

Analysis of OCR Models: Surya vs. GCP Internship Report

Name : SONIA YADAV

Enrollment : 2022BITE030

Institute : National Institute of Technology Srinagar

Internship : IIT MADRAS

Abstract

This research demonstrates comparative analysis of Optical Character Recognition (OCR) models **Surya and Google Cloud Platform (GCP)** to assess their performance in recovering text from images. The experiment was conducted by presenting **25 images in three languages: Telugu, Odia, and Gujarati**. The two models were tested over these images to recover textual information, which was then evaluated for accuracy, readability, error types, and consistency.

The results indicated that **GCP OCR performed better than Surya**, with greater accuracy and word segmentation in all three languages. Nevertheless, GCP had some problems, including additional spaces, incorrect character separation, and missing numbers, which affected its dependability. Conversely, Surya OCR was more readable with fewer spacing errors but with more text misprediction.

Based on this comparison, GCP is suggested as the better OCR model because it supports multiple languages, is more accurate, and structures words more properly, although spacing and numerical accuracy need to be improved further. Future research involves fine-tuning both models on specialized data and applying post-processing methods for improved output.

Introduction:

Optical Character Recognition (OCR) technology is transforming the process of digitizing text from images and has become an indispensable tool for document processing, language translation, and artificial intelligence (AI)-based automation. Yet, all OCR models are not alike—particularly when used to recognize complex scripts such as Telugu, Odia, and Gujarati. This research examines a comparative study of two OCR models—Surya and Google Cloud Platform (GCP)—to identify which model is best in terms of accuracy, readability, and consistency.

To compare their performance, 25 text-containing images of the three languages were passed through both models. The obtained text was carefully examined to see if there are common mistakes such as misidentified characters, spacing that is unnecessary, omitted punctuation, and problems with segmentation. Through the determination of these, we hope to find out which model gives the most accurate and well-organized text output.

The results of this research not only point out the merits and demerits of both OCR models but also open doors for further development of multilingual OCR systems. For research in academia, business automation, or applications of AI, this study provides a useful guide to choose the optimal OCR model for Indian language scripts.

Setup and Environment:

To evaluate the performance of the **Surya OCR** model, I set up the required dependencies in **Google Colab Notebook**. The following steps were followed to install and configure the necessary packages:

1 Install Python and Torch:

To install Surya, it's essential to ensure two prerequisites are met:

1. Python version 3.9 or higher is required.

2. PyTorch must be installed on the system.

To ensure compatibility with Surya OCR, the latest version of **PyTorch** was installed along with its dependencies:



```
!python --version
!pip install -U torch
!pip3 install torch --upgrade torchvision torchaudio --index-url
https://download.pytorch.org/whl/cu118
```

2 Install Surya OCR

After setting up PyTorch, **Surya OCR** was installed using the following command:



```
!pip install surya-ocr
```

3 Execution in Google Colab

The entire analysis and text extraction process were conducted in **Google Colab**, ensuring easy access to GPU acceleration and a streamlined Python environment.

GCP OCR Setup

To utilize **Google Cloud Platform (GCP) OCR**, I followed the official documentation and setup instructions provided by Google Cloud. The step-by-step guide can be accessed here:

✂ **GCP OCR Documentation:** [Google Cloud Vision OCR Guide](#)

Link : <https://cloud.google.com/vision/docs/drag-and-drop>

This documentation provided guidance on using **GCP Vision OCR** for extracting text from images efficiently.

Methodology

The analysis was conducted by comparing extracted text from both models against the original text. The evaluation criteria included:

Accuracy: How closely the extracted text matches the original.

Error Types: Common mistakes made by each model.

Readability: The clarity and usability of extracted text.

Consistency: Frequency of errors across multiple samples.

Excel Sheet for Analysis:

The dataset and analysis used for evaluating the models can be accessed in the **Google Docs Excel sheet and Link is given below:**

<https://docs.google.com/spreadsheets/d/1EPlpC0yaOwmFxHK59AYqwxq9d-ihYgbF/edit?usp=sharing&oid=110672771518185269583&rtpof=true&sd=true>

GitHub Repository

All the work done during my **internship at AI4Bharat** is available on GitHub. This repository contains:

- 1.OCR Model Analysis (Surya vs. GCP)
- 2.Code & Implementation used for text extraction(used Google Colab).
- 3. Datasets & Results for Telugu, Odia, and Gujarati(Images of all three languages).
- 4. Excel Sheet & Report Documentation

🔗 **GitHub Repository Link:** <https://github.com/ya-sonia/Surya-and-GCP-OCR-Model-Comparison>

This repository provides a comprehensive overview of the project, including the methodologies used, extracted text analysis, and performance evaluation of both OCR models.

