CV Content Recognition and Organization Framework based onYOLOv8 and Tesseract-OCR Deep Learning Models

Amany M. Sarhan
amany_sarhan@f-eng.tanta.edu.eg
Tanta University
Hesham A. Ali
Delta University for Science and Technology
Mariam Wagdi
Delta University for Science and Technology
Bassant Ali
Delta University for Science and Technology
Aliaa Adel
Delta University for Science and Technology
Rahf Osama
Delta University for Science and Technology

Research Article

Keywords: Optical Character Recognition (OCR), You Only Look Once (Yolov8), Deep learning, Tesseract-OCR, Object Detection, CV Recognition

Posted Date: September 23rd, 2024

DOI: https://doi.org/10.21203/rs.3.rs-4947322/v1

Additional Declarations: No competing interests reported.

CV Content Recognition and Organization Framework based onYOLOv8 and Tesseract-OCR Deep Learning Models

Amany M. Sarhan[1], Hesham A Ali[2,3], Mariam Wagdi[2], Bassant Ali[2], Aliaa Adel[2], Rahf Osama[2]

[1]Department of Computer and Control Engineering, Tanta University, Tanta, Egypt
[2]Department of A I, Faculty of A I, Delta University for science and technology, Gamassa, Egypt.
[3]Department of Computer and Systems Engineering, Faculty of Engineering, Mansoura University, Mansours, Egypt
**Email: amany_sarhan@f-eng.tanta.edu.eg*

ABSTRACT

The growing number and variety of resumes in the job market makes it necessary to have better systems for sorting them efficiently. The task of finding suitable candidates for an open job position would be a repetitive and time-consuming task, especially from a large pool of candidates. The same task could be an actual fair screening and shortlisting though. Indeed, it is not acceptable to lose the chance to employ top skilled candidates because of the tardiness of the whole census or the poor selection caused by human fault. In order to help automating this cycle, this study introduces a new CV recognition system that combines advanced technologies: You Only Look Once (YOLO) for detecting important sections in the CV files, Tesseract-OCR for extracting text in each section, and an efficient post processing steps for correcting possible faults in recognized text. We also propose to automatically organize the data within the CV into a database to facilitate performing any data analytics or searching processes. To evaluate the system, we used a dataset of 1,300 resumes in JPEG, PNG, and JPG formats from various sources, showcasing different formats, languages, and quality levels. Preprocessing steps were taken to ensure the data is high quality and consistent. This approach saves time and improves the efficiency and accuracy of sorting resumes, helping HR teams focus on more strategic tasks and making the hiring process easier and less stressful. Results support the validity of the proposed system through the experiments with this diverse dataset. The system's effectiveness was confirmed through experiments with this diverse dataset, achieving a mean Average Precision (mAP) of 92.1%, a precision rate of 92.2%, and a recall rate of 86.0%.

Keywords

Optical Character Recognition (OCR), You Only Look Once (Yolov8), Deep learning, Tesseract-OCR, Object Detection, CV Recognition.

1. Introduction

Currently, the recruiting process in companies and organizations entails going through massive volumes of largely formal and informal written CVs. Though CVs provide valuable information for the applicants, they vary in their structure and yet HR personnel must go through them for analysis. Recruiters often manually examine and get important information from CV data, as it is difficult to process automatically. Furthermore, employers may gather CVs from unofficial networks such as LinkedIn, or as an alternative, there may be a site that collects CVs for people searching for work, where a recruiter can collect potential applications and contact them if they meet the job description. Recruiters take information from CVs by looking for certain elements and inserting them into a database or system. The major problem of such step is that it is a time-consuming process, moreover, also prone to errors caused by humans.

1.1 Background and Significance

In a report in USA on 2023, companies—of all sizes and industries, in virtually every state—are having unprecedented difficulty finding enough workers to fill available positions. The most recent statistics reveals that there are 8.5 million job opportunities in the United States, but only 6.5 million jobless individuals. On the other hands, in European Union, jobs of certain type are seeking workers while others are filled, as shown in Fig. 1. To make the right match between the applicant or job seekers and a certain job, the organizations should rely on a concise database of CVs and a strong tool for matching job description with applicants features, which could of course be automated through the currently advances in machine learning tools.

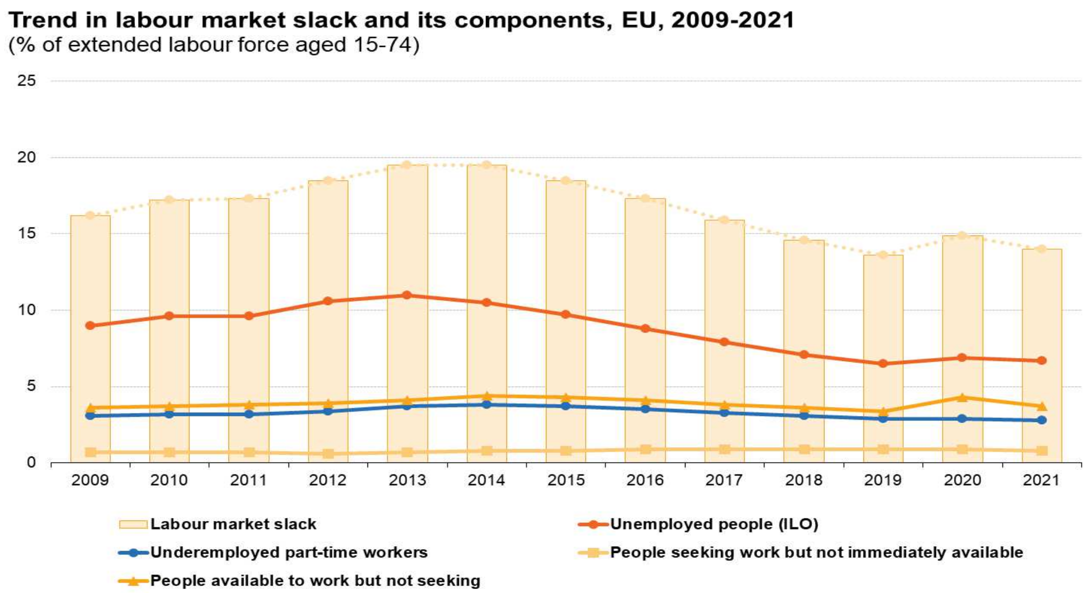

Figure 1: Job market analysis in EU from 2009-2022.

Optical Character Recognition (OCR) [13] is the process of converting images of characters to digital texts. In the age of Deep learning and the development of the field of computer vision, OCR has seen significant performance enhancements. However, it is not fully matured, in which, for example, natural language processing (NLP) and Information extraction (IE) are the main educational fields. Post-OCR parsing is one of the hardest tasks in this area. In this task, the semantic labels given prior to the OCR output would be predicted, which is a highly uncertain process. Optical character recognition (OCR) is the most widely used technology in multiple application areas including IE in online education, operation of toll gates, industrial automation, and robotics. OCR is the ability of a machine to recognize text that comes from an extensive range of images from scanned paper documents to text in the wild. The technology possesses a long track of research and has been successfully applied with high accuracy [5].

Human beings can easily detect and identify objects in their surroundings, without consideration of their circumstances, no matter what position they are in and whether they are upside down, different in color or texture, partly occluded, etc. Therefore, humans make object detection look trivial. The same object detection and recognition with a computer require a lot of processing to extract some information on the shapes and objects in a picture. In computer vision, object detection refers to finding and identifying an object in an image or video. You Only Look Once (YOLO) is a viral and widely used Object Detection algorithm [1]. YOLO is famous for its object detection characteristic. In 2015, Redmon et al. gave the introduction of the first YOLO version [2]. In the past years, scholars have published several YOLO subsequent versions described as YOLO V2, YOLO V3, YOLOV4, and YOLO V5 [3,10]. There are a few revised-limited versions, such as YOLO-LITE [11]. This research aims to explore the integration of YOLO's object detection capabilities with OCR to further enhance text recognition and post-OCR parsing tasks.

1.2 Objective and Scope

The main objective of the paper is to employ artificial intelligence to simplify the process of CV information extraction and data structuring through the use of machine learning methods. As a result, reports on the best candidates can be obtained by feeding the database with a specific job description. For HR departments and recruiting firms that use the vast amount of information contained in resume databases to enhance decision-making in the initial stages of hiring, this can be a useful tool. It is especially interesting in hiring periods when the amount of information is so large that it hinders HR departments' ability to make better decisions about resume selection. The huge memory system of machine learning systems allows for a larger and more complex database than traditional or craft databases. This scope is not without its limitations and implications.

The research work presented used different ways for automating information extraction and resume assessment, each one at different levels of success. In [15], techniques to enhance keyword recognition attained good results. The model could also handle keywords within the job description as well as the potentiality of mistakes with quotations marks in the source code. But she admitted that her translation is imperfect. Avisha Anand's system [6], on the other hand, provided an overview of the most important issues of the course with a high level of performance in the direction of the required workload both in quality and quantity. The system is not always right because it is based on the style of the source. Yang-Yang's model [17] is mostly good for gathering information from texts and images but it has a very slow pace and the automation in the first place is not good among its peers. In addition, the designs in [18, 19] have an outstanding accuracy rate, but they lack ideas for practical database integration. The OCR Tesseract engine [30] performs well in the case of text extraction but still it drops out with scans of low quality and manual processing. Overall, these studies have made some progress, however, they still need to reach the goals of full automation and database integration efficiently.

In this article, we will explore an automatic CV recognition and organization system that utilizes YOLOv8 for detecting various classes within an image. We use a CV dataset that has not been explored yet in the literature [32] that contains images of collected CV in different file format written in different languages, which makes it a challenging problem in such filed. The proposed CV content recognition and organization system can be considered as integrated system that processes the CV in an innovative way to generate a ready-to use output format. The system will identify and create bounding boxes for 11 different classes using YOLO8, which quickly identifies key sections in resumes, such as personal information, education, work history, and skills, even with different layouts. Additionally, the proposed system will employ PyTesseract to recognize and extract words within these bounding boxes as it can accurately extract the text, ensuring no important details are missed. The system also checks for and corrects any errors before saving the cleaned data into a database. This automated process efficiently handles different resume formats, saving time and reducing the chances of mistakes. As a result, HR professionals can focus on more important tasks, making the hiring process smoother, more efficient, and less stressful.

