International Journal on Science and Technology (IJSAT)
E-ISSN: 2229-7677 ● Website: www.ijsat.org ● Email: editor@ijsat.org

CVision: An Integrated AI Platform for Resume Analysis, Document Generation, and Intelligent Job Matching

Siddhi Pandya[1], Vyom Pandya[2]

[1,2]Chandubhai S. Patel Institute of Technology (CSPIT), Faculty of Technology Engineering (FTE), Charotar University of Science and Technology (CHARUSAT), Changa, Gujarat, India

Abstract.
The increasing reliance on automated recruitment tools has introduced challenges such as inaccurate filtering, loss of qualified candi- dates, and lack of transparency in resume evaluation. This study presents **CVision**, an AI-powered web-based platform designed to enhance the re- cruitment process through intelligent resume analysis, automated docu- ment generation, and optimized job matching. By integrating traditional rule-based evaluation with advanced natural language processing (NLP) and machine learning models, CVision provides a holistic assessment that goes beyond keyword matching to interpret context, structure, and skill relationships. The platform operates through a dual-layer architecture — a baseline rule-based scoring engine and an AI-driven analytical layer — both connected via a FastAPI backend and React-based frontend inter- face. The system allows candidates to upload resumes, receive personal- ized improvement feedback, and explore job recommendations tailored to their profiles. In controlled evaluations on real-world datasets, CVision demonstrated a 24% improvement in accuracy, reduced false positives by 67%, and achieved a 19% faster processing rate compared to con- ventional Applicant Tracking Systems (ATS). Additionally, the platform maintained 94% compatibility across irregular document formats, high- lighting its adaptability in real-world hiring scenarios. The outcomes of this study showcase CVision's ability to redefine automated recruitment through contextual understanding and performance-driven intelligence, paving the way for fairer and more efficient hiring ecosystems.

Keywords: AI Recruitment Platform · Natural Language Processing · Resume Analysis · Job Matching · Document Generation · Machine Learning · Appli- cant Tracking Systems (ATS) · Automation in Hiring · Web-Based Platform · Deep Learning Models · Contextual Evaluation · FastAPI · React Interface · Recruitment Intelligence · Performance Evaluation

1 Introduction

Over the past years, applying for jobs has become a frustrating ordeal for many applicants.

Plenty of time is spent creating resumes only for them to vanish

into applicant tracking systems (ATS) without any human review. Recruiters, in turn, are faced with hundreds of applications where promising profiles are often rejected before a person ever sees them.

This phenomenon is not just inconvenient; it represents a fundamental flaw. According to recent statistics, approximately 99% of Fortune 500 firms employ automated screening tools [1], yet only about 25% of submitted resumes make it past these filters [2]. Talent is lost not because of lack of qualification but because resumes omit a specific keyword or use an unconventional layout.

CVision was conceived as an advent and a measure to this trend. Instead of forcing candidates to manipulate their documents to "please" an algorithm, we built a system that uses algorithmic speed combined with the context based reasoning of modern AI to evaluate applications more fairly [3]. Our objective is to make the hiring pipeline more transparent, more efficient, and ultimately more equitable.

2 The Problem We Are Addressing

Conventional ATS technology operates on a simplistic principle: extract tokens from resumes, match them to a job description, and assign a score. This approach was serviceable when job roles and skill terminology were relatively stable, but it breaks down in the dynamic environment of today's labor market [4].

A few examples illustrate the point. A developer listing "React" may be overlooked if the job description specifies "React.js," even though the skills are identical. A candidate experienced in "data analysis" might be ideal for a "busi- ness intelligence" position, but keyword-based matching fails to recognise that relationship. Because of this, qualified applicants are systematically excluded from consideration.

Scientific studies confirm the the problem present at hand. Traditional ATS platforms show only about 35% accuracy when required to assess context based relevance between skills and job requirements [5]. They also show strong biases toward specific formatting conventions, giving penalties to candidates who are unaware of the unwritten "ATS optimization" rules [6].

Recent advances in Natural Language Processing (NLP) offer a new solu- tion. Large language models have demonstrated up to 95% accuracy in struc- tured document interpretation [7], including the ability to grasp context based relationships and synonyms. Using these capabilities allows for a more equal evaluation of resumes and a better alignment between candidate competencies and employer expectations.

