren

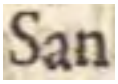


san

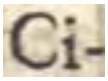


virillo

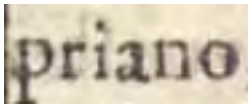
y



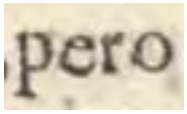
san



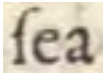
ci



priano

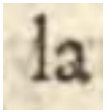


pero

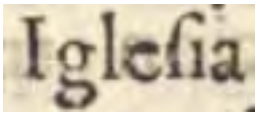


fea

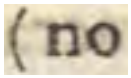
en



la

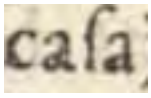


lgléa

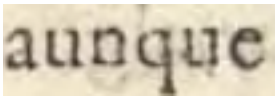


(no

en

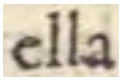


cala

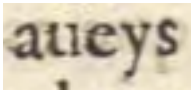


aunque

en

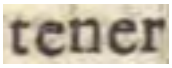


ella

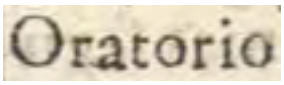


aveys

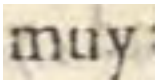
de



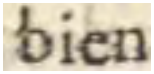
tener



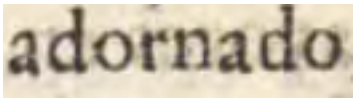
gratio



muy

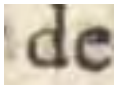


bien

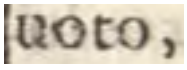


adornado

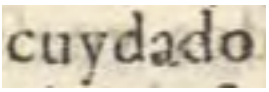
yy



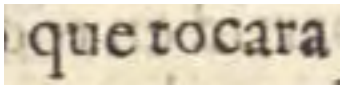
de



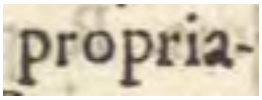
uotos



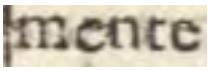
cuygado



quetocara

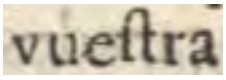


propria



mente

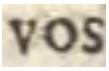
a



vuerra

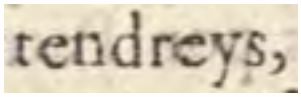
mugeres

pero



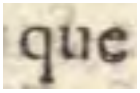
vos

ile

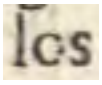


tendreys

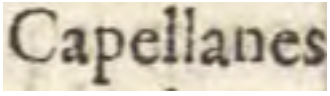
de



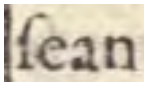
que



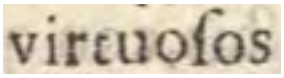
los



capellanes



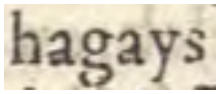
fean



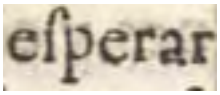
virtuofos

y

no

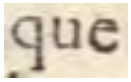


hagays



eperar

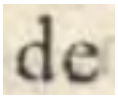
al



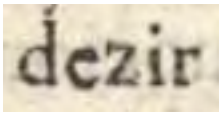
que

os

ha

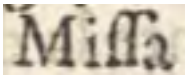


de

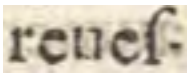


dezir

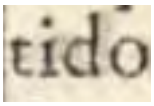
la



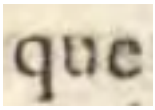
milla



reuef

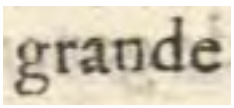


tido

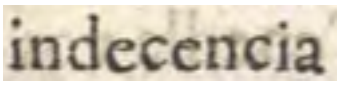


que

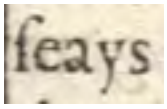
es



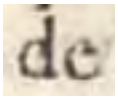
grande



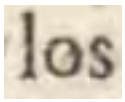
indecencia



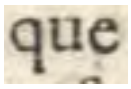
feays



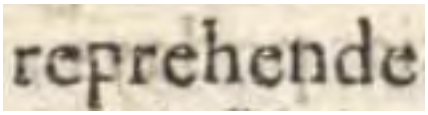
do



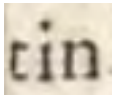
los



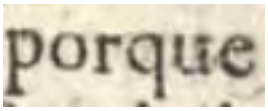
que



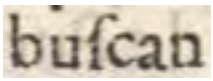
reprende



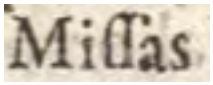
tin



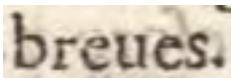