We can highlight the main contributions of this research in points as:

1. Introducing an integration of image detection mechanism based on YOLOv8 and Optical Character recognition approach based on Tesseract-OCR**:** This research successfully combines the YOLOv8 model with Tesseract-OCR, creating a robust CV details recognition system capable of detecting and recognizing text within images.
2. Proposing suitable image enhancement techniques: Implementing advanced image pre-processing techniques, such as contrast adjustment, noise reduction, and sharpening, to significantly improve the clarity of resume images. This step enhances the visibility of text and features, facilitating more effective character detection and reducing the error rate in subsequent processing stages.
3. Developing a Text Recognition approach: Utilize PyTesseract, an open-source Optical Character Recognition (OCR) tool, to extract text from within the YOLOv8-detected bounding boxes.
4. Proposing an Automated Error Correction approach: The system includes an automated error-checking mechanism that identifies and corrects mistakes in several languages after detecting the CV language.
5. Introducing a new approach of saving the CV contents into an organized manner: The content of each section of the CV is stored in the appropriate field of a database enabling the system user to manage the CV data in a reliable and productive manner. The extracted data is then organized into a database, reducing manual work and errors.

The rest of the paper is organized as follows. The next section discusses optical character recognition (OCR) and its different types along with the details of Yolov8 architecture and main characteristics. This section ends with the previous research in CV text recognition. In section 3, we present the proposed CV recognition in details. In section 4, we summarize the experimental results and analysis. Section 5 gives the conclusion and possible future work.

2. Back ground and Related Work

Information extraction form unstructured files (image/pdf) automatically into pre-defined categories or extracting specific pieces of information has long been identified as an important goal laying the foundation of a multitude of business processes. The first widely recognized effort at solving these challenges is the Information Extraction, or IE, architecture developed by text analysis researchers. Based on natural language understanding, the proprietary system demonstrated state-of-the-art extraction accuracy using a series of statistical and lexical classification processes.

Text extraction has continued to be a challenging and active area of research, in part due to the huge amount of new OCR technology that fuels the creation of unstructured documents in the form of word processing files, PDF files, image content, and miscellaneous content within the database. One recurring theme behind the current research is the use of techniques to understand documents and analyze their contents. Text analysis technology can be used to answer company-specific questions related to internal operations, business development, and people selection. The custom system is based on well-established technologies and is capable of analyzing large amounts of incoming unstructured text data, and has a wide range of applications in today's competitive business environment. The next subsections focus on shedding light in a simplified manner on the necessary basics of some of the techniques that were employed in this research, with a review of the latest recent attempts in the scope of autobiography analysis.

2.1 Optical Character Recognition

Another widespread application of computer vision in industry is Optical Character Recognition (OCR) [4]. Contemporary OCR techniques are primarily rooted in deep learning and aim to recognize various characters or numerals in images [7]. These techniques have been increasingly effective, even for complex scripts such as Urdu [6]. OCR technologies can identify characters in different types of documents, such as printed stickers, and can be utilized for Printed Character Recognition (PCR) and Handwritten Character Recognition (HCR) [7]. Recent advancements in HCR have used deep learning for recognizing handwritten words in documents, significantly improving accuracy in this area [8]. PCR is comparatively simpler than HCR due to the limited number of available fonts in PCR as opposed to the diverse handwriting styles found in HCR [9]. However, it is often necessary to identify documents or printed stickers containing arbitrary characters, such as hexadecimal identifiers or passwords with special characters [9]. Some of Optical Character Recognition Tools shown in fig. 2. We will discuss each of them in detail.

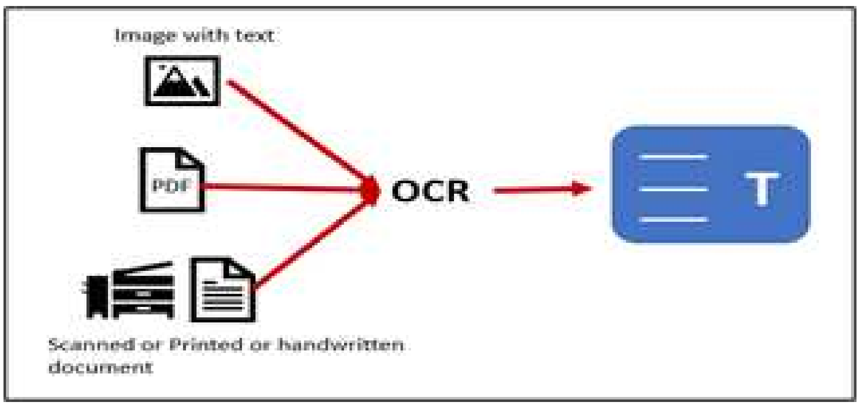

Figure 2: Optical Character Recognition Process

2.2.1 Tesseract Optical Character Recognition

Tesseract [12] is an open-source OCR engine designed to extract printed or written text from images. Originally developed by Hewlett-Packard and later maintained by Google, it is now known as “Google Tesseract OCR.” The engine supports over 100 languages out of the box. Tesseract 4.0 introduces AI integration through LSTM neural networks, which enhances its ability to detect and recognize text of various sizes. Operating without a built-in GUI, Tesseract can be integrated with multiple programming languages and frameworks through wrappers available on GitHub. It can process both large documents and single text lines, utilizing its internal layout analysis or an external text detector as needed. The update to version 4.0 includes a neural network subsystem based on OCRopus' LSTM model, reengineered in C++ and compatible with TensorFlow's Variable Graph Specification Language (VGSL). While LSTM networks excel at sequence learning, they may be slower with large state spaces. Tesseract 3.x used a multi-stage approach involving word finding, line finding, and character classification, whereas Tesseract 4.x modernizes this process by applying LSTM models for more efficient line-by-line text recognition

2.2.2 Easy Optical Character Recognition

EasyOCR [14] is an open-source, ready-to-use OCR tool with support for over 80 languages, developed by Jaded AI and built on the PyTorch library. It features a comprehensive architecture consisting of feature extraction, sequence labeling, and decoding to effectively recognize text from images. For feature extraction, EasyOCR employs ResNet and VGG networks to convert images into usable features; ResNet uses residual connections to address challenges in deeper networks, while VGG relies on simple, stacked convolutional layers. Sequence labeling is performed by Long Short-Term Memory (LSTM) networks, which handle temporal dependencies and long sequences with memory cells and gating mechanisms. Decoding is managed by Connectionist Temporal Classification (CTC), which processes variable-length predictions by computing loss over all possible alignments between input sequences and target text. While EasyOCR provides robust text recognition capabilities with the option to train models on custom data, it does not offer as many customization options as Tesseract. The framework, based on the deep-text-recognition-benchmark, ensures EasyOCR versatility and effectiveness in text recognition tasks.

2.2.3 Paddle Optical Character Recognition

PaddleOCR [28] is an ultra-lightweight Optical Character Recognition (OCR) system designed to be practical and efficient. Developed by Baidu Inc., the system addresses the challenges of diverse text appearances and the need for computational efficiency. PP-OCR features a compact model size—3.5M for recognizing Chinese characters and 2.8M for alphanumeric symbols—achieving this through a set of strategic enhancements. These include using a lightweight backbone and head for text detection, optimizing the text direction classifier, and employing efficient strategies for text recognition. The system also incorporates pre-trained models for both Chinese and English, and has been validated on additional languages like French, Korean, Japanese, and German. All models and code are open-source, available on GitHub, making PP-OCR accessible for a wide range of OCR applications. Table 1 summarizes a comparison between OCR famous tools.

Table 1: OCR Tools Comparison

Feature	Paddle-OCR	Tesseract-OCR	Easy-OCR
Accuracy	High, supports a wide range of languages and scripts	Good for basic OCR tasks; accuracy may vary	Good accuracy, handles diverse text styles
Ease of Use	Can be complex to set up; requires familiarity with Paddle	Relatively easy to integrate; open source	Simple to use; easy integration
Language Support	Extensive language and script support	Multiple languages supported	Supports a variety of languages
Features	Text detection, recognition, and layout analysis	Basic OCR; needs tuning for complex documents	Good for general OCR tasks
Community & Support	Growing community, but less widespread compared to Tesseract	Strong community support and documentation	Growing community; less extensive than Tesseract
Customization	High customization potential, but more complex	Limited customization; basic configurations	Limited customization; user-friendly

2.2.4. Preprocessing Techniques

The initial operation of an Optical Character Recognition (OCR) system involves several key steps to prepare the image for accurate text recognition. First, preprocessing techniques such as resizing the image, cropping unnecessary parts, reducing noise, and adjusting contrast are applied to enhance image quality. These steps make the text clearer and easier for the system to recognize. After extracting the text, Natural Language Processing (NLP) tools, like those in NLTK, are used to correct any errors such as spelling or grammar mistakes. Finally, the cleaned and corrected text is saved into a database, making it easier to analyze and retrieve information later. This process ensures that the system produces accurate and reliable results for future use.

2.3 Previous Work on CV detection

The rapid growth of the labor market also brought a rise in labor competition. Many new graduates pour into the labor market every year, and there is fierce competition when looking for a job. Nowadays, many companies use automatic filtering when job hunting. The number of employers with human intervention is gradually decreasing. These factors have stimulated the growth of online job-hunting platforms. One of the important functions of the job-hunting platform is to hold resumes of job seekers. The user uploads his resume corresponding to his job characteristics, abilities, experience, and other factors on the platform. With the help of the platform, users can learn about the latest employment information, access job information, and obtain some guidance for job-seeking.

According to official statistics, millions of new individuals enter the employment field every month at an increasing rate every year. The most challenging factor that makes shortlisting possible profiles for the required occupations tedious and time-consuming is the absence of a uniform CV structure and style [12]. Effective resume screening necessitates the evaluation of a profile's suitability and application to the position by a subject matter expert. Making a short list is challenging and need to be automated to reduce the possibility of errors. The automation of such process is gaining attraction around the research community due its importance to the job market.

