iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 Issue: V Month of publication: May 2026

DOI: https://doi.org/10.22214/ijraset.2026.82265

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue V May 2026- Available at www.ijraset.com

Development of a Knowledge-Based Python AddIn/Scripting for Automated CNC Tool Path Generation in Stone Fabrication Using Autodesk Fusion 360 CAM

Prashant P. Atram, Prof. Mithlesh Pande, Prof. Swapnil Choudhary, Dr. Dilip Gangwani
Student of MTech CAD/CAM, Assistant Professor, Department of Mechanical Engineering WCEM Nagpur, India

Abstract: Computer Numerical Control (CNC) machining has become an essential manufacturing process in the stone fabrication industry for producing countertops, vanity slabs, sink cutouts, edge profiles, and architectural stone components with high dimensional accuracy. However, conventional CAM programming for stone machining remains highly manual, requiring repetitive selection of machining tools, feed rates, spindle speeds, and depth parameters for every individual operation. This not only increases programming time but also introduces dependency on operator experience and inconsistency in machining standards. To overcome these limitations, the present research proposes a Knowledge-Based Engineering (KBE) framework for automation of CNC tool path generation in Autodesk Fusion 360 using Python-based scripting.

The developed system captures machining expertise in the form of rule-based knowledge including material-specific machining parameters, slab thickness rules, edge profile tool mapping, and operation-wise tool assignment logic. A custom Python AddIn/Scripting was developed using Fusion 360 CAM API that automatically reads CAM operations, identifies machining intent, selects the appropriate tool from a JSON-based digital tool library, applies predefined spindle speed, feed rate, and step-down values, renames operations, and regenerates CNC tool paths automatically. The proposed system significantly reduces repetitive CAM programming effort while ensuring standardization of machining parameters across stone fabrication jobs.

Experimental implementation on multiple machining scenarios such as edge profiling, sink cutout machining, and drilling operations demonstrates that the proposed KBE system offers a practical and scalable solution for intelligent CAM automation in the stone manufacturing sector. The research establishes an effective bridge between domain knowledge capture and industrial CNC programming automation.

Keywords: Knowledge-Based Engineering, CNC Tool path Automation, Fusion 360 API, Stone Fabrication, Python Scripting, CAM Automation, Rule-Based Manufacturing.

I. INTRODUCTION

The stone fabrication industry has witnessed substantial growth in recent years due to increasing demand for customized kitchen countertops, washbasin vanity tops, decorative wall panels, staircases, and precision architectural stone components. Materials such as granite, marble, quartz, and Dekton are commonly machined using CNC routers and CNC bridge saw systems to achieve accurate dimensions, smooth profiles, and repeatable production quality. In order to manufacture these components, Computer Aided Manufacturing (CAM) software is employed to generate machining tool paths corresponding to edge finishing, drilling, sink cutouts, faucet holes, and contour profiling operations.

Despite advancements in CAM software, the actual process of generating CNC machining operations for stone remains largely dependent on manual decision making by CAM programmers. For each machining job, the programmer is required to manually inspect machining operations, determine suitable tooling, assign spindle speed and feed values according to material properties, define depth of cut, and regenerate each toolpath individually. When a single stone slab contains multiple machining features, this process becomes highly repetitive and time intensive. Moreover, due to dependency on individual programmer expertise, inconsistencies frequently arise in tool selection and machining parameter assignment.

Knowledge-Based Engineering (KBE) offers an effective methodology to automate repetitive engineering decisions by embedding expert manufacturing knowledge into computational rule systems. In KBE, practical shop-floor machining experience is translated into explicit logical rules that can automatically drive software decisions without repeated human intervention.

Such an approach is particularly beneficial in CNC stone machining where tooling and parameter decisions often follow standard repetitive industrial patterns.

The present research focuses on the development of a Python-based KBE automation system integrated with Autodesk Fusion 360 CAM environment. The proposed system automates CNC tool path generation by capturing stone machining knowledge related to material type, slab thickness, edge profile, and machining feature type. Based on these user inputs, the developed AddIn/Scripting automatically assigns the appropriate machining tool, applies optimized spindle speed and feed rate values, modifies machining depth, updates CAM operation naming, and regenerates all machining tool paths.

The major objective of this work is to reduce CAM programming dependency on manual operator decisions and to establish a scalable intelligent automation framework specifically for stone fabrication industries. Unlike conventional macro scripting that only modifies isolated parameters, the present work develops a domain-specific rule-based inference engine capable of performing integrated machining decision automation inside a commercial CAM platform.