porque



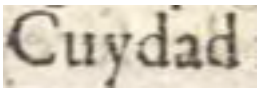
bufdan



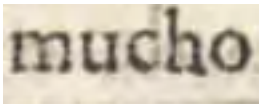
milias



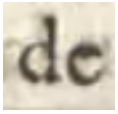
breues



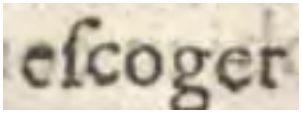
cuydad



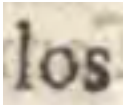
mucho



de



eleger



los

sa

[ ]: 'que proceda de vuo ferimientó interior la pafion de christo y le ria bien retitaros a vu conuento aquellos ocino dias serid lla comi do el lueues santo a doze pobres lla vandoles defpues los pies belan dofelos cofumbre lóle de todos los reyes christianos oyreys milla cada dia fin que aya ocupacion que os lo eforue que fon infinitas las ganancias defi to como dizeren san virillo y san ci priano pero fea en la lgléa (no en cala aunque en ella aveys de tener gratio muy bien adornado yy de uotos cuygado quetocara propria mente a vuerra mugeres pero vos ile tendreys de que los capellanes fean virtuofos y no hagays eperar al que os ha de dezir la milla reuef tido que es grande indecencia feays do los que reprende tin porque bufdan milias breues cuydad mucho de eleger los sa'

[ ]:
```
page_14 = result14_1 + " " + result14_2
page_14
```

[ ]: 'si por quitar un pecado morta aueys de poner vuetra vida en pe ligro arrigala que es el mejor emplo que della podeys hazer y devuéra hazienda para efte fin en redemir cautuos y facar mugeres de pecado dotandolas liberalmen le caton dixó nunca hagas el bien porque fe fepa dad pues vos hin bueno aqualquiera obra con que huyreys de ha hiporéa pero tami poco efcondays las que han defer de

buien exemplo pues es obliga cion de perfoñas tales el darlo y lo contranio tentacion en algunos nó hagays profellon de fantero pe ro íi de buen christiano no apro ueyśmas tampoco reproueys fauti dades mudofas fino efimad las cier tas y aprobadas y a effo toca elno fer mibalrero acordaos del rey s luys que no quillo ver con los ojos lo que mejor veya con la fe si por quitar un pecado morta aueys de poner vuetra vida en pe ligro arrigala que es el mejor emplo que della podeys hazer y devuéra hazienda para efte fin en redemir cautuos y facar mugeres de pecado dotandolas liberalmen le caton dixó nunca hagas el bien porque fe fepa dad pues vos hin bueno aqualquiera obra con que huyreys de ha hiporéa pero tami poco efcondays las que han defer de buien exemplo pues es obliga cion de perfoñas tales el darlo y lo contranio tentacion en algunos nó hagays profellon de fantero pe ro íi de buen christiano no apro ueyśmas tampoco reproueys fauti dades mudofas fino efimad las cier tas y aprobadas y a effo toca elno fer mibalrero acordaos del rey s luys que no quillo ver con los ojos lo que mejor veya con la fe'