This work in [15] presents a mixed way to automatically retrieve information from Polish resumes during IT recruitment. The multi-module system incorporates dictionary approaches and named entity recognition (NER) tools, which significantly enhance information extraction accuracy. Experiment results revealed that key word recognition in different sections of CV is 60%–160% higher than with separate tools. The hybrid solution achieved an accuracy of 30.92% for detecting educational institutions and 19.68% for company names, outstripping single modules. Such a system can operate as a pre-processor and analyzer for unstructured CV data, making the recruitment process more efficient with minimum human failure

An automated CV screening system that uses machine learning (ML) and natural language processing (NLP) to reduce the recruitment process is the subject of the work in [16]. The system analyzes the resumes to find matches between candidate profiles and job descriptions. In this way, it not only saves time but also reduces the effort of the recruiters significantly. The way to make use of the system is through the provision of resumes, keyword extraction, and scoring of the candidates based on their skills, education, and experience. The strategy applied can uncover the relationship between textual data interpretations and the cosine similarity score, as well as spaCy for the training dataset. The model's precision was verified by using 220 resumes, of which 200 were for training and 20 for testing, with an accuracy of 85%. The system is expected to make the process of candidate pre-selection faster, thus becoming more advanced than traditional e-recruitment platforms.

The research in [17] introduces the Divergence-Oriented Multi-Modal Fusion Network (DOMFN), which is an original approach to resume assessment that merges textual content with visual page style data to enhance the evaluation process. Unlike traditional techniques that are confined to the text, DOMFN uses multi-modal fusion, but it only merges the desired modality based on the divergence between different modalities to escape from the degrading performance. The machine learning model generates a cost score, which the AI considers the most important thing when it comes to the fusion of two (imaging and textual) modalities and thus leads to better predictions. The research findings confirm DOMFN's better performance compared with the comparison models, which is in the evaluation of resumes for the UI Designer and Enterprise Service positions. DOMFN gets an accuracy of 0.785, which is a better output as compared to other models in other metrics, including AUC, precision, recall, and macro-F1.

In [18], TSHD (Topic Segmentation based on Headings Detection) algorithm is presented, which identifies and segments the subjects within resumes. It is a collection of 105 resumes consisting of 4733 lines downloaded from various websites. TSHD's performance has reached a high point with F1-scores of approximately 96%; both values of precision and recall are close to 99%. The error rates in the segmentation are down to 2% for the specific thresholds. The study emphasizes the use of the algorithm in its purpose to develop resume data processing algorithms by providing the same through the resume structure. As a result, the algorithm has shown that it is possible to detect the potential of specific unformatted resumes that have distinctive and diverse content models.

In [19], a generative AI-based resume analysis system is designed to serve as a tool for HR to improve their efficiency. This system involves the use of entity and relation extraction, skill and experience extraction, and cross-encoder methods to process resumes and match them with job descriptions. It takes scores at an aggregate level and shows you the most relevant resumes, and thus, there are savings in terms of both time and cost due to a decrease in recruiting campaigns. The system is based on a private dataset of around 6,000 resumes and shows a considerable increase in matching accuracy as well as efficiency, with 60% more of the staff of HR not participating in the analysis of time. The statistics of the classification show us that the system reaches a percentage of close to 99% in accuracy and, thus, an F1 score of about 96%, which is good enough to show the effectiveness of the system.

In [30], Tesseract OCR engine is used to retrieve and computerize text from business cards, tackling the problem of effective information management. Due to its cost-effectiveness and extensive language support, Tesseract conducts different steps on images, such as preprocessing them or converting color ones into grayscale. The research examines how accurate this is by achieving a word-level accuracy of 98.44% for black and white images and 95.70% for colored images, thus identifying the limitations of dealing with poorly scanned and complex background documentation using it. Experimental results confirm an overall word-level accuracy of 96.87%. Summary of the previous work is given in Table 3. This table shows that effort is required to fully automate the CV recognition process, including additional improvements to the extracted data, either in reviewing the validity of the generated text or converting the CV text into a proper product ready for the next phase of the CV screening process. As a summary, Table 2 provides a summary of the pervious related methods in CV Recognition

Table 2: Pervious related methods in CV Recognition

Paper	Dataset	Algorithm	Accuracy
Agnieszka Wosiak (2021) [15]	Polish resumes in IT recruitment	Hybrid approach with dictionary methods and NER	30.92% (educational institutions), 19.68% (company names)
Avisha Anand (2022) [16]	220 resumes (200 training, 20 testing)	ML and NLP (spaCy)	85%

Yang Yang (2022) [17]	Various resume datasets	Divergence-Orientated Multi-Modal Fusion Network (DOMFN)	0.785 (Accuracy), superior in AUC, Precision, Recall, Macro-F1
Majd E. Tannous (2023) [18]	105 resumes with 4733 lines	TSHD (Topic Segmentation based on Headings Detection)	~96% F1 score, ~99% accuracy
Thanh Tung Tran (2023) [19]	Private dataset of ~6,000 resumes	Generative AI-based system with entity and relation extraction	~99% accuracy, ~96% F1 score
Abrar Al Sayem (2023) [30]	Business cards	Tesseract OCR	96.87% overall, 98.44% (black-and-white), 95.70% (colorful)

2.4 Problem formulation

The existing research on CV recognition systems faces several challenges. Some systems, like the one using a hybrid approach with dictionary methods and NER, have low accuracy in identifying important information like educational institutions and company names. Others, such as the ML and NLP-based systems, do not handle different CV formats well and only match candidates to set criteria without offering any recommendations. The DOMFN approach, although better at integrating text and visual data, still has room for improvement in its accuracy and reliability. Systems using topic segmentation or generative AI also face limitations in processing unstructured resumes and may miss important details. Overall, these systems often lack flexibility and fail to provide deeper insights beyond basic filtering and matching.

2.5 Plane of solution

In this work, we propose a framework for analyzing resume information by combining onYOLOv8 and Tesseract-OCR Deep Learning. This study structured and parsed resume information to mine deeper job-related information for increasing the ability of accurately matching job seekers with employers. First, the input resume is preprocessed to reduce and eliminate noise which may be affected the important information (personal information, contact information, job objectives, education background, practical experience, personal quality, and training experience). Then, the YOLOv8 model is used to obtain the associated entity important information. Then, we evaluated the validity of the method by applying it to samples of working relations in evaluation software.

3. Proposed CV recognition and Organization System

The fields of computer vision and natural language processing have seen significant advancements in recent years, leading to the development of powerful tools and algorithms for image and text analysis. The aim of this work is to design and develop an automated system for CV text recognition and organization. The architecture of the proposed system is given in Fig. 3. The steps involves preprocessing of the input CV image, detection of the CV sections, extraction of the text inside each section, and post processing of the generated text.

To achieve the goal of CV text recognition, two such tools are employed; YOLO (You Only Look Once), a state-of-the-art object detection system, and PyTesseract, a Python wrapper for the Tesseract OCR engine. By combining these two technologies, we can create a robust system capable of not only detecting objects within images but also extracting and processing textual information from these objects.

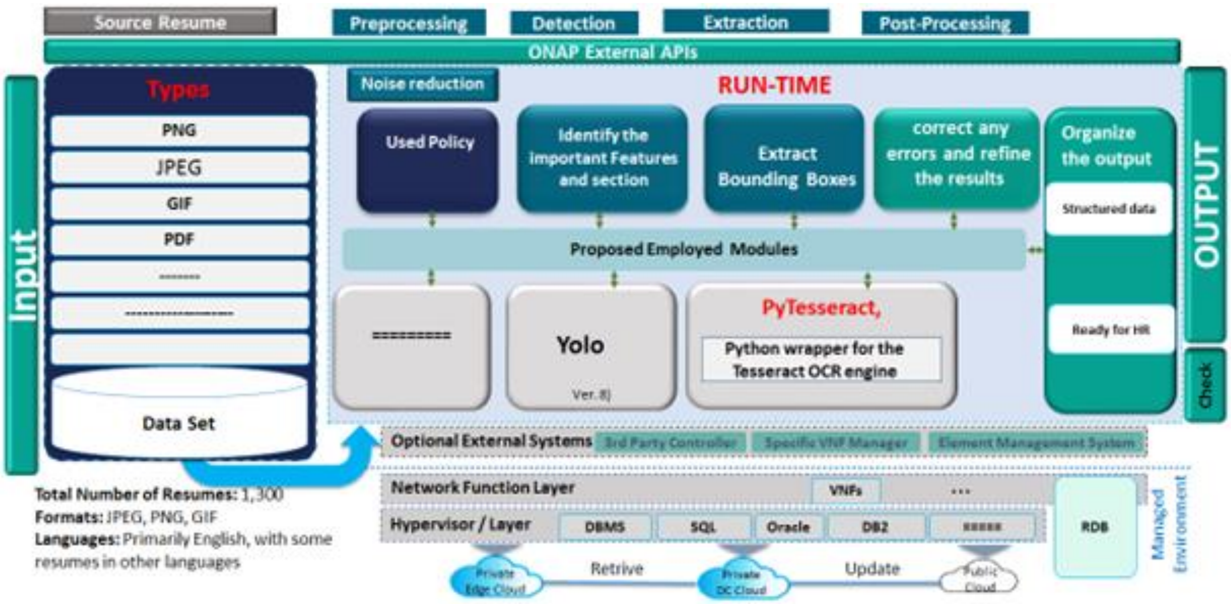

Figure 3: Proposed architecture of CV recognition and organization system

The integration of these two technologies offers several advantages:

- **Efficiency:** YOLO's ability to detect multiple objects in a single pass makes the system highly efficient, suitable for real-time applications.
- **Accuracy:** Combining object detection with OCR allows for precise extraction of information from specific regions of an image.
- **Versatility:** The system can be applied to a wide range of documents, from resumes to certificates and beyond, making it a versatile tool for various industries.

This section delves into the details of the integration of YOLO8 and PyTesseract to address a specific use case: the detection and extraction of information from CV images. In such scenario, we need to identify and extract details related to education, certifications, skills, and languages found in the CV. By using YOLO8, we can detect these sections within the document. Then, with PyTesseract, we can extract the textual content of these sections, which can subsequently be corrected and processed for further analysis. The overall framework for the proposed CV recognition and organization system is given in Fig 4.

