INFORMATION EXTRACTION FROM PRODUCT LABELS: A MACHINE VISION APPROACH

Hansi Seitaj and Vinayak Elangovan

Computer Science program, Penn State Abington, Abington, PA, USA

ABSTRACT

This research tackles the challenge of manual data extraction from product labels by employing a blend of computer vision and Natural Language Processing (NLP). We introduce an enhanced model that combines Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) in a Convolutional Recurrent Neural Network (CRNN) for reliable text recognition. Our model is further refined by incorporating the Tesseract OCR engine, enhancing its applicability in Optical Character Recognition (OCR) tasks. The methodology is augmented by NLP techniques and extended through the Open Food Facts API (Application Programming Interface) for database population and text-only label prediction. The CRNN model is trained on encoded labels and evaluated for accuracy on a dedicated test set. Importantly, our approach enables visually impaired individuals to access essential information on product labels, such as directions and ingredients. Overall, the study highlights the efficacy of deep learning and OCR in automating label extraction and recognition.

KEYWORDS

Optical Character Recognition (OCR); Machine Vision; Machine Learning; Convolutional Recurrent Neural Network (CRNN); Natural Language Processing (NLP); Text Recognition; Test Classification; Product Labels; Deep Learning; Data Extraction.

1. INTRODUCTION

Automating visual inspection on product labels offers several important advantages and benefits, which make it a valuable and essential process in various industries. Automated inspection systems can process large volumes of product labels quickly and consistently, reducing the need for manual labor and accelerating production processes. An efficient system can identify defects, inconsistencies, and deviations in labels, ensuring that products meet regulatory standards and quality requirements. Along the same lines, Optical Character Recognition (OCR) technology is widely used to recognize and extract text and characters from images, allowing for the automated processing and analysis of textual information on various objects, including product labels, documents, and images.

In this research paper, the focus is on addressing a significant societal challenge related to manual data extraction from product labels. This process is known to be tedious, error-prone, and poses accessibility problems for visually impaired individuals. The issue is compounded by the vast array of variations in product labels, including differences in design, colors, text fonts, and sizes, as well as variations in lighting conditions during image capture [1-4]. Traditional OCR methods have been limited in their ability to handle these variations effectively. However, the initiative by researchers in [2] underscores a significant advancement in automatic information extraction from cylindrical prescription labels, with potential applications to assist the elderly and individuals with visual disabilities. To address these challenges, this research introduces an

enhanced, automated OCR system that leverages advanced machine vision and learning methods. The proposed solution makes use of a Convolutional Recurrent Neural Network (CRNN) model, combining the feature extraction capabilities of Convolutional Neural Networks (CNNs) with the sequence recognition strengths of Recurrent Neural Networks (RNNs).

This model is further enhanced through the integration of the Tesseract OCR engine, enhancing its robustness and accuracy [5]. Advanced image processing techniques are applied for efficient preprocessing and noise reduction, while Natural Language Processing (NLP) techniques are integrated to enhance context understanding and semantics.

The implementation of this research has wide-ranging implications across various industries. In healthcare, it can help prevent medication errors by accurately extracting and recognizing medication labels [6]. Food safety organizations can efficiently monitor product ingredients, allergens, and expiration dates, and consumer goods companies can streamline product information management [7, 8]. Although many recent approaches utilizing deep learning models like CNNs and RNNs have shown promise, they have not fully integrated computer vision and NLP techniques [9 - 12]. Importantly, Optical character recognition (OCR) systems provide persons who are blind or visually impaired with the capacity to scan printed text and then have it spoken in synthetic speech or saved to a computer file [13].

Our research also extends data analysis through the Open Food Facts API, enriching the database and fulfilling a text only RNN model for improved label prediction. This comprehensive approach not only revolutionizes the field of text recognition but also makes significant contributions to OCR applications. The research aims to offer a solution that automates information extraction from product labels and does so with exceptional efficiency and accuracy. In summary, this research presents an enhanced solution that effectively integrates computer vision and NLP techniques to automate the extraction and recognition of textual information from product labels, thereby addressing the limitations of existing OCR methods and enhancing the efficiency and accuracy of the process.

2. LITERATURE REVIEW

The challenge of OCR, particularly in unconstrained environments like product labels, has been an important point in the domain of computer vision and text recognition. Traditional OCR methods have been found to have limitations when dealing with text in such unconstrained environments due to geometrical distortions, complex backgrounds, and diverse fonts [15]. One of the significant steps in this domain has been the introduction of CRNN which combines the feature extraction capabilities of CNNs with the sequence recognition capabilities of RNNs [16]. Also, the arrival of deep learning has brought new methodologies that aim to tackle OCR challenges more effectively. However, as Ignat et al. (2022) point out, these methods, particularly end-to-end transformer OCR models, require substantial amounts of annotated data, which poses challenges for low-resource languages. They emphasize the necessity of OCR in improving machine translation for these languages, highlighting a critical gap in OCR research [17].

The beginning of deep learning has brought a new wave of methodologies that aim to tackle these challenges more effectively. Further advancements have been seen with the integration of external OCR engines like Tesseract, which when combined with deep learning methodologies, have shown promise in enhancing OCR capabilities [18]. These integrated models aim to enhance the accuracy and efficiency of text recognition in OCR systems. Moreover, the behavior of transformer models like BERT in language processing has been a subject of recent research. Jawahar et al. (2019) investigated what BERT learns about the structure of language, providing valuable insights into the model's linguistic capabilities [19]. Additionally, Guarasci et al. (2022)

conducted a computational experiment on the syntactic transfer capabilities of BERT in Italian, French, and English, underscoring the adaptability of transformer models across various languages [20].

The application of deep learning techniques for text detection and extraction has also been explored, with algorithms like EAST being utilized to transform text from images or scanned documents into machine-readable form [21]. Furthermore, the OCR is identified as an important branch in the field of machine vision, encompassing pattern recognition, image processing, digital signal processing, and artificial intelligence showcasing the multi-dimensional facets of OCR technologies [22]. Moreover, the evolution of text recognition software has revolutionized text extraction processes, presenting new techniques and technologies that continue to advance the field of OCR. This integration aims to provide a robust solution to text extraction and recognition challenges posed by diverse and complex imagery found on product labels. In fact, the visual online detection method is proposed to ensure the label quality in real time during industrial production, indicating the practical application of OCR in industrial settings [23]. Additionally, there is a growing interest in the application of OCR in various industry sectors. For instance, a deep learning approach to drug label identification has been proposed to address challenges in drug control and distribution, ensuring the provision of safe and approved drugs to consumers and healthcare professionals [23, 24].

Despite the maturity of optical character recognition (OCR) technology, reading data from screens such as 7 segments and LCD screens possess some challenges that illustrate the ongoing challenges in OCR technology [25]. Also, in terms of precision, a system designed for OCR is noted to be capable of interpreting captured images of hard disk drive and solid-state drive labels with high accuracy, emphasizing the progression towards higher accuracy in OCR systems [26 - 27]. Along the same lines, this paper proposition holds of a camera-based assistive text reading framework to help blind persons read text labels and product packaging from hand-held objects in their daily lives underlining the societal impact and accessibility improvements offered by OCR technologies [28]. In the recent past years, there is a significant shift away from traditional OCR methods toward more integrated and advanced approaches that utilize deep learning and external OCR engines to tackle the inherent challenges in text recognition and extraction. This paper aims to improve the extraction of information from product labels, an important aspect for industries like healthcare, food safety, and consumer goods. Conventional optical character recognition (OCR) techniques, which are predominantly rule-based and depend on manually created features and heuristic rules, have demonstrated their limitations, particularly when it comes to managing intricate document layouts, one-of-a-kind fonts, and complex formatting, which may lead to the extraction of crucial data being lost. In industries like finance, law, healthcare, and government, where documents frequently have different formatting requirements, this problem is especially significant [29]. Due to the variation in handwriting styles and quality, these methods also have trouble identifying handwritten text. This can result in mistakes and inaccuracies, which presents serious problems in situations where handwritten documents are common [30].

