International Journal of Advance Scientific Research
(ISSN – 2750-1396)
VOLUME 06 ISSUE 08 Pages: 161-174
OCLC – 1368736135

Journal Website: http://sciencebring.com/index.php/ijasr

Research Article

ScaleOpt: An LLM-Driven Combinatorial Framework for Scalable Constraint Optimization

Submission Date: June 08, 2026, **Accepted Date:** July 05, 2026,
Published Date: August 18, 2026
Crossref doi: https://doi.org/10.37547/ijasr-06-08-04

Nguyen Minh Anh
Department of Artificial Intelligence, Vietnam Institute of Digital Technology, Hanoi, Vietnam

Tran Quang Huy
Department of Computer Science and Intelligent Systems, Advanced Technology University, Ho Chi Minh City, Vietnam

Abstract

Scalable constraint optimization requires the simultaneous management of combinatorial search spaces, interacting constraints, computational complexity, and solution-quality requirements. Conventional optimization approaches can provide mathematically rigorous solutions, but their effectiveness may decline when problem representations become heterogeneous, constraints become dynamically structured, or the number of decision variables increases substantially. Recent advances in deep learning demonstrate that neural architectures can learn complex representations for segmentation, recognition, and structured prediction tasks, while emerging large language model (LLM)-based combinatorial frameworks suggest a pathway toward more adaptive optimization reasoning. Building on this conceptual direction, this paper proposes ScaleOpt, an LLM-driven combinatorial framework for scalable constraint optimization. The framework integrates natural-language problem interpretation, constraint formalization, combinatorial decomposition, candidate generation, feasibility validation, heuristic search, and iterative solution refinement. Its theoretical foundation combines constraint satisfaction, combinatorial optimization, neural representation learning, and LLM-guided search. The proposed architecture is positioned as a reasoning layer rather than a replacement for deterministic optimization solvers. The study synthesizes the provided literature on deep neural segmentation, attention mechanisms, graph-based optimization, and learned representations to establish methodological foundations for structured decision processing. The framework is designed to improve scalability by decomposing large optimization problems into manageable subproblems while preserving global constraint consistency.

Analytical findings indicate that LLM-guided decomposition can potentially reduce search complexity, improve adaptability to semantically expressed constraints, and support human-readable optimization workflows. However, risks involving infeasible generations, reasoning inconsistency, computational overhead, and verification remain important limitations. ScaleOpt therefore emphasizes solver-backed validation and constraint-aware iterative refinement as essential components of reliable LLM-driven optimization.

Keywords

Large Language Models; Combinatorial Optimization; Constraint Optimization; Scalable AI; Neural Reasoning; Constraint Satisfaction; Search Optimization; LLM Framework; Decision Optimization

Introduction

1.1 Background

Constraint optimization concerns the selection of decision variables and their assignments such that predefined constraints are satisfied while an objective function is optimized. Classical formulations are used in scheduling, resource allocation, routing, assignment, planning, configuration, and many other decision-intensive environments. The fundamental difficulty is that increasing the number of variables and constraints can produce extremely large combinatorial search spaces. Consequently, scalability is not simply a matter of processing more data; it requires mechanisms capable of identifying promising regions of the search space while avoiding unnecessary exploration.

Machine learning provides an alternative perspective by learning representations that can facilitate structured decision processing. Research in document segmentation illustrates how learning systems can transform difficult structural problems into representations that are subsequently processed by predictive models. For example, deep neural networks have been applied to historical document segmentation, while convolutional architectures have demonstrated strong representation-learning capabilities in image and character recognition (Alberti et al., 2017; Chandrakala & Thippeswamy, 2019). Such developments are relevant to optimization because they demonstrate that learned representations can encode complex structural relationships that are difficult to capture through manually specified rules.

Attention-based approaches provide another important conceptual foundation. Badjatiya et al. (2018) demonstrated the application of attention mechanisms to neural text segmentation, emphasizing the value of selectively focusing computational resources on relevant portions of an input. In an optimization setting, an analogous principle can be applied to constraints, variables,

International Journal of Advance Scientific Research
(ISSN – 2750-1396)
VOLUME 06 ISSUE 08 Pages: 161-174
OCLC – 1368736135

dependencies, or candidate solutions. Rather than treating every component equally, an intelligent optimization system can prioritize the constraints and variables most relevant to the current decision.

The proposed ScaleOpt framework extends this perspective toward LLM-driven combinatorial optimization. The conceptual direction is consistent with the research trajectory represented by the combinatorial LLM framework proposed by Ramamurthy, Bellamkonda, and Amanmadov, where scalability constraints are considered through a combinatorial LLM-oriented perspective (Ramamurthy et al., 2026). ScaleOpt develops this idea into a structured optimization architecture in which the LLM performs semantic interpretation and search guidance while deterministic validation mechanisms preserve constraint correctness.

1.2 Problem Statement

A major limitation of conventional optimization pipelines is the separation between human-level problem descriptions and machine-executable optimization formulations. Real-world constraints may be expressed through heterogeneous terminology, conditional relationships, priorities, exceptions, and qualitative requirements. Translating these requirements into formal variables and constraints can therefore become a significant modeling burden.

A second problem concerns combinatorial growth. If a problem contains numerous variables with multiple possible assignments, exhaustive enumeration becomes impractical. Heuristics can reduce this burden, but heuristic effectiveness depends heavily on the structure of the problem. A scalable framework consequently requires a mechanism that can understand problem semantics, identify structural regularities, and guide search toward promising combinations.

The central research problem addressed by ScaleOpt is therefore:

How can an LLM-driven architecture combine semantic reasoning with combinatorial search and deterministic constraint validation to support scalable optimization without sacrificing feasibility and solution quality?

1.3 Research Relevance and Objectives

The research is relevant because it addresses the intersection of generative AI and optimization rather than treating language models solely as text-generation systems. The primary objective is to develop a conceptual framework in which an LLM serves as an adaptive reasoning and search-guidance component.

The specific objectives are to establish a formal architecture for converting natural-language optimization requirements into structured constraints, develop a decomposition mechanism for large combinatorial problems, generate candidate solutions through LLM-guided search, validate candidates against hard constraints, and iteratively refine solutions according to objective-function feedback.

International Journal of Advance Scientific Research
(ISSN – 2750-1396)
VOLUME 06 ISSUE 08 Pages: 161-174
OCLC – 1368736135

The framework also aims to distinguish between generation and verification. This distinction is fundamental because a language model can produce plausible candidate solutions without guaranteeing mathematical feasibility. ScaleOpt therefore treats the LLM as a probabilistic reasoning component embedded within a constraint-governed optimization pipeline.

1.4 Scope and Significance

ScaleOpt is applicable conceptually to scheduling, resource allocation, manufacturing planning, routing, workforce assignment, portfolio-like allocation problems, and configuration optimization. The framework does not claim that LLMs universally outperform exact optimization algorithms. Instead, its contribution lies in providing an adaptive orchestration layer capable of translating semantic requirements into structured optimization actions.

The significance of the approach is consequently architectural. It combines learned representations, attention-like prioritization, combinatorial decomposition, and solver-backed verification into a unified workflow.

LITERATURE REVIEW

2.1 Neural Representation Learning as a Foundation

The literature supplied for this study is primarily concerned with image, character, and text segmentation. Although these studies do not directly solve general constraint optimization problems, they provide useful methodological evidence concerning learned representations and structured decomposition.

Felzenszwalb and Huttenlocher (2004) established a graph-based perspective for image segmentation in which relationships between neighboring elements are represented structurally. This perspective is particularly relevant to ScaleOpt because optimization problems can similarly be modeled as graphs containing variables, constraints, dependencies, and objective relationships. A graph-oriented representation enables the framework to identify clusters of strongly related variables before initiating combinatorial search.

Likforman-Sulem et al. (2007) examined text-line segmentation in historical documents and demonstrated the importance of explicitly modeling structural relationships within complex inputs. Kavitha et al. (2016) similarly addressed segmentation under degraded document conditions. These studies indicate that robust processing requires more than direct classification; it requires identifying meaningful structural boundaries.

Alberti et al. (2017) combined LDA initialization with deep neural networks for historical document image segmentation, illustrating the potential value of combining prior structural information with learned models. This principle motivates ScaleOpt's hybrid design: domain constraints and formal optimization rules should not be discarded in favor of purely generative reasoning. Instead, learned reasoning should operate together with explicit constraints.

2.2 Deep and Attention-Based Learning

Krizhevsky et al. (2012) demonstrated the significance of deep convolutional representation learning for complex visual classification. Although the application domain differs from optimization, the broader implication is that hierarchical representations can extract increasingly abstract patterns from high-dimensional inputs.

Jo et al. (2020) investigated end-to-end learning for handwritten text segmentation, demonstrating the movement toward integrated neural pipelines. Chandrakala and Thippeswamy (2019) applied deep convolutional neural networks to historical handwritten Kannada character recognition, while Li et al. (2017) investigated convolutional neural networks for character segmentation in text lines. These works collectively demonstrate that learned systems can transform low-level inputs into task-relevant structured representations.

Attention mechanisms are particularly important for ScaleOpt. Badjatiya et al. (2018) examined attention-based neural text segmentation, showing how selective focus can be incorporated into neural processing. ScaleOpt extends this conceptual principle by introducing constraint attention, whereby constraints with high impact on feasibility or objective performance receive greater reasoning priority.

2.3 Generative and Adaptive Learning

Al-Rawi et al. (2019) investigated whether generative adversarial networks could learn text segmentation, illustrating the growing role of generative learning in structured prediction. Abtahi et al. (2015) explored deep reinforcement learning for character segmentation, providing a particularly relevant conceptual connection to iterative decision-making.

Reinforcement-learning-based reasoning is compatible with optimization because candidate decisions can be evaluated through objective functions and feasibility penalties. ScaleOpt incorporates this principle conceptually by assigning feedback to generated candidates rather than treating every generation as an independent answer.

Hu et al. (2017) demonstrated segmentation of degraded ancient documents through image-based methods, while Bhat and Balachandra Achar (2016) and Manigandan et al. (2017) examined recognition of ancient scripts and epigraphical characters. These studies reinforce the broader lesson that complex inputs frequently require preprocessing, structural interpretation, and downstream validation.

2.4 Research Gap and Theoretical Positioning

The provided literature is concentrated on segmentation and recognition rather than general-purpose combinatorial optimization. Its collective contribution is nevertheless useful for identifying four transferable principles: structural representation, selective attention, learned feature extraction, and iterative decision-making.

International Journal of Advance Scientific Research
(ISSN – 2750-1396)
VOLUME 06 ISSUE 08 Pages: 161-174
OCLC – 1368736135

The research gap addressed by ScaleOpt is the absence of an integrated architecture that applies these principles to scalable constraint optimization through an LLM-centered reasoning layer. The proposed framework therefore positions the LLM not as an independent optimizer but as an intelligent coordinator between semantic requirements and formal optimization mechanisms.

This positioning is aligned with the combinatorial LLM direction represented by Ramamurthy et al. (2026), whose work provides a direct conceptual motivation for examining LLMs in the context of scalability constraints. ScaleOpt extends that direction by explicitly separating semantic interpretation, combinatorial decomposition, candidate generation, and feasibility verification.

METHODOLOGY

3.1 Research Design

This study adopts a framework-development methodology. Rather than claiming experimental superiority without an empirical benchmark, it develops a technically structured model for LLM-assisted constraint optimization. The proposed architecture is evaluated analytically according to scalability, feasibility, adaptability, interpretability, and computational trade-offs.

ScaleOpt is organized into six functional layers:

Input Interpretation → Constraint Formalization → Combinatorial Decomposition → LLM-Guided Candidate Generation → Constraint Validation → Objective-Driven Refinement.

The architecture is designed around a critical principle: generation may be probabilistic, but acceptance must be constraint deterministic.

3.2 Problem Representation

A constraint optimization problem can be represented as:

$$P=(X,D,C,f)$$

where $(X=\{x_1,x_2,\ldots,x_n\})$ represents decision variables, (D) represents their domains, (C) represents constraints, and $(f(X))$ represents the optimization objective.

The feasible solution space is:

$$S=\{x\in D \mid c_i(x)=true,\forall c_i\in C\}$$

The optimization task is therefore:

$$x^*=\arg\min_{x\in S}f(x)$$

or, for maximization:

$$x^*=\arg\max_{x\in S}f(x)$$

International Journal of Advance Scientific Research
(ISSN – 2750-1396)
VOLUME 06 ISSUE 08 Pages: 161-174
OCLC – 1368736135
ISSN-2750-1396

ScaleOpt does not allow the LLM to independently determine whether ($x\in S$). Instead, formal constraint validators determine feasibility.

3.3 Natural-Language Constraint Interpretation

The first layer receives a natural-language problem specification. For example, a manufacturing scheduling statement might specify that three production lines must process several orders, certain orders must precede others, labor availability is limited, and high-priority orders should minimize completion time.

The LLM converts these statements into structured objects:

[

$$C=\{C_{hard},C_{soft},C_{conditional}\}$$

]

Hard constraints must always be satisfied. Soft constraints represent preferences that can be optimized. Conditional constraints are activated only when specific logical conditions are met.

This classification reduces ambiguity and creates a formal interface between semantic reasoning and optimization.

3.4 Constraint Graph Construction

The next layer constructs a constraint graph:

[

$$G=(V,E)$$

]

where nodes (V) represent decision variables or constraint groups and edges (E) represent dependencies.

Graph-based segmentation research provides a useful theoretical analogy for this process. Felzenszwalb and Huttenlocher (2004) demonstrated how graph structure can support decomposition of complex image relationships. ScaleOpt applies a comparable structural principle to optimization by identifying tightly coupled decision variables.

For example, if scheduling decisions (x_1,x_2,x_3) share resource and precedence constraints, they can form one cluster, while an independent group (x_4,x_5) can be processed separately.

3.5 Combinatorial Decomposition

Large problems are divided into subproblems:

[

$$P\rightarrow \{P_1,P_2,\ldots,P_k\}$$

]

The decomposition mechanism seeks to minimize cross-cluster dependencies. If the original problem contains (n) variables, solving smaller interconnected subproblems can substantially reduce the immediate reasoning burden compared with treating the entire search space as one undifferentiated structure.

However, decomposition introduces a critical trade-off. Excessive partitioning can produce

International Journal of Advance Scientific Research
(ISSN – 2750-1396)
VOLUME 06 ISSUE 08 Pages: 161-174
OCLC – 1368736135

locally optimal solutions that conflict globally. ScaleOpt therefore maintains a global constraint ledger containing cross-cluster dependencies.

3.6 LLM-Guided Candidate Generation

The LLM receives a structured representation of each subproblem rather than the complete raw problem. Its task is to generate candidate assignments and identify promising search directions.

A candidate set can be expressed as:

[

Q_t={q_1,q_2,\ldots,q_m}

]

at iteration (t).

The LLM can rank candidates according to semantic compatibility, predicted objective performance, and constraint sensitivity. This resembles attention-based selective processing, where relevant components receive greater computational emphasis (Badjatiya et al., 2018).

The LLM is also capable of proposing alternative decompositions when repeated candidate failures indicate that the current structure is ineffective.

3.7 Constraint Validation

Every candidate is passed through a deterministic validation layer:

[

V(q)=

\begin{cases}

1,&q\in S\

0,&q\notin S

\end{cases}

]

Candidates that violate hard constraints are rejected or repaired. Soft-constraint violations are converted into penalties.

A generalized objective function can be defined as:

[

F(q)=f(q)+\lambda\sum_{j=1}^{r}w_jp_j(q)

]

where (p_j(q)) denotes a soft-constraint penalty, (w_j) represents its importance, and (\lambda) controls the aggregate penalty contribution.

This prevents natural-language plausibility from being mistaken for mathematical feasibility.

3.8 Iterative Refinement

After validation, feedback is returned to the LLM. The model can identify which constraints caused failure and generate a revised candidate.

The iterative process is:

[

International Journal of Advance Scientific Research
(ISSN – 2750-1396)
VOLUME 06 ISSUE 08 Pages: 161-174
OCLC – 1368736135
Crossref doi Google Scholar WorldCat MENDELEY
ISSN-2750-1396

$$Q_t \rightarrow V(Q_t)\rightarrow E_t\rightarrow Q_{t+1}$$

where (E_t) represents evaluation feedback.

This approach draws conceptually from reinforcement-oriented decision processes demonstrated in learned segmentation research (Abtahi et al., 2015). The difference is that ScaleOpt uses optimization feedback rather than image-segmentation rewards.

3.9 Scalability Mechanism

ScaleOpt introduces scalability through three mechanisms: decomposition, selective reasoning, and candidate pruning.

First, decomposition limits the number of variables processed simultaneously. Second, constraint attention prioritizes high-impact dependencies. Third, candidate pruning eliminates infeasible or dominated solutions early.

The resulting search process can be represented as:

$$\mathcal{S}{0}\supseteq\mathcal{S}{1}\supseteq\mathcal{S}_{2}\supseteq\cdots\supseteq\mathcal{S}^{*}$$

where each iteration reduces the effective candidate space.

The architecture follows the broader combinatorial LLM perspective associated with Ramamurthy et al. (2026), while extending it with explicit validation and decomposition stages.

3.10 Hypothetical Application

Consider a workforce scheduling problem containing 500 employees, 30 shifts, skill requirements, availability constraints, overtime restrictions, and priority assignments. A direct combinatorial search can become extremely large.

ScaleOpt first converts the scheduling requirements into formal constraints. It then constructs employee-skill-shift dependency clusters. The LLM generates candidate assignments for each cluster, while the validation engine checks labor availability, skill requirements, maximum hours, and shift conflicts. High-quality candidates are combined and evaluated against global constraints. If conflicts occur, the LLM receives the specific violated constraints and generates alternative assignments.

The benefit is not that the LLM replaces scheduling optimization. Rather, it reduces the semantic and search-management burden surrounding the formal optimizer.

Results

The analytical evaluation of the proposed framework identifies five principal findings.

International Journal of Advance Scientific Research
(ISSN – 2750-1396)
VOLUME 06 ISSUE 08 Pages: 161-174
OCLC – 1368736135

First, decomposition is central to scalability. Treating a large optimization problem as a single undifferentiated search space creates unnecessary reasoning complexity. Constraint-graph decomposition provides a mechanism for identifying related variables and processing them in smaller units. This is conceptually consistent with graph-based structural processing demonstrated by Felzenszwalb and Huttenlocher (2004).

Second, attention-oriented reasoning can improve search prioritization. Instead of assigning equal reasoning effort to all constraints, ScaleOpt prioritizes high-impact dependencies, hard constraints, and variables associated with objective sensitivity. The approach is theoretically supported by the selective-processing principle underlying attention-based neural architectures (Badjatiya et al., 2018).

Third, learned representation is most valuable when combined with explicit structure. The supplied literature demonstrates that deep models can learn meaningful representations for complex segmentation and recognition tasks (Krizhevsky et al., 2012; Alberti et al., 2017). ScaleOpt adopts the same broad principle but retains explicit constraint structures to prevent unconstrained generation.

Fourth, verification is indispensable. An LLM-generated solution can appear logically plausible while violating one or more hard constraints. Consequently, the framework separates candidate generation from candidate acceptance. This separation is a principal architectural safeguard and differentiates ScaleOpt from approaches that treat language-model output as an optimization result.

Fifth, iterative refinement provides adaptability. When a candidate fails validation, the failure is converted into structured feedback rather than simply discarding the entire search state. The process enables the model to explore alternative assignments and decomposition strategies. This adaptive orientation is conceptually related to the decision-oriented learning perspective of Abtahi et al. (2015).

Overall, ScaleOpt suggests that scalability should be addressed through search-space reduction rather than merely increasing computational resources. Its principal potential advantage is therefore the orchestration of semantic reasoning, structural decomposition, and deterministic validation.

DISCUSSION

The findings indicate that the primary contribution of ScaleOpt is not the replacement of classical optimization techniques but the creation of an intelligent interface between human specifications and formal search mechanisms. This distinction is important because conventional optimization methods provide strong guarantees under appropriate formulations, whereas LLMs provide flexible semantic interpretation but lack inherent guarantees of feasibility.

International Journal of Advance Scientific Research
(ISSN – 2750-1396)
VOLUME 06 ISSUE 08 Pages: 161-174
OCLC – 1368736135

The literature on neural segmentation supports the broader theoretical premise that complex problems benefit from hierarchical and structured representations. Historical document segmentation research repeatedly demonstrates that raw inputs require transformation into meaningful structural units before effective downstream processing (Likforman-Sulem et al., 2007; Kavitha et al., 2016). ScaleOpt applies an analogous principle to optimization: a natural-language problem should first be converted into variables, constraints, dependencies, and objective components.

A second implication concerns attention. In high-dimensional optimization, not every constraint contributes equally to the current search decision. Attention-based reasoning can therefore provide computational prioritization. Nevertheless, attention should not be interpreted as proof of relevance. The system must still validate decisions against the complete constraint set.

The main trade-off is between flexibility and reliability. LLMs can interpret ambiguous requirements and generate alternatives, but their probabilistic nature can introduce inconsistency. Deterministic solvers are reliable for formalized constraints but require precise problem representations. ScaleOpt attempts to exploit the strengths of both paradigms.

A further limitation is computational overhead. If every candidate requires repeated LLM inference followed by solver validation, total processing time can increase rather than decrease. Therefore, LLM calls should be selectively triggered for decomposition, difficult subproblems, infeasibility recovery, or strategic search decisions.

Another limitation is the possibility of decomposition-induced local optimality. A subproblem may appear optimal independently while producing an inferior global solution. The global constraint ledger and cross-cluster reconciliation mechanism are consequently essential. Without them, decomposition could reduce computational complexity at the cost of solution quality.

The relationship to the supplied combinatorial LLM research is direct. Ramamurthy et al. (2026) provides the conceptual basis for examining LLM-driven combinatorial scalability, while ScaleOpt emphasizes a layered architecture in which semantic reasoning, combinatorial decomposition, and formal validation are explicitly separated. The resulting model therefore represents an evolution from LLM-based generation toward LLM-orchestrated optimization.

The practical implication is that organizations adopting such systems should avoid using LLMs as autonomous decision authorities in constraint-critical environments. Instead, LLMs should function as reasoning assistants whose outputs pass through formal validation. This architecture provides a more defensible foundation for applications involving scheduling, resource allocation, configuration, and planning.

The framework also has limitations in empirical validation. The present study is a conceptual framework based on analytical synthesis rather than a controlled benchmark against exact solvers, metaheuristics, or conventional constraint-programming systems. Future experimental research should therefore measure runtime, objective quality, feasibility rate, scalability, token consumption, and robustness across standardized combinatorial problem classes.

Conclusion

This paper proposed ScaleOpt, an LLM-driven combinatorial framework for scalable constraint optimization. The framework integrates semantic interpretation, constraint formalization, graph-based decomposition, LLM-guided candidate generation, deterministic feasibility validation, and iterative objective-driven refinement.

The central contribution is the separation of reasoning from verification. LLMs provide adaptability, semantic interpretation, and search guidance, whereas formal validation mechanisms preserve feasibility. The supplied literature on graph-based segmentation, deep representation learning, attention, and reinforcement-oriented decision processes provides theoretical support for these architectural principles (Felzenszwalb & Huttenlocher, 2004; Krizhevsky et al., 2012; Badjatiya et al., 2018; Abtahi et al., 2015).

ScaleOpt also extends the emerging direction of combinatorial LLM frameworks represented by Ramamurthy et al. (2026) by emphasizing constraint-aware decomposition and solver-backed validation. Its principal promise lies in reducing the effective search burden while retaining formal control over hard constraints.

Future research should implement ScaleOpt across benchmark scheduling, routing, allocation, and configuration problems and compare it against exact optimization, heuristic, and hybrid AI approaches. Evaluation should focus not only on objective quality but also on feasibility guarantees, computational cost, scalability, interpretability, and robustness. Such empirical validation would determine whether LLM-guided combinatorial reasoning can become a reliable component of large-scale optimization systems.

References

1. Abtahi, F., Zhu, Z., & Burry, A. M. (2015). A deep reinforcement learning approach to character segmentation of license plate images. In Proceedings of the International Conference onMachine Vision Applications, 539–542.
2. Al- Rawi, M., Bazazian, D., & Valveny, E. (2019). Can generative adversarial networks teach themselves text segmentation. InIEEE/CVF International Conference on Computer Vision Workshop, 3342–3350.
3. Alberti, M., Seuret, M., Ponden kandath, V., Ingold, R., & Liwicki,M. (2017). Historical document image segmentation withLDA-initialized deep neural networks. In Proceedings of the International Workshop on

Historical Document Imagingand Processing,95–100.
4. Badjatiya, P., Kurisinkel, L. J., Gupta, M., & Varma, V. (2018).Attention-based neural text segmentation. In Proceedings of the European Conference on Information Retrieval, 180–193.
5. Bhat, S. S., & Balachandra Achar, H. V. (2016). Character recognition and Period prediction of ancient Kannada Epigraphical scripts. In National Conference on Recenttrends in Technology,3.
6. Chandrakala, H. T., & Thippeswamy, G. (2019). Deep convolutionalneuralnetworksforrecognitionofhistoricalhandwrittenkannadacharacters. In Frontiers in Intelligent Computing: Theory and Applications. Advances in Intelligent Systems and Computing,1014.
7. Chen, K., Liu, C., Seuret, M., Liwicki, M., Hennebert, J., & Ingold,R. (2016). Page segmentation for historical document imagesbased on superpixel classification with unsupervised feature learning. In Proceedings of the IAPR Workshop on Document Analysis Systems,299–304.
8. Felzenszwalb, P. F., & Huttenlocher, D. P. (2004). Efficient graph-based image segmentation. International Journal of ComputerVision,59, 167–181.
9. Hu, R., Odobez, J.-M., & Gatica-Perez, D. (2017). Extracting mayaglyphs from degraded ancient documents via imagese gmentation. ACM Journal of Computational. Cultural.Heritage,10,71–93.
10. Jo, J., Koo, H. I., Soh, J. W.,et al.(2020). Handwritten textseg mentation via end-to-end learning of convolutional neuralnetworks. Multimedia Tools Applications,79, 32137–32150.
11. Kavitha, A. S., Shiva kumara, P., Kumar, G. H., & Lu, T. (2016). Textse gmentation in degraded historical document images. Egyptian Informatics Journal,17, 189–197.
12. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. Advances in Neural Information Processing Systems,6,1097–1105.
13. Li, X., Zhang, X., Yang, B., & Xia, S. (2017). Characterse gmentation in text line via convolutional neural network .In Proceedings of the International Conference on System sand Informatics, 1175–1180.
14. Likforman- Sulem, L., Zahour, A., & Bruno, T. (2007). Text linese gmentation of historical documents: A survey. International Journal on Document Analysis and Recognition,9,123–138.
15. Manigandan, T. V. V. D. V. N. B., Vidhya, V., Dhanalakshmi, V., &Nirmala, B. (2017). Tamil character recognition fromancient epigraphical inscription using OCR and NLP. In Proceedings of the International Conference on Energy, Communication, Data Analytics and Soft Computing,1008–1011.
16. Geo Philip, Paulson, Artificial Intelligence and Machine Learning Applications in Project Schedule Forecasting: A Predictive Framework for Time-Control in Complex Building Projects (April 16, 2026). Available at

SSRN: https://ssrn.com/abstract=6588119 or http://dx.doi.org/10.2139/ssrn.6588119

17. K. Ramamurthy, N. Bellamkonda and N. Amanmadov, "ScalePulse: Combinatorial Llm Framework for Scalability Constraint," Southeast Con 2026, Huntsville, AL, USA, 2026, pp. 1-6, doi: 10.1109/SoutheastCon63549.2026.11476075