The details of the stages in the proposed framework are given below:

3.1 Data Collection

3.1.1 Image Preprocessing and segmentation

Several preprocessing algorithms are employed for this step to prepare the input image for the subsequent steps, such as grayscale conversion, Gaussian blur, thresholding, and edge detection using Canny or Sobel algorithms. These techniques help enhancing the quality of the image, making it easier to process in later stage.

3.1.3 Image Resizing

This algorithm is used to adjust the size of the CV image to a standard dimension to ensure consistent processing, Common algorithms used for resizing include bilinear interpolation, nearest neighbor, and bicubic interpolation. The resizing process can be represented as:

$$I_{Resized} = Resize(I, (Width, Hight))$$

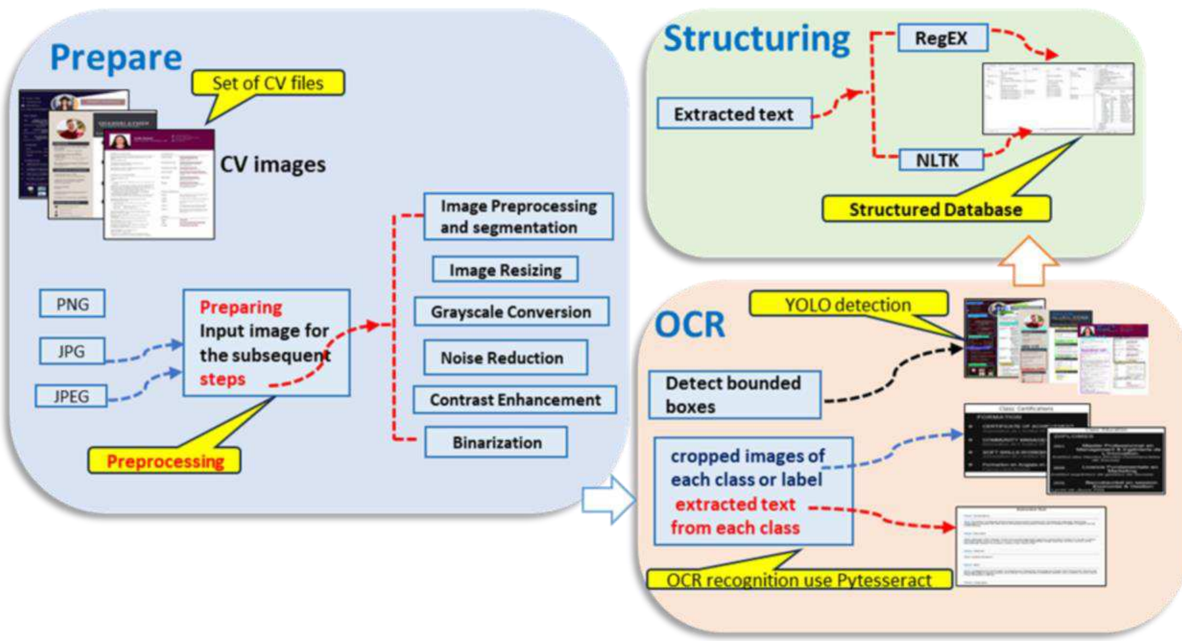

Figure 4: Proposed CV recognition and organization system framework

where $\boldsymbol{I}$ is Original CV image, $\boldsymbol{I_{Resized}}$ is Resized CV image, $Resize$ is function used to change the image size and $(Width, Hight)$ is the target dimensions for resizing typically set to 416x416 pixels.

3.1.4 Grayscale Conversion

The image is converted to grayscale to simplify the detection and recognition processes.

$$I_{Gray} = Convert\_To\_GrayScale(I_{Resized})$$

where $\boldsymbol{I_{Gray}}$: Grayscale CV image and $Convert\_To\_GrayScale$ is Function to convert an image to grayscale.

3.1.5 Noise Reduction

Filters are used to remove any unwanted noise or artifacts from the image.

$$I_{Denoised} = Filter(I_{Gray})$$

where $\boldsymbol{I_{Denoised}}$ is Denoised CV image and $Filter$ is a function to apply noise reduction such as Median filtering or Gaussian filtering.

3.1.6 Contrast Enhancement:

Improve the contrast of the image to make the text and elements more distinguishable.

$$I_{Enahnced} = Enhance\_Contrast(I_{Denoised})$$

where $\boldsymbol{I_{Enahnced}}$: Contrast-enhanced CV image and $Enhance\_Contrast$ is a function to improve image contrast.

3.1.6 Binarization

The grayscale image is converted to a binary image (black and white) to highlight text and important features.

$$I_{Binary} = Binarize(I_{Enhanced})$$

where $\boldsymbol{I_{Binary}}$ is Binary CV image and $Binarize$ is Function to convert an image to binary format.

3.2 Object Detection using YOLOv8

In this stage, we utilize YOLOv8 deep learning model, to detect the key elements within the CV image like skills, experience, and education. The detection function in YOLOv8 applies a function *D(I)* to the image *I* which produces a set of detected elements, each with a bounding box and a class label:

$$D(I) = \{(b_i, c_i)\}_{i=1}^{N}$$

where *D(I)* is the detection function applied to the image *I*, b_i is the bounding box for the detected element *i*, c_i is the class label for the detected element *i*, and *N* is the number of detected elements. Then the localization of each element in the image is performed by identifying and marking the locations of these elements within the image.

$$L = \{\ b_i\}_{i=1}^{N}$$

where $\boldsymbol{L}$ is Set of bounding boxes.
Bounding boxes are drawn around the detected elements for further processing.

$$B = \{b_i\ for\ i \epsilon 1 \dots N\}$$

where $\boldsymbol{B}$ is Set of all bounding boxes.
Afterwards each detected class is assigned a label or class (e.g., skills, experience).

$$C = \{c_i\ for\ i \epsilon 1 \dots N\}$$

where $\boldsymbol{C}$ is the set of all class labels.

3.2.1 Proposed YOLO Deep Learning Model

YOLO, which stands for You Only Look Once series algorithms [20,21,22] and single shot multi-box detector (SSD) algorithms [23,24], is a cutting-edge algorithm in the field of computer vision that specializes in object detection (fig. 5). Unlike traditional object detection methods that use a two-stage approach, YOLO employs a single-stage architecture that processes the entire image at once, enabling real-time detection with high accuracy. This innovative method allows YOLO to simultaneously predict bounding boxes and class probabilities for multiple objects within an image, significantly speeding up the detection process.

YOLOv8 is the most advanced version of the YOLO algorithm. It surpasses previous versions by incorporating several sophisticated enhancements, such as spatial attention mechanisms, feature fusion techniques, and context aggregation modules. These advancements enable YOLOv8 to better understand and integrate spatial relationships and contextual information, leading to more precise and robust object detection. alternative methods exist to significantly reduce the model's parameters and computational complexity, for example, by substituting the backbone of YOLOv8 with the lighter MobileNet v3 [25]

The spatial attention mechanism in YOLOv8 helps the network focus on relevant parts of the image, improving its ability to distinguish between closely packed objects. Feature fusion allows the algorithm to combine information from different layers of the neural network, enhancing its ability to detect objects at various scales. Context aggregation modules provide the network with a broader understanding of the scene by integrating contextual information, further boosting detection accuracy.

Figure 5: TimeLine of YOLO Advancement

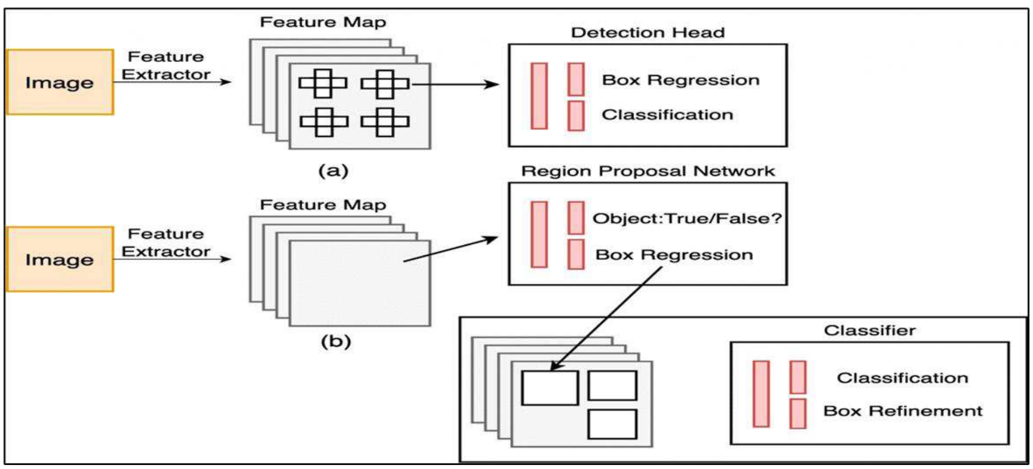

Figure 6: Object Detection and Classification Process in YOLO

These improvements make YOLOv8 not only faster but also more reliable in diverse and challenging environments, cementing its position as one of the leading object detection algorithms in the field of computer vision. Its capability to perform real-time detection with high precision makes it particularly valuable for applications in autonomous driving, surveillance, robotics, and various other domains where rapid and accurate object detection is crucial. The analysis of the YOLOv8 repository [26] and its documentation [27] highlights several key advancements that contribute to its superior performance. Object detection and classification process in YOLO is summarized in Fig. 6.

3.2.2 Key Features of YOLOv8

Multiple features are to be focused on in YOLOv8. Here are some key features of YOLOv8:

1. Improved Accuracy: YOLOv8 improves object detection accuracy compared to its predecessors by incorporating new techniques and optimizations.