II. LITERATURE REVIEW

The concept of Knowledge-Based Engineering has been extensively explored in modern manufacturing as a means of embedding domain knowledge into automated computational systems. KBE systems are primarily designed to reduce repetitive engineering tasks by capturing expert knowledge in the form of rules, databases, and inference logic. In CNC machining applications, KBE has demonstrated strong potential in reducing manual process planning and improving consistency in machining parameter selection.

Various researchers have investigated CAD/CAM integration for machining automation. Early works focused on feature recognition methods where geometric entities such as pockets, holes, slots, and planar surfaces were identified directly from CAD models and linked to predefined machining strategies. These systems attempted to reduce manual CAM planning by associating recognized geometric features with corresponding machining operations. However, implementation complexity and software interoperability limitations often restricted industrial adoption.

Recent studies have shifted toward rule-based automation within commercial CAM platforms through scripting interfaces and API-level customization. Python scripting, in particular, has emerged as an effective approach due to its flexibility in handling user-defined logic, database interaction, and software automation commands. Several manufacturing studies reported successful use of scripting for automatic parameter modification, tool path regeneration, and process standardization in aerospace and metal cutting applications.

Despite these developments, very limited research has been reported in the field of stone fabrication CNC automation. Stone machining differs significantly from metal machining due to brittle material behavior, varying hardness, edge finishing requirements, and the necessity for specialized diamond profile tools. Machining parameters for granite, marble, quartz, and sintered stone materials cannot be generalized and require material-specific empirical knowledge. In addition, operations such as sink cutouts, faucet drilling, bevel edging, and bullnose profiling involve repetitive but highly experience-dependent CAM decisions.

Existing commercial CAM software provides flexible machining capabilities but lacks an embedded intelligent rule engine specifically designed for stone manufacturing workflows. As a result, tool selection and machining parameter tuning remain manual even in advanced software environments. This creates an opportunity for developing a practical KBE-based automation framework that can capture stone machining expertise and apply it directly inside a commercial CAM environment.

The present study addresses this research gap by developing a Python-scripted Fusion 360 AddIn/Scripting capable of integrating machining knowledge, digital tool libraries, and automated parameter assignment into a single knowledge-driven CNC tool path generation framework for stone fabrication applications.

III. KNOWLEDGE ACQUISITION FROM INDUSTRIAL CAM PRACTICE

Since stone machining decisions in practical workshops are largely experience-driven, the first stage of the present research involved systematic observation of industrial CAM programming procedures and extraction of repetitive decision rules followed by skilled programmers. Particular attention was given to commonly repeated tasks such as manual edge tool replacement, sink machining tool assignment, spindle/feed modification, and depth setting for different slab materials.

The collected industrial knowledge was then translated into explicit rule statements suitable for software implementation. This knowledge acquisition stage served as the foundation for development of the proposed Knowledge-Based Engineering system.

IV. PROBLEM STATEMENT AND EXISTING MANUAL CAM WORKFLOW

In conventional stone CNC programming practice, CAM engineers perform the majority of machining preparation tasks manually within the CAM software environment. After generating a machining setup, each individual operation such as contour cutting, sink cutout profiling, faucet drilling, edge polishing, bevel edging, or bullnose shaping must be manually edited by the programmer. The operator first determines the suitable tool based on machining intent, then searches the required tool from the tool library, assigns the tool to the selected operation, manually enters spindle speed, feed rate, and step-down values according to stone material, and finally regenerates the machining path.

For jobs containing numerous machining operations, the above procedure is repeated operation by operation, resulting in substantial programming time consumption. In addition to time inefficiency, this manual process creates several industrial limitations:

1) Repetitive manual intervention for each CAM operation.
2) Heavy dependency on operator machining knowledge.
3) Lack of machining parameter standardization.
4) Increased probability of incorrect tool assignment.
5) Difficulty in maintaining consistency across multiple jobs.

Furthermore, when different materials such as granite, marble, quartz, or Dekton are processed, the programmer must again manually revise spindle speed, cutting feed, and depth values according to practical machining experience. Similarly, if the customer requires a different edge profile such as bevel edge, full bullnose, or pencil edge, the edge tool must be manually replaced in all relevant operations.

This conventional manual workflow demonstrates a clear need for an intelligent automation framework capable of converting repetitive machining decisions into a standardized computational process. Therefore, the present research proposes a Knowledge-Based Engineering automation system that minimizes manual CAM editing and transforms operator knowledge into software-executable machining rules.

