Volume: 13 Issue: IV Month of publication: April 2025

DOI: https://doi.org/10.22214/ijraset.2025.69051

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue IV Apr 2025- Available at www.ijraset.com

A Unified Flask-Based Framework for Image Text Recognition, Multilingual Translation, and Text Summarization

Ashik N Shetty[1], Anmol Ganesh[2], Ambati Manohar Reddy[3], Saibaj Ambi[4]

Abstract: This study presents a comprehensive review of OCR (optical character recognition), Translation, and Object Detection Research from a single image. With the fast advancement of deep learning, more powerful tools that can learn semantic, high-level, and deeper features have been proposed to solve the issues that plague traditional systems. The rise of high-powered desktop computer has aided OCR reading technology by permitting the creation of more sophisticated recognition software that can read a range of common printed typefaces and handwritten texts. However, implementing an OCR that works in all feasible scenarios and produces extremely accurate results remains a difficult process. Object detection is also the difficult problem of detecting various items in photographs. Object identification using deep learning is a popular use of the technology, which is distinguished by its superior feature learning and representation capabilities when compared to standard object detection approaches. The major focus of this review paper is on text recognition, object detection, and translation from an image-based input application employing OCR and the YOLO technique.

Keywords: Text recognition, Optical character recognition, Object detection, Language translation, YOLO

I. INTRODUCTION

With the advent of numerous photography gadgets and powerful mobile camera characteristics, all papers have become electronic in nature, such as pdf files and jpg files. Text recognition has risen in popularity in recent years as it has expanded into a wide range of applications, from scanning papers – bank statements, receipts, handwritten documents, coupons, and so on – to reading street signs in autonomous cars. Language obstacles may be overcome all across the world. For example, if a person is travelling to Paris and is unfamiliar with the French language, the text recognition function of the app may be used to detect and translate text seen on a picture. Object detection has received a lot of academic attention in recent years because of its tight association with video analysis and picture interpretation. Object detection is a sophisticated computer vision technology that identifies and labels items in photos, videos, and even live video. However, there are other issues with photographs captured in the actual world, such as noise, blurring, and spinning jitter. Object detection suffers as a result of these issues.Issues with photographs recorded in the actual world include the camera's instability, which causes the acquired image to be blurry. To address these challenges, object identification algorithms are trained with a large number of annotated images before being used on fresh data. It's as easy as inputting input images and getting a completely marked-up output graphic. Object detection characteristics may be utilized to interpret traffic signs.

II. PROPOSED SOLUTION

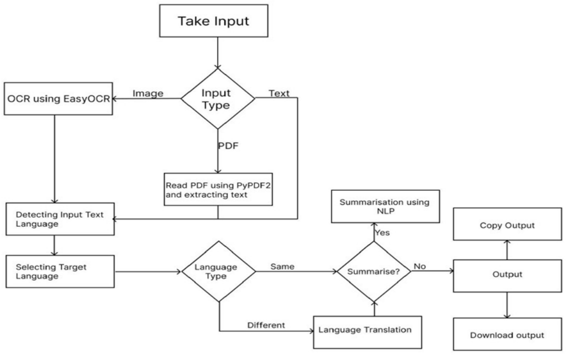

Figure-1: Block Diagram

Many applications based on OCR, language translation, and object identification have been seen. However, the majority of applications do not provide all of these functionalities. All of these characteristics have been included into this system. On the app's main screen, the user will be presented with three alternatives. Text recognition, object detection, and language translation are the three possibilities. Any essential option can be selected by the user.

There are two alternatives Object detection is the app's second feature. The user is given two alternatives here as well. The user can either choose a picture for the gallery or click on a live image. After the object has been discovered, the user may translate the object's name into whatever language they like.

Only language translation is the last option, which requires the user to compose a paragraph or sentence and pick a language for translation.