Another significant drawback of traditional OCR is its reliance on image quality; noise introduced by low-resolution images or dim lighting can make it difficult to recognize characters accurately. Because of its dependence on high-quality images, OCR is less useful when dealing with a variety of image qualities, font variations, and layouts [31]. This is especially true in scenarios that pose difficulties for the visually impaired. Finally, the challenges in data collection, labeling, and regulatory compliance, particularly in sensitive sectors like finance and healthcare, underscore the continuous need for advancements in OCR systems to handle real-world applications more effectively [32].

3. PROPOSED METHODOLOGIES

This section provides a detailed description of the approaches used to process and analyze product labels. It covers the methods, strategies, and algorithms employed, emphasizing the experimental protocols as well as the logic behind the choice of each approach. The following sections are organized as follows: section 3.1 addresses applied methodology; section 3.2 addresses machine vision technique; section 3.3 addresses techniques comparison; section 3.4 addresses similar applications and the limitations.

3.1. Applied Methodology

Figure 1 provides an overview of the comprehensive workflow of the system, covering the entire process from initial data collection to the final assessment of the model. The creation of an efficient automated system for processing and analyzing product labels employed a structured methodology, as depicted in Figure 1. Initially, approximately fifty product label photos from diverse industries were manually selected using an iPhone 11 Pro Max (Apple Inc., Cupertino, California, USA), encompassing both '.jpg' and '.png' formats. To augment and enhance the dataset, additional textual data was obtained from the Open Food Facts API using a free developer account. Integration of this API data was crucial to broaden the dataset's scope, specifically incorporating a diverse range of non-standardized product labels. This approach facilitated the analysis of a more extensive dataset, comprising at least 500 lines, thereby enhancing the depth of the study.

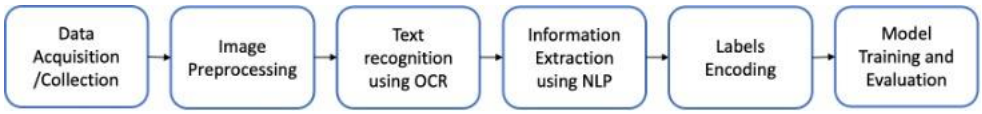


Figure 1. Overall System Overflow

Subsequently, Tesseract OCR was employed for optical character recognition to extract text from the images. In parallel, a PyTorch model was devised to process these images and extract considerable information. Prior to feeding the images into the PyTorch model, image preprocessing steps were undertaken to enhance image quality making them conducive for the model. Furthermore, the text data embedded within the product labels required standardization, hence several text preprocessing techniques were applied using Natural Language Processing (NLP) methods. The methodology included the transformation of all text to lowercase to maintain consistency, removal of punctuation to enhance focus on words and meaningful content, and the elimination of common stop words such as 'and', 'the', 'in' to reduce noise and augment the relevance of the information extracted.

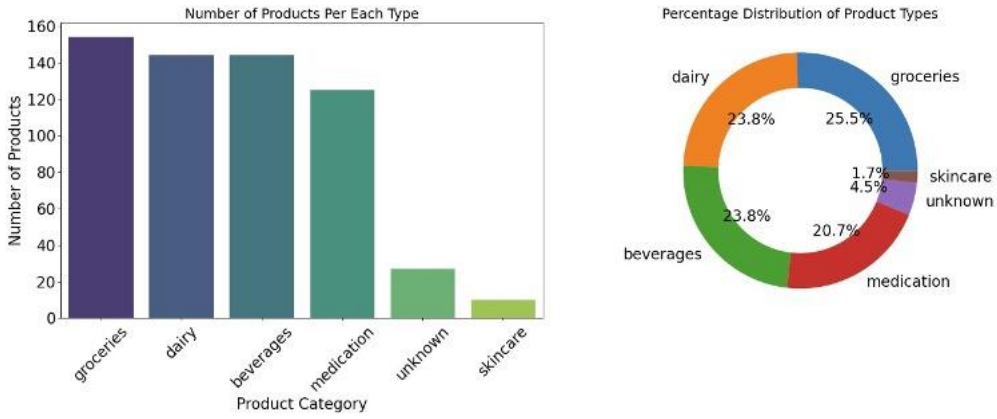
To enhance the processing speed, parallel threading was employed during image processing. Furthermore, Figure 2 shows a visualization of the product categories to further clarify the nature of the dataset derived post image processing. The data was then categorized into distinct features like 'product_type', 'directions', 'supplements_or_elements', and 'warnings', based on predefined keywords. For instance, the 'product_type' feature was discerned by matching keywords related to groceries, skincare, or medication. Moreover, based on the length of the extracted text, the corresponding image was categorized either in the 'extracted_images' or the 'not_extracted_images' directories.

As shown in Figure 2, a step further was taken to group the products into four categories: 'food', ‘medication’, and 'skincare' or an 'unknown', based on keywords associated with each category found in the product's name or ingredients. In addition, each row of the frame presents an image reference, its corresponding category, also the product description or ingredients list and other information extracted.

images/IMG-4249.jpg	food		unknown	foot e abet Sack k See j od teat lute ome
images/IMG-4261.jpg	medication		unknown	DEVELOPED WITH DERMATOLOGISTS Daily Moisturizi...
images/IMG-4262.jpg	medication	aquawatereau glycerin capryliccapric triglycer...		ceraVe Daily Moisturizing Lotion Developed der...
images/IMG-4263.jpg	skincare		unknown	VALUE SIZE fe g HUE Po Bat att ak Hydrating Fa...
images/IMG-4264.jpg	skincare	aquawatereau glycerin cetearyl alcohol peg ste...		CeraVe Hydrating Facial Cleanser Developed der...
images/IMG-4265.jpg	medication	aquawatereau cocamidopropyl hydroxysultaine gl...		en ee ro pe pe ccerave developed essen oil der...
images/IMG-4266.jpg	skincare		unknown	DEVELOPED WITH DERMATOLOGISTS Renewing A Clean...

Figure 2. Data frame representation

After preprocessing, the processed images and the written information taken from Open Food Facts API were combined to create a dataset, as shown in Figure 3. Along the same lines, the visualization provides a concrete illustration of the process of implementing keyword-based product categorization as well as the initial difficulties encountered when selecting a consistent dataset. The Data Frame structure facilitates more efficient data processing and analysis in accordance with the objectives of the study. Following the classification, data cleaning steps were performed to handle missing values, format inconsistencies, and other anomalies, leading to a cleaned and structured dataset converted into a Data Frame. Figure 3a shows the total quantity of products under each category using a bar chart, with "groceries" having the higher number of products. A donut chart in Figure 3b, shows the percentage distribution of these product types. This visual representation highlights the frequency of product types in our dataset and facilitates comprehension of its composition.



a. bar chart representing total quantity of products b. products percentage distribution.