CAD Model Import
↓
Create CAM Setup
↓
Select Operation Manually
↓
Select Tool Manually
↓
Enter RPM/Feed Manually
↓
Edit Depth of Cut
↓
Regenerate Toolpath
↓
Repeat for Every Operation

Fig. 1 Existing Manual CNC Tool path Preparation Workflow in Stone Fabrication

V. PROPOSED KNOWLEDGE-BASED AUTOMATION FRAMEWORK

To overcome the inefficiencies of conventional CAM programming, a Knowledge-Based Engineering framework was developed in the present work for automating repetitive CNC machining decisions in stone fabrication. The proposed framework converts human machining expertise into a computational inference system capable of performing tool selection, machining parameter assignment, and CAM operation regeneration automatically.

The developed framework is composed of four major functional layers:

A. *User Input Layer*

The first layer captures essential high-level manufacturing information from the CAM programmer through a custom Fusion 360 AddIn/Scripting dialog. Three principal inputs are provided by the user:

- Stone Material Type
- Slab Thickness
- Default Edge Profile

These inputs act as the primary decision variables governing subsequent machining logic.

B. Knowledge Base Layer

The second layer consists of a digital knowledge repository containing two categories of manufacturing knowledge:

1) Machining Parameter Rules

Material and thickness dependent spindle speed, feed rate, and step-down values.

2) Tool Knowledge Database

A JSON-based structured tool library containing all available stone machining tools such as:

- Sink Cutout Tool
- Faucet Hole Tool
- Sinkit Hole Tool
- Eased Edge Tool
- Bevel Edge Tool
- Pencil Edge Tool
- Half Bullnose Tool
- Full Bullnose Tool

Each tool entry stores description labels and Fusion-compatible tool geometry data.

C. Inference Engine Layer

This is the core decision-making engine of the KBE system. Once the user provides inputs and the CAM setup is loaded, the inference engine performs:

- scanning of all CAM operations,
- identification of machining intent from operation names,
- matching of machining feature with corresponding tool,
- extraction of machining rules from the knowledge base,
- automated assignment of tool and machining parameters.

The inference engine effectively replaces repetitive human decision making with deterministic rule execution.

D. CAM Execution Layer

The final layer communicates with Autodesk Fusion 360 CAM API to implement the inferred decisions directly into the machining environment. This layer performs:

- replacement of operation tool,
- spindle speed modification,
- feed rate modification,
- depth of cut modification,
- operation renaming,
- regeneration of complete tool paths.

Thus, the developed framework forms a closed automation loop from user knowledge input to final CNC-ready CAM output.

User Inputs
(Material + Thickness + Edge)
↓
Knowledge Rule Base
(Material Rules + Tool Rules)
↓

Python Inference Engine
↓
Fusion 360 CAM API Automation
↓
Tool Assignment + Parameter Apply + Rename + Tool path Generate

Fig. 2 Proposed Knowledge-Based Automation Architecture for Stone CAM Programming

VI. KNOWLEDGE RULE FORMULATION AND REPRESENTATION

A crucial stage in Knowledge-Based Engineering is the transformation of practical machining expertise into formal executable rules. In the present work, machining knowledge was acquired through observation of stone fabrication programming practices and converted into explicit IF–THEN production rules.

The developed rule base contains two major classes of knowledge: machining parameter rules and operation-to-tool mapping rules.

A. *Mathematical Basis for Machining Rule Development*

The machining parameter values used in the developed knowledge base were not assigned arbitrarily, but were derived from standard CNC machining relationships commonly adopted in stone routing and abrasive tool machining. The spindle speed, feed rate, and depth-of-cut values were selected by considering cutting speed recommendations, tool diameter, abrasive engagement, and brittle fracture behavior of stone materials.

(a) Spindle Speed Relation

$$n=\frac{1000 \times V_C}{\pi \times D}$$

Where,
n = spindle speed (rpm),
(V_C) = cutting speed (m/min),
D = tool diameter (mm).

(b) Feed Rate Relation

$$f_{t=n \times z \times} f_z$$

Where,
(f_t) = table feed rate (mm/min),
z = number of cutting segments,
(f_z) = feed per segment.

(c) Material Removal Rate

$$MRR = a_{p \times} a_e \times f_t$$

Where,
a_p is the axial depth of cut
a_e is the radial step-over.
(f_t) = table feed rate (mm/min),

The above machining relations were used as baseline engineering references and subsequently tuned according to industrial stone fabrication practice to generate the final rule values embedded in the KBE system.