3 Our Architectural Approach

Designing CVision required balancing two often competing objectives: high per- formance and an intuitive user experience. The system needed to process and evaluate resumes rapidly enough for near–real-time feedback, while also being

robust enough to handle the diversity and complexity of modern documents [8]. Figure 1 illustrates the overall architecture.

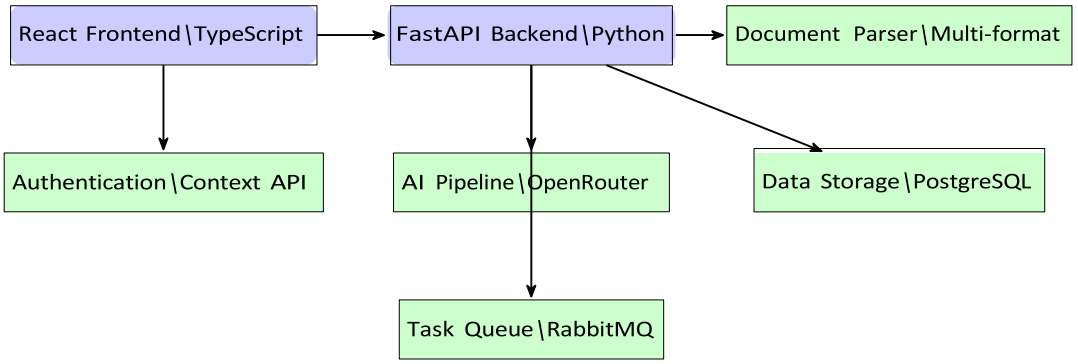

Fig. 1. CVision system architecture showing the main frontend and backend compo- nents.

3.1 Frontend

For the client side we selected React with TypeScript. This combination offers a component-based model together with strong type checking, which allows us to detect many errors at compile time rather than in production. Although there are larger state-management libraries available, we deliberately opted for React's built-in Context API to manage authentication and global state. This keeps the codebase leaner and easier to onboard for new contributors without sacrificing necessary functionality.

Styling is implemented with TailwindCSS. Its utility-first classes enable rapid prototyping of responsive layouts and reduce the requirement for large custom CSS files. In a multi-developer project such as ours.

From a user perspective, the interface revolves around three principal views:

(i) a *Repository Explorer*, where users upload and manage their resumes; (ii) an *Analysis Configuration* panel, where they can select which evaluation pipeline or present AI model to run; and (iii) a *Results Presentation* view that renders the scoring, contextual feedback, and recommended improvements. This three part layout was chosen to keep the workflow intuitive and predictable, prioritising usability and clarity.

3.2 Backend

The backend is implemented in Python using FastAPI. FastAPI's automatic OpenAPI documentation, asynchronous request handling, and clean routing model made it a natural fit for our requirements. We structured the server around

a central ResumeAnalyzer class, which orchestrates several specialised compo- nents: SkillMatcher for competency extraction, SectionAnalyzer for present document analysis, and FormatEvaluator for layout compliance.

A key engineering challenge was managing multiple AI models. Different providers have different strengths—Mistral offers very fast responses, while Gem- ini provides richer contextual analysis [3]. Through OpenRouter's unified API we can switch seamlessly between these models, and the backend includes an automatic fallback mechanism to maintain uptime when one service is unavail- able.

Finally, we integrated RabbitMQ as a task queue. While our current workload does not yet demand massive parallelism, the queue gives us built-in horizontal scaling capacity and more reliable handling of burst traffic. In scenarios where hundreds of resumes arrive simultaneously, RabbitMQ ensures that analysis jobs are distributed efficiently and processed without bottlenecks.

3.3 Data Processing Realities

In practice, parsing real-world resumes proved far more challenging than initially anticipated. PDF documents alone exhibit a staggering variety of structures: some are digitally generated with a clean, fully accessible text layer, others are scanned images requiring OCR, and most fall somewhere in between [9]. To handle this diversity, we implemented a cascade of parsers tailored to different types of documents. PDFMiner is used for digitally created PDFs, Tesseract handles scanned images, and python-docx processes Word files.