2. Enhanced Speed: YOLOv8 achieves faster inference speeds than other object detection models while maintaining high accuracy.
3. Multiple Backbones: YOLOv8 supports various backbones, such as EfficientNet, ResNet, and CSPDarknet, giving users the flexibility to choose the best model for their specific use case.
4. Adaptive Training: YOLOv8 uses adaptive training to optimize the learning rate and balance the loss function during training, leading to better model performance.
5. Advanced-Data Augmentation: YOLOv8 employs advanced data augmentation techniques such as MixUp and CutMix to improve the robustness and generalization of the model.
6. Customizable Architecture: YOLOv8's architecture is highly customizable, allowing users to easily modify the model's structure and parameters to suit their needs.
7. Pre-Trained Models: YOLOv8 provides pre-trained models for easy use and transfer learning on various datasets.

3.2.3 YOLO Architecture

YOLO architecture is composed of 3 essential blocks which are: Backbone, Neck and Head as shown in Fig. 7. YOLOv8 features a new backbone network which is a modified version of the CSPDarknet53 architecture [10] which consists of 53 convolutional layers and employs a technique called cross-stage partial connections to enhance gradient flow and reduce computational complexity. The backbone, also known as the feature extractor, is responsible for extracting meaningful features from the input. It captures simple patterns in the initial layers, such as edges and textures. It has multiple scales of representation, capturing features from different levels of abstraction. This provides a rich, hierarchical representation of the input.

The second block is the neck layer which in YOLOv8 incorporates the FPN [11] + PAN [12] and acts as a bridge between the backbone and the head, performing feature fusion operations and integrating contextual information. Basically, the Neck assembles feature pyramids by aggregating feature maps obtained by the Backbone, in other words, the neck collects feature maps from different stages of the backbone. It performs concatenation or fusion of features of different scales to ensure that the network can detect objects of different sizes. It integrates contextual information to improve detection accuracy by considering the broader context of the scene. This reduces the spatial resolution and dimensionality of resources to facilitate computation, a fact that increases speed but can also reduce the quality of the model.

The final block is the head, which is responsible for generating the network’s outputs, such as bounding boxes and confidence scores for object detection. It generates bounding boxes associated with possible objects in the image. Then it assigns confidence scores to each bounding box to indicate how likely an object is present. Thus, it sorts the objects in the bounding boxes according to their categories. The overall process of the YOLO architecture and responsibilities of the three blocks are summarized in Fig. 7.

Figure 8 shows the segmented image after performing the steps of Yolov8, while Fig. 9 shows samples of the labeling of segments of the segmented image.

3.3 Text Recognition with Pytesseract

This stage employs one of the most famous and robust optical character recognition (OCR) techniques, Pytesseract, to extract the text within each part of the segmented image generated by YOLOv8. Algorithm 1 summarizes the steps involved in this phase. It includes text extraction from detected segments, machine-readable format conversion, text storage and accuracy check.

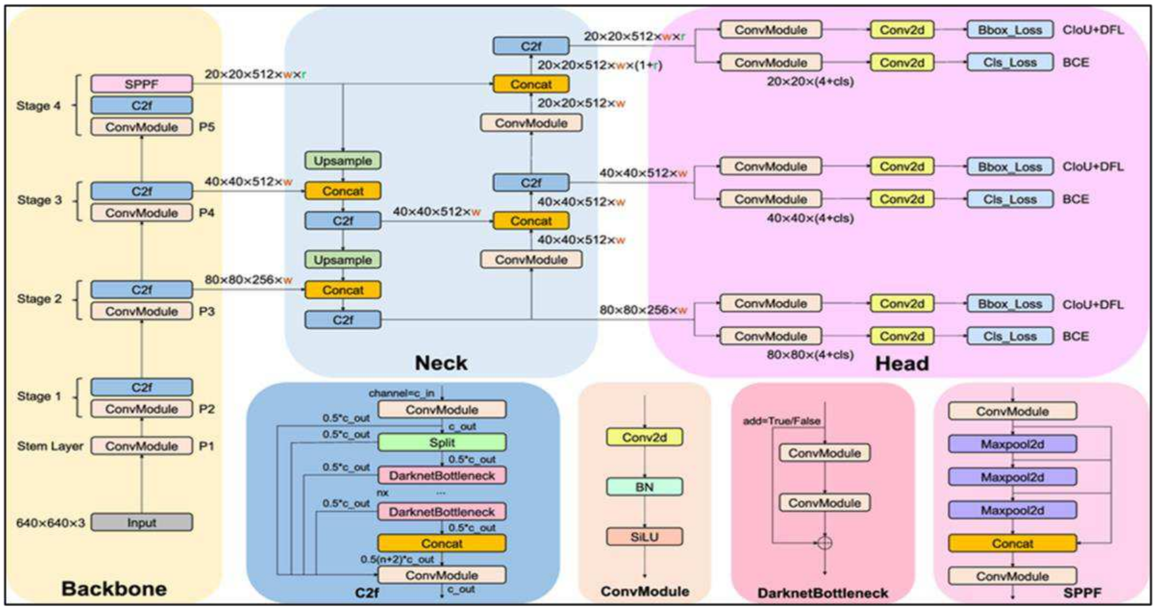

Figure 7: YOLO Model Architecture

(a) (b)

Figure 8: (a) original image and (b) segmented image using Yolov8

Figure 9: Samples of segmented and labeled elements

Algorithm 1: Text Recognition and Organization

Inputs:

Segmented image parts

Output:

Text for each part

Steps:

// Segmented Employee CV

1. *Read the text from detected regions.*

$$\boldsymbol{T}(\boldsymbol{b_i}) = \boldsymbol{t_i}$$

Where $\boldsymbol{T}(\boldsymbol{b_i})$*: OCR function applied to bounding box* $\boldsymbol{b_i}$ *and* $\boldsymbol{t_i}$*: Extracted text from bounding box* $\boldsymbol{b_i}$

//Text Extraction

2. *Extract the text within the bounding boxes identified by YOLOv8 using Pytesseract.*

$$\boldsymbol{T}(\boldsymbol{B}) = \{\boldsymbol{t_i} \boldsymbol{for}\ \boldsymbol{i} \in \mathbf{1..}\boldsymbol{N}\}$$

Where $\boldsymbol{T}(\boldsymbol{B})$ *is a set of extracted texts from all bounding boxes.*

// Image Preprocessing

3. *The image is processed, including adaptive thresholding.*

4. ***Character Recognition with Tesseract:***
   i. *Binary Image Creation: The image is converted into a binary format.*
   ii. *Connected Component Analysis: Character outlines are identified.*
   iii. *Character Segmentation: Characters are segmented and grouped into words.*
   iv. *Word Recognition: Words are recognized using dictionaries and clustering.*

.

5. ***Information Extraction: Key information such as*** *skills, experience, education, certifications, contact, languages*

Figure 10 shows a sample of text extraction generated with PyTesseract. Each class is identified and the text within it is converted from image to text.

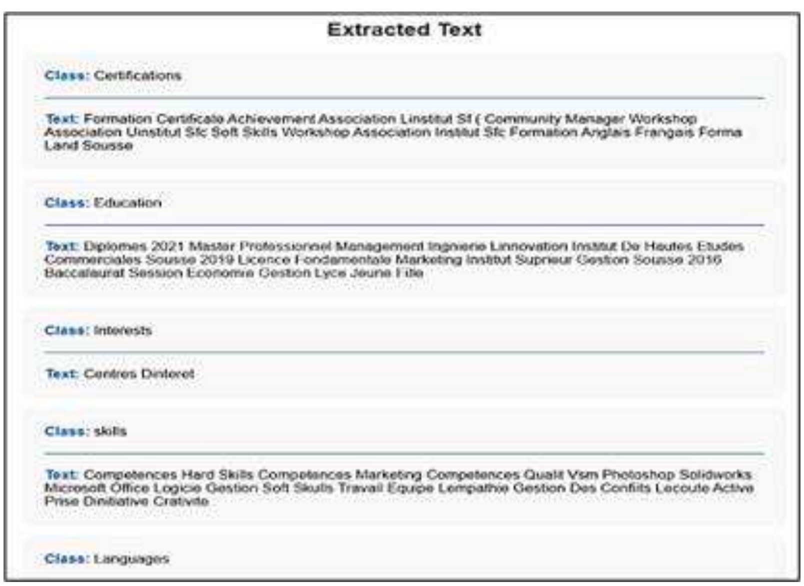

Figure 10: Sample of Text Extraction using PyTesseract

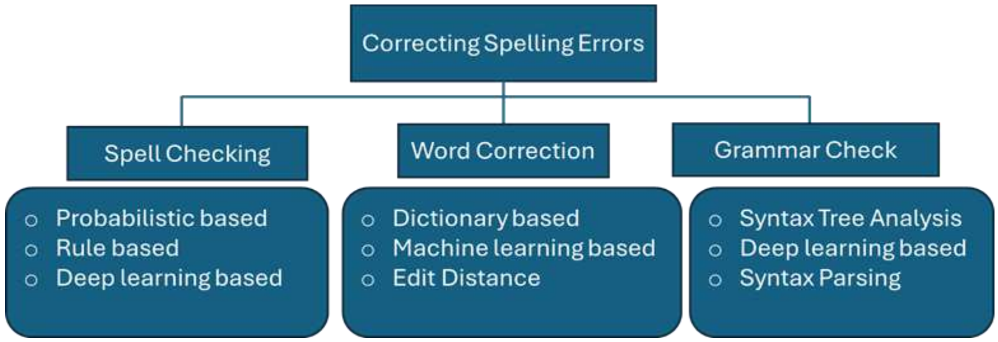

Figure 11: Text Correction and Post-Processing

3.4 Generated Text Error Checking and Correction

As the accuracy of the OCR process could be inaccurate, we apply different post processing steps to enhance the quality of the recognized text (as shown in fig. 11).

1. **Spell checking**: Use a spell checker to find any spelling mistakes in the extracted
2. **Word Correction**: Automatically correct the identified spelling errors
3. **Grammar Check**: Ensure the text is grammatically correct.

3.5 Database Storage

Finally, the generated text in each section, after being reviewed and improved, is saved in a corresponding column in the database for future use by the system owners. In the database, we've created a table called "extracted_text" with 11 columns to contain the information retrieved from CV images. Each column corresponds to a specific class of information detected from the images, such as skills, experience, education, and profile descriptions. This structured approach allows for organized storage and retrieval of the extracted data, facilitating seamless integration with other applications or services. The structure of the database table is defined as follows:

DB_CV_Table = {*skills, experience, education, certifications, contact, languages, projects, other_details*}.

