

# Report

## Image-to-Text Conversion and Text Correction System Report

### Introduction

In this report, we present a project aimed at extracting text from images using EasyOCR and performing spell checking using the PySpellChecker library. Additionally, we have developed a web application using Streamlit to provide a user-friendly interface for utilizing these functionalities.

### Project Overview

Our project encompasses the following main components:

1. Text extraction from images using EasyOCR.
2. Spell checking using PySpellChecker.
3. Development of a web application using Streamlit.

### Methodology

#### Text Extraction with EasyOCR

EasyOCR is a Python library that allows for easy and efficient text extraction from images. We utilize its capabilities to extract text from images uploaded by the user through the web application.

#### Spell Checking with PySpellChecker

PySpellChecker is a Python library that provides efficient spell checking functionality. After extracting text from images, we pass it through PySpellChecker to identify and correct any spelling errors present in the text.

#### Web Application Development with Streamlit

Streamlit is a popular framework for building web applications with Python. We leverage Streamlit to create a user-friendly interface where users can upload images, extract text, perform spell checking, and view the corrected text.

## Implementation Details

### Required Libraries:

```
#!/pip3 install torch torchvision torchaudio --index-url https://
!pip install easyocr
!pip install --user opencv-python
pip install pyspellchecker
```

### Import Libraries:

```
import easyocr
import cv2
from matplotlib import pyplot as plt
import pandas as pd
from spellchecker import SpellChecker
```

### Text Extraction

```
# Provide your image path to image_path variable
image_path= r"C:\Users\Dell\OneDrive\Pictures\Screenshots\Screenshot 2024-04-26 030414.png"

reader= easyocr.Reader(['en'],gpu=False)
result=reader.readtext(image_path)
print(result)
```

```
#result contain String and co-ordinate of the text
```

## Covert it to Data Frame to better understand :

```
#Covert the ouput into df to better understand the text
df = pd.DataFrame(result, columns= ['Cor','text','confidence'])
s = [str(x) for x in df["text"] ]
s1 = " ".join(s)
#s1 contain the Text which we extracted from the image
```

## Spell Checking

```
spell = SpellChecker()
l = ""
for word in s1.split() :
    l = l + " " + spell.correction(word)
print(corrected_text)
```

## Streamlit Web Application

```
pythonCopy code
# Code snippet for Streamlit web application
import streamlit as st

def main():
    # Title and description
    st.title("Image to Text with Text Correction")
    st.write("Upload an image and get the text with spelling correction!")
```

```

# Define directory to store uploaded images
upload_dir = "uploaded_images"
os.makedirs(upload_dir, exist_ok=True)

# File uploader
uploaded_file = st.file_uploader("Choose an image...", type=["jpg", "png"])

if uploaded_file is not None:
    # Store the uploaded file in the specified directory
    image_path = os.path.join(upload_dir, uploaded_file.name)

    with open(image_path, "wb") as f:
        f.write(uploaded_file.getvalue())

    # Display uploaded image
    image = Image.open(uploaded_file)
    st.image(image, caption="Uploaded Image", use_column_width=True)

    # Store full path of the uploaded image
    full_image_path = os.path.abspath(image_path)

# Passing full image to our easyocr module
reader = easyocr.Reader(['en'], gpu=False)
result = reader.readtext(full_image_path)

# converting our text into data frame to better understand
df = pd.DataFrame(result, columns=['Cor', 'text', 'confidence'])
s = [str(x) for x in df["text"]]
s1 = " ".join(s)
st.title("Extracted Text from the image: ")
st.code(s1)

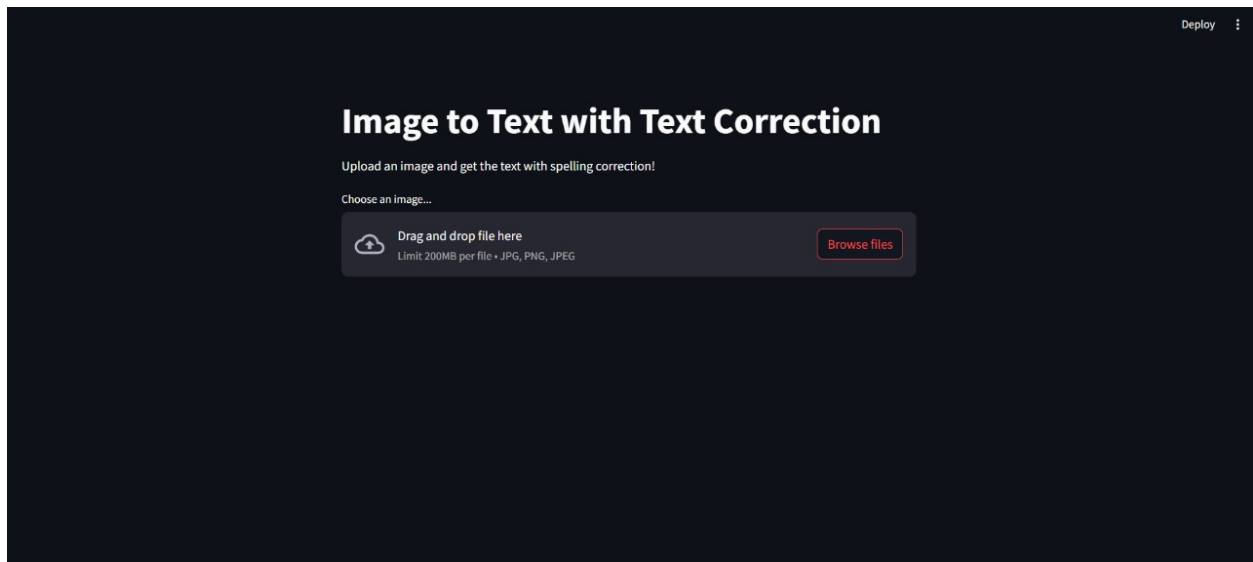
```

```
# spellijg checker
spell = SpellChecker()
l = ""
for word in s1.split():
    l = l + " " + spell.correction(word)
st.title("Spelling Corrected text after Extraction:
")
st.code(l)

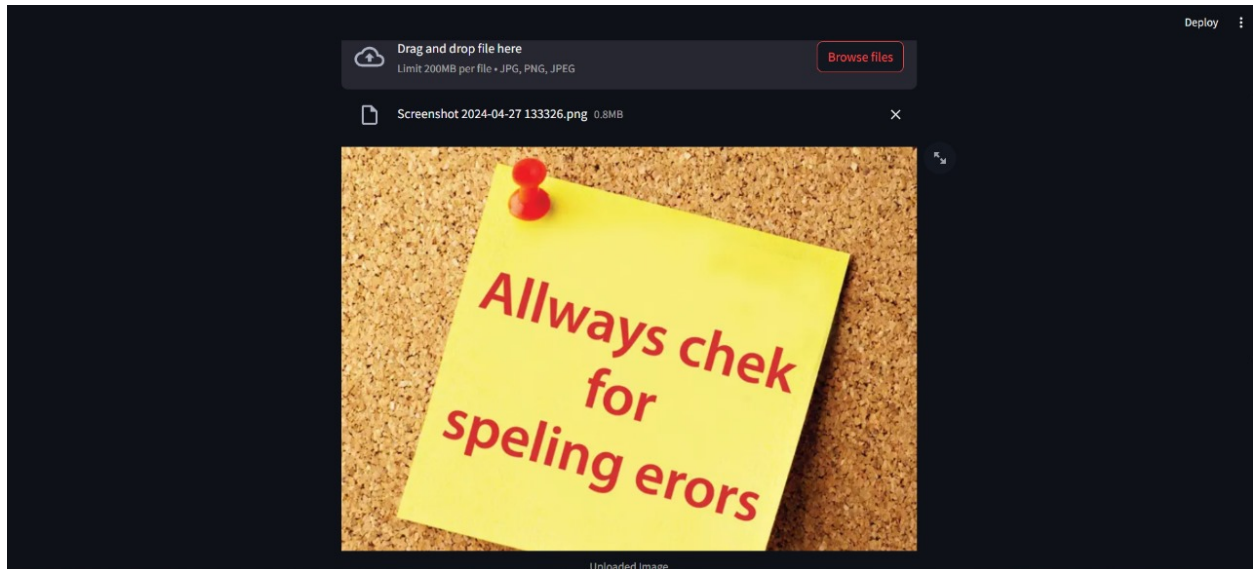
if __name__ == "__main__":
    main()
```