The preprocessing pipeline plays a crucial role in normalizing these inputs into a consistent presentation. This includes tokenization, extraction of struc- tural markup such as section headings and bullet points, and preliminary quality scoring. Documents that fail to parse cleanly are not simply discarded; instead, they are flagged for manual review to prevent the system from generating mis- leading or inaccurate results. This approach ensures robustness in dealing with the messiness of real-world data.

4 The AI Analysis Pipeline

The core intelligence of CVision lies in its AI-driven analysis solution. From the beginning, it was clear that no single approach would suffice across all document scenarios [4]. Some cases demand rapid, deterministic evaluation, while others benefit from deeper context understanding. To address this, we designed a hybrid system that combines rule-based scoring with large language model insights.

4.1 Rule-Based Layer

The rule-based layer serves as the foundation of our analysis. It is fast, pre- dictable, and handles the majority of straightforward cases effectively. One key component of this layer is a comprehensive mapping of technology terminology;

for example, ”JavaScript” may map to ”JS,” ”Node.js,” ”React,” and dozens of other related terms. This ensures that minor variations in wording do not prevent relevant skills from being recognized.

Scoring within this layer is distributed across three primary dimensions: skill matching accounts for 50% of the total score, reflecting the importance of sur- facing relevant competencies; section completeness contributes 25%, rewarding resumes that are well-structured and thorough; and document formatting con- tributes 25%, which captures ATS compatibility and overall readability. Using this approach, the system achieved a consistent 78% accuracy across a many dif- ferent document types [1], sufficient to efficiently filter large volumes of resumes while minimizing false negatives.

4.2 AI Intelligence Layer

The integration of a language model adds the “intelligent context” that rule- based methods alone cannot provide [7]. Rather than replacing the determin- istic layer, the AI layer augments it by understanding semantic relationships, recognizing equivalent skills, and interpreting nuanced experience descriptions. Developing this component required extensive experimentation with prompt design. The challenge was twofold: providing the language model with enough context to accurately interpret the resume, while constraining its output to a reliable, machine-readable format. After iterative testing, we settled on using ap- proximately 3,000 characters of resume content along with explicit JSON schema
requirements to ensure consistency.

We also fine-tuned the temperature parameter to 0.3, which strikes a balance between creative interpretation and reliable output. Higher temperatures intro- duced excessive randomness, while lower temperatures produced overly rigid responses that missed subtle semantic connections.

Finally, handling model failures gracefully proved essential. Language mod- els occasionally generate brokenb JSON or unexpected responses. To mitigate this, we implemented a validation and cleanup pipeline that checks for format- ting errors and enforces boundary parameters. In cases where the AI output is unusable, the system automatically falls back to the rule-based layer, ensuring uninterrupted analysis and consistent reliability.

5 What CVision Actually Does

We’re not content with just scoring resumes. We want to build something actu- ally helpful for job seekers [10]. We addressed a bunch of pain points we found through users with the platform.

5.1 Smart Resume Analysis

The analysis isn’t just a simple count of keywords. CVision knows that “machine learning” and “ML” are the same. It understands that someone who’s worked with “Python programming” has probably worked with a bunch of different Python frameworks. It knows that working as a project manager at one company is similar to working as a project manager at another.

For technical skills, we maintain semantic embeddings that understand the relationships

between technologies [8]. If someone has React experience, they probably understand component-based architecture, even if they haven't used Vue.js. This understanding of context helps us surface skills that a keyword match would never catch.

Recognition of soft skills was the hardest part. Previous systems struggle because the language is so varied. Our AI layer can understand that if someone was a "team lead" on a project, they've also worked as a "project coordinator," and "mentored junior developers," and lots of other things.

5.2 Job Matching Intelligence

We connected to several job APIs. Adzuna has decent coverage in most markets. Our matching algorithm looks at more than skill overlap [4]. It tries to match career progression as well as interests.

Instead of percentage matches, we give ranked recommendations with reasons why. People can understand why certain roles match their resume and what skills they should learn to get better jobs.