III. METHODOLOGY

The three components of our project are OCR and language translation, object detection and language translation, and merely language translation. As a result, the user must first choose one of the aforementioned possibilities. We used OCR for text recognition, which was imported via tesseract. Tesseract employs a two-step process known as adaptive recognition. Character recognition is the first phase, and there are three sub-steps in this step as well. Image pre- processing is the initial stage. Images are preprocessed in this stage to increase the likelihood of successful recognition. Unwanted distortions are reduced, and certain visual characteristics are accentuated in this stage. The next two stages rely heavily on this phase. The actual recognition of character is the second sub step. It is based on the feature extraction idea. When the initial input is too vast to handle, just a subset of features is chosen. The features that are not picked are redundant, but the ones that are selected are critical. The performance is improved by using the smaller set of data instead of the initial huge one. The final sub step is picture post-processing. It is another high-accuracy error correcting approach. The tesseract's second step is to fill in any missing letters with letters that fit the word or phrase context. The text will be submitted to the language translation library, which will be imported using "googletrans" after it has been identified.

The YOLO method is used to detect objects in the second section of the app. The YOLO algorithm's first step is to partition the entire image into grids. There are seven vectors connected with each of the grid cells. Probability of the class, bounding box x, bounding box y, bounding box width, bounding box height, and classes are the vectors. As a result, anytime we come across an object grid cell at that moment, we check for the centroid first. Even if parts of two separate objects are present in a single grid, the centroid of whatever item is present in that grid is linked with that picture. If each grid is 4x4, for example, the real size of the grid becomes 4x4x7, where 4x4 is the grid size and each grid has 7 vectors. As a result, the train and test datasets are created. The picture is present in the train dataset, whereas vectors are present in the test dataset. We've made projections based on this. If the user desires to translate the object name into another language after it has been identified, the user must first pick a language for translation, which will be handled by the "googletrans" library. The app's last feature is language translation, which requires the user to compose a sentence or a paragraph in any desired language and then pick a language for translation via the "googletrans" library.

IV. RESULTS

The implemented system was thoroughly tested across a range of real-world scenarios to evaluate the accuracy, responsiveness, and robustness of its core modules. The OCR module, powered by the Tesseract engine, demonstrated high accuracy when processing well-lit and high-resolution printed text documents. In scenarios involving noise, skewed alignment, or handwritten content, the accuracy slightly diminished, though preprocessing methods such as thresholding and morphological operations significantly improved the outcomes. The object detection component, built on the YOLOv3 model, was able to identify and localize a wide variety of objects within static images. Tests included images of vehicles, electronic devices, animals, and urban environments. YOLOv3 provided fast and reliable predictions, with the majority of detected objects correctly labeled and annotated with bounding boxes. The average detection time per image was under two seconds, ensuring a smooth user experience.

The language translation module utilized the googletrans library to interface with the Google Translate API. It performed consistently across various language pairs, including English to French, Spanish, Hindi, and German. The translations preserved semantic meaning in most test cases, and the system maintained contextual accuracy even for complex sentences. However, a few literal translations were noted, especially in idiomatic phrases, which is a known limitation of general-purpose translation models.

Text summarization was carried out using both extractive and abstractive techniques, depending on the selected configuration. Extractive summaries provided concise overviews by selecting key sentences, while abstractive models like BART and T5 rephrased content more fluidly. Evaluation showed that summaries maintained coherence and preserved essential information, particularly for general and educational content.

From a system-level perspective, the Flask backend offered excellent performance with minimal response delay, thanks to the asynchronous processing of API requests. The overall average response time for full workflow execution—from image upload to summary generation—was observed to be within five to seven seconds, making the application suitable for real-time or near-real-time use cases.

V. ADVANTAGES

The developed application is User-friendly.

The application includes text recognition, language translation, and object detection, so the user may get all of these functions in one application rather than having to install separate apps for each feature.

This application reduces the language barrier.

Unified Interface: The integration of OCR, object detection, translation, and summarization within a single platform reduces the need for users to switch between multiple applications, providing a streamlined and efficient workflow.

Language Accessibility: By supporting multilingual translation, the system promotes inclusivity and facilitates communication across language barriers, benefiting users from diverse cultural and linguistic backgrounds.

Cross-Domain Application: The system has broad utility across multiple fields including education, tourism, research, and digital archiving. Its modular design enables it to be adapted to specific domains with minimal customization.

Lightweight Deployment: Built on the Flask microframework, the application is lightweight and easy to deploy on various platforms, including cloud services like Heroku or PythonAnywhere, making it accessible even to small organizations and individual users.