Figure 3. Dataset Visualization

The data frame was improved by using data from product labels that were retrieved through the Open Food Facts API. Complete product details, such as names, brands, ingredients, and countries of origin, were included in the information. However, the abundance of the descriptors found in the product labels extracted text from the API improved the keyword-based product grouping system. A comprehensive dataset comprising five hundred lines was analyzed, as illustrated in the figure 3, leading to a more precise classification of products into categories like "groceries," "dairy," "beverages," "medication," and "skincare," as well as accounting for

"unknown" classifications in situations where the data was ambiguous or not present. The histogram and pie chart, which present the number and percentage distribution of product types graphically, respectively, demonstrate the diversity and size of the dataset made possible by this integration. Finally, using the Open Food Facts API improved the quantity and quality of available data, allowing for the making of more informed and superior decisions based on information. The machine learning techniques implementation included the use of three neural network architectures: CNN (Convolutional Neural Network), RNN (Recurrent Neural Network), and CRNN (Convolutional Recurrent Neural Network). The CNN network is primarily responsible for processing image data.

In addition, the images of product labels are first preprocessed and then fed into CNN. This network extracts features from these images, which are crucial for understanding visual patterns in the product labels. Additionally, the RNN processes sequential data as the textual data extracted from the product labels. The sequential nature of RNNs makes them well-suited for text as it considers the order and context of words. Lastly, CRNN combines the features of both CNN and RNN. Furthermore, the network takes the image features extracted by CNN and the text processed by the RNN to perform a comprehensive analysis of the product labels. This approach enables the model to understand both the visual and textual content of the labels, enhancing the accuracy of your classification or analysis task.

For instance, PyTorch was used to create a deep learning model for classifying product labels. With class weights, a Label Encoder and CrossEntropyLoss as the loss function were used to encode the text data. The performance of the CRNN model on untested data was assessed following the training phase using a testing Data Loader. Additionally, the database enrichment made room for the development of an RNN implementation that just accepted text input. Also, we leveraged a text-only Recurrent Neural Network (RNN) model, combining data from image extractions and the Open Food Facts API into a singular Data Frame. Our RNN and Gated Recurrent Unit (GRU) models, built with PyTorch, processed the textual information with an embedding layer followed by recurrent layers to accommodate sequential data. These models utilized LSTM and GRU units to better capture dependencies in text data, which is crucial for classification tasks.

To assess the model's classification performance, evaluation metrics such as accuracy and precision were calculated. The weights of the trained CRNN model were stored in a file called "crnn_model_new.pth" in case they were needed again. Finally, the model underwent several training phases at different learning rates. Lastly, the performance metrics for each learning rate were stored in a metrics dictionary and visualized using seaborn to elucidate the relationship between learning rates and performance, enabling a comprehensive understanding of the model's behavior and performance under different conditions.

3.2. Machine Vision Techniques

To enhance the OCR process, we incorporated machine vision techniques. Image preprocessing techniques such as image normalization, denoising, and contrast enhancement were applied to improve the quality of label images. Additionally, image analysis algorithms, including contour detection and segmentation, were employed to isolate and extract individual characters or text regions from the labels.

Data Extraction Algorithm

The OCR algorithm takes a single image and returns the extracted information in these ordered steps:

1. Image Loading: The image is loaded from the specified path using OpenCV.
2. Color Conversion: Convert the image from BGR to RGB color space.
3. Image Resizing: Resize image proportionally to standardize the images, which helps in normalizing the variations in dimensions.
4. LAB Color Space Conversion and Splitting: The resized image is converted to the Lab color space, and the channels are split. This space is chosen for its ability to separate grayscale information from color data.
5. CLAHE: Apply CLAHE to the L channel to improve contrast. Merge enhanced L with A and B channels.
6. RGB Conversion and Sharpening: Convert LAB image back to RGB. Sharpen the RGB image.
7. Grayscale Conversion and Thresholding: Convert sharpened image to grayscale. Otsu's thresholding is applied to binarize the image, simplifying the detection of text regions.
8. Black/White Ratio Calculation: Calculate ratios of black and white pixels.
9. Image Inversion: If the image is mostly black or white, invert the binary image.
10. Contour Detection and Extraction: For color images, dilate the image, find contours, and detect text regions. Extract contour with the maximum area and crop the image.
11. Text Extraction: Extract text from the cropped image using Pytesseract.
12. Data Extraction and Image Saving: Extract data from the text and save images based on text length.
13. Exception Handling: Handle any errors during image processing, log the error message.

The settings for filters, such as the Contrast Limited Adaptive Histogram Equalization (CLAHE), and threshold values, are chosen based on their proven effectiveness in similar image processing tasks. Specifically, the CLAHE settings are tuned to enhance the contrast of the L channel in the LAB color space without amplifying noise, which is critical for improving OCR accuracy on unevenly lit images. The clip limit and tile grid size parameters for CLAHE are selected to balance contrast enhancement and detail preservation. For binary thresholding, Otsu's method is employed due to its automatic calculation of the optimal threshold value, which separates the pixel intensity distribution into two classes, foreground, and background. This method is particularly effective for document images where there is a clear distinction between text (foreground) and the rest of the image (background). Along the same lines, the color space notation, the LAB color space is represented as Lab or Lab*. This notation reflects the three axes of the color space: L for lightness and a and b for the color-opponent dimensions. Using the correct notation is not just a matter of accuracy in terms; it also ensures adherence to the established standards in image processing and color science literature.

Figure 4 showcases a few sample images after preprocessing and machine vision techniques on label images. These three sample images, ranging from a mouthwash product, a medicinal label, to a nutritional fact sheet of ground beef, depict the potential challenges faced during OCR extraction, such as complex backgrounds, warped perspectives, and varying text densities.

a: Mouthwash label b: Vaseline label c: Food label

Figure 4. Preprocessed Images.

Figure 4a (Mouthwash Label) is likely to be full of text and various symbols, including certification marks which could complicate text recognition. The complexity comes from both the amount of textual information and the variety of formats such as logos, small print, and potentially decorative elements in the background that are common on personal care products. Figure 4b (Vaseline Label) presents a different challenge due to the curvature of the surface on which it is attached. Labels on curved surfaces can cause distortions in the text, making it harder for OCR to accurately read the information. The label's text may be stretched or compressed due to the shape of the container, and the lighting in the image may create reflections or shadows that interfere with text detection. Lastly, Figure 4c (Food Label) holds the nutritional information label for ground beef showcases the need for OCR to accurately capture numerical data and percentages. This is crucial for consumers who rely on this information for dietary and health reasons. The label's text density and the importance of precision in capturing the data make it a good example of the need for advanced OCR techniques, which may include steps like color conversion and contour detection to improve text extraction accuracy.

Each label requires a nuanced approach to OCR pre-processing to handle issues like complex backgrounds, warped perspectives from curved surfaces, and high text density. These examples highlight the necessity of a multi-step algorithmic method. As mentioned, the algorithm includes various stages such as image loading, pre-processing steps like color conversion, grayscale conversion, noise removal, and more sophisticated steps like contour detection and perspective transformation to handle warped text. Exception handling is also crucial to ensure that anomalies or unexpected issues don't lead to incorrect text extraction.

3.3. Natural Language Processing (NLP) in Product Labels

The NLP framework within our methodology utilizes an ensemble of techniques to interpret and categorize the text extracted from product labels. The process starts with the application of a lemmatizer, which reduces words to their base or dictionary form. This is particularly useful in handling the various inflected forms of words encountered in product descriptions, ensuring that 'runs', 'running', and 'ran' are all analyzed as 'run'.

Stop words, typically the most common words in a language that carry minimal unique information (like 'the', 'is', 'at'), are filtered out to focus on the more meaningful content. Removal of these words is a standard practice in NLP to reduce noise and computational load.