B. Material and Thickness Based Machining Rules

Different stone materials exhibit significantly different machinability characteristics. Hard materials such as Dekton require lower feed and spindle values due to high brittleness, while granite and quartz permit relatively aggressive machining conditions. Similarly, slab thickness influences permissible step-down depth.

The following machining rules were encoded into the Python automation engine.

Table 1. Material-Based Machining Rule Set

Rule No.	IF Condition	THEN Spindle Speed	THEN Feed Rate	THEN Step-down
R1	Granite + 20mm	4500 rpm	600 mm/min	2 mm
R2	Granite + 30mm	4200 rpm	500 mm/min	1.5 mm
R3	Marble + 20mm	3500 rpm	800 mm/min	5 mm
R4	Marble + 30mm	3200 rpm	700 mm/min	3 mm
R5	Quartz + 20mm	4800 rpm	900 mm/min	2 mm
R6	Quartz + 30mm	4500 rpm	800 mm/min	2 mm
R7	Dekton + 20mm	2500 rpm	300 mm/min	1 mm
R8	Dekton + 30mm	2200 rpm	250 mm/min	0.8 mm

C. Operation Feature Detection Rules

To automate tool assignment, the system uses keyword-based operation feature recognition. Existing CAM operation names are analyzed and compared with predefined logical identifiers.

Table 2. Operation Mapping Rule Set

Rule No.	IF Operation Name Contains	Assigned Tool
R9	sinkit	Sinkit Hole Tool
R10	faucet	Faucet Hole Tool
R11	sink	Sink Cutout Tool
R12	pencil	Pencil Edge Tool
R13	eased	Eased Edge Tool
R14	bevel	Bevel Edge Tool
R15	half bull	Half Bullnose Tool
R16	full bull	Full Bullnose Tool
R17	No keyword match	User selected default edge tool

These mapping rules allow the AddIn/Scripting to automatically determine the machining intent of every CAM operation without requiring manual operation editing.

D. Knowledge Rule Execution Logic

The implemented KBE engine follows a sequential decision logic:

1) Read user selected material, thickness, and edge profile.
2) Load material machining rule corresponding to selected condition.
3) Load JSON digital tool database.
4) Scan all operations from Fusion 360 CAM setup.
5) Detect operation keyword.
6) Match corresponding tool.

7) Assign tool to CAM operation.
8) Apply spindle speed, feed rate, and step-down values.
9) Rename operation according to applied knowledge.
10) Regenerate updated toolpath.

This converts implicit machining know-how into a formal software-executable rule chain.

VII. PYTHON-BASED ADDIN/SCRIPTING IMPLEMENTATION IN FUSION 360

The practical implementation of the proposed KBE system was carried out through Autodesk Fusion 360 API using Python scripting. A dedicated custom AddIn/Scripting named **"KBE Stone Automation"** was developed and integrated into the Manufacture workspace toolbar.

The AddIn/Scripting architecture consists of three major program modules:

A. *User Interface Module*

When the AddIn/Scripting button is executed, a custom command dialog is launched containing dropdown selections for:

- Material Type
- Slab Thickness
- Default Edge Profile

This provides a simple human-machine interaction layer through which the CAM programmer inputs only essential manufacturing conditions while all lower-level machining decisions are delegated to the automation engine.

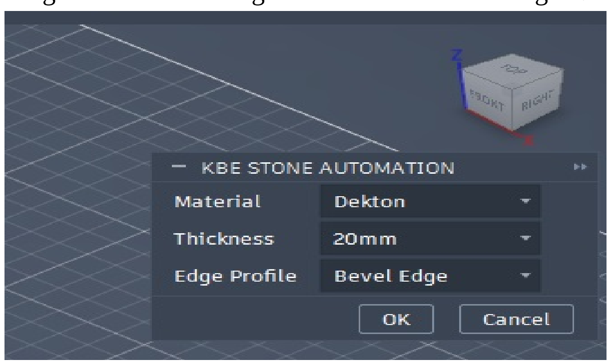


Fig. 3 Custom Fusion 360 AddIn/Scripting Interface for User Input

B. *Tool Database Loading Module*

A JSON-based digital tool file named *Stone_Tool.json* is loaded during execution. This file stores all predefined Fusion-compatible stone machining tools. The script reads the tool descriptions and dynamically converts the JSON entries into Fusion 360 CAM Tool objects using Python API commands.

This database-driven approach offers two advantages:

- Tool geometry can be modified without rewriting core code.
- New tools can be added to the knowledge base easily.