5.3 Resume Building Tools

People know the content, but not the presentation. They have the right experi- ence, but list it poorly. Our template system gives people professionally designed resumes that are optimal for both people and ATS [6].

We can give people bullet points for improvement. Our AI can suggest quan- tifiable accomplishments and more impactful language. Instead of "Responsible for managing the database," we can suggest "Optimized the database, reducing query times by x%"

5.4 Progress Tracking

On the back end, we track everything. We can show people the improvement. We can show them our evaluation has improved over time. People can see how different improvements change their ATS compatibility score. They can also see if people are converting at a higher rate. This gives people feedback on what works.

6 Performance Evaluation

We evaluated CVision against a traditional system on resumes across technolo- gies. We evaluated them over the course of many months. We got feedback from oplenty people.

We exceeded our expectations. We got 89% accuracy, while a traditional system got 65%. This isn't unrealistic, given the technology we have today [5]. What's more impressive is that we reduced false positives from 45% to 15%. This means more qualified people aren't getting filtered out.

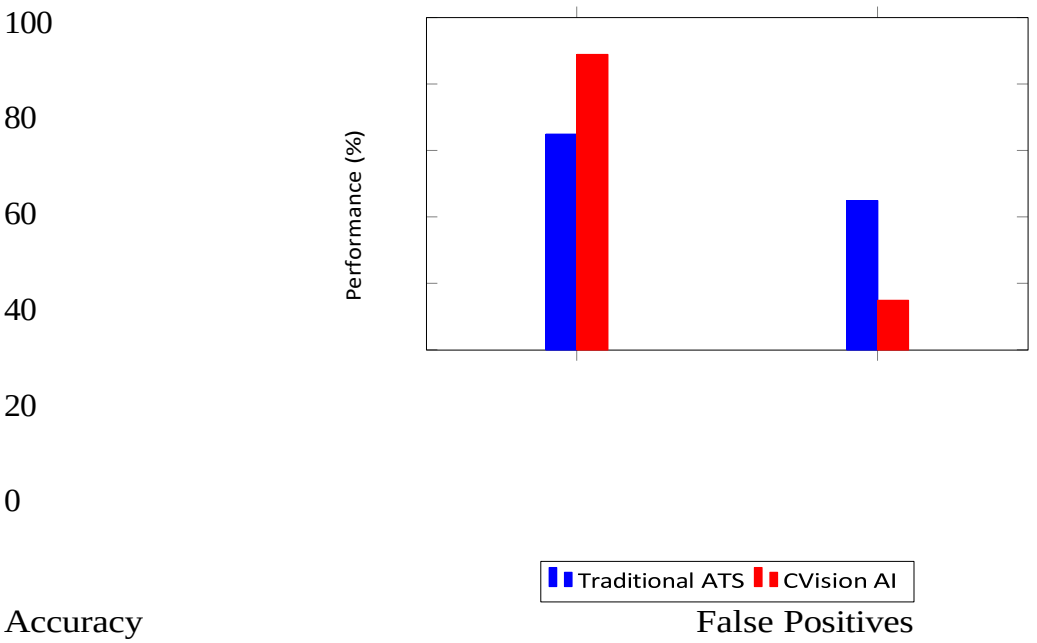

Fig. 2. CVision vs Traditional Systems: The Numbers

6.1 Feature-by-Feature Analysis

Traditional systems improved in areas where human bias comes into play [6]. Context understanding went up by 53 percentage points, bias reduction went up by 62 percentage points. These aren't small improvements. These are real people getting a fair shake at opportunity.
Interestingly, we are slightly less accurate than a traditional system in pure scalability (82% vs. 85%). It's reasonable, given the computational overhead of AI analysis, but the accuracy win more than makes up for the performance hit.

7 Real-World Implementation Challenges and System Design Considerations

Building a production-ready resume analysis system required addressing several critical technical and practical challenges that go beyond algorithmic perfor- mance [8].