The text extraction process includes several customized steps:

1. Applied text lowercasing: Converting all characters to lowercase to maintain uniformity and prevent the same words in different cases from being treated as distinct.
2. Keyword Extraction: Identifying specific keywords such as 'ingredients', 'directions', and 'warnings' using a combination of string operations and regular expressions. The extraction is sensitive to the context, searching for these terms within the relevant sections of the text.
3. Tokenization: Breaking down text into individual terms or tokens, which are then analyzed for patterns or used as input for further processing like classification.
4. Lemmatization: Applying the lemmatizer to reduce tokens to their lemmas, thus aiding in the generalization and reducing the complexity of the textual data.
5. Text Cleaning: Implementing a series of regular expressions to remove non-alphanumeric characters, excess whitespace, and numbers where they are not part of the meaningful text. This step also includes encoding and decoding operations to handle non-ASCII characters, which can often be found in product labels due to internationalization.
6. Advanced Text Processing: Utilizing NLP libraries like Text Blob to correct spelling, which is crucial for ensuring the accuracy of keyword-based categorization.
7. Feature Extraction: From the cleaned and tokenized text, key features are extracted based on predefined categories, which are then used to train machine learning models. This step involves identifying the presence of specific terms that are indicative of product types, such as differentiating food items from skincare products.

In summary, the described NLP techniques are deeply integrated into the overall architecture, playing an essential role in the system's ability to accurately categorize and understand the product labels. Each technique contributes to refining the text data, which is essential for the subsequent machine learning stages.

3.4. Technique Comparison with Traditional OCR Methods

We implemented and compared our approach with traditional OCR methods, specifically LSTM, transformer-based models, and CRNN. LSTM (Long Short-Term Memory) is a recurrent neural network (RNN) model apt for OCR (Optical Character Recognition) tasks due to its memory cells' ability to retain long-term information. Unlike traditional neural networks, LSTM incorporates feedback connections, allowing it to process entire sequences of data, not just individual data points. This makes it highly effective in understanding and predicting patterns in sequential data like time series, text, and speech [33]. Its workflow includes preprocessing input images, text extraction using OCR techniques like Tesseract, feature extraction through keyword matching, and training an LSTM model using encoded label features. The model's architecture can include embedding layers, LSTM layers, and fully connected layers. It is trained on labeled data, validated, and evaluated for accuracy on a test set.

Transformer-based models, like those in the EasyOCR library, have shown promising results in OCR tasks due to their self-attention mechanisms and parallel processing capabilities. The workflow mirrors that of the LSTM approach, except for text extraction, where a transformer-based model like EasyOCR is used. Furthermore, transformer-based OCR is an end-to-end transformer-based OCR model for text recognition, this is one of the first works to jointly leverage pre-trained image and text transformers [34]. These models require large datasets and computational resources for effective training, and their performance is evaluated through metrics like accuracy, precision, and recall.

CRNN (Convolutional Recurrent Neural Network) leverages both CNN (Convolutional Neural Networks) and RNN's strengths for image and sequence recognition. The workflow is akin to the

previous approaches, with the training involving a CNN for image feature extraction and an RNN (such as LSTM or GRU) for sequence modeling. This combined model learns visual representations and captures label sequence dependencies. The CRNN model is described as a unified framework that integrates feature extraction, sequence modeling, and transcription, which does not require character segmentation and can predict label sequences with relationships between characters [35].

Our comparison with conventional OCR techniques showed that transformer-based models outperform LSTM models because of their self-attention processes, while LSTM models are resilient when processing sequential data. Nevertheless, our results show that CRNN provides an improved method for OCR tasks by combining CNN's capability to extract hierarchical visual characteristics with LSTM's sequential data processing. This hybrid model is the most promising for improving OCR skills because it can capture the subtleties of temporal and spatial correlations in text data. It excels at interpreting text from a variety of font styles and complex backdrops that are seen in real-world settings.

3.5. Similar Applications and the Limitations

3.5.1. OCR Space website API

The OCR space website supplies an OCR (Optical Character Recognition) API for extracting text from images. OCR space provides a simple and convenient OCR (Optical Character Recognition) API for parsing images and multi-page PDF documents, returning the extracted text results in a JSON format. The API offers three tiers: Free, PRO, and PRO PDF. The Free OCR API plan is suitable for users who want to explore and test the OCR functionality. It comes with a rate limit of 500 requests per day per IP (Internet Protocol) address to prevent unintended spamming [36]. This plan allows developers to get started without any financial commitment.

The OCR space website supplies an OCR (Optical Character Recognition) API for extracting text from images. However, there are limitations associated with using the free version of the API. One limitation is that the API commands can only be used every 600 seconds (about 10 minutes). Additionally, the image size must be less than 1 MB to be processed by the API. However, there are other payable deals that offer great solutions to general image OCR implementations.

It is also important to note that while the Free OCR API plan has certain limitations such as the rate limit and file size limit of 1 MB, the PRO plans offer expanded capabilities, including larger file size support, higher request volumes, and customizable OCR servers. Finally, the PRO OCR API can be purchased as locally installable OCR software, providing organizations with the flexibility to have their OCR servers set up at a location of their choice. This option is suitable for users with specific data security or compliance requirements.

3.5.2. MMOCR

MMOCR is an OCR toolbox that provides a comprehensive set of tools for optical character recognition. To use MMOCR, Microsoft Visual C++ 14.0 or greater must be installed on the system to compile the pycocotools package, a dependency for MMOCR.

The installation process for MMOCR involves installing several dependencies using the "openmim" package manager. These dependencies include "mmengine," "mmcv" (version 2.0.0rc1 or later), and "mmdet" (version 3.0.0rc0 or later). More detailed installation instructions can be found on the MMOCR documentation website [37].

MMOCR offers two main components for OCR: detection and recognition. It is worth noting that while MMOCR offers high accuracy in text recognition, it has some limitations. The approach struggles with images of different scales, often resulting in extremely small images. Resizing the images manually can help overcome this issue. Furthermore, based on the MMOCR official documentation, also as a drawback, the training of the model can take up to 400 phases to be trained and tested to provide high accuracy which will require much more powerful computer with high speed. Additionally, the performance of the MMOCR approach is slower compared to other OCR methods.

In conclusion, MMOCR provides a powerful OCR toolbox with efficient detection and recognition capabilities [37]. By following the installation instructions and utilizing the approach examples, users can leverage MMOCR to extract text from images effectively. However, it is important to consider the limitations, such as the need for manual resizing of images and the slower processing speed, when using MMOCR for OCR tasks.

3.5.3. EAST and CRAFT

The implementation of text detection using the CRAFT (Character Region Awareness for Text Detection) algorithm performs text detection on both black-and-white and color images. The CRAFT algorithm involves several steps, including text region detection using the CRAFT model and subsequent preprocessing and OCR steps. The CRAFT algorithm involves steps such as text region detection using the CRAFT model, contrast enhancement, noise removal, CLAHE for even lighting, and Otsu's thresholding. CRAFT's accuracy surpasses other methods like CTPN, EAST, and MSER, especially in complex text scenarios, although it may have a longer processing time compared to these methods [38].

Similarly, the EAST algorithm relies on a fully convolutional neural network (FCN) for text detection. However, the EAST approach also faces issues with optimization and modularity, as indicated by the encountered module errors in C++ from the OpenCV library. The limitations of the EAST algorithm for product label OCR include challenges in capturing complex designs and layouts accurately, difficulty in detecting small text or fine details, and potential interference from background noise or clutter.