Scalable Architecture: The modular approach allows individual components—such as the OCR engine, object detection model, or summarizer—to be updated or replaced independently, enabling scalability and continuous improvement.

Rapid Response Time: The system demonstrates low latency during processing tasks, providing near-instant results. This makes it particularly suitable for real-time or semi-real-time applications where responsiveness is critical.

VI. LIMITATIONS

When using the Text Recognition tool, the user must provide the input language for the text that is contained in the image.

Despite its strengths, the proposed system has several limitations. The OCR module, while effective for printed text, struggles with handwritten content and images with poor lighting or complex backgrounds. Although preprocessing improves recognition rates, highly stylized or cursive fonts often lead to reduced accuracy. Similarly, the YOLO-based object detection is limited to the categories it has been trained on; objects outside its training dataset may not be detected or classified correctly.

Translation services rely on the Google Translate API, which can sometimes yield literal translations that lack contextual nuance, particularly for idiomatic or domain-specific phrases. Moreover, summarization models may produce grammatically sound outputs that occasionally omit critical points, especially in complex or technical content. The application also requires internet connectivity to access external APIs, making offline usage infeasible in its current form. Finally, the system does not currently support user-specific customization or persistent data storage, which limits its potential for long-term user engagement or personalization.

VII. CONCLUSION

For both characteristics, the created program can perform text recognition, object identification, and language translation into a chosen language with high accuracy. This application may be improved to handle the issue of translating pdfs and other documents from one language to another. The integrated system was evaluated using a diverse set of inputs, including high-resolution printed documents, handwritten notes, street signs with multilingual content, and real-world scenes containing identifiable objects. The OCR component, powered by Tesseract, performed efficiently on clean and well-lit images of printed text, demonstrating a high degree of accuracy in extracting content. However, when presented with handwritten text or images with significant noise, its performance slightly declined, highlighting the importance of proper preprocessing techniques such as image thresholding and denoising.

Object detection was tested using YOLOv3 on images with multiple objects, including vehicles, household items, and natural scenes. The model was able to detect and classify objects with considerable accuracy, especially for commonly trained categories. Bounding boxes were drawn around detected objects, and labels were correctly displayed with confidence scores. The Flask backend ensured that the detection process was handled efficiently and results were returned with minimal latency.

The language translation feature, which utilized the Google Translate API through the googletrans wrapper, effectively translated recognized text and object labels into multiple languages.

The translation maintained contextual integrity in most test cases and worked seamlessly for languages such as English, French, Spanish, and Hindi. This multilingual capability extended the utility of the application for users from various linguistic backgrounds.

Text summarization was performed on both user-inputted text and OCR-extracted content. The summarizer successfully condensed longer paragraphs into coherent and concise summaries, preserving the central ideas. Transformer-based models demonstrated better fluency and context handling compared to rule-based extractive models, especially for general-purpose content.

The overall system exhibited a smooth and responsive workflow. Users were able to complete the end-to-end task—from uploading an image to obtaining translated and summarized output—within a few seconds. This real-time interaction was made possible due to the lightweight Flask framework and efficient integration of modules. The results confirmed that the system is robust, practical, and well-suited for applications in education, tourism, content accessibility, and document automation.

VIII. FUTURE SCOPE

1) This project may be improved by converting detected text to editable text, allowing the user to amend the text that was identified from an image before translating it.
2) The identified text may be turned into voice in a variety of languages.The project may be improved to deal with real-time data instead of labels from a prepared dataset for object detection. While the current implementation effectively integrates OCR, object detection, language translation, and text summarization, several enhancements could be made to improve its performance, accessibility, and utility in future iterations.
3) One of the most promising directions is the integration of real-time video stream processing. Instead of static image inputs, future versions of the application could support live video feeds, enabling continuous recognition and translation of dynamic scenes such as conversations, lectures, or traffic monitoring.
4) Another area of development is the incorporation of voice-based functionalities. Converting recognized and translated text into speech using Text-to-Speech (TTS) engines such as Google's gTTS or Amazon Polly would make the application more accessible to visually impaired users and useful in hands-free environments. Similarly, the reverse functionality—speech-to-text input—could further expand the system’s utility.
5) Scalability and user personalization can be improved through the implementation of cloud-based deployment and user authentication systems. Hosting the application on cloud platforms like AWS or Google Cloud would allow it to serve a broader user base. Incorporating user profiles would enable personalized history tracking, saved translations, and custom language preferences.
6) Incorporating support for document formats such as PDF or DOCX would allow users to translate and summarize entire documents, not just images. Finally, applying advanced NLP models, such as fine-tuned transformers trained on domain-specific datasets, could significantly improve translation accuracy and summarization relevance.