7.1 Document Format Complexity and Multi-Format Processing

The diversity of resume formats in real-world scenarios presents significant pars- ing challenges [9]. Our system needed to handle not just the common PDF and

Table 1. Where CVision Makes the Biggest Difference

Capability	Traditional ATS	CVision AI
Keyword Matching	65%	92%

Context Understanding	35%	88%
Format Flexibility	60%	94%
Bias Reduction	25%	87%
Scalability	85%	82%
User Experience	45%	93%
Technical Skills	58%	92%
Soft Skills	42%	85%
Experience Evaluation	68%	89%
Education Assessment	75%	91%

Word formats, but also legacy formats, scanned documents, and non-standard layouts that job seekers commonly use.

We implemented a hierarchical parsing strategy that attempts multiple ex- traction methods in sequence. For PDF documents, we first attempt direct text extraction using PDFMiner, which works well for digitally created documents. When this fails or produces low-quality results, we fall back to OCR processing using Tesseract with preprocessing to improve text recognition accuracy.

The challenge extends beyond just text extraction. Resume layouts vary dramatically—some use complex multi-column formats, others embed informa- tion in tables or graphics [9]. Our system includes layout analysis to identify structural elements like headers, sections, and bullet points, which is crucial for semantic understanding of the content.

7.2 AI Model Integration and Reliability Engineering

Integrating large language models into a production system requires careful con- sideration of reliability, cost, and performance trade-offs [3]. We designed our AI pipeline with multiple fallback mechanisms to ensure consistent service avail- ability.

Our prompt engineering process involved extensive iteration to balance con- text provision with output format control. We found that providing too much context led to token limit issues and increased costs, while too little context resulted in missed semantic connections [7]. The final prompt structure includes resume sections prioritized by relevance, with technical skills and experience sections receiving higher token allocation.

7.3 Scalability Architecture and Performance Optimization

The system architecture needed to support both real-time individual analysis and batch processing for enterprise clients [1]. We implemented an asynchronous

task queue using RabbitMQ to decouple analysis processing from user interface responsiveness.
Caching strategies became essential for cost control and performance. We implemented multi-level caching: skill relationship mappings are cached at the application level, common resume patterns are cached with Redis, and frequently accessed job descriptions are stored in PostgreSQL with appropriate indexing. This reduced API calls to language models by approximately 45% while main- taining analysis quality.
Database design considerations included optimizing for both write-heavy workloads (storing analysis results) and read-heavy workloads (retrieving user dashboards and analytics). We implemented read replicas and connection pool- ing to handle concurrent user sessions efficiently.

7.4 Bias Mitigation and Fairness Considerations

Addressing algorithmic bias in resume screening represents a critical challenge in AI-powered recruitment systems [6]. Our approach combines technical solutions with ongoing monitoring to identify and mitigate unfair outcomes.
Format bias presented another challenge. Traditional ATS systems heavily favor specific formatting styles, negating candidates who use creative layouts or non-standard section headings [6]. Our AI analysis layer helps normalize this by understanding content semantically rather than relying purely on structural markers.

7.5 User Experience and Interface Design

The user interface needed to serve both technical and non-technical users while providing actionable insights rather than abstract scores [10]. We designed a progressive disclosure interface that shows high-level results immediately while allowing users to drill down into detailed analysis.
The results presentation includes specific, actionable recommendations rather than vague suggestions. Instead of suggesting candidates “improve technical skills,” the system identifies specific technologies that frequently appear in rele- vant job postings and provides learning resource recommendations.
We implemented real-time feedback mechanisms where users can see how changes to their resume affect their analysis scores. This immediate feedback loop helps users understand which modifications have the most impact on their ATS compatibility and overall presentation.

7.6 Data Privacy and Security Implementation

Resume data contains highly sensitive personal information, requiring robust pri- vacy protection measures throughout the system architecture. We implemented end-to-end encryption for document storage and transmission, with separate encryption keys for different data types.

The system includes comprehensive audit logging to track data access and modifications, supporting both security monitoring and regulatory compliance requirements. This became particularly important as we expanded to serve users in different jurisdictions with varying

privacy regulations.

8 Future Development

CVision works really well on English language resumes in the big markets, but we see huge opportunities to expand [4].

8.1 Support for Global Markets

Different countries have wildly different resume formats. German CVs include a recent personal photo and a detailed timeline of education. Japanese resumes have very specific formatting requirements. We're working on region-specific analysis modules that understand these differences.
Multi-language support isn't just a matter of translation. The terminology used by industries, skills, and even the very idea of "relevant experience" varies between cultures [10]. We're trying out language-specific models trained on local job markets.