Both the CRAFT and EAST algorithms lack semantic understanding of text regions, treating all regions equally without considering their specific context or structure. This limitation is particularly relevant for product label OCR, where different text regions often convey diverse types of information. To overcome these challenges, alternative approaches tailored for product label OCR can be explored. These approaches may involve combining text detection methods that account for the unique characteristics of product labels, employing specialized preprocessing techniques, utilizing domain-specific models, and incorporating techniques for semantic understanding of the text content. Customizing the OCR pipeline to address the specific challenges of product labels can improve the accuracy and reliability of OCR results.

In conclusion, while both CRAFT and EAST offer valuable capabilities in text detection, their limitations point to the need for further development in OCR technology. This involves combining text detection methods tailored for specific applications, utilizing specialized preprocessing techniques, and incorporating models that understand the semantic context of text content [21]

3.5.4. BLOB/Boxes

The approach implements text detection using the BLOB (Binary Large Object) approach. It includes functions for extracting bounding boxes, cropping images, applying non-maximum suppression, rescaling boxes, and filtering boxes based on specific criteria.

In BLOB detection, parameters such as binarization threshold levels, minimum and maximum pixel height and width are set to identify the desired text regions. The choice of these parameters significantly influences the detection process. A higher binarization threshold can lead to the detection of more dominant white pixels, while a lower threshold might result in a larger number of undetected regions if the image has more dominant black pixels. Adjusting these parameters allows the developer to target specific blob characteristics, but it requires a balance to achieve accurate detection without overwhelming the system with too many potential regions [39].

However, a major limitation of this approach is that it encounters an issue where there are more than 1600 boxes per image, causing the algorithm to get stuck in a loop for each box and resulting in extremely slow performance. This significantly decreases the practical usability of the BLOB approach. To address this limitation, an alternative contour-based approach can be considered. The contour-based approach involves detecting contours in the image and filtering them based on size, shape, or hierarchy. This technique significantly reduces the number of regions to be processed, leading to improved algorithm speed. The contour-based approach is more efficient in managing complex images, as it targets specific features of the text regions, thus reducing the likelihood of processing irrelevant or excessive data [40].

In summary, the BLOB approach provides functions for text detection but suffers from poor performance due to the substantial number of boxes per image. To overcome this limitation, exploring the contour-based approach and incorporating contour filtering and edge detection techniques can offer a more efficient and accurate text detection solution.

3.5.5. Google Vision API

The Google Vision API is a notable possibility for label OCR, offering powerful capabilities for text recognition. It utilizes advanced machine learning models to accurately extract and interpret text from images, making it an excellent choice for various OCR tasks, including label extraction from product images.

The API incorporates optical character recognition, natural language processing, and computer vision algorithms, ensuring reliable and accurate results. It handles complexities such as different fonts, sizes, orientations, and backgrounds, providing a comprehensive solution for label OCR [41].

However, there are important considerations when opting for the Google Vision API. Firstly, it is a paid service, requiring users to create a developer account and adhere to Google's pricing plans. The costs associated with using the API depend on factors such as the volume of image requests and the specific features utilized. This aspect of implementation may pose challenges for researchers without technical expertise or limited development resources. Additionally, utilizing the Google Vision API requires substantial computational resources. The API relies on cloud-based infrastructure and powerful computing resources to process and analyze images effectively, potentially resulting in additional computational expenses, particularly for high-volume or real-time applications.

Despite these considerations, the Google Vision API remains a reliable choice for label OCR due to its robust capabilities and high accuracy. It offers advanced features and comprehensive tools for extracting text from images. Users need to be aware of the associated costs, the need for a developer account, and the potential computational expenses involved when considering the Google Vision API as an OCR solution. Evaluating these factors will enable users to make informed decisions based on their specific requirements and available resources.

In summary, the comparison of OCR tools in Table 1 provides an overview of various OCR technologies, highlighting their key features and limitations. Furthermore, this table serves as a quick reference for understanding the strengths and weaknesses of each OCR tool, aiding in informed decision-making for specific OCR tasks.

Table 1. Comparison of OCR Tools

OCR Tool	Key Features	Limitations
MMOCR	Multi-model OCR, Modular, OpenMMLab ecosystem	Struggles with scale, High computational need
EAST	Convolutional network, Complex layout handling	Limited semantic understanding, Modularity issues
CRAFT	Character-level detection, Text region focus	Similar limitations as EAST
BLOB/Boxes	Bounding box extraction, Image manipulation	Poor with many regions, Slow
OCR Space API	User-friendly, Free and paid versions	Rate/file size limits on free, Paid for full features
Google Vision API	Advanced ML models, Versatile OCR	Paid service, High computational resources

4. EXPERIMENTAL RESULTS

4.1. Text Recognition and Classification

In this section, the performance of the OCR system extraction and classification of textual data from two sample product labels were analyzed.

4.1.1. Analysis of Eye Drop Product Label

The processed image as shown in Fugure-5 (The eye drops image/text) demonstrates how the OCR result offers several insights. The image showcases a label from a bottle of eye drops. The label contains multiple sections providing varied information, a list of active ingredients and their respective purposes, as well as warnings about potential contraindications and side effects. Throughout the label, text fonts and sizes vary to emphasize certain sections over others. The main extracted text demonstrates how the OCR system can recognize important parts like “Drug Facts,” “Active ingredients,” “Uses,” “Warnings,” and “Directions.” In addition, the system classifies information, demonstrating its contextual comprehension skills, going beyond mere text extraction.

For instance, “Instill 1 or 2 drops in the affected eye(s) as needed” is the distinct usage instructions, but it has a few little errors, including the prefix “m”. The product category is appropriately identified as “medication.” The ‘supplements_or_elements’ label for the active

substances raises the possibility of improving the naming standards. Finally, the important user warning instructions are retrieved with small distortions and prefixed aberrations such as “m” and “Mm”. Lastly, the file path for the image was also captured by the system, and this information may be crucial for traceability and additional research.

The text that was retrieved shows how well the system identified the main content pieces. Still, the existence of twisted words—for example, “LUDriCant” rather than “Lubricant”—indicates regions where the OCR still needs further refining. If random letters like “m” appear before certain words, there may be noise in the image, or the algorithm can be refined even further to get rid of these prefixes. Classifying parts such as ‘directions’, ‘warnings’, and ‘product_type’ indicates that the algorithm has a comprehension of the context of the text.


```
{'directions': 'm instill 1 or 2 drops in the affected eye(s) as needed',
 'extracted_text': 'Drug Facts Active ingredient Purpose Glycerin ss LUDriCant '
                   'Propylene glycol 10ss0LUDricant Uses temporary relief '
                   'burning irritation due dryness eye prevents irritation '
                   'Warnings For external use use solution change color '
                   'becomes cloudy When using product touch tip container '
                   'surface avoid contamination remove contact lens using '
                   'replace cap use Stop use ask doctor experience eye pain '
                   'change vision continued redness irritation eye condition '
                   'worsens persists 72hours Keep reach child swallowed get '
                   'medical help contact Poison Control Center right away '
                   'Directions instill drop affected eye needed',
 'image_path': 'C:/Users/hseit/Desktop/SummerMCREU/images/IMG-4237.jpg',
 'product_type': 'medication',
 'supplements_or_elements': 'Active',
 'warnings': 'For external use only  Do not use m if solution changes color or '
             'becomes cloudy  When using this product  m do not touch tip of '
             'container to any surface to avoid contamination  m remove '
             'contact lenses before using  m replace cap after use  Stop use '
             'and ask a doctor if Mm you experience eye pain, changes in '
             'vision,  *  continued redness or irritation of the eye m '
             'condition worsens or persists for more than 72.hours  Keep out '
             'of reach of children. If swallowed, get medical help or contact '
             'a Poison Control Center right away'}
```


Figure 5. Eye Drop Label.