GCP OCR for Telugu

1. GCP adds unnecessary spaces between words (common error):

- **Example:**నీ కు కు ది రి న ప్పు డు కా దు , ఎ దు టి వా డి కి అ వ స ర మై న ప్పు డు చే స్తే దా న్ని సా యం అం టా రు . . .(Text extracted by GCP)
- **Issue:** Words are split into individual characters or contain excessive spacing.
- **Expected Text:** "నీకు కుదిరినప్పుడు కాదు, ఎదురిటివాడికి అవసరమైనప్పుడు చేస్తే దాన్ని సహాయం అంటారు."

2. Incorrect segmentation of characters:

- **Issue:** Instead of recognizing full words, GCP sometimes segments characters incorrectly, making the text hard to read.

3. Mixing similar-looking Telugu characters:

- **Example:**"వె స్త న నూ ప్ ర మో ద సం ద"
- **Issue:** Incorrectly recognizing similar-looking letters (like "న" and "త") or missing diacritics.
- **Expected Text:** "వేస్తనను ప్రమోద సంద"

4. Incorrect handling of punctuation marks:

- **Example:**"ఇదే మనకు తేలింది , మనం దీని మీద బాగా ఆలోచించాలి . . ."
- **Issue:**
 - Unnecessary spaces before punctuation.
 - Ellipsis ("...") not properly recognized.
- **Expected Text:** "ఇదే మనకు తేలింది, మనం దీని మీద బాగా ఆలోచించాలి..."

5. Incorrect Letter Prediction

- Some Telugu letters are misrecognized or predicted incorrectly.
- **Example: Incorrect:** "స ఇది తప్పు పాఠం" (with extra letters)
- **Correct:** "సరైన వాక్యం"

6.Extra/ Missing Characters

- Sometimes, extra characters are added, or some characters are omitted.
- **Example: Incorrect:** "ర ఇది అదనపు వర్ణం" (Added 'ర' wrongly)
- **Correct:** "ఇది సరైన వర్ణం"

7.Loss of Meaning Due to Jumbled Words

- GCP sometimes rearranges words, leading to loss of meaning.

- **Example: Incorrect:** "ఆ నం దం గా వుం టా డు ఓ డి న వా డు"
- **Correct:** "ఆనందంగా ఉంటాడు, ఓడినవాడు"

8.Missing Digits:

Some numbers are **completely missing**, affecting the accuracy of numerical data in documents.

- Missing digits: **1, 4, 5, 6** were not extracted. **(Image Name: 19_tel.png)**
- Incorrect date extraction:

eg: "2 3 - 0 4 - 2 0 2 2" was extracted,

It should be "23-04-2022" (extra spaces between digits).

Solutions :

1. **Post-Processing:** Use regex and NLP-based text normalization to remove unnecessary spaces and correct errors.
2. **Fine-Tuning OCR Models:** Train custom models using Tesseract, Google AutoML Vision, or Transformer-based OCR (e.g., Donut, TrOCR).
3. **Better Datasets:** Improve training data with diverse Telugu fonts, synthetic text generation, and real-world document samples.

Future Work:

- Integrate **NLP models** (like BERT for Telugu) for spell correction.
- Use **hybrid OCR** (GCP + Tesseract + AI-based OCR) for higher accuracy.

Surya OCR for Telugu

1.Extra Text Included

- **Example: Extracted Text:** "నీకు కుదిరినప్పుడు కాదు మరియు ఇది ప్రత్యేకమైన విషయం."
- **Correct Text:** "నీకు కుదిరినప్పుడు కాదు."
- **Issue:** Surya sometimes **adds extra words or phrases** that were not present in the original text.

2. Predicting Wrong Text

- **Example: Extracted Text:** "ప్రమో త సంద"
- **Correct Text:** "ప్రమోద సంద"
- **Issue:** Some Telugu letters are **incorrectly predicted**, changing the meaning of the word.

3. Repeated Words

- **Example: Extracted Text:** "ఇది ఇది సరైన మార్గం."
- **Correct Text:** "ఇది సరైన మార్గం."
- **Issue:** Certain words are **repeated unnecessarily**, which affects readability.

4. Missing Punctuation

- **Example: Extracted Text:** "ఇది సరైనది మనం దీన్ని పాటించాలి"
- **Correct Text:** "ఇది సరైనది. మనం దీన్ని పాటించాలి."
- **Issue: Missing full stops (.), commas (,), and other punctuation marks**, making sentences harder to read.

