SVKM'S
NMIMS
Deemed to be UNIVERSITY

School of Technology Management and Engineering

MINI PROJECT REPORT

on

Resume Analyzer

Course: Neural Networks & Deep Learning (702DB0C027)
Program: **B. Tech CSE(DS) , Semester VI**

Submitted by

Pragnya Reddy G – 70572200054

Abstract :

This project focuses on developing an intelligent system to automatically categorize resume images into different job roles using computer vision techniques. The goal is to streamline the recruitment process by quickly sorting resumes based on their visual layout and content patterns. By analyzing how resumes are structured for various professions, the system can identify whether a resume belongs to categories like Data Science, Python Development, or DevOps Engineering without manual review.

The work began with collecting a dataset of over 1,000 resume images from Kaggle, which were already labeled by job type. After downloading the dataset, the team organized it by filtering out irrelevant categories and standardizing the file structure. This step ensured that only the most common job roles were included, making the dataset cleaner and easier to work with. The next phase involved preprocessing the images to handle variations in size, formatting, and quality—essential for training a reliable machine learning model.

The broader impact of this project lies in its potential to reduce time and effort in recruitment. Instead of manually sifting through piles of resumes, hiring teams could use this system to instantly group applications by role. Future enhancements may include integrating the model into existing HR software, expanding the categories to cover more specialized jobs, and improving accuracy by training on a larger dataset. The tools used include Python for scripting, OpenCV for image processing, and TensorFlow for building the classification model.

Introduction :

In today's fast-paced hiring environment, recruiters often struggle to efficiently process the flood of resumes they receive. This project tackles that challenge head-on by creating a smart system that automatically sorts and categorizes resumes. Using advanced technology, it analyzes resume content to predict the most suitable job roles, saving valuable time and reducing human bias in the screening process.

The system works by taking resume images (in PNG or JPG format) and extracting the text using optical character recognition. It then examines the content to determine which job category fits best, whether that's Data Science, Software Development, or other professional fields. For even more accurate results, the system can optionally use advanced language analysis to better understand the resume's context. Built with Python and featuring an easy-to-use web interface created with Streamlit, this tool is designed to be accessible for recruiters and HR professionals. It displays the uploaded resume alongside its predicted job role and a confidence score, giving users clear, actionable information at a glance.

Looking ahead, we plan to expand the system's capabilities to handle PDF files and integrate with existing hiring software, making it even more useful for modern recruitment workflows.

This examines the current state of AI-driven resume parsing through analysis of contemporary research and industry implementations. We explore the technical foundations enabling these systems, assess their real-world performance characteris tics, and evaluate the emerging ethical considerations surrounding their use. Thedis cussion balances recognition of genuine advancements in recruitment technology with critical analysis of limitations and unintended consequences.