4.1.2. Analysis of CeraVe Daily Moisturizing Product Label

The “CeraVe Daily Moisturizing Lotion” label was subjected to the Optical Character Recognition (OCR) method. First, the product’s title and description: “CeraVe Daily Moisturizing Lotion” was clearly identified by the OCR. Additionally, the algorithm found a significant description, which highlights the lotion’s special formulation created in collaboration with specialists and its effectiveness in repairing the skin barrier. Figure 6 presents a label from a CeraVe Daily Moisturizing Lotion bottle. It begins with the product name prominently displayed, followed by a brief description emphasizing its unique formula developed with dermatologists The middle section provides detailed directions for the lotion’s application and a comprehensive list of ingredients. Icons and varied text formatting are used to visually distinguish different sections of the label.

Furthermore, the algorithm effectively emphasized several of the product’s characteristics. For example, there is included the MVE Delivery Technology, being oil-free, having hyaluronic acid, not containing fragrance, and having undergone allergy testing. The algorithm did a commendable job of extracting the lotion’s application recommendations.
However, the limitations make the algorithm not perfect in which shows a few minor issues such as: “Avoid direct contact eye” was meant to say, “Avoid direct contact with eyes,” however there was a small misunderstanding.

A comprehensive list of all the recognized ingredients in the lotion's recipe was provided. Characters weren't always appropriately distinguished or divided, though. For instance, "CERAMIDE CERAMIDE CERAMIDE EOP" felt like an over-extraction, and "CAPRYLICCAPRIC TRIGLYCERIDE" should ideally be "CAPRYLIC/CAPRIC TRIGLYCERIDE."


```
{'directions': ': Apply liberally as often as needed, or as directed by a '
               'physician. Avoid direct contact with eyes. If contact occurs '
               'rinse thoroughly with water.  20215004V01 -',
 'extracted_text': 'ceraVe Daily Moisturizing Lotion Developed dermatologist '
                   'unique formula essential ceramides moisturizes help '
                   'restore protective skin barrier Technology Oil Free '
                   'Controlled release Noncomedogenic day hydration wont clog '
                   'pore Hyaluronic Acid mae Helps retain skin natural '
                   'moisture Fragrance Free avoid fragrance irritation Gentle '
                   'skin Allergy tested Noncomedogenic Directions Apply '
                   'liberally often needed directed physician Avoid direct '
                   'contact eye contact occurs rinse thoroughly water '
                   '20215004V01 INGREDIENTS AQUAWATEREAU GLYCERIN '
                   'CAPRYLICCAPRIC TRIGLYCERIDE CETEARYL ALCOH CETYL ALCOHOL '
                   'POTASSIUM PHOSPHATE CERAMIDE CERAMIDE CERAMIDE EOP '
                   'CARBOMER DIMETHICONE CETEARETH20 BEHENTRIMONIUM '
                   'METHOSULFATE METHYLPARABEN SODIUM LAUROYL LACTYLATE '
                   'CHOLESTEROL DISODIUM EDTA DIPOTASSIUM PHOSPHATE '
                   'PROPYLPARABEN HYDROLYZED HYALURONIC ACID PHYTOSPHINGOSINE '
                   'XANTHAN GUM POLYSORBATE POLYGLYCERYL3 DIISOSTEARATE CODE '
                   'FLL D2157632 CeraveLte New York MWWEisa registered '
                   'trademark DFB Technology Ltd ade USA andor Imported '
                   'Ingredients WWceravecom Uestions comment',
 'image_path': 'C:/Users/hseit/Desktop/SummerMCREU/images/IMG-4262.jpg',
 'supplements_or_elements': ': AQUA/WATER/EAU, GLYCERIN, CAPRYLIC/CAPRIC '
                            'TRIGLYCERIDE, CETEARYL ALCOH OL, CETYL ALCOHOL, '
                            'POTASSIUM PHOSPHATE, CERAMIDE NP, CERAMIDE AP, '
                            'CERAMIDE EOP, CARBOMER, DIMETHICONE, '
                            'CETEARETH-20, BEHENTRIMONIUM METHOSULFATE,  | '
                            'METHYLPARABEN, SODIUM LAUROYL LACTYLATE,  _ '
                            'CHOLESTEROL, DISODIUM EDTA, DIPOTASSIUM '
                            'PHOSPHATE, PROPYLPARABEN, HYDROLYZED HYALURONIC '
                            'ACID, PHYTOSPHINGOSINE, XANTHAN GUM,  ___ '
                            'POLYSORBATE 20, POLYGLYCERYL-3 -v*  | _ '
                            'DIISOSTEARATE (CODE FLL. D215763/2) N  po ——— '
                            '(-)  | a  ee —-)  | Cerave@Lte, 0  New York, NY '
                            '10001 -$-$-$———s(_)  MWWEisa registered '
                            'trademark =  of DFB Technology, Ltd, ae  ade in '
                            'USA of US and/or ™) Imported'}
```


Figure 6. CeraVe Daily Moisturizing Lotion Label.

In conclusion, the OCR system has demonstrated admirable performance in text classification and recognition from the image of these sample product labels. A significant amount of data has been successfully recovered, the existence of minor errors suggests that more improvements are required, including in the areas of noise reduction, special character identification, and contextual comprehension.

4.2. Online Platform Comparison

When comparing OCR systems, our custom algorithm and OnlineOCR.net both demonstrated remarkable effectiveness in extracting text from images, even with certain distinctive characteristics. Both solutions were able to capture and transcribe significant elements of text from the source image. In addition, the image below shows the comparison between the words extracted from these two algorithms. For instance, the test image is a photo taken by an iPhone 11 Pro Max (Apple Inc., Cupertino, California, USA) device and the product name is 'Ice Spice Deodorant'.

Figure 7. OCR Accuracy Evaluation.

The efficiency of our unique OCR method is seen in Figure 7 when compared to the OnlineOCR.net solution. Based on a target picture of the "Ice Spice Deodorant" label, the transcribing results from both platforms are shown in the attached image. Visual examination reveals that both methods can capture a sizable percentage of the text, but with varying degrees of reliability and completeness. In the aspect of data volume, both OCR systems were able to successfully extract a considerable amount of text from the image. Even though OnlineOCR.net gathered more specifics, our proprietary OCR algorithm proved competent by drawing out most of the text components. Notably, both solutions precisely identified and transcribed crucial segments such as product details, usage instructions, warnings, and ingredient lists.

Despite variances in extraction precision, it's noteworthy that our proprietary OCR algorithm showcased significant efficiency. For instance, text strings like "48 HOUR ODOR PROTECTION," "DIRECTIONS: Twist up product. Apply to underarms only, Use daily for best results," and "INGREDIENTS: DIPROPYLENE GLYCOL, WATER, PROPYLENE GLYCOL, SODIUM STEARATE, POLOXAMINE 1307, FRAGRANCE, PPG-3 MYRISTYL ETHER, TETRASODIUM EDTA, VIOLET 2, GREEN 6" were accurately transcribed by both OCR platforms, indicating that our proprietary OCR algorithm was able to match the precision of OnlineOCR.net in specific aspects.

Regarding the output presentation, both solutions delivered results in a format that was easy to read and accessible. Although there's potential for refining the structure in our proprietary OCR algorithm, it still demonstrated parallel performance in terms of generating a user-friendly output.