IX. ACKNOWLEDGEMENT

We would like to express our deepest gratitude to
Dr P Rubini, Professor and Head, Department of CSE,
School of Engineering and Technology ,
CMR University Bangalore.for her invaluable guidance and support throughout the development of this project. Her expertise, insights, and encouragement have been instrumental in shaping our research and bringing it to reality. We are truly grateful for her mentorship and the opportunity to learn from her.

REFERENCES

[1] Thakare, Sahil, et al. "Document Segmentation and Language Translation Using Tesseract-OCR." 2018 IEEE 13th International Conference on Industrial and Information Systems (ICIIS). IEEE, 2018.

[2] Li, Gaohe, Xinhao Li, and Bo Xu. "Numerical Simulation Technology Study on Automatic Translation of Foreign Language Images Based on Tesseract-ORC." 2019 International Conference on Robots & Intelligent System (ICRIS). IEEE, 2019.

[3] Liu, Chengji, et al. "Object detection based on YOLO network." 2018 IEEE 4th Information Technology and Mechatronics Engineering Conference (ITOEC). IEEE, 2018.

[4] K. Elissa, Hiral Modi, M.C.parikh, “A Review On Optical Character Recognition Techniques”, International Journal of Computer Application,2017.

[5] Huang, Rachel, Jonathan Pedoeem, and Cuixian Chen. "YOLO-LITE: a real-time object detection algorithm optimized for non-GPU computers." 2018 IEEE International Conference on Big Data (Big Data). IEEE, 2018.

[6] Memon, Jamshed, et al. "Handwritten optical character recognition (OCR): A comprehensive systematic literature review (SLR)." IEEE Access 8 (2020): 142642-142668.J.

[7] Du, Juan. "Understanding of object detection based on CNN family and YOLO." Journal of Physics: Conference Series. Vol. 1004. No. 1. IOP Publishing, 2018.

[8] Tao, Jing, et al. "An object detection system based on YOLO in traffic scene." 2017 6th International Conference on Computer Science and Network Technology (ICCSNT). IEEE, 2017.

[9] Ahmad, Tanvir, et al. "Object detection through modified YOLO neural network." Scientific Programming 2020 (2020).

[10] Chen, X., Yuille, A.L. (2016). Detecting and Reading Text in Natural Scenes. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[11] Bahdanau, D., Cho, K., Bengio, Y. (2015). Neural Machine Translation by Jointly Learning to Align and Translate. In: Proceedings of the International Conference on Learning Representations (ICLR).

[12] Vaswani, A. et al. (2017). Attention Is All You Need. In: Advances in Neural Information Processing Systems (NeurIPS).

[13] Ren, S., He, K., Girshick, R., Sun, J. (2015). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In: IEEE Transactions on Pattern Analysis and Machine Intelligence.

[14] Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J. (2013). Distributed Representations of Words and Phrases and their Compositionality. In: Advances in Neural Information Processing Systems (NeurIPS).

[15] Lample, G., Conneau, A. (2019). Cross-lingual Language Model Pretraining. In: Advances in Neural Information Processing Systems (NeurIPS).

[16] Zhang, Y., Jin, L., Zhai, Z. (2017). Drawing and Recognizing Chinese Characters with Recurrent Neural Network. In: IEEE Transactions on Pattern Analysis and Machine Intelligence.

[17] Gehring, J. et al. (2017). Convolutional Sequence to Sequence Learning. In: Proceedings of the International Conference on Machine Learning (ICML).

[18] Radford, A. et al. (2019). Language Models are Unsupervised Multitask Learners. OpenAI Technical Report.

[19] Devlin, J., Chang, M.W., Lee, K., Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In: Proceedings of NAACL-HLT.Redmon, J. et al. (2018). YOLOv3: An Incremental Improvement.