5. Some Text is Missing

- **Example: Extracted Text:** "ఆనందంగా ఉంటాడు"
- **Correct Text:** "గెలిచినవాడు ఆనందంగా ఉంటాడు, ఓడినవాడు లోచిస్తూ ఉంటాడు."
- **Issue:** Important parts of the text are **completely missing**, leading to loss of meaning

Solutions:

- Train Surya OCR with **more diverse Telugu datasets** to reduce incorrect predictions.
- Use **language models** for post-processing to fix extra/missing text and punctuation errors.
- Implement **image enhancement techniques** to improve character recognition.
- Apply **rule-based methods** to detect and correct repeated words and missing punctuation.

Future Work:

- Fine-tune OCR for **better handling of Telugu conjunct letters & diacritics**.
- Improve **Telugu numeral & date recognition** to avoid missing digits.
- Develop a **hybrid approach (ML + rule-based correction)** for better accuracy.

GCP OCR for Gujarati

1. Unnecessary Spaces Between Words (Common Error)

- **Example (Extracted Text by GCP):** " ગુજરાત બજેટનીવાંચોમહત્વનીજાહેરાત"
- **Issue:** Extra spaces between words.

2. Incorrect Character Segmentation

- **Issue:** Characters are wrongly segmented instead of forming proper words.

This is due to→ Some scripts have intricate letter connections , poor image quality

3. Mixing Similar-looking Gujarati Letters

- **Example (Extracted Text by GCP):** "મહેતવિશાલસિંહ"
- **Issue:** Confusing letters like "ત" and "ટ", "શ" and "ષ", etc.
- **Expected Output:** "મહેતાવિશાલસિંહ"

4. Missing Punctuation Marks

- **Example (Extracted Text by GCP):** "તુંસારાકામોકરલોકોતનેયાદકરશે"
- **Issue:** Missing commas or full stops.
- **Expected Output:** "તુંસારાકામોકર, લોકોતનેયાદકરશે."

5. Incorrect Word Prediction

- **Example (Extracted Text by GCP):** "મારાઘવલમકાનછે"
- **Issue:** "ઘવલ" is a misrecognition of "નવલ".
- **Expected Output:** "મારાનવલમકાનછે."

6. Extra or Missing Characters

- **Example (Extracted Text by GCP):** "મારગમતોખાધ"
- **Issue:** The correct phrase should be "મારુંગમતુંખાધું". Missing "ું".
- **Expected Output:** "મારુંગમતુંખાધું."

7. Repeated Words

- **Example (Extracted Text by GCP):** "અમેઅમેભેગામળ્યા"
- **Issue:** The word "અમે" is repeated unnecessarily.
- **Expected Output:** "અમેભેગામળ્યા."

8. Missing Digits or Incorrect Number Extraction

- **Example (Extracted Text by GCP):** "મારોફોનનંબર: ૯૮૭૬૫૪"

- **Issue:** Missing digits at the end.
- **Expected Output:** "મારો ફોન નંબર: ૯૮૭૬૫૪૩૨૧૦."

Solutions:

Improve training datasets with high-quality Gujarati text, fine-tune OCR models for better segmentation, and use NLP-based post-processing for error correction. Implement dictionary-based corrections and confidence scoring for better accuracy.

Future Work:

Develop AI-powered auto-correction tools, train Transformer-based OCR models for Gujarati, and enhance digit recognition with specialized number datasets. Rule-based filtering and context-aware algorithms can further refine extracted text.

Surya OCR for Gujarati

Extra Text Inclusion:

- Example: Some predictions contain **extra words or phrases** that are not in the original text.

Wrong Text Prediction:

- Example: "હુકમાને" is incorrect; the correct word should be "હુકમાન".

Repeated Words:

- Example: "અનુસાર કામગીરી" appears with extra spacing or duplication.

Punctuation Errors:

- Some Gujarati punctuation marks like **full stops, commas, or dandas (|)** are missing or incorrectly placed.

Missing Text:

- Some sentences or specific words are **completely missing**, affecting the meaning.

Solutions & Future Work:

- Implement **language-specific models** with better handling of Gujarati script nuances.
- Improve **segmentation and spacing corrections** to reduce repeated or missing words.
- Enhance **punctuation recognition** to correctly extract **Gujarati punctuation marks**.
- Use **post-processing techniques** like dictionary-based correction for **common OCR errors**.
- Train the model on **more Gujarati datasets** to improve accuracy.