To conclude, the proprietary OCR algorithm showed impressive capability in text extraction from images, equating to OnlineOCR.net in various dimensions. Its accuracy in transcribing certain vital text components was particularly impressive, which illustrated the considerable efficacy of our algorithm. While there were instances where OnlineOCR.net showed a minor advantage, the review emphasized the potential of our custom solution. This analysis highlights the areas for improvement in our OCR system, such as enhanced error management, fine-tuned pre-processing methodologies, and post-processing spelling corrections. Despite slight differences when contrasted with a tool like OnlineOCR.net, our OCR algorithm is a well-established, generic solution.

4.3. Implementation in the Web

In a bid to further reveal and demonstrate "LabelOCRWebsite," serves as an interactive interface allowing users to upload images of product labels and retrieve the extracted textual information

therein. The source code and assets for this platform are publicly accessible on GitHub under the repository “LabelOCRWebsite”.

The “LabelOCRWebsite” is structured as a Flask application, integrating HTML and Bootstrap for the front-end design, thereby ensuring a user-friendly and intuitive interface. The back-end logic, written in Python, harnesses the OCR model to process uploaded images, extract textual data, and present the results to the user in a readable format. This platform exemplifies a real-world application of the OCR system, showcasing its potential in addressing the challenges of manual data extraction from product labels.

Furthermore, to supply a comprehensive understanding and an easy-to-follow guide on how the OCR system was developed and implemented, a Google Collab Notebook was prepared. Although the contents of the notebook could not be accessed directly, it’s presumed to have a step-by-step walkthrough of the OCR model’s training, evaluation, and deployment process, possibly alongside relevant code snippets and explanations. This notebook is intended to serve as an educational resource, helping the reproducibility of the research and offering insights into the methodologies employed.

The synergy between the “LabelOCRWebsite” platform and the Google Collab notebook presents a holistic approach to not only proposing a solution to a societal challenge but also ensuring that the community can access, learn from, and build upon this research. The interactive platform demonstrates the practical utility of the OCR system, while the Collab notebook provides a detailed exposition of the technical underpinnings and procedures involved in realizing this solution.

5. CONCLUSIONS

This research marks a significant advancement in the domain of Optical Character Recognition (OCR), specifically in the context of product label data extraction. By integrating a Convolutional Recurrent Neural Network (CRNN) model, this study bridges the gap between conventional OCR methods and the demands of modern data processing.

The CRNN model, integrating the feature extraction process of CNNs with the sequence recognition strength of RNNs, demonstrated promise, achieving a peak validation accuracy of 47.06% over ten epochs. However, this performance level indicates that there is substantial room for optimization and improvement. The intricacies of OCR tasks, especially in diverse and complex environments like product labels, necessitate ongoing advancements and refinements in model architecture and training methodologies.

The customized approach to OCR, leveraging the renowned Tesseract OCR engine, enhances the system’s text recognition capabilities. The integration of NLP techniques acts as a catalyzer, further boosting the model’s overall efficiency and performance. Another groundbreaking facet of this research is the use of the Open Food Facts API. The incorporation of data from the Open Food Facts API provided a multi-dimensional approach to text extraction, ensuring a more comprehensive and accurate analysis of product labels. This approach underscores the potential of integrating various data sources and techniques in OCR tasks.

While LSTM and other traditional OCR methods made significant strides, they faced challenges, particularly with image variability. Transformer-based models, on the other hand, though advanced, demanded substantial computational resources. The proposed CRNN model ingeniously bridges the gap, amalgamating the strengths of both CNN and RNN architectures. In a subsequent exploration using a GRU model, the system showed a sharp increase in its fifth

epoch, achieving a validation accuracy of 58.82%. Finally, this performance discrepancy accentuates the intricate nature of OCR and the necessity for sustained research.

This research not only addresses the immediate challenge of extracting data from product labels but also sets the stage for future innovations. The implications of this work extend to critical sectors such as healthcare and food safety. Moreover, it opens avenues for enhancing accessibility for visually impaired individuals, highlighting the societal impact of advancements in OCR technology. In practice, the results of this study can be implemented in various applications where accurate and efficient text extraction from images is crucial. These include consumer goods labeling, healthcare information systems, and automated data entry processes in various industries. The findings also pave the way for future research, particularly in refining deep learning models for OCR, exploring the integration of diverse data sources, and adapting OCR technologies to different languages and contexts.

This model has the following limitations: 1. The output of the OCR technique depends on the clarity of the image, 2. Currently, the system works on uploaded images for processing and 3. The developed techniques are not tested on labels with vertical texts. In future, the developed systems can be integrated together to form a functioning software which can process real-time video captured from a mobile phone at a desirable distance and output text describing the product label and answer queries by user. The techniques developed by our previous work as described in [2] can also be combined to accommodate cylindrically distorted images.

6. Abbreviations

OCR: Optical Character Recognition
CNN: Convolutional Neural Network
RNN: Recurrent Neural Network
CRNN: Convolutional Recurrent Neural Network
GRU: Gated Recurrent Unit
LSTM: Long Short-Term Memory
NLP: Natural Language Processing
API: Application Programming Interface

REFERENCES

[1] Chauhan, A. Drawing Bounding Box Method in Image Processing. Medium. 2022. Available online: https://pub.towardsai.net/drawing-bounding-box-method-in-image-processing-ec7487393cfa (accessed on 31 July 2023).

[2] Gromova, K.; Elangovan, V. Automatic Extraction of Medication Information from Cylindrically Distorted Pill Bottle Labels. Mach. Learn. Knowl. Extr. 2022, 4, 852-864. https://doi.org/10.3390/make4040043

[3] Grubert, J.; Gao, Y. Mobile OCR: A Benchmark for Publicly Available Recognition Systems. Stanford University. 2023. Available online: https://stacks.stanford.edu/file/druid:bf950qp8995/Grubert_Gao.pdf

[4] How to Find the Bounding Rectangle of an Image Contour in OpenCV Python. Available online: https://www.tutorialspoint.com/how-to-find-the-bounding-rectangle-of-an-image-contour-in-opencv-python (accessed on 19 June 2023).

[5] Tesseract OCR. Tesseract Documentation. Tesseract OCR. 2023. Version 5.0.0. Available online: https://tesseract-ocr.github.io/tessdoc/Home.html (accessed on 31 July 2023).

[6] Institute of Medicine (US), Roundtable on Environmental Health Sciences, Research, and Medicine. Global Environmental Health in the 21st Century: From Governmental Regulation to Corporate Social Responsibility: Workshop Summary. Washington (DC): National Academies Press (US);