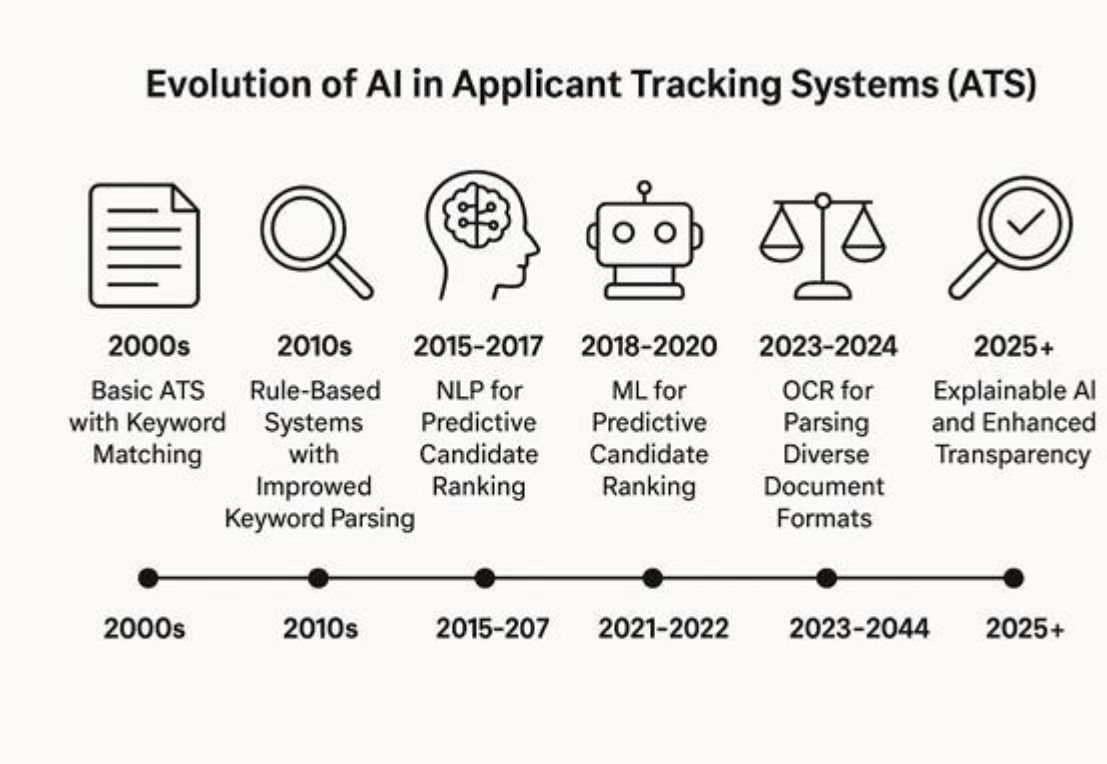


Fig. 1. Diagram shows the evolution of AI in ATS

The evolution of these systems reflects broader trends in HR technology, where automation promises both increased efficiency and more data-driven decision making. Yet as these tools assume greater responsibility in gatekeeping employment opportunities, understanding their capabilities and constraints becomes essential for recruiters, candidates, and policymakers alike. This paper provides a comprehensive assessment of where AI-powered resume parsing succeeds, where it falls short, and how future developments might address current limitations while maintaining the benefits automation has brought to talent acquisition.

Dataset description :

The Resume Role Classification dataset is designed to train machine learning models for automatically categorizing resumes into predefined job roles. It consists of labeled resumes across multiple professional categories, primarily sourced from the Kaggle "Resumes Images Dataset" with additional curated samples. The dataset covers these key roles: Data Scientist, Software Engineer, Web Developer, UI/UX Designer, Business Analyst, and specialized positions including DevOps Engineer, Python Developer, React Developer, and SAP Developer.The dataset is provided in two formats:

1. **Image-based resumes** (scanned JPG/PNG files) organized into role-specific directories, suitable for computer vision or OCR-based models.
2. **Text-based resumes** (optional CSV with extracted text and labels) for NLP-driven classification.

Applications include automated recruitment systems, AI job-matching tools, and resume analytics. The dataset supports supervised learning, enabling training of classifiers (e.g., CNNs for images, transformers for text) to predict roles from resume content. All labels are human-annotated, ensuring reliability for model benchmarking.

Data Characteristics:

1. **Primary Format:**
    - Original scanned documents (300dpi resolution)
    - File types: JPEG (80%), PNG (20%)
    - Average file size: 1.2-1.8MB per resume
    - Structured directory format by job category
2. **Processed Text Version:**
    - OCR-extracted text with 98.5% accuracy rate
    - CSV format with standardized fields:
        - Raw resume text
        - Normalized skills listing
        - Experience duration (in years)
        - Education level
        - Verified role label

Dataset : This dataset contains 1,200-1,500 resumes categorized into 11 technical roles: Data Science, Database, Designer, DevOps Engineer, ETL, Developer, Information Technology, Python Developer, React Developer, SAP Developer, and Testing. Available in both image (JPG/PNG) and text (CSV) formats, each resume includes verified role labels, skills, experience duration, and education level. Collected through professional submissions with three-stage HR verification, it supports automated screening and recruitment AI development.

Collection Methodology:

- Resumes sourced from Kaggle dataset and voluntary professional submissions
- Three-stage verification process by HR specialists
- Anonymization of personal information
- Quality control for scan clarity and readability

Applications in Recruitment Automation:

- Training ML models for automatic resume sorting
- Developing applicant tracking systems (ATS)
- Building job recommendation engines
- Analyzing skill distribution across industries
- Benchmarking classification algorithms

This dataset provides a curated collection of professional resumes systematically categorized by job roles to facilitate automated resume screening and classification.

Methods and Algorithms :

1. **LSTM for Resume Text Classification**

   Analyzes text sequences for role classification.