Fig. 4 Digital JSON-Based Tool Knowledge Repository Used by Developed AddIn/Scripting

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue V May 2026- Available at www.ijraset.com

C. CAM Operation Automation Module

After loading user inputs and tool knowledge, the AddIn/Scripting reads all machining setups and recursively scans every operation present in the CAM browser. For each operation, the script performs the following automated tasks:

- Detects target machining type from operation name,
- Selects corresponding tool object,
- Assigns selected tool to operation,
- Overwrites spindle speed,
- Overwrites cutting feed rate,
- Overwrites lead and ramp feed rates,
- Overwrites maximum step-down value,
- Renames the operation according to applied tool and material condition.

After all operations are updated, Fusion 360 CAM API is instructed to regenerate all modified tool paths automatically.

D. Algorithmic Workflow of Developed AddIn/Scripting

The computational sequence of the implemented Python automation engine can be represented as follows:

Algorithm 1: KBE Stone CAM Automation

1) Start AddIn/Scripting execution
2) Accept user inputs (material, thickness, edge profile)
3) Load JSON tool knowledge base
4) Read all CAM setups
5) For each CAM operation
- Detect machining keyword
- Identify required tool
- Assign Fusion tool object
- Apply machining rules
- Rename operation

6) Regenerate setup toolpaths
7) Display automation completion report
8) End program

The above algorithm demonstrates that the developed AddIn/Scripting performs not merely parameter editing, but a complete rule-based machining decision automation process.

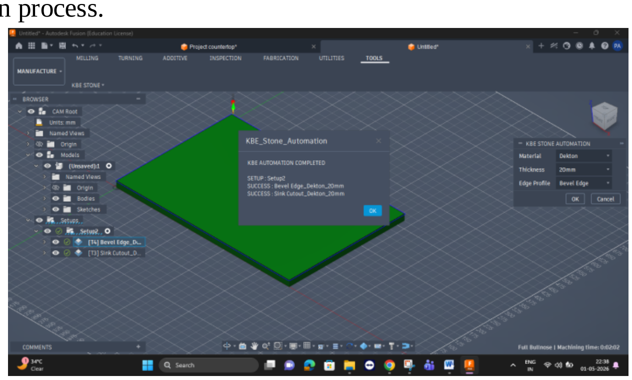


Fig. 5 Automation Completion Report Generated by Developed Fusion 360 AddIn/Scripting

VIII. EXPERIMENTAL RESULTS AND VALIDATION

To evaluate the effectiveness of the developed Knowledge-Based Automation system, experimental implementation was carried out on representative stone machining CAM jobs created in Autodesk Fusion 360 Manufacture workspace. Multiple machining operations including edge profiling, sink cutout machining, faucet drilling, and contour finishing were considered for validation.

The primary objective of validation was to compare the conventional manual CAM programming workflow with the developed KBE-based automated workflow in terms of programming effort, operator intervention, machining parameter consistency, and tool path preparation efficiency.

A. Validation Methodology

To verify the robustness of the developed automation system, the AddIn/Scripting was executed on multiple representative CAM setups containing combinations of drilling, sink cutout, contour edging, and profile finishing operations. Different material-thickness combinations were tested to ensure that the knowledge engine correctly applied the corresponding machining rules and edge tool substitutions under varying fabrication scenarios

.

B. Validation Case Studies

Three representative stone fabrication machining cases were selected:

Case Study 1: Granite Countertop with Full Bullnose Edge

This case involved contour edge finishing operations and faucet hole machining on a 20 mm granite slab.

Case Study 2: Quartz Vanity Top with Bevel Edge and Sink Cutout

This job contained sink cutout profiling, contour machining, and bevel edge finishing on a 30 mm quartz slab.

Case Study 3: Dekton Utility Slab with Sinkit and Drilling Operations

This case involved high-hardness Dekton machining requiring conservative feed and spindle conditions.

These cases were selected to ensure evaluation across different stone materials, slab thicknesses, and machining feature combinations.

C. Manual Programming versus KBE Automated Programming

In the conventional CAM workflow, the programmer manually edits every machining operation individually by selecting the proper tool, entering spindle speed, entering cutting feed, adjusting depth of cut, and regenerating the operation toolpath. The same sequence is repeated for all operations present in the machining setup.

In contrast, under the developed KBE framework, the programmer only selects:

- Material type,
- Slab thickness,
- Default edge profile,

after which the AddIn/Scripting performs all lower-level machining decisions automatically.