2007. 5, Corporate Social Responsibility. Available from: https://www.ncbi.nlm.nih.gov/books/NBK53982/ (accessed on 31 July 2023).
[7] Open Food Facts - World. Available online: https://world.openfoodfacts.org (accessed on 31 July 2023).
[8] OpenCV. Documentation. OpenCV. 2023. Version 4.8.0. Available online: https://docs.opencv.org/4.x/ (accessed on 31 July 2023).
[9] PIL (Pillow). Pillow (PIL Fork) Documentation. Read the Docs. 2023. Version 9.5.0. Available online: https://pillow.readthedocs.io/en/stable/index.html (accessed on 31 July 2023).
[10] PyTorch. PyTorch Documentation. Version 2.0.1+cpu. PyTorch. 2023. Available online: https://pytorch.org/docs/stable/index.html (accessed on 31 July 2023).
[11] Rosebrock, A. OpenCV Text Detection (EAST Text Detector). PyImageSearch. 2018. Available online: https://pyimagesearch.com/2018/08/20/opencv-text-detection-east-text-detector/ (accessed on 31 July 2023).
[12] Sun, H.; et al. Spatial Dual-Modality Graph Reasoning for Key Information Extraction. arXiv. 2021. Available online: http://arxiv.org/abs/2103.14470 (accessed on 31 July 2023).
[13] "Optical Character Recognition Systems." American Foundation for the Blind. Accessed November 2, 2023. https://www.afb.org/node/16207/optical-character-recognition-systems.
[14] Li Zhang, Chee Peng Lim, Jungong Han, "Complex Deep Learning and Evolutionary Computing Models in Computer Vision", Complexity, vol. 2019, Article ID 1671340, 2 pages, 2019. https://doi.org/10.1155/2019/1671340
[15] Namysl, Marcin, and Iuliu Konya. "Efficient, lexicon-free OCR using deep learning." 2019 international conference on document analysis and recognition (ICDAR). IEEE, 2019.https://arxiv.org/abs/1906.01969 (accessed on 31 July 2023).
[16] Sydorenko, Iryna. "OCR with Deep Learning: The Curious Machine Learning Case." Label Your Data, May 25, 2023. https://labelyourdata.com/articles/ocr-with-deep-learning
[17] Ignat, Oana, et al. “OCR Improves Machine Translation for Low-Resource Languages.” arXiv preprint arXiv:2202.13274 (2022).
[18] Deep Learning Based OCR Text Recognition Using Tesseract. LearnOpenCV. Available online: https://www.learnopencv.com/deep-learning-based-cor-text-recognition-using-tesseract (accessed on 31 July 2023).
[19] Jawahar, Ganesh, Benoît Sagot, and Djamé Seddah. “What does BERT learn about the structure of language?.” ACL 2019-57th Annual Meeting of the Association for Computational Linguistics. 2019.
[20] Guarasci, R., Silvestri, S., De Pietro, G., Fujita, H., & Esposito, M. (2022). BERT syntactic transfer: A computational ex-periment on Italian, French and English languages. Computer Speech & Language, 71, 101261.
[21] Zhou, Xinyu & Yao, Cong & Wen, He & Wang, Yuzhi & Zhou, Shuchang & He, Weiran & Liang, Jiajun. (2017). EAST: An Efficient and Accurate Scene Text Detector. https://www.researchgate.net/publication/316015737_EAST_An_Efficient_and_Accurate_Scene_Text_Detector
[22] R. Mittal and A. Garg, "Text extraction using OCR: A Systematic Review," 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, 2020, pp. 357-362, doi: 10.1109/ICIRCA48905.2020.9183326. https://ieeexplore.ieee.org/document/9183326
[23] Liu, X., Meehan, J., Tong, W. et al. DLI-IT: a deep learning approach to drug label identification through image and text embedding. BMC Med Inform Decis Mak 20, 68 (2020). https://doi.org/10.1186/s12911-020-1078-3
[24] Drug Label Extraction using Deep Learning. Section.io. Available online: https://www.section.io/engineering-education/drug-labels-extraction-using-deep-learning/ (accessed on 31 July 2023).
[25] A. Yadav, S. Singh, M. Siddique, N. Mehta and A. Kotangale, "OCR using CRNN: A Deep Learning Approach for Text Recognition," 2023 4th International Conference for Emerging Technology (INCET), Belgaum, India, 2023, pp. 1-6, doi: 10.1109/INCET57972.2023.10170436.
[26] V. Wati, K. Kusrini and H. A. Fatta, "Real Time Face Expression Classification Using Convolutional Neural Network Algorithm," 2019 International Conference on Information and Communications Technology (ICOIACT), Yogyakarta, Indonesia, 2019, pp. 497-501, doi: 10.1109/ICOIACT46704.2019.8938521.

[27] V. E. Bugayong, J. Flores Villaverde and N. B. Linsangan, "Google Tesseract: Optical Character Recognition (OCR) on HDD / SSD Labels Using Machine Vision," 2022 14th International Conference on Computer and Automation Engineering (ICCAE), Brisbane, Australia, 2022, pp. 56-60, doi: 10.1109/ICCAE55086.2022.9762440.
[28] C. Yi, Y. Tian and A. Arditi, "Portable Camera-Based Assistive Text and Product Label Reading From Hand-Held Objects for Blind Persons," in IEEE/ASME Transactions on Mechatronics, vol. 19, no. 3, pp. 808-817, June 2014, doi: 10.1109/TMECH.2013.2261083.
[29] Tensorway. "What Is Optical Character Recognition (OCR): Its Working, Limitations, and Alternatives." Tensorway. https://www.tensorway.com/post/optical-character-recognition
[30] Potrimba, Petru. "What is Optical Character Recognition (OCR)?" Roboflow Blog, November 21, 2023.https://blog.roboflow.com/what-is-optical-character-recognition-ocr/
[31] Tripathi, Pankaj. "Major Limitations Of OCR Technology and How IDP Systems Overcome Them." Docsumo Blog, October 9, 2023. https://www.docsumo.com/blog/ocr-limitations
[32] Bais, Gourav. "Building Deep Learning-Based OCR Model: Lessons Learned." Neptune.ai. September 12, 2023. https://neptune.ai/blog/building-deep-learning-based-ocr-model.
[33] Analytics Vidhya. "Introduction to Long Short-Term Memory (LSTM)." Analytics Vidhya Blog, March 2021. https://www.analyticsvidhya.com/blog/2021/03/introduction-to-long-short-term-memory-lstm/.
[34] Infrrd. "Transformer-Based OCR Model: How OCR Decoder Works." Infrrd AI Blog. Accessed November 7, 2023. https://www.infrrd.ai/blog/transformer-based-ocr.
[35] B. Shi, X. Bai and C. Yao, "An End-to-End Trainable Neural Network for Image-Based Sequence Recognition and Its Application to Scene Text Recognition," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 11, pp. 2298-2304, 1 Nov. 2017, doi: 10.1109/TPAMI.2016.2646371.
[36] a9t9 software GmbH. "Free OCR API." OCR.space. Last modified 2023. Version 3.52. Accessed June 19, 2023. https://ocr.space/OCRAPI#local.
[37] MMOCR Contributors. (2020). OpenMMLab Text Detection, Recognition and Understanding Toolbox (Version 0.3.0) [Computer software]. https://github.com/open-mmlab/mmocr
[38] Baek, Youngmin, Bado Lee, Dongyoon Han, Sangdoo Yun and Hwalsuk Lee. “Character Region Awareness for Text Detection.” 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019): 9357-9366
[39] Acoba, Jonathan James. "Computer Vision Tutorial: Blob Detection." CodeMentor. Accessed December 19, 2023. https://www.codementor.io/@jonathanjamesacoba/computer-vision-tutorial-blob-detection-130qpyy6hr.
[40] "Text Detection and Recognition." MathWorks. Accessed December 19, 2023. https://www.mathworks.com/help/vision/text-detection-and-recognition.html.
[41] Google Cloud. Vision AI | Cloud Vision API. Available online: https://cloud.google.com/vision (accessed on 31 July 2023).

AUTHORS

Hansi Seitaj received his Bachelor of Science in Computer Science in 2023 from Pennsylvania State University, Abington. His research interests are in developing real-world applications of computer science and machine learning algorithms. He is an interdisciplinary researcher and technologist who combines social responsibility with technical expertise.

Dr. Vinayak Elangovan is an Assistant Professor of Computer Science at Penn State Abington. His research interest includes computer vision, machine vision, multi-sensor data fusion and activity sequence analysis with keen interest in software applications development and database management.