## Results

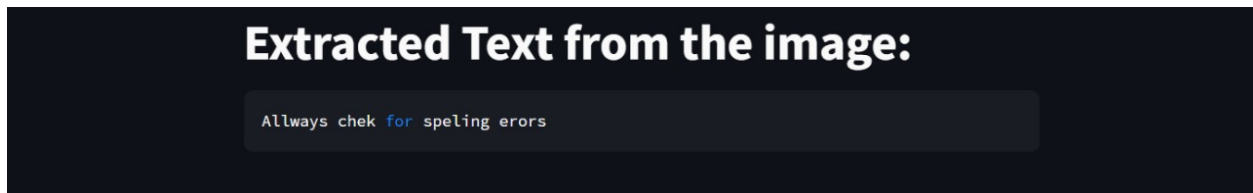
### Web Application Interface



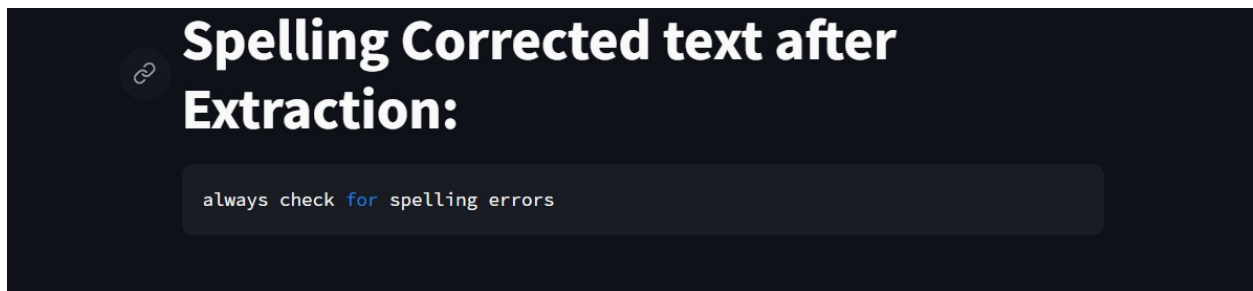
### Uploading Image



## Text Extraction Output



## Spell Checking Output



## Web Application Screenshots



Uploaded Image

### Extracted Text from the image:

Allways chek for speling errors

### Spelling Corrected text after Extraction:

always check for spelling errors

## Conclusion

Our project successfully demonstrates the integration of EasyOCR for text extraction, PySpellChecker for spell checking, and Streamlit for web application development. The combination of these technologies provides a powerful tool for extracting and correcting text from images in a user-friendly manner.