Table 3. Comparative Evaluation of Manual and Automated Workflow

Evaluation Parameter	Conventional Manual CAM	Proposed KBE Automation
Tool Selection	Manual for each operation	Automatic
RPM Assignment	Manual	Automatic
Feed Rate Assignment	Manual	Automatic
Step-down Setting	Manual	Automatic
Operation Renaming	Manual/Not standardized	Automatic
Tool path Regeneration	Manual each time	Automatic
Programmer Dependency	High	Low
Process Standardization	Variable	High

D. Observed Performance Improvement

Experimental implementation indicated a substantial reduction in repetitive CAM programming effort. The developed automation engine was able to update all operations in a machining setup within a single AddIn/Scripting execution cycle, whereas conventional manual preparation required repeated editing of each individual operation.

The following practical observations were recorded during testing:

- Manual CAM editing steps were reduced significantly,

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue V May 2026- Available at www.ijraset.com

- Machining parameter consistency was maintained across all operations,
- Tool assignment errors were minimized,
- Edge profile replacement was achieved through a single user selection,
- Complete setup toolpaths were regenerated automatically.

Although exact programming time may vary depending on number of operations, it was observed that the developed KBE system reduced overall CAM preparation workload by a considerable margin while simultaneously improving repeatability.

E. Tool Assignment and Browser Update Validation

The developed AddIn/Scripting was further validated by observing Fusion 360 CAM browser changes after execution. It was confirmed that:

- Operation tool references were successfully replaced by the KBE-selected tools,
- Spindle speed values were overwritten according to selected material,
- Feed rates and step-down values were updated,
- Operation names reflected the applied edge or machining condition,
- Warning symbols were cleared after regeneration in valid cases.

This demonstrated successful API-level communication between the knowledge engine and Fusion 360 CAM environment.

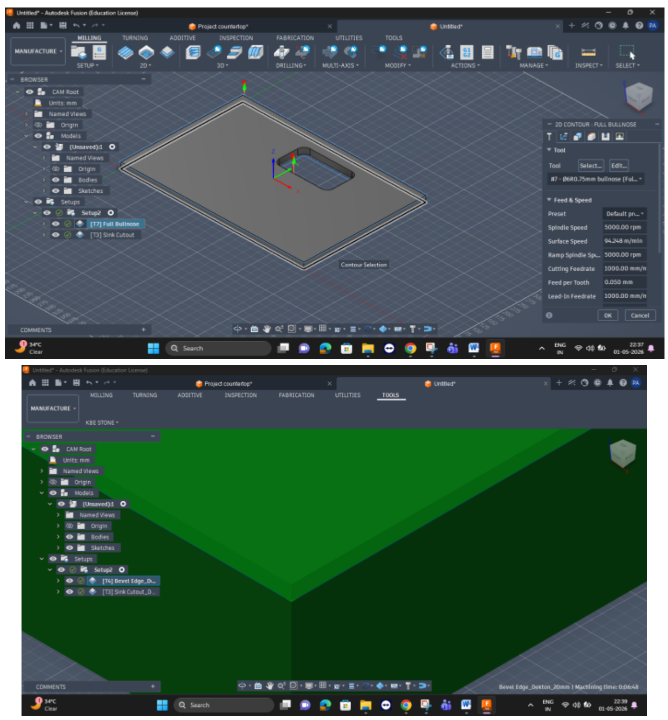
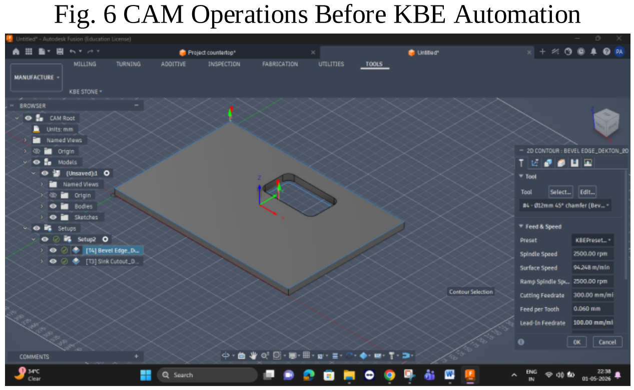