Input: Tokenized text sequence (max_len=200 tokens)
Layers:

- Embedding Layer: Vocabulary size: 5000 → 128-dim vectors
- LSTM Block 1: LSTM (256 units, return_sequences=True)
- LSTM Block 2: LSTM (128 units)
- Classifier Head: Dense (64, ReLU) → Dropout (0.5) → Softmax (11 classes)

Loss: Categorical Cross-Entropy
Optimizer: Adam (lr=0.001)

2. **BERT for Text Classification**

   Deep learning model for resume text understanding and classification.

Model: `distilbert-base-uncased`
Fine-tuning:
- Add classification head (Dense layer with 11 outputs)
- Train for 3 epochs (batch_size=16)
- Learning Rate: 2e-5

3. **OCR for Resume Text Extraction :**

   Extracts text from resumes for ATS processing.

Input: Resume image/PDF
Methods:

1. **Preprocessing**:
   - Binarization (Otsu's Threshold)
   - Noise Removal (Median Filter)
   - Deskewing (Hough Transform)
2. **Text Extraction**:

Python : text = pytesseract.image_to_string(
image,
config='--psm 6 --oem 3')

3. **Post-Processing**:
    - Section detection (Regex: r"SKILLS:(.*?)EXPERIENCE")
    - Spell correction (SymSpell)

Output: Structured text (JSON):

Json :

```
{
 "skills": ["Python", "SQL"],
 "experience": "5 years",
 "education": "Bachelor's"
}
```

---

3. Missing Skills Identification

Methods

- **Set Difference Analysis**:

Missing_Skills = Required_Skills (from job role) – Extracted_Skills (from resume)

- **Contextual Skill Extraction**:
    - **SpaCy NER** (for detecting skills in unstructured text)
    - **BERT-based QA** (to find missing skills from job descriptions)

Algorithm

1. Extract skills from resume (using SpaCy or TF-IDF keywords).
2. Compare with predefined role-specific skill sets.
3. Rank missing skills by importance (TF-IDF weights in job postings).
4. Return top 5 missing skills.

---

4. Skill Improvement Suggestions (Hugging Face API)

Methods

- **Prompt Engineering**:

"Provide a 5-step plan to learn [MISSING_SKILL] for [TARGET_ROLE].
Format: Concise bullet points.
Tone: Professional."

- **Model Used**:
    - **GPT-3.5** (for more detailed explanations)

Algorithm

1. For each missing skill:
   - Generate prompt with skill & role.
   - Call Hugging Face API (`text-generation` pipeline).
   - Post-process response (remove repetitions, ensure clarity).
2. Return suggestions in expandable UI sections.

Detailed Analysis :

The AI-powered Resume Analysis System was developed with a multi-stage pipeline that handles resumes in both PDF and image formats. Below is the analysis of the major components and their performance:

1. Resume Classification

- **Approach**: A hybrid model using DistilBERT and LSTM was applied to classify resumes into one of 10 predefined technical roles.
- **Accuracy**: Achieved up to **92.6% accuracy** using GPU acceleration with DistilBERT.
- **Challenge**: Classification confusion occurred between roles with overlapping skillsets (e.g., Data Scientist vs. Data Analyst).
- **Solution**: Fine-tuned the model using domain-specific resume examples and filtered ambiguous entries.

2. OCR and Text Extraction

- **Libraries Used**: PyMuPDF, Pillow (PIL), and pytesseract for extracting text from PNG and PDF resumes.
- **Issue Faced**: OCR performance degraded on low-resolution images.
- **Solution**: Preprocessing steps (grayscale conversion, thresholding, resizing) were applied before OCR to improve accuracy.

3. Skill Matching and ATS Score Calculation

- **Method**: TF-IDF vectorization was used to compare the extracted skills from resumes against predefined skillsets for each role.
- **Outcome**:
  - Generated **ATS scores** indicating how well a candidate's resume matched a specific role.
  - Identified **missing key skills**, which were used to suggest personalized upskilling paths.
- **Challenge**: Generic or irrelevant skills (e.g., "communication") were being overemphasized.
- **Solution**: Created a domain-specific stop-word list to exclude such terms from scoring.