Figure 12 shows a sample of the database after storing one of the CVs. By integrating object detection, image preprocessing, text recognition, spell checking, and database storage, our project provides a comprehensive solution for extracting and processing information from CV images. Automation of these processes streamlines workflows and enhances the accuracy and efficiency of data extraction tasks, ultimately benefiting HR teams and other users in managing and analyzing CV data more effectively.

Model Training and Validation

Figure 13 illustrates the detailed steps involved in the training and testing pipelines for our CV Recognition System, which integrates YOLOv8 for object detection and Tesseract-OCR for text recognition

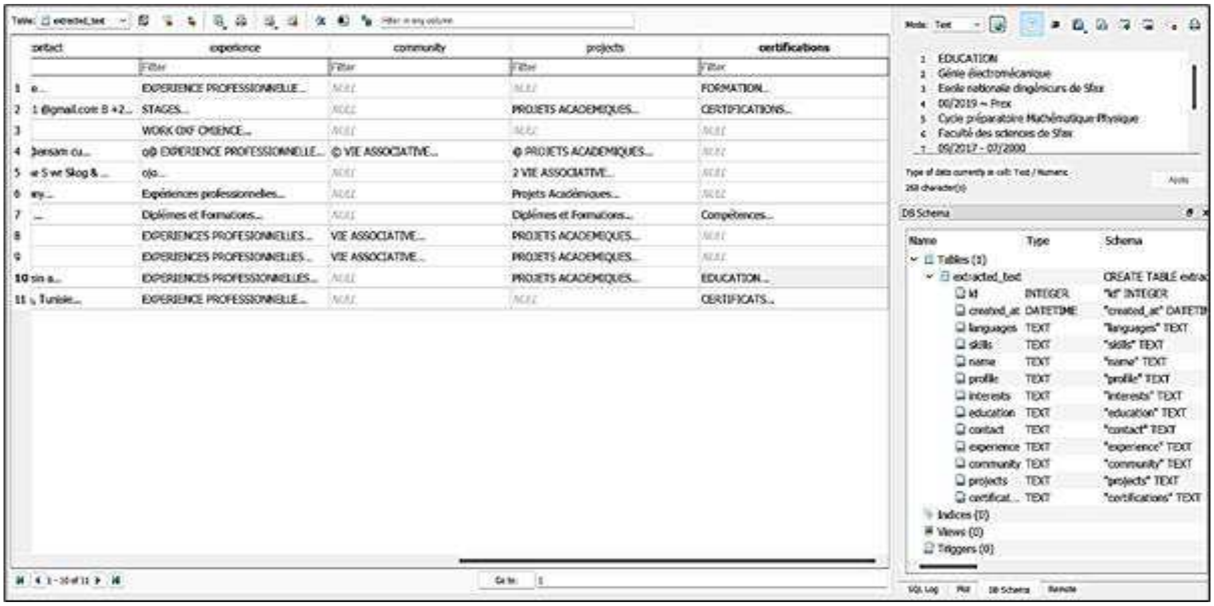

Figure 12: Extracted text stored in the database table

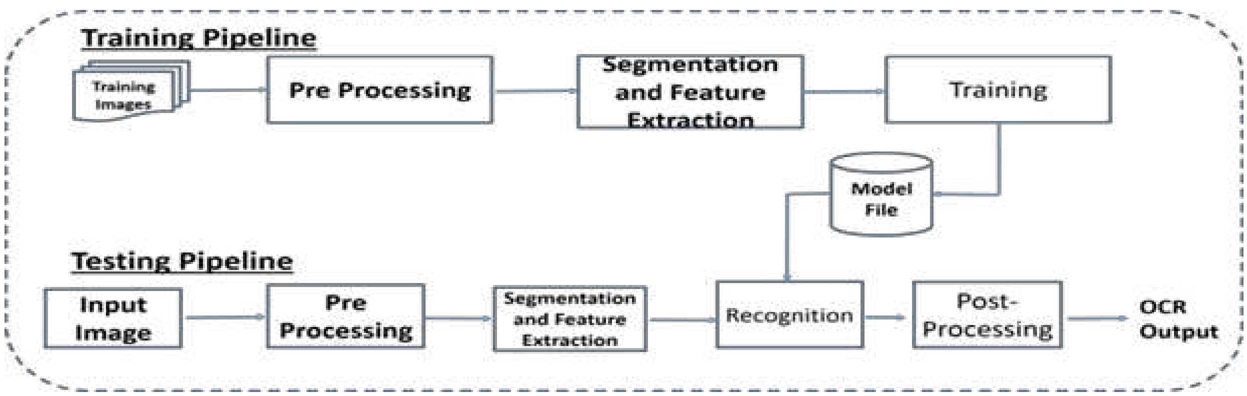

Figure 13: Training and Testing Pipelines for CV Recognition System

4. Experimental results and analysis

In this section, we present the results of our CV recognition system, which integrates YOLOv8 for object detection and Tesseract-OCR for text extraction. This system is designed to streamline the process of reviewing resumes by automatically identifying and extracting key information such as personal details, education history, work experience, and skills. By leveraging these advanced technologies, we aim to reduce the time and effort required for manual resume screening and increase the accuracy and reliability of the information extracted. The overall implementation code is given below.

```
1.  #Load the image and initialize the YOLO model
        image = cv2.imread(image_path)   model = YOLO("best (3).pt")
2.  #Perform object detection using YOLO
        results = model(image)
3.  #Initialize a DataFrame to store extracted text
            data = {'Class': [], 'Extracted Text[] :'}
4.  #Get the bounding box coordinates and class name
            cls = int(r.boxes.cls[i])        xmin, ymin, xmax, ymax = r.boxes.xyxy[i]
            class_name = model.names[cls]
5.  #Crop and preprocess the region of interest (ROI)
            roi = image[int(ymin):int(ymax), int(xmin):int(xmax)]
            roi = cv2.resize(roi, None, fx=2, fy=2, interpolation=cv2.INTER_CUBIC)
6.  #Perform OCR using Tesseract
            custom_config = r'--oem 3 --psm 6 -l eng+fra'
            text = pytesseract.image_to_string(roi, config=custom_config)
7.  # Store the extracted text and class name
            data['Class'].append(class_name)       data['Extracted Text'].append(text)
```

4.1 Dataset Overview

The dataset is essential for evaluating our CV recognition system [32]. It includes 1.300 resumes from various sources, showcasing different formats, languages, and quality levels. Below is a detailed overview of the dataset, the types of data it contains, and the preprocessing steps taken to ensure quality and consistency. Fig. 14 shows samples of the dataset.

- **Total Number of Resumes:** 1,300
- **Formats:** JPEG, PNG, GIF
- **Languages:** Primarily English, with some resumes in other languages

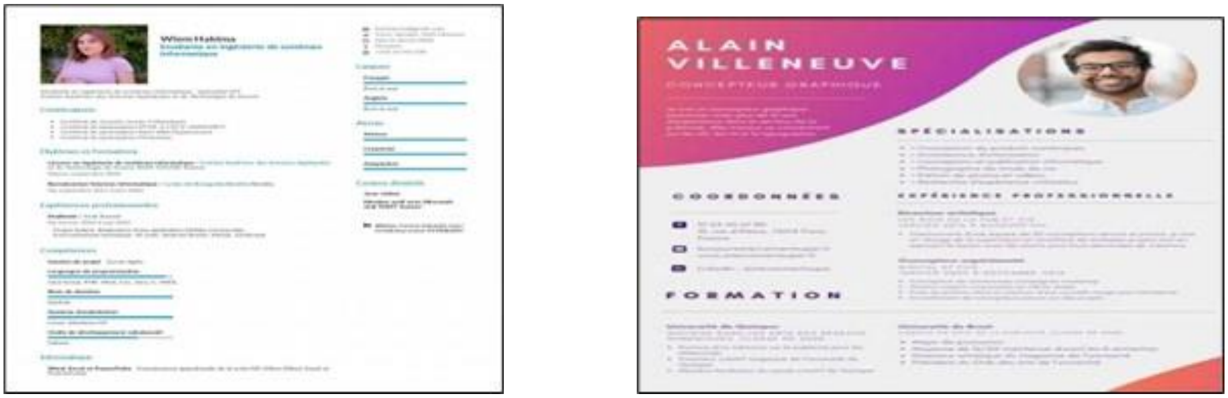

Figure 14: Samples of Resume Dataset

4.2 Preprocessing and Standardization of CV Dataset

To prepare the dataset for analysis, several preprocessing steps were applied as shown in Algorithm 2.

4.3 Experimental Results

4.3.1 Evaluation Metrics

To evaluate our model, we have used various performance metrics such as accuracy, F1 Score, Precision, and Recall which are calculated as below [29] [31]:

$$\text{Recall (Sensitivity)} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$\text{Accuracy} = \frac{correct\ prediction}{total\ prediction} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

$$\text{F}_1\text{\_score} = 2 * \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Where true negative (TN), true positive (TP), false negative (FN), and false-positive (FP).

$$mAp = \frac{1}{N} \sum_{i=1}^{N} AP_{\text{i}}$$

Where mean Average Precision (mAP) is the mean of the average precision scores for each class, Average Precision (AP) for each class is the area under the precision-recall curve.

$$\text{IOU} = \frac{Area\ of\ Overlap}{Area\ of\ Union}$$

Where Intersection over Union (IoU) is a metric used to evaluate the overlap between the predicted bounding box and the ground truth bounding box, Area of Overlap =Area of intersection between the predicted and grounded truth bounding box, Area of union = Total area covered by both the predicted and grounded truth bounding box minus the area of overlap.