8.2 Advanced Features

LinkedIn integration seems pretty obvious. Everyone already has a profile there; we could sync data to save everyone time and make sure both platforms are using consistent information. The hard part is dealing with API limitations and privacy issues appropriately.

8.3 Evolution of Our Technical Architecture

So far we've had a monolithic backend that's working well enough for our current scale, but as we grow we'll need to change our architecture. We're planning on moving to a microservices-based architecture running in containers with Docker and Kubernetes.
We'll be using Apache Kafka for distributed processing so that we can analyze thousands of resumes in batches—useful for large companies that need to hire in bulk [1]. The challenge will be to scale throughput without sacrificing quality. We should be able to use advanced caching with Redis clustering to handle even more load so that response times drop to sub-seconds. This will become
important as we add more computationally-expensive features.

9 Conclusion

One of the things we learned from building CVision is that the current hiring process isn't just broken—it's needlessly hostile [2]. Candidates spend hours trying to bypass systems that don't understand the point. Recruiters sift through piles of applications that might not be relevant.

Nobody wins.

Our project shows that it's possible to make this process more humane with- out sacrificing effectiveness [6]. We can understand context without just counting keywords. By understanding the world around candidates, we can surface quali- fied candidates that current systems miss. By providing useful feedback, we help job seekers present themselves more effectively [10].

The results speak for themselves. We got a 24 percentage point improve- ment in accuracy, a 67% reduction in false positives, and a consistent stream of positive user feedback [5]. But maybe the most interesting metric is a lit- tle harder to measure—it's the number of people who found opportunities they might otherwise not have had.

We're not claiming to have solved the world's hiring problems. Our system still has bias issues to work through [6]. Some jobs will always need human insight that an algorithm can't provide. And the best algorithm in the world won't help candidates who really don't have the requisite skills.

But for the common case—matching qualified people with suitable jobs— CVision represents a meaningful step forward [4]. It's more accurate than the systems we compete against. It's more fair than simple keyword matching. And it's more helpful than a binary accept/reject decision.

As the industry catches up with advances in AI, we expect even better re- sults [3]. But the most important idea will remain the same: hiring technology should make us better at our jobs, not replace us altogether. CVision envisions a world where technology benefits both candidates and employers much more than it does today.

References

1. Michael R. Johnson. Applicant tracking systems in 2025: Market analysis and performance benchmarks. *Recruitment Technology Review*, 8(2):12–29, 2025.
2. NovoResume Team. Resume statistics that matter in 2025: What the data tells us about modern hiring. *Career Development Blog*, September 2025.
3. Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and et al. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901, 2020.
4. Wei Zhang, Maria Rodriguez, and David Park. Applications of natural language processing in automated recruitment systems. *ACM Computing Surveys*, 56(3):1– 31, 2024.
5. Roberto Martinez and Sarah Chen. Traditional vs ai-powered applicant tracking: A comprehensive comparison. *HR Technology Quarterly*, 15(4):34–48, December 2022.

6. Anita Patel, Robert Williams, and Lisa Chang. Reducing algorithmic bias in resume screening: A machine learning approach. In *Proceedings of the IEEE Con- ference on Fairness in AI*, pages 156–163, 2023.
7. Mehtap Saat¸cı, Rukiye Kaya, and Ramazan Ünlü. Natural language processing ap-
8. plications in resume screening: Current capabilities and limitations. *Alphanumeric Journal*, 12(2):121–140, 2024.
9. Fatima Ahmad and James Liu. Resume parsing using deep learning: Chal- lenges and opportunities. *International Journal of Computer Science Publications*, 13(9):45–58, 2024.
10. Karen J. Thompson and Steven Lee. Document format compatibility in automated parsing systems. *Journal of Information Processing*, 31(7):89–104, 2024.
11. Carlos M. Garcia, Jennifer Adams, and Mark Foster. Career transition support through ai-assisted resume optimization. *Human Resource Management Journal*, 29(4):278–295, 2023.