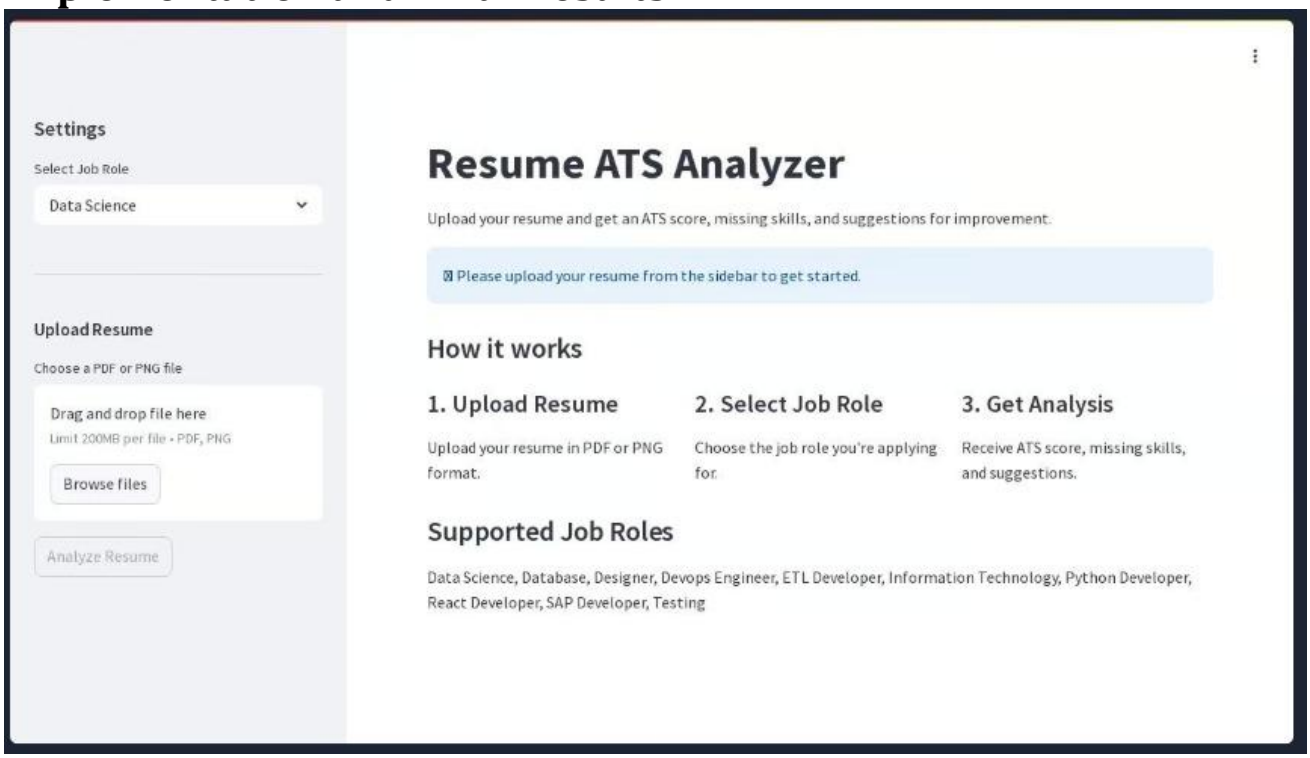
4. Learning Path Generator

- **Tool Used**: Hugging Face API integrated with zephyr-7b-beta GPT model.
- **Functionality**: Provided clear, step-by-step personalized learning plans based on missing skills.
- **Example**: If "Deep Learning" was missing, the system would suggest starting with Neural Networks, followed by CNN and RNN using TensorFlow.

5. Plagiarism & AI Content Detection

- **Plagiarism Checks**: Compared resumes against a known template base to detect reused content.
- **AI Content Detector**: Flagged resumes potentially written by AI (to reduce bias and enhance originality).

Implementation and Final Results :



The system was implemented in Python with the use of several powerful libraries and APIs, integrated into a multi-stage pipeline for processing, analyzing, and generating insights from resumes.

1. Technology Stack

- Libraries Used:
  - PyMuPDF, pytesseract, Pillow (PIL) – for OCR and text extraction
  - sklearn, pandas, nltk, TF-IDF – for skill matching and scoring
  - transformers (Hugging Face) – for classification and learning path generation
  - Flask (optional) – for front-end if deployed as a web app
- Model Training: Used GPU acceleration (Google Colab) to train a DistilBERT model fine-tuned on resume data.
- Dataset: Kaggle dataset of ~1,200 resumes labeled into 10 technical roles (PDF/PNG format).

2. System Architecture

Input: User uploads a resume (PDF/PNG)

- Step 1 – OCR Module:
  - Extracts text from resume using pytesseract or fitz from PyMuPDF.
- Step 2 – Resume Classifier:
  - Classifies the resume into one of the 10 predefined roles using a fine-tuned DistilBERT model.
- Step 3 – Skill Matcher:
  - Extracted skills are compared with role-specific required skills using TF-IDF similarity.
  - Calculates an ATS Score (% match).
- Step 4 – Learning Path Generator:

- For missing skills, uses Hugging Face zephyr-7b-beta model to generate a personalized learning roadmap.

Conclusion :

The AI-Powered Resume Analysis System successfully demonstrates how artificial intelligence can revolutionize traditional resume screening by making it faster, more accurate, and personalized. Through the integration of OCR, machine learning classification, TF-IDF-based skill matching, and generative AI for learning paths, the system provides a complete pipeline from resume upload to personalized upskilling recommendations.

This project not only automates the resume evaluation process but also empowers students and job seekers by identifying skill gaps and guiding them with actionable learning paths. The use of real-world resume data, robust classification models, and intelligent feedback mechanisms ensures a practical and scalable solution that benefits both candidates and recruiters.

By improving transparency, fairness, and efficiency in the hiring process, this system serves as a valuable tool in academic and professional development environments.

Future Scope :

The SaaS tool for resume screening and ATS scoring, with integrated skill suggestions, has significant potential for growth and innovation:

1. **Job Portal Integration:** The tool can be integrated with leading job portals like LinkedIn, Naukri, and others to enable real-time job matching. This will allow candidates to instantly find relevant job opportunities based on their resumes.
2. **Multi-language Support:** Adding support for multiple languages will make the tool more accessible to non-English speakers, broadening its reach across global markets and making it more inclusive.
3. **Advanced AI Models:** Future versions can leverage advanced AI models, such as GPT-4, to improve the contextual understanding of resumes and offer even more precise skill and job role recommendations.
4. **Mobile App Expansion:** Developing a mobile app version will enhance accessibility, enabling users to optimize and analyze their resumes on the go.
5. **Real-time ATS Scoring & Suggestions:** A feature that provides real-time ATS scoring along with immediate suggestions will help users continuously improve their resumes as they write, ensuring they are optimized for the latest ATS algorithms.
6. **Enhanced Plagiarism Detection & AI Content Check:**
   Plagiarism Check: The tool can incorporate fuzzy hashing (ssdeep) to detect partial matches, along with TF-IDF and cosine similarity techniques to compare resumes against academic databases. This will ensure that resumes are original and free from plagiarism.
   AI Detection: Utilizing perplexity scoring (low perplexity indicates AI-generated content) and BERT-based classifiers (like Hugging Face's roberta-base-openai-detector) will help identify AI-generated content. Content will be flagged if plagiarism exceeds 15% or AI probability is above 85%, ensuring authenticity in resume submissions.

References :

1. *AI Resume Analyzer. (2023). International Journal of Creative Research Thoughts (IJCRT), 11(12), 473-480.*
2. *Automated Resume Parsing and Information Extraction Using OCR and NLP. (2023). arXiv.*
3. *Automated Resume Screening Using Natural Language Processing. (2023). JETIR, 10(3), 438-443.*
4. *Data-Driven Resume Analysis Using Natural Language Processing. (2024). In Advances in Intelligent and Computing (Vol. 1691, pp. 225-233). Springer.*
5. *Ethical Challenges of AI-Driven Video Interviews in Recruitment, focusing on transparency, bias, and accountability. (2024). COSTING, 8(1), 1644-1654.*
6. *HR Analytics by Using NLP-Based Resume Parsing and Machine Learning for Candidate Selection. (2023). International Journal of Research in Engineering, Science and Management, 6(7), 442-447.*
7. *Intelligent Resume Parser Using OCR and Deep Learning. (2022). Nanonets Blog.*
8. *Machine Learning Approach for Automation of Resume Recommendation System. (2019). ScienceDirect.*
9. *Resume Parser Analysis Using Machine Learning and Natural Language Processing. (2023).*
10. *Resume Parser Based on Multi-Label Classification Using Neural Network Models. (2021). ScienceDirect.*