Algorithm 2: Proposed System Training/Testing Cycle

Steps

1. **File Conversion:** Converted all resumes to a standard high-resolution image format (300 DPI) for uniform processing.
2. **Noise Reduction:** Processed images to remove noise and enhance text clarity, improving OCR accuracy.
3. **Layout Analysis and Segmentation:** Analyzed and segmented each resume into sections (personal details, education, work experience, etc.) for targeted data extraction.
4. **Text Normalization:** Normalized the extracted text to correct variations in font, size, and formatting, ensuring consistency.
5. **Data Organization:** After preprocessing and extraction, the data was organized into a structured format using Python's Pandas library. This structured dataset was then exported to an Excel file for further analysis.
   //Training Pipeline Steps:
6. **Training Images**: We start with a collection of training images, each containing various classes of text and objects found in resumes.
7. **Pre-Processing**: These images undergo pre-processing to enhance their quality. This step involves noise reduction, normalization, and other techniques to improve image clarity and make feature extraction more efficient.
8. **Segmentation & Feature Extraction**: The pre-processed images are segmented into different regions, and features are extracted from these segments. This step is crucial for identifying the distinct classes within the images, such as headings, names, and contact information.
9. **Training**: The extracted features are used to train the YOLOv8 model. During this phase, the model learns to detect and classify the 11 different classes within the resumes.
10. **Model File**: Once the training is complete, the model is saved as a **model file. This file contains all the learned parameters and is essential for the recognition phase in the testing pipeline.**
    // Testing Pipeline Steps:
11. **Input Images: The testing phase begins with new input images, which are resumes that** need to be processed and analyzed.
12. Pre-Processing: Similar to the training pipeline, these input images undergo pre-processing. This ensures that the images are in optimal condition for feature extraction and recognition.
13. Segmentation & Feature Extraction: The pre-processed images are segmented, and features are extracted. This step identifies and isolates the different text regions and objects within the resumes.
14. Recognition: Using the trained model file, the system recognizes and classifies the text and objects in the images. YOLOv8 detects the bounding boxes for the 11 classes, and Tesseract-OCR extracts the text within these boxes.
15. **Post-Processing**: The recognized text undergoes post-processing to correct any errors and refine the results. This step ensures that the extracted text is accurate and ready for further use.
16. **OCR Output**: The final output is the OCR result, which contains the clean and verified text extracted from the resumes. This data is then saved into a database.

4.3.2 YOLOv8 Detection Results

The YOLOv8 model validation results reveal a high level of performance across a diverse set of object detection classes as shown in Table 3. The model achieved an impressive overall mean Average Precision (mAP) of 92% at 50% Intersection over Union (IoU) and 89.1% for the more stringent mAP50-95, which averages mAP at IoU thresholds from 50% to 95%. For individual classes, the model consistently demonstrated high precision and recall:

Table 3: OCR Evaluation Metrics and Object Detection Performance

Class	Images	Instances	Box (P)	Box (R)	mAP50	mAP50-95
All	333	2550	0.901	0.797	0.891	0.8
Certifications	333	45	0.840	0.560	0.788	0.643
Community	333	43	0.756	0.581	0.782	0.674
Contact	333	339	0.940	0.875	0.962	0.845
Education	333	313	0.927	0.927	0.975	0.912
Experience	333	332	0.978	0.924	0.981	0.951
Interests	333	140	0.916	0.776	0.893	0.802
Languages	333	333	0.926	0.866	0.960	0.845
Name	333	333	0.926	0.866	0.960	0.845
Profile	333	109	0.926	0.866	0.960	0.845
Projects	333	53	0.875	0.604	0.772	0.738
Skills	333	333	0.901	0.869	0.939	0.868

Confusion matrices are a good way to see which classes are confused the most. The confusion matrix in Figure 15 illustrates the YOLOv8 model's strong performance, with high true positive rates across most classes, indicating accurate detection and classification. Notably, classes such as Contact, Education, Experience, and Languages show excellent results, with minimal false positives and false negatives. For instance, the Contact class correctly identifies 328 out of 339 instances, highlighting the model's precision. The low error rates and high precision and recall scores across various classes confirm the robustness and effectiveness of the detection system. These results validate the quality and reliability of the proposed YOLOv8 model for object detection tasks.

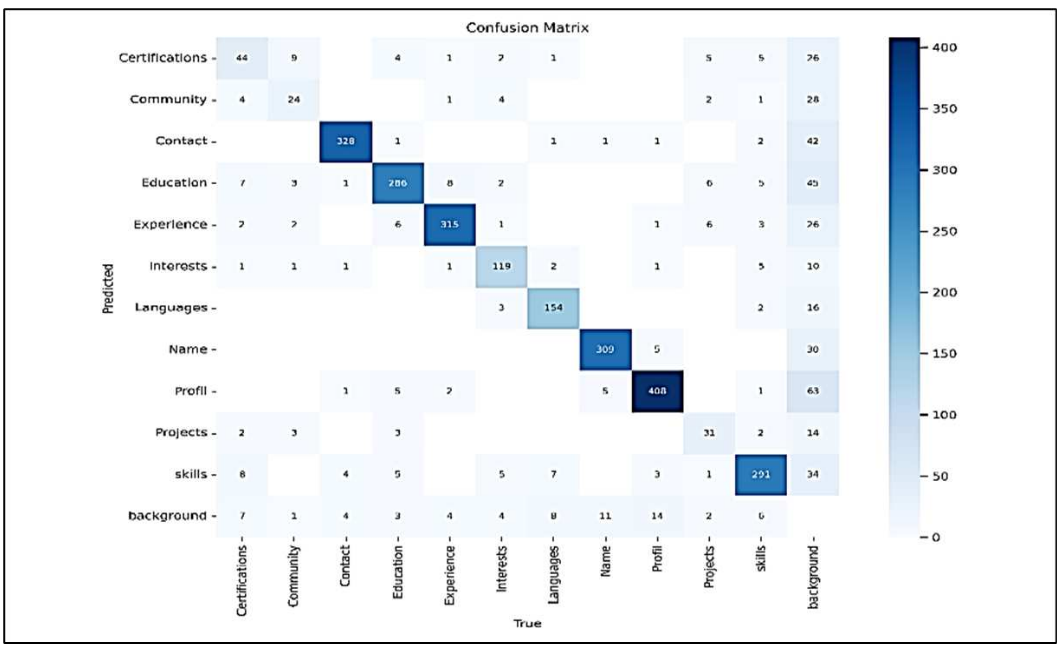

Figure 15: Confusion Matrix for identified classes using YOLOv8

4.3.3 OCR Results

In addition to the object detection metrics, we evaluated the quality of words extracted by Tesseract OCR. The focus was on maintaining the integrity and clarity of the detected words to ensure they are suitable for further natural language processing tasks. In this section, we perform OCR error analysis on the character level and word level. Furthermore, we analyze post-correction results on the word level. The extracted text was checked against a corpus of language to evaluate and potentially modify the words based on the following metrics:

- **Word Accuracy**: Percentage of correctly identified and classified words. It is computed as:

$$WA = \frac{Number\ of\ correct\ words}{Total\ number\ of\ ground\ truth\ words} X100$$

- **Character Error Rate (CER)**: Number of incorrect characters divided by the total number of characters in the ground truth, which is computed as:

$$CER = \frac{Number\ of\ incorrect\ characters}{Total\ number\ of\ ground\ truth\ characters} X100$$

The reported results by our system are: Word Accuracy without error modification = 81.5% and with error modification = 93.22%. The Character Error Rate (CER) = 7.6%.

Figure 16 shows Word Accuracy of three sample images from the dataset after correcting text errors. As seen, they exhibit almost 100% accuracy.

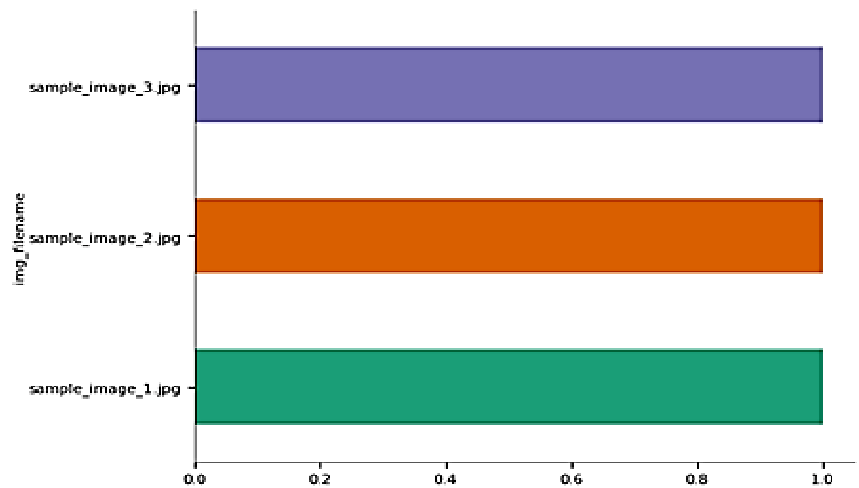

Figure 16: Word Accuracy of Three Sample Images from the Dataset

4.3.5 Results Analysis

In this subsection, we discuss the segmentation results obtained by YOLOv8 and the OCR results obtained by Tesseract presented in the paper, moreover, the effect of the post-processing step. The comparison with similar previous work demonstrates a substantial improvement in performance of our system. The current implementation of YOLOv8 combined with Tesseract OCR achieves an accuracy of 93.22% a significantly high compared to similar systems in the literature. The precision and recall also show marked improvements, confirming the enhanced capability of the proposed system in accurately detecting and extracting relevant information. The post-processing step increased the accuracy of the generated output text from 81.5% to be 93.22%. These improvements are indicative of the model's high quality and effectiveness, making it a strong candidate for real-world applications in resume parsing and other text extraction tasks. The employed post processing steps that involved the correction of the generated text alleviated the problems associated with the errors of detection of letters caused by Tesseract OCR. The confusion matrix indicates the ability of the proposed system of correctly classifying the components of the CV which reached in many classes equals 0 false negative or positive which in an incredible result for such real-time application.

5. Conclusion

The integration of advanced technologies such as YOLO (You Only Look Once) and PyTesseract has demonstrated a significant potential for automating complex tasks involving image and text analysis. This project specifically aimed to develop a robust system for detecting and extracting information from resumes, showcasing the combined power of these tools in practical applications. Throughout the previous chapters, we have meticulously explored the capabilities of YOLO for real-time object detection and PyTesseract for optical character recognition (OCR), culminating in a system designed to streamline resume parsing and analysis.

Key Achievements:

- **Efficiency**: The YOLO model's ability to detect multiple objects in a single pass has proven to be highly efficient, making it suitable for real-time applications. This efficiency is critical in scenarios where quick processing is required, such as in high-volume resume screening.
- **Accuracy**: By combining YOLO's precise object detection with PyTesseract's robust text extraction capabilities, the system achieved high accuracy in identifying and extracting relevant sections from resumes. This accuracy is further enhanced through preprocessing techniques and text correction algorithms.
- **Versatility**: The system's versatility is evident in its ability to handle various document formats, languages, and quality levels. This makes it a valuable tool for a wide range of applications beyond resume parsing, including document management and information retrieval.

Throughout the development process, several challenges were encountered, which present opportunities for further refinement:

- **Variability in Document Formats**: The diverse formats and layouts of resumes posed a challenge for consistent object detection and text extraction. Future work could focus on developing adaptive algorithms that can dynamically adjust to different document structures.
- **OCR Accuracy**: The accuracy of text extraction was sometimes affected by factors such as font styles and image quality. Implementing advanced OCR techniques and continuous model training on varied datasets can mitigate these issues.

5.1 Future Work

While the developed system has shown promising results, there are several avenues for future work that could further enhance its capabilities and extend its applicability. Below are some key areas for potential improvement and exploration:

5.1.1 Enhancing Model Performance

Model Training: Future work could involve training the YOLO model on a more extensive and diverse dataset to improve its detection accuracy for a broader range of document types and formats. This would help in better generalizing the model to various real-world scenarios.

Advanced Preprocessing Techniques: Incorporating more sophisticated image preprocessing techniques, such as deep learning-based image enhancement and denoising, could further improve OCR accuracy, especially for low-quality images.

5.1.2 Improving Text Extraction and Correction

Contextual Understanding: Implementing advanced natural language processing (NLP) techniques to understand the context of the extracted text could significantly improve the accuracy of information extraction. For instance, using Named Entity Recognition (NER) to identify specific details such as names, dates, and locations within resumes.

Error Correction: Developing more sophisticated text correction algorithms that leverage contextual information and language models (e.g., using transformer-based models like BERT) could enhance the accuracy of the extracted text.

5.1.3 Expanding Use Cases

Document Variety: Extending the system to handle a wider variety of documents beyond resumes, such as invoices, contracts, and academic papers, would increase its utility in different industries.

Real-Time Applications: Optimizing the system for real-time applications, such as mobile scanning apps or live document analysis tools, could open up new possibilities for on-the-go document processing.

References

1) F. Sultana, A. Sufian, P. Dutta, "A Review of Object Detection Models based on Convolutional Neural Network," in Intelligent Computing: Image Processing Based Applications, Advances in Intelligent Systems and Computing, vol. 1157, pp. 1-16, (2020).
2) Zhiqiang Wang and Jun Liu, "A review of object detection based on convolutional neural network," in 2017 36th Chinese Control Conference (CCC), July 26-28, 2017, Dalian, China. IEEE,
3) Zhong-Qiu Zhao, Peng Zheng, Shou-Tao Xu, Xindong Wu, "Object Detection With Deep Learning: A Review," *IEEE Transactions on Neural Networks and Learning Systems*,vol. 30, no. 11, pp. 3212-3232, November (2019).
4) S. Srivastava, A. Verma, and S. Sharma, "Optical Character Recognition Techniques: A Review," in Proceedings of the 2022 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS), Bhopal, India, 19-20 February 2022, pp. 1-6.
5) Jeonghun Baek, Geewook Kim, Junyeop Lee, Sungrae Park, Dongyoon Han, Sangdoo Yun, Seong Joon Oh, and Hwalsuk Lee, "What is wrong with scene text recognition model comparisons? Dataset and model analysis," CoRR, vol. abs/1904.01906, (2019).

6) Jain, M.; Mathew, M.; Jawahar, C. Unconstrained OCR for Urdu using deep CNN-RNN hybrid networks. In Proceedings of the2017 4th IAPR Asian Conference on Pattern Recognition (ACPR), Nanjing, China, 26–29 November 2017; pp. 747–752.
7) Hossam Magdy Balaha, Hesham Arafat Ali, Mahmoud Badawy, "Automatic recognition of handwritten Arabic characters: a comprehensive review " , April 2021, Neural Computing and Applications (33) , 3011-3034
8) Daraee, F.; Mozaffari, S.; Razavi, S.M. Handwritten keyword spotting using deep neural networks and certainty prediction.Comput. Electr. Eng. 2021,92, 107111.
9) Vinjit, B.M.; Bhojak, M.K.; Kumar, S.; Chalak, G. A Review on Handwritten Character Recognition Methods and Techniques. Proceedings of the 2020 International Conference on Communication and Signal Processing (ICCSP), Chennai, India, 28–30 July2020; pp. 1224–1228.
10) Junyan Lu, Chi Ma, Li Li, Xiaoyan Xing, "A Vehicle Detection Method for Aerial Image Based on YOLO," *Journal of Computer and Communications*, vol. 6, no. 11, pp. 98-107, January (2018).
11) Rachel Huang, Jonathan Pedoeem, and Cuixian Chen, "YOLO-LITE: A Real-Time Object Detection Algorithm Optimized for Non-GPU Computers," IEEE international Conference on Big Data (Big Data), (2018).
12) B. Eddine, "Build a Trained Data Model for Tesseract OCR Engine for Tifinagh Script Recognition," Data & Metadata, vol. 2, no. 185, February (2024),
13) Thi Tuyet Hai Nguyen, Adam Jatowt, Mickael Coustaty, and Antoine Doucet, "Survey of Post-OCR Processing Approaches," *ACM Computing Surveys (CSUR)*, vol. 54, no. 6, article no. 124, pp. 1-37, (2021).
14) B. Eddine, "Text Detection and Recognition from Scene Images Using RCNN and EasyOCR,".In *IOT with Smart Systems*, pp. 75-85. (2023).
15) Agnieszka Wosiak, "Automated extraction of information from Polish resume documents in the IT recruitment process," Procedia Computer Science, vol. 192, pp. 2432-2439, (2021).
16) Avisha Anand, Sandeep Dubey, "CV Analysis Using Machine Learning," International Journal for Research in Applied Science & Engineering Technology, DOI:10.22214/ijraset.2022.42295, Corpus ID: 248877934, (2022).
17) Yang Yang, Jingshuai Zhang, Fan Gao, Xiaoru Gao, Hengshu Zhu, "DOMFN: A Divergence-Orientated Multi-Modal Fusion Network for Resume Assessment," MM '22: Proceedings of the 30th ACM International Conference on Multimedia, pp. 1612-1620, (2022).
18) Majd E. Tannous, Wassim H. Ramadan, and Mohanad A. Rajab, "TSHD: Topic Segmentation Based on Headings Detection (Case Study: Resumes)," Advances in Human-Computer Interaction, Article ID 6044007, 12 pages vol. (2023)
19) Thanh Tung Tran, Truong Giang Nguyen, Thai Hoa Dang, and Yuta Yoshinaga, "Improving Human Resources' Efficiency with a Generative AI-Based Resume Analysis Solution," Future Data and Security Engineering. Big Data, Security and Privacy, Smart City and Industry 4.0 Applications (FDSE 2023), pp. 352-365. First Online: 17 November (2023).
20) M. Yuan, Q. Zhang, Y. Li, Y. Yan, Y. Zhu, A suspicious multi object detection and recognition method for millimeter wave SAR security inspection images based on multi-path extraction network, Rem Sens 13 (2021).
21) J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only Look once: unified, real-time object detection, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Las Vegas, 2016, pp. 779e788,
22) C.-Y. Wang, A. Bochkovskiy, H.-Y.M. Liao, C.-Y. Wang, A. Bochkovskiy, H.-Y.M. Liao, YOLOv7: trainable bag-offreebies sets new state-of-the-art for real-time object detectors, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVF, Vancouver. (2023), pp. 7464e7475.
23) B. Leibe, J. Matas, N. Sebe, M. Welling, Computer Vision e ECCV, first ed., Springer International Publishing, Cham. (2016)
24) M. Maktab, M. Razaak, P. Remagnino, Enhanced single shot small object detector for aerial imagery using super-resolution feature fusion and deconvolution, Sensors 22 (2022)
25) Howard, A.; Sandler, M.; Chu, G.; Chen, L.-C.; Chen, B.; Tan, M.; Wang, W.; Zhu, Y.; Pang, R.; Vasudevan, V. Searching for mobilenetv3. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 1314–1324.
26) Jocher, G., Chaurasia, A., and Qiu, J. 2023. "YOLO by Ultralytics (Version 8.0.0)."Computer software.
27) Ultralytics. 2023. "YOLOv8 Docs." https://docs.ultralytics.com/. Accessed April 27,2023.
28) Yuning Du, Chenxia Li, Ruoyu Guo, Xiaoting Yin, Weiwei Liu, Jun Zhou, Yifan Bai, Zilin Yu, Yehua Yang, Qingqing Dang, Haoshuang Wang " PP-OCR: A Practical Ultra Lightweight OCR System" (2020)
29) Ž. Đ. Vujović, "Classification Model Evaluation Metrics," International Journal of Advanced Computer Science and Applications, vol. 12, no. 6, pp. 599-606, (2021).

30) Abrar Al Sayem, Asiful Islam Chowdhury, Shahriar Hossain Shojol, Md Humaion Kabir Mehedi, and Annajiat Alim Rasel, "Implementing the Tesseract Method for Information Extraction from Images," Procedia Computer Science, vol. 227, pp. 97-101, (2023).
31) Oona Rainio, Jarmo Teuho, and Riku Klén, "Evaluation metrics and statistical tests for machine learning," Scientific Reports, vol. 14, no. 1, March 2024.
32) Open source resume dataset, https://universe.roboflow.com/kkkk-mlls6/resum, 2023 (last accessed 7-8-2024).