Fig. 6 CAM Operations Before KBE Automation



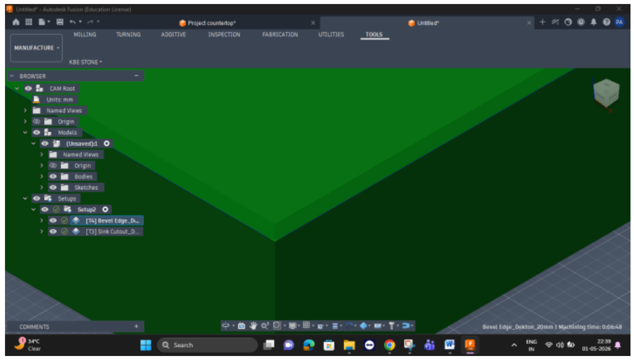


Fig. 7 CAM Operations After KBE Automation

IX. DISCUSSION

The results obtained from the developed system indicate that Knowledge-Based Engineering can serve as an effective practical automation methodology for CNC stone machining. In conventional CAM programming, the programmer repeatedly performs low-level machining decisions that largely follow standard industrial rules. Such repetitive tasks do not require continuous human creativity but instead require consistent application of known machining knowledge.

The present research demonstrates that this repetitive engineering knowledge can be successfully formalized into executable Python rules and deployed within a commercial CAM software platform. By doing so, the role of the programmer shifts from manual parameter editor to supervisory decision provider.

A major advantage observed in the developed system is standardization. Since all machining rules are predefined and automatically applied, the resulting CNC programs maintain consistent spindle speed, feed rate, and edge tool selection independent of programmer variability. This is particularly beneficial in stone industries where operator skill differences often produce machining inconsistency.

Another important contribution is maintainability. Because the developed tool database is stored externally in JSON format, any future modification in tool geometry, tool length, or physical dimensions can be incorporated without rewriting the inference logic. This makes the proposed system industrially scalable.

Furthermore, the present work establishes a practical bridge between Knowledge-Based Engineering theory and real CAM software implementation. Rather than limiting KBE to conceptual process planning, the developed system performs direct executable machining automation within Fusion 360, which strengthens its industrial applicability.

X. CONCLUSION

The present research successfully developed and implemented a Knowledge-Based Automation framework for CNC tool path generation in stone fabrication using Python-based scripting in Autodesk Fusion 360. The developed system captures expert machining knowledge related to stone material, slab thickness, edge profile selection, and operation-wise tooling decisions, and translates this knowledge into a rule-based computational inference engine.

A custom Fusion 360 AddIn/Scripting was created to automatically scan machining operations, load tools from a digital JSON library, assign corresponding tools, apply machining parameters, rename operations, and regenerate tool paths without repetitive manual intervention. Experimental validation confirmed that the proposed automation significantly improves CAM programming consistency, reduces operator dependency, and simplifies repetitive machining preparation tasks.

Thus, the research demonstrates that Knowledge-Based Engineering can be effectively integrated with modern commercial CAM environments to achieve practical intelligent manufacturing automation for stone fabrication industries.

XI. FUTURE SCOPE

Although the present work successfully establishes a robust rule-based CAM automation framework, further developments can enhance the intelligence and industrial scope of the system. Future research may include:

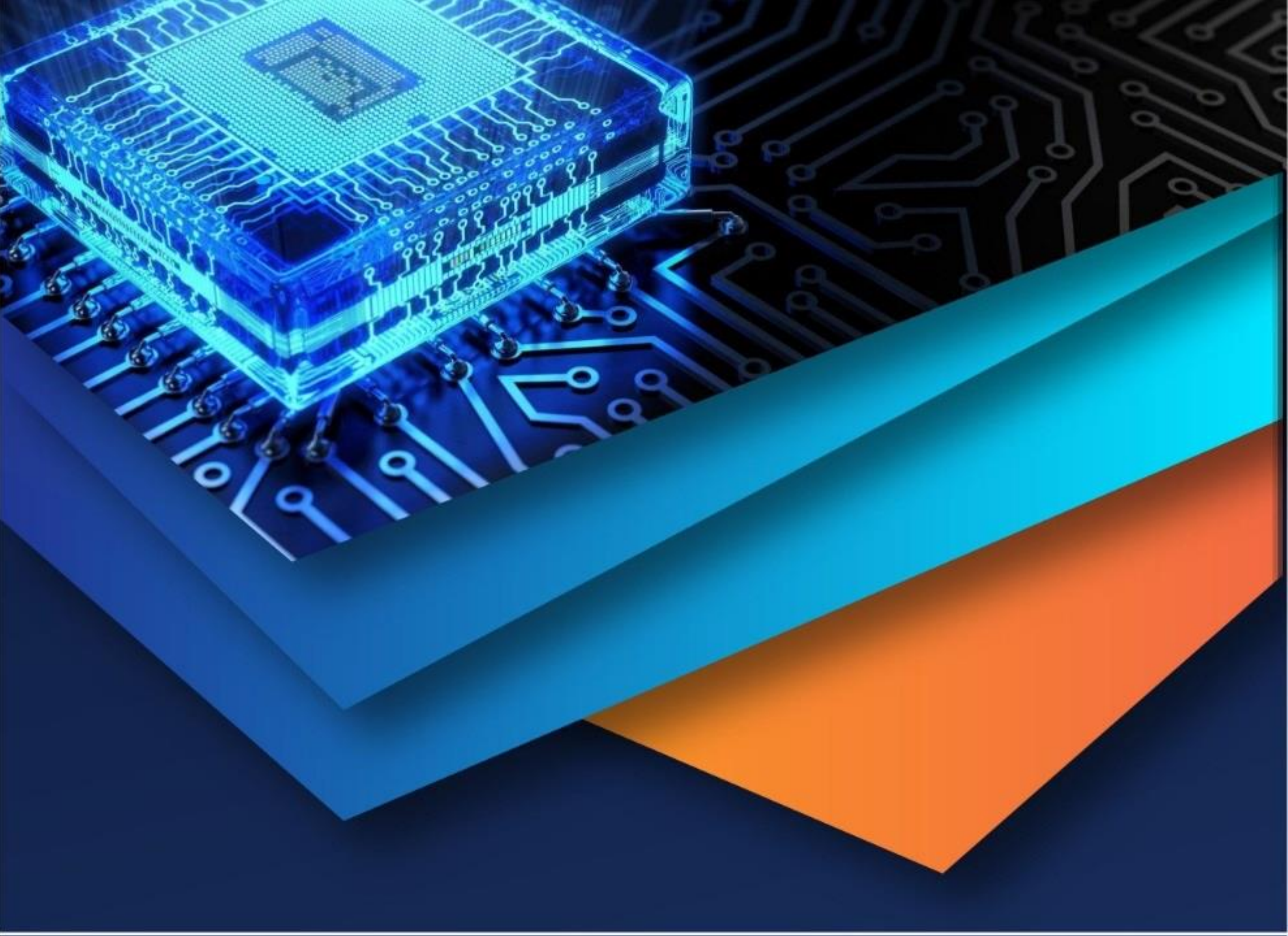
1) Automatic CAD feature recognition for direct geometry-driven operation creation.
2) Integration of a larger enterprise-level stone tool database.
3) Adaptive machining parameter optimization based on tool wear and machine load.

4) Automatic NC code post-processing and machine-specific output generation.
5) Integration with artificial intelligence for self-learning machining recommendations.

These future enhancements can transform the developed KBE system into a fully autonomous smart manufacturing platform for digital stone fabrication.

REFERENCES

[1] J. E. C. P. de Souza, C. R. Diniz, L. E. C. de Araújo, and R. J. R. de Oliveira, "Automated Unsupervised 3D Tool-Path Generation Using Stacked 2D Image Processing Technique," J. Manuf. Mater. Process., vol. 3, no. 4, p. 84, Sep. 2019.
[2] J. Liao and Z. Huang, "Data model-based tool path generation techniques for CNC milling machines," Frontiers in Mechanical Engineering, vol. 10, Mar. 2024.
[3] D. Calkins and C. Scholz, "Knowledge-Based Engineering (KBE) Design Methodology at the Undergraduate and Graduate Levels," International Journal of Engineering Education, vol. 16, no. 1, pp. 21-38, Jan. 2000.
[4] M. G. S. O'Sullivan and C. J. L. A. Marra, "Enhancing product development through knowledge-based engineering (KBE)," J. Manuf. Technol. Manag., vol. 20, no. 8, pp. 1070-1083, 2009.
[5] I. Fidan and A. ElSawy, "The Development of a Knowledge-Based Engineering Tool for CNC Machining," International Journal of Engineering Education, vol. 18, no. 6, pp. 696-702, Dec. 2002.
[6] Autodesk, "Fusion Help | Introduction to the CAM API." Retrieved from https://help.autodesk.com/view/fusion360/ENU/?guid=GUID-7B5A90C8-E94C-48DA-B16B-430729B734DC.
[7] P. Rainsberry, "Automating Fusion 360 with the API," Autodesk University Class, MFG225552, 2018.
[8] Camm Metals. (2017). Top 5 Challenges In CNC Machining Explained. Retrieved from https://www.cammmetals.com/blog.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY