A Vision-Language Model Agent for Building Code Compliance

Zeyang Zheng
yuq8cp@virginia.edu
University of Virginia
Charlottesville, Virginia, USA

Seyed Hamidreza Nabaei
fgx9eq@virginia.edu
University of Virginia
Charlottesville, Virginia, USA

Somayeh Asadi
rkn3gr@virginia.edu
University of Virginia
Charlottesville, Virginia, USA

Brad Campbell
bradjc@virginia.edu
University of Virginia
Charlottesville, Virginia, USA

Arsalan Heydarian
ah6rx@virginia.edu
University of Virginia
Charlottesville, Virginia, USA

Abstract

Traditional methods for ensuring building code compliance often demand substantial time and resources and are prone to human error, leading to inconsistent evaluations of critical residential systems. Such inconsistencies can result in overlooked safety hazards and costly future repairs. To address these challenges, this paper introduces an innovative Vision-Language Model (VLM) agent specifically designed for building code compliance. The proposed agent combines advanced reasoning and action capabilities with specialized tools. It leverages a knowledge base comprising key building codes, including the International Residential Code (IRC) and the International Plumbing Code (IPC), and employs Retrieval-Augmented Generation (RAG) to identify relevant standards tailored to specific compliance requirements. An interactive interface enables users to submit both images and text, which the agent systematically analyzes. The VLM agent detects critical components, such as P-traps, and retrieves corresponding building code references. The system then generates a comprehensive report summarizing identified issues, assessing their severity, and citing relevant code sections. We use four distinct building components from real home inspection reports to evaluate the system's performance. The VLM agent achieves an average 96.25% similarity with the human-created inspection report. This research demonstrates a practical application of VLM agents, significantly enhancing the accuracy, accessibility, and reliability of building code compliance processes.

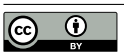
CCS Concepts

• **Computing methodologies** → **Artificial intelligence**.

Keywords

Vision-Language Model, Agent System, Smart Building, Smart City, Cyber-Physical Systems

ACM Reference Format:
Zeyang Zheng, Seyed Hamidreza Nabaei, Somayeh Asadi, Brad Campbell, and Arsalan Heydarian. 2025. A Vision-Language Model Agent for Building Code Compliance. In *The 12th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation (BUILDSYS '25), November 19–21, 2025, Golden, CO, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3736425.3770118



BUILDSYS '25, Golden, CO, USA

ACM ISBN 979-8-4007-1945-5/25/11
https://doi.org/10.1145/3736425.3770118

1 Introduction

Building code compliance is critical for ensuring safe and healthy living environments. However, code violations are often missed or ignored because they are not immediately visible, the issue developed well after the last inspection, or the building code was updated. Ensuring code compliance presents significant challenges, requiring specialized knowledge, considerable time, and the involvement of costly experts. For instance, a single manual inspection averages three hours to complete and can cost more than $500, all while being prone to human error as inspectors must interpret complex, evolving codebooks. As a result, compliance is frequently neglected [15], leading to building deterioration, increased problems, and negative impacts on residents' quality of life and well-being. Moreover, the home inspection industry faces a significant demographic challenge, as the majority of inspectors are aging, with a shrinking pool of new entrants. Additionally, despite recommendations to conduct thorough inspections at least once every two years, many property owners neglect regular inspections due to cost, accessibility, and awareness constraints.

As current methods for verifying compliance are largely manual, labor-intensive, and rely heavily on expert interpretation and physical inspections [2], this process is subjective and difficult to scale. Automation opens new possibilities for streamlining complex processes such as building inspections. In particular, advances in artificial intelligence (AI) have enabled the translation of visual semantic relationships and spatial configurations into structured text-based formats, which can be systematically compared and verified against regulatory standards.

Recent developments in computer vision have demonstrated significant potential for partially addressing these informational gaps in smart building applications. For instance, Pérez et al. [13] employed convolutional neural networks (CNNs) to successfully detect and localize common building defects such as mold, stains, and structural deterioration. Building on this, research has scaled to the urban level, where Gouveia et al. [5] utilized CNN-based models to classify entire buildings using Google Street View images, showcasing the potential for large-scale building stock analysis. To further accelerate progress in this domain, Kottari et al. [6] introduced a crucial public benchmark dataset covering six common types of building defects. While these studies validate the use of

AI for visual identification, they primarily focus on classification and detection tasks, stopping short of interpreting these findings within a complex regulatory framework. Similarly, Mathur et al. [8] developed an autonomous pipeline using UAVs specifically for detecting cracks on high-rise building façades. Moreover, much of this work has focused on building exteriors, leaving the complex and highly varied interior environments—where occupants spend the vast majority of their time—comparatively under-explored.

The emergence of large language models (LLMs) and vision-language models (VLMs) highlights their potential to automate building inspections by providing contextual, spatial, and ambient knowledge that traditional methods often fail to capture. These models advance semantic and spatial understanding, enabling recognition of subtle relationships among building components. Radford et al. [14] proposed CLIP (Contrastive Language-Image Pre-Training), a VLM pre-trained on 400 million image-text pairs that demonstrates robust zero-shot capabilities. Li et al. [7] introduced BLIP-2, achieving high performance on vision-language tasks with significantly fewer trainable parameters.

Beyond passive analysis, LLM agent systems also show promise for complex reasoning tasks. Wei et al. [17] introduced Chain-of-Thought (COT) prompting, enhancing LLM reasoning through intermediate steps. Yao et al. [18] developed ReAct, combining reasoning and tool interaction dynamically. Niu et al. [11] created ScreenAgent, a VLM agent capable of interacting with graphical user interfaces effectively.

Despite these parallel advancements in semantic building models, computer vision, and LLM agents, to our knowledge, no existing study has explored VLM agents for building code compliance. This paper addresses critical research gaps: (1) How can a VLM agent streamline and enhance building code compliance? (2) What categories of building codes should be included in the knowledge base for retrieval-augmented generation (RAG)?

To answer these questions, we introduce an innovative VLM agent system for building code compliance. Our primary contribution lies in constructing a knowledge base of building codes and developing a VLM agent system that leverages reasoning and specialized tools to retrieve relevant information. Users provide both images and text as inputs, and the agent analyzes them to generate comprehensive compliance reports.

2 Methodology

We introduce a VLM agent system designed to evaluate building code compliance. The overview of our approach is in Figure 2. The user captures image data and creates an instruction for the system. The VLM agent interprets the input, and queries the knowledge base for building codes relevant to the input data. Then, it generates an inspection report by evaluating if the relevant building codes are met in the input image.

2.1 Inspection Report Knowledge Base

Retrieval-Augmented Generation (RAG) plays a crucial role in addressing the limitations of LLMs, including hallucinations and outdated internal knowledge [4]. RAG enables an LLM to retrieve relevant information from an external knowledge base based on an

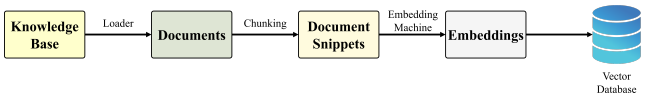


Figure 1: Overview of RAG Setup.

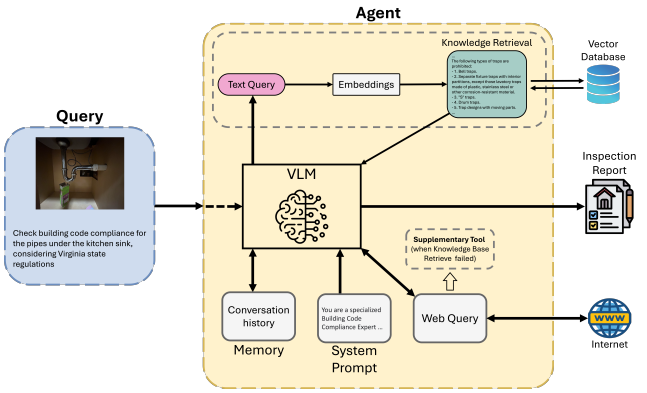


Figure 2: Framework of VLM Agent System.

input query. It then combines the query with the retrieved information before passing it to the LLM, leading to more accurate and contextually relevant outputs. Through RAG, an LLM can access external knowledge that may be up to date and not included in its original training data. This external information can be sensitive or difficult to obtain for direct model training. The RAG setup is illustrated in Figure 1.

To build our knowledge base, we collected the International Residential Code (IRC) 2021, International Plumbing Code (IPC) 2021, and Virginia Residential Code 2021. The Virginia Residential Code adopts the IRC 2021 with amendments and represents the current code in Virginia. For use with building code compliance, our knowledge base contains building codes that are the rules used to evaluate input text and images for compliance. We convert these documents from PDF to Markdown format using Docling [16] and segment the documents into smaller chunks. Text chunks are then transformed into vector representations and stored in a vector database based on the BGE M3-Embedding model [3].

When a text query is created by the agent, it is also converted into vectors using the same embedding model. A similarity search is performed to retrieve the most relevant information from the database. Finally, the query and retrieved content are provided to the agent to generate a precise response.

2.2 Tool Box

The system includes two tools: a web search tool and a knowledge base retrieval tool.

The web search tool, built on Tavily, achieves real-time information search from the internet. The knowledge base retrieval tool, based on RAG, locates relevant information from the internal knowledge repository.

2.3 Agent System

The agent system accepts both image and text inputs, enabling it to reason and perform actions by invoking tools, conducting retrieving information from the knowledge base and web searches. The knowledge base retrieval tool accesses relevant information based on the input query. When this tool fails to retrieve relevant information, the web search tool queries online sources. The system also maintains memory to preserve contextual information throughout interactions.

The overall framework can be expressed mathematically as:

$$\mathbf{A} = \mathcal{S}(\mathcal{P}, \mathcal{H}, \mathcal{Q}; \mathcal{T}) \tag{1}$$

Where $\mathbf{A}$ denotes the answer produced by the agent system, $\mathcal{S}$ represents the agent system itself, $\mathcal{P}$ refers to the system prompt, $\mathcal{H}$ denotes to the conversation history, $\mathcal{Q}$ indicates the input , and $\mathcal{T}$ represents the set of tools utilized during reasoning.

3 Experiment Design

We used Gradio [1] to develop a user interface that enables interaction between the user and the agent. The designed interface allows users to input queries and receive responses directly from the agent.

We curated a collection of 86 detailed home inspection reports from public online sources. Our evaluation focuses on case studies from Virginia as an example to align our experiments with the agent's specialized knowledge base, which was populated with the Virginia Residential Code. Representative examples of building components, such as kitchen sinks, vinyl siding, chimney crowns, and coils, were selected as input. Relevant text was combined with these components to form multimodal queries for the VLM agent system. Figure 3 illustrates examples of multimodal inputs used to assess compliance with applicable building codes.

To evaluate the performance of our VLM agent, we compare the similarity of the report generated by our system with expert-provided analysis from the inspection report. To measure the similarity, we use GPT-4o [12] to produce the overall percentage of our report that is similar to the ground truth inspection report.

4 Result

Based on user input, including images or text, the agent system performs reasoning and takes actions through tool invocation to complete code compliance analysis. Figure 4 illustrates a full report generated by the VLM agent system for a kitchen sink case study (Figure 3a). The agent first identifies the components in the image, including a kitchen sink drain with a P-trap and waste disposer, as well as an electrical junction box located beneath the sink. It then retrieves relevant sections and citations from the knowledge base and the National Electrical Code (NEC) using RAG and the web search tool, such as P3201.4 and P3114.7. In the third step, the VLM agent systematically compares findings within a structured table, highlighting several violations. These include a "double trap" (two U-bends in series) violating IPC P3201.4, an improper "trap adapter coupling," and a potentially missing "trap vent," each supported by corresponding code references. The agent also identifies that the electrical junction box is inaccessible due to its placement behind plumbing, violating NEC 314.29. Finally, the agent synthesizes these findings into a compliance determination, summarizing the violations and providing a prioritized list of corrective actions with clear, step-by-step instructions to address the plumbing and electrical issues.

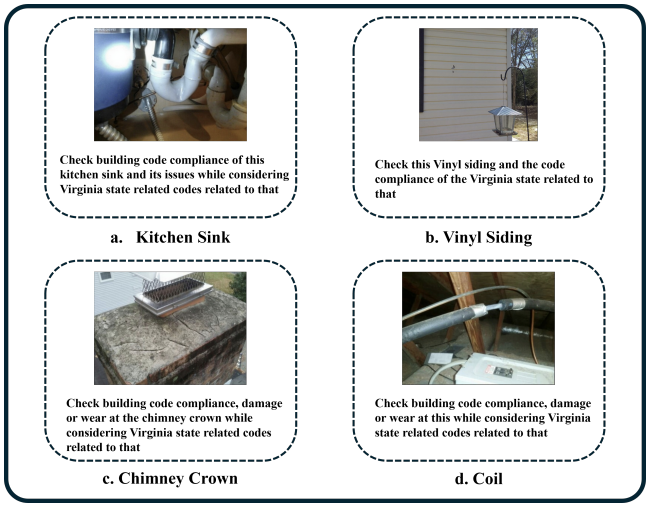


Figure 3: Examples of Multimodal Input.

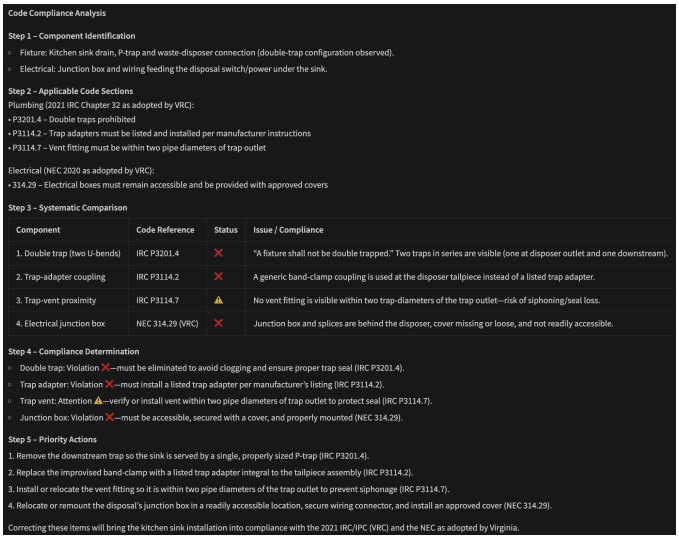
To quantitatively validate the agent's performance, we compare its generated reports against the ground-truth evaluations from the original home inspection reports. The results, summarized in Table 1, show a high degree of similarity across all case studies, with the system achieving an average similarity of 96.25%. The agent demonstrated 100% similarity on the more visually distinct cases of vinyl siding and the chimney crown. Similarity remained high for the mechanically complex HVAC coil scene (95%) and the kitchen sink scene (90%), with the latter's score reflecting the multiple, overlapping plumbing and electrical issues present in a confined space.

Table 1: Similarity of the VLM agent system's answers compared with home inspection reports

Object	Evaluation Similarity
Kitchen Sink	90%
Vinyl Siding	100%
Chimney Crown	100%
Coil	95%
Average	**96.25%**

5 Discussion

The evaluation results of the agent system demonstrate that the agent system effectively analyzes user queries and establishes strong connections between the provided image of building components and the corresponding building codes. Through reasoning and tool invocation, the system generates a detailed report on the inspected building component in Virginia. The report describes the detected components in detail, identifies compliant elements, highlights areas requiring attention or posing safety risks, reference the relevant



Code Compliance Analysis

Step 1 – Component Identification

- Fixture: Kitchen sink drain, P-trap and waste-disposer connection (double-trap configuration observed).
- Electrical: Junction box and wiring feeding the disposal switch/power under the sink.

Step 2 – Applicable Code Sections

Plumbing (2021 IRC Chapter 32 as adopted by VRC):

- P3201.4 – Double traps prohibited
- P3114.2 – Trap adapters must be listed and installed per manufacturer instructions
- P3114.7 – Vent fitting must be within two pipe diameters of trap outlet

Electrical (NEC 2020 as adopted by VRC):

- 314.29 – Electrical boxes must remain accessible and be provided with approved covers

Step 3 – Systematic Comparison

Component	Code Reference	Status	Issue / Compliance
1. Double trap (two U-bends)	IRC P3201.4	✗	"A fixture shall not be double trapped." Two traps in series are visible (one at disposer outlet and one downstream).
2. Trap-adapter coupling	IRC P3114.2	✗	A generic band-clamp coupling is used at the disposer tailpiece instead of a listed trap adapter.
3. Trap-vent proximity	IRC P3114.7	⚠	No vent fitting is visible within two trap-diameters of the trap outlet—risk of siphoning/seal loss.
4. Electrical junction box	NEC 314.29 (VRC)	✗	Junction box and splices are behind the disposer, cover missing or loose, and not readily accessible.

Step 4 – Compliance Determination

- Double trap: Violation ✗—must be eliminated to avoid clogging and ensure proper trap seal (IRC P3201.4).
- Trap adapter: Violation ✗—must install a listed trap adapter per manufacturer's listing (IRC P3114.2).
- Trap vent: Attention ⚠—verify or install vent within two pipe diameters of trap outlet to protect seal (IRC P3114.7).
- Junction box: Violation ✗—must be accessible, secured with a cover, and properly mounted (NEC 314.29).

Step 5 – Priority Actions

1. Remove the downstream trap so the sink is served by a single, properly sized P-trap (IRC P3201.4).
2. Replace the improvised band-clamp with a listed trap adapter integral to the tailpiece assembly (IRC P3114.2).
3. Install or relocate the vent fitting so it is within two pipe diameters of the trap outlet to prevent siphonage (IRC P3114.7).
4. Relocate or remount the disposal's junction box in a readily accessible location, secure wiring connector, and install an approved cover (NEC 314.29).

Correcting these items will bring the kitchen sink installation into compliance with the 2021 IRC/IPC (VRC) and the NEC as adopted by Virginia.

Figure 4: Example of the VLM Agent System outputs.

building codes, and proposes corrective actions. Furthermore, incorporating icons in the reports improves clarity and emphasizes critical information.

The over 98% similarity observed in the vinyl siding and chimney crown scenes is likely attributable to their visual simplicity, which enables the agent to analyze these components more effectively. In contrast, the coil scene appears more complex, resulting in a slightly lower similarity of 95%. The kitchen sink scene records the lowest similarity at 90%, possibly due to increased visual or contextual complexity. These findings highlight the need to enhance the agent's ability to understand and reason about more intricate building components. Upgrading the current VLM agent system to a multi-agent framework may help address this challenge.

Despite its strong overall performance, the agent system exhibits several limitations. The current knowledge base includes only the Virginia building code for state-level regulations in the United States, even though each state typically maintains its own building codes. Moreover, the generated reports occasionally include errors, such as incorrect building code references. These issues may stem from deficiencies in the web search and knowledge base retrieval tools, which sometimes fail to access accurate or comprehensive information. This suggests that the model lacks sufficient domain-specific knowledge of building code compliance. Expanding the knowledge base to include additional building codes and improving the RAG component could mitigate these shortcomings.

6 Conclusion and Future Work

This study introduces a VLM agent designed to support building code compliance. The agent establishes a bridge connecting images and building codes. By reasoning and collaboratively invoking specialized tools, the system analyzes multimodal inputs (images and text) and generates detailed reports that help users understand the condition and compliance status of building-related objects depicted in the images. Four different building object scenes were used to evaluate the VLM agent system. GPT-4o was employed to calculate the similarity accuracy between the outputs generated by the VLM agent and the corresponding evaluation results from home inspection reports. The system achieves an average similarity of 96.25% and produces a detailed report for each scene.

Future enhancements could further improve system performance. Expanding the knowledge base to include additional building codes, such as the NEC and the International Fuel Gas Code (IFGC), would enable broader and more precise information retrieval across different types of buildings (e.g., residential, commercial, and industrial). Incorporating a more advanced embedding model, adopting an improved similarity search algorithm, or introducing a reranking model to better organize retrieved information could enhance the accuracy of the RAG component. Additionally, evolving the system into a multi-agent architecture may benefit building code compliance tasks by decomposing complex processes into subtasks, with each agent specializing in a particular aspect of the analysis. Another significant future direction would be the integration of human expertise through a human-in-the-loop framework. This would involve using datasets annotated by certified inspectors to fine-tune the agent and creating an interactive system where inspectors can validate, correct, and enrich the agent's findings.

This work lays the foundation for future research on VLM agents for building code compliance and offers insights that can inspire broader research in smart buildings and smart cities.

7 Acknowledgments

This material is based upon work supported by the U.S. National Science Foundation under Award. Nos. 2326408 and 1823325 [9, 10]. Identification of funding sources and other support, and thanks to individuals and groups that assisted in the research and the preparation of the work should be included in an acknowledgment section, which is placed just before the reference section in your document.

References

[1] Abubakar Abid, Ali Abdalla, Ali Abid, Dawood Khan, Abdulrahman Alfozan, and James Zou. 2019. Gradio: Hassle-Free Sharing and Testing of ML Models in the Wild. *arXiv preprint arXiv:1906.02569* (2019).

[2] Ildar Baimuratov and Denis Turygin. 2025. Representing Normative Regulations in OWL DL for Automated Compliance Checking Supported by Text Annotation. arXiv:2504.05951 [cs] doi:10.48550/arXiv.2504.05951

[3] Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. BGE M3-Embedding: Multi-Lingual, Multi-Functionality, Multi-Granularity Text Embeddings Through Self-Knowledge Distillation. arXiv:2402.03216 [cs.CL]

[4] Wenqi Fan, Yujuan Ding, Liangbo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, and Qing Li. 2024. A Survey on RAG Meeting LLMs: Towards Retrieval-Augmented Large Language Models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (Barcelona Spain, 2024-08-25). ACM, 6491–6501. doi:10.1145/3637528.3671470

[5] Feliz Gouveia, Vítor Silva, Jorge Lopes, Rui S. Moreira, José M. Torres, and Maria Simas Guerreiro. 2024. Automated Identification of Building Features with Deep Learning for Risk Analysis. 6, 9 (2024), 466. doi:10.1007/s42452-024-06070-2

[6] Praveen Kottari and Pandarasamy Arjunan. 2024. BD3: Building Defects Detection Dataset for Benchmarking Computer Vision Techniques for Automated Defect Identification. In *Proceedings of the 11th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation* (Hangzhou China, 2024-10-29). ACM, 297–301. doi:10.1145/3671127.3698789

[7] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. BLIP-2: Bootstrapping Language-Image Pre-Training with Frozen Image Encoders and Large Language Models. In *Proceedings of the 40th International Conference on Machine Learning* (2023-07-03). PMLR, 19730–19742. https://proceedings.mlr.press/v202/li23q.html

[8] Prayushi Mathur, Charu Sharma, and Syed Azeemuddin. 2024. Autonomous Inspection of High-Rise Buildings for Façade Detection and 3D Modeling Using

UAVs. *IEEE Access* 12 (2024), 18251–18258. doi:10.1109/ACCESS.2024.3360209

[9] National Science Foundation. 2020. U.S. National Science Foundation Award No. 1823325. https://www.nsf.gov/awardsearch/showAward?AWD_ID=1823325. [Accessed: 2025-04-29].

[10] National Science Foundation. 2023. U.S. National Science Foundation Award No. 2326408: Future of Digital Facility Management (Future of DFM). https://www.nsf.gov/awardsearch/showAward?AWD_ID=2326408. [Accessed: 2025-04-29].

[11] Runliang Niu, Jindong Li, Shiqi Wang, Yali Fu, Xiyu Hu, Xueyuan Leng, He Kong, Yi Chang, and Qi Wang. 2024. *ScreenAgent: A Vision Language Model-Driven Computer Control Agent.* arXiv:2402.07945 [cs] doi:10.48550/arXiv.2402.07945

[12] OpenAI. 2024. GPT-4o System Card. arXiv preprint arXiv:2410.21276. doi:10.48550/arXiv.2410.21276

[13] Husein Perez, Joseph H. M. Tah, and Amir Mosavi. 2019. Deep Learning for Detecting Building Defects Using Convolutional Neural Networks. 19, 16 (2019), 3556. Issue 16. doi:10.3390/s19163556

[14] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning Transferable Visual Models From Natural Language Supervision. In *Proceedings of the 38th International Conference on Machine Learning* (2021-07-01). PMLR, 8748–8763. https://proceedings.mlr.press/v139/radford21a.html

[15] Sebastian Seiß, Judith Fauth, Yuan Zheng, and Aurica Poetz. 2025. Understanding and Conceptualizing Inspections in the Context of Building Permits. *Smart and Sustainable Built Environment* (May 2025). doi:10.1108/SASBE-11-2024-0492

[16] Deep Search Team. 2024. *Docling Technical Report.* Technical Report. arXiv:2408.09869 doi:10.48550/arXiv.2408.09869

[17] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (Eds.), Vol. 35. Curran Associates, Inc., 24824–24837.

[18] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. *ReAct: Synergizing Reasoning and Acting in Language Models.* arXiv:2210.03629 [cs] doi:10.48550/arXiv.2210.03629