GCP OCR for Odia

Unnecessary Spaces in Words:

- Example: "ଭୁମର" → Extracted as "ଭୁ ମ ର" (wrong spacing).
- Expected: "ଭୁମର"

Incorrect Character Segmentation:

- Example: "ଘାରିଘ" → Extracted as "ଘା ରି ଘ"
- Expected: "ଘାରିଘ"

Misrecognition of Characters:

- Example: "ଅମ୍ବୁଗ୍ଘ" → Extracted as "ଅ ଯୁ ଗ୍ଘ"
- Expected: "ଅମ୍ବୁଗ୍ଘ"

Missing or Extra Letters:

- Example: "ରାଜଗୋପାଳଙ୍କ" → Extracted as "ରା ଜ ଗୋ ପା ଲ ଣ୍ କ"
- Expected: "ରାଜଗୋପାଳଙ୍କ"

Punctuation and Formatting Issues:

- Incorrect spacing before/after punctuation marks.
- Some commas and full stops missing.

Solutions and Future Work:

- **Improve Preprocessing:** Use better segmentation techniques to avoid unnecessary spacing.
- **Fine-tune OCR Model:** Train with more Odia-specific datasets for better character recognition.
- **Post-Processing Corrections:** Implement a spell-check or language model to auto-correct misrecognized words.
- **Enhance Punctuation Handling:** Train OCR to better recognize Odia punctuation rules.

Surya OCR for Odia

Incorrect Word Segmentation

- Some words are split incorrectly, making the text harder to read.
- **Example:**
 - **Extracted:** "ଯୁଗ୍ମ ବୟସ ରେ ବିବାହ"
 - **Expected:** "ଯୁଗ୍ମ ବୟସରେ ବିବାହ"

Extra Spaces Between Words

- Additional spaces appear between words or within numbers.
- **Example:**
 - **Extracted:** "ଶ୍ରୀମତୀ ପ୍ୟାରିସ ଅଲିମ୍ପିକ୍ସ ୨ ୦ ୨ ୪"
 - **Expected:** "ଶ୍ରୀମତୀ ପ୍ୟାରିସ ଅଲିମ୍ପିକ୍ସ ୨୦୨୪"

Missing Text or Characters

- Some words or letters are completely missing from the extracted text.
- **Example:**
 - **Extracted:** "ବର ର ଅଯୁଗ୍ମ ବୟସ"
 - **Expected:** "ବର ଏବଂ ଜନ୍ମାର ଅଯୁଗ୍ମ ବୟସ"

Repeated Words

- Some words appear multiple times unnecessarily.
- **Example:**
 - **Extracted:** "ଭଲ ଭଲ ମନେ ହେଉଛି"
 - **Expected:** "ଭଲ ମନେ ହେଉଛି"

Punctuation Errors

- Some punctuation marks (commas, full stops) are missing.
- **Example:**
 - **Extracted:** "ତୁମେ କାହିଁକି ଅପେକ୍ଷା କର"
 - **Expected:** "ତୁମେ କାହିଁକି ଅପେକ୍ଷା କର?"

Solutions & Future Work:

- **Improve OCR Model Training:** Train the model on a larger dataset specific to Odia script.
- **Post-Processing:** Implement language rules to auto-correct errors (e.g., fixing extra spaces, missing punctuation).
- **Dictionary Matching:** Use an Odia dictionary to validate words and correct wrong predictions.
- **Better Segmentation Algorithms:** Improve text segmentation to reduce errors in word splits.

GCP Vs Surya

Best Overall Model: Google Cloud Vision (GCP) OCR

Based on the analysis of OCR errors in Telugu, Gujarati, and Odia, **Google Cloud Vision (GCP) OCR** is the best overall model due to its **higher accuracy, better word segmentation, and multi-language support** compared to Surya. However, GCP still has **issues with missing digits, incorrect spacing, and punctuation handling**, which can impact its reliability for certain applications.

Why GCP is the Best?

- **Higher Accuracy** – Fewer missing words/digits compared to Surya
- **Better Word Segmentation** – Preserves word structure better in Telugu, Gujarati, and Odia
- **Fewer Repeated Words** – Less duplication of words/characters than Surya
- **Multi-Language Support** – Consistent performance across all languages

However, GCP **still makes errors** (e.g., extra spaces in Telugu, missing digits in Odia, incorrect character recognition in Gujarati).

For **specific languages**, alternative models perform better:

- **Telugu:** Fine-tuned **TrOCR** is recommended for handling complex script spacing.
- **Gujarati:** **Microsoft Azure OCR** is better at recognizing conjunct characters and punctuation.
- **Odia:** **Tesseract (fine-tuned)** can improve accuracy by leveraging a domain-specific dataset.