[ ]:
```
page_15 = result15_1 + " " + result15_2
page_15
```

[ ]: 'imente mas ederrama por efa cau faflu fangreyque elel auerla heredado itan generolá y vivio defeminado ide ino perder ocacion en fervirle cumplir flu voluntad a vuestro confer que elege reys eipiritual docto yy hombre dé gran talento tened mucho refeto y dadle autoridad para que os diga libremente quantas verdades ave frra alma importanten en las colas tos cantes a ella obedecedle enterá mente con todo rendimiento y tal que no admitays razon para ló que os ordenerre en ellas materias por no perder el merito dela fe yy obe diencia clega que aqui la deue auer) tomad lu confejo pues eologiendó le con las partes dichas uo aura pelli gro de qúe defo metiendole en el gobierno de todo y querien do confequirl lo que pidiéri julto injulto que es propiedad dé igno santes no muy epiritales yy mas de huyr defó ganareys el dar credito oy autoridad atodas vuetras acciones eigiendole con ellas si bie lla negocacion de las colas pias toca principalmente al conferior el qual tiendo á propõo no muda reys fluo armas no poder y aqui os aduierto que aunqu a todas las reli giones ten gays el amor que efa dí cho en ẽa materia no os aleys a ninguna eleged confer doñde lo halleys mas conveniente que un fuyeto lée ha de bufcar para efõo y no toda la religion el confẽaros podria fer cada ocho dias y comulgar quado al confer parezca defeñan dosaellá mañava de qualquiera otra ocupacion y no vieys sie almo hada en efas ocallones bien me parecia rezafes el oficio divino ĩi las ocupaciones obligatorias os diedien lugarpora lo meuos el dela virgen y flu rĩar'

[ ]:
```
page_16 = result16_1 + " " + result16_2
page_16
```

[ ]: 'pro cada dia cuya deuocior feles hu cio obien arlos a los imperadores heri icos illy villya otros muchos los llunes oficio de difuntos los vier des los plallos pontenciales y of cio dela Cruz y viralo de oracion procurad no perderle

```
penfando en jelfin para que fuyfes criado cumplis con las obligaciones
christiano y de vuestro Eitado que el superador carlos quinto ocui pava cada dia
dos horas ren efte exercicio en medio de fus grandes negocios conociendo fer
elmas imem porante gadohecha hazed cuentas con dios y examen de vuetra concien
ciaji pues no fabeys flu aunareceres en el otro mundo como fuedio al muchos
semana santa ola quarefmá mofrad con particularidad qúl loys christiano
celebrando conl veftido fiedpre segro ferblante que proceda de vuo ferimientó
interior la pafion de christo y le ria bien retitaros a vu conuento aquellos
ocino dias serid lla comi do el lueues santo a doze pobres lla vandoles defpues
los pies belan dofelos cofumbre lóle de todos los reyes christianos oyreys
milla cada dia fin que aya ocupacion que os lo eforue que fon infinitas las
ganancias defi to como dizeren san virillo y san ci priano pero fea en la lgléa
(no en cala aunque en ella aveys de tener gratio muy bien adornado yy de uotos
cuygado quetocara propria mente a vuerra mugeres pero vos ile tendreys de que
los capellanes fean virtuofos y no hagays eperar al que os ha de dezir la milla
reuef tido que es grande indecencia feays do los que reprende tin porque bufdan
milias breues cuydad mucho de eleger los sa'
```

6.2 Conculusion

In this notebook I develop OCR models from scratch and fine tune the model on small datasets. To improve accuracy, I use pretrained OCR model. Some of the bad predictions on test pages are due to incorrect detection of the db50 text detection model, which struggles with capturing some words and has difficulties with spacing and punctuation. If needed, I can focus on enhancing text detection to achieve more accurate results.

I would appreciate any feedback. Thank you for your time and consideration.

###Code Reference Fine tune TrOCR on IAM Handwriting Database

https://sushantjha8.medium.com/lets-train-image-to-text-transformer-846150b632ef

https://medium.com/quantrium-tech/text-extraction-using-doctr-ocr-471e417764d5

https://keras.io/examples/vision/image_captioning/

- TrOCR paper: https://arxiv.org/abs/2109.10282
- TrOCR documentation: https://huggingface.co/transformers/master/model_doc/trocr.html