IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 3, March 2026

DOI: 10.17148/IJARCCE.2026.153107

Self-Evolving Cognitive Architectures: Bridging Neural Optimization and Consciousness Simulation for AGI

Adhithyan A[1], Vivek S[2], Febin K James[3], Muhammed Abnan[4], Asst. Prof. Jinu L[5]

Department of Computer Science, College of Engineering, Kottarakkara, Kollam, Kerala, India[1-5]

Abstract: This paper presents the design and theoretical foundations of a Self-Evolving Cognitive Architecture (SECA), a unified framework that combines autonomous neural architecture evolution with consciousness-inspired cognitive mod- eling. The proposed system leverages evolutionary algorithms, particularly customized genetic operators, to iteratively redesign and optimize neural network architectures without human intervention. In parallel, it incorporates Theory of Mind (ToM) principles, recursive self-modeling, and agent-based logic to simulate subjective awareness, emergent be- havior, and adaptive reasoning. Conventional approaches to artificial intelligence typically emphasize either structural optimization, as seen in automated architecture search, or cognitive emulation, as pursued in consciousness simulators. SECA addresses this divide by providing a dual layered paradigm in which neural substrates evolve continuously while cognitive processes develop recursively. This synthesis offers both computational adaptability and higher order reasoning capabilities, enabling AI systems to dynamically restructure their architectures while simultaneously reflecting on their internal states, predicting the intentions of other agents, and adapting behavior to complex, dynamic environments. By integrating architectural self-optimization with consciousness simulation, SECA represents a significant advancement to- ward Artificial General Intelligence (AGI). The framework facilitates the creation of AI systems that are not only efficient and scalable but also capable of introspection, adaptive learning, and human-like social cognition.

Keywords: Self-Evolving Neural Architectures, Consciousness Simulation, Theory of Mind, Evolutionary Algorithms, AGI, Adaptive AI

I. INTRODUCTION

Artificial intelligence has made tremendous progress in the past decade due to developments in deep learning and optimization technology. Though such models are highly efficient in their respective tasks of image recognition, natural language processing, and playing games, these models lack broad intelligence since most of these models rely on predefined architectures and significantly require human involvement to make them work in a new environment. “The human brain, in contrast, shows self-adaptation, learning, and reasoning processes carried out continually, and this continues to be enabled by brain plasticity and consciousness.” Motivated from such biological tenets, researchers have suggested the theory of Self-Evolving Cognitive Architectures (SECA), thereby developing AI models with the ability for self-evolution of their nervous system, along with the development of cognitive skills like self-introspection and theory of mind. SECA integrates various areas of research work such as evolutionary algorithms, neuroevolution, cognitive modeling, and meta- learning studies. With the processes of evolution and self-feedback analysis, SECA hopes to develop an AI model that is capable of re-modeling its own internal structuration dynamically and adjust to fresh tasks. With this feature, it is hoped that it will represent a marked development towards reaching Artificial General Intelligence. In this study, the development of SECA and its architectural elements are reviewed and discussed for assessing the future development of intelligent systems and overcoming the current limitations of AI. The current developments and limitations of AI are discussed based on various study reports and analyses available for consideration.

II. LITERATURE REVIEW

Classifier-Assisted Hybrid Evolutionary NAS [1]
Methodology: CHENAS is a hybrid evolutionary NAS framework that combines evolutionary algorithms with gradient-based local refinement. It predicts Pareto dominance using a classifier and a contrastive learning-based autoencoder that maps architectures to a continuous latent space. Evaluation metrics: Experiments on benchmark datasets validate solution quality and convergence speed, implicitly measuring the proposed dominance classifier and contrastive learning’s effectiveness against state-of-the-art NAS approaches. Limitations: NAS is computationally expensive with slow

DOI: 10.17148/IJARCCE.2026.153107

convergence and rank disorder in existing predictor-assisted evolutionary methods, requiring a specialized dominance classifier. Over- come in our Design: Utilize a Hybrid Evolutionary classifier-assisted prediction model to efficiently rank and discard non-optimal deep learning architectures, reducing search costs for high-quality, low-latency biometric solutions. Contrastive learning autoencoders can guide efficient architecture representation for the final lightweight model.

Cognitive Models for Machine Theory of Mind [2]
Methodology: This work suggests that cognitive models using ACT-R and instance based learning can meet the criteria for a Machine Theory of Mind (MToM). These models personalize behavior through mechanisms like knowledge and model tracing. Evaluation metrics: We predict individual cognitive load, estimate error probability for behavioral predictions, provide player self-efficacy estimates, and calculate trust calibration metrics for human-machine interaction. Limitations: This Minecraft model showcases decision-making in search and rescue, ad dressing Machine Theory of Mind (MToM). Its success hinges on the mechanistic frame work (e.g., ACT-R), which may have limitations. The model needs to predict intervention effects and manage adaptive human behavior. Overcome in our Design: Personalized cognitive modeling can be adapted for few-shot palm vein verification, even with limited data. Explainability is crucial for high-security biometrics like PULSEID to build trust. Diagnosing underlying causes can improve er ror detection (e.g., False Rejection). Future work could apply cognitive principles to model user interaction, reducing errors like hand misplacement by managing adaptive user placement.

Multi-Objective Evolutionary NAS with Weight-Sharing Su pernet [3]
Methodology: This framework uses a weight-sharing supernet for multi-objective evolutionary NAS. It optimizes net-work and architecture parameters alternately and employs a bi-population MOEA/D to avoid the "small model trap," merging EC-NAS with gradient-based NAS. Evaluation metrics: Validated algorithm efficiency and transferability on real-world image datasets, including CIFAR-10 and Mini-ImageNet. Performance was evaluated against various NAS methods. Limitations: Limitations include: search limited to CNN architectures, population evaluation remains time-consuming, performance lags compared to some manual networks, and the evolutionary algorithm risks premature convergence to small models. Overcome in our Design: This document outlines efficient deep learning strategies for biometric identification (hand vein, PulseID) using weight-sharing supernets, bi population optimization, surrogate models, and NAS framework transferability for NIR images.

Multi-Objective Evolutionary NAS for Recurrent Neural Networks [4]
Methodology: MOE/RNAS proposes a modular RNN architecture with approximate network morphisms to optimize complexity during evolution. Evaluation metrics: We evaluated model accuracy (e.g., character-level prediction) and architecture complexity (e.g., parameters, inference time). Pareto Front generation aided trade-off analysis. Empirical analysis across three datasets validated effectiveness. Limitations: Multi-objective NAS optimizes feed-forward NNs. NN architecture design is complex and time-consuming, balancing accuracy and parameters. Traditional RNNs have vanishing gradients. Overcome in our Design: Develop a multi-objective deep learning model for real-time systems, balancing high accuracy and low computational cost. Use efficient architectures (e.g., LSTM/GRUfor biometric data) and Neural Architecture Search (NAS) to automate model optimization and deployment.

Universal Knowledge Model and Cognitive Architecture for AGI [5]
Methodology: A new cognitive architecture for intelligent systems, aiming for AGI, is proposed. Researchers identified 42 existing architectures lacking necessary functional blocks and suggest Archigraph models (annotated metagraphs) for universal knowledge representation. Evaluation metrics: The article compares its proposed AGI architecture with 42 existing ones, evaluating its functional blocks, universal knowledge model, and structural integrity for AGI capabilities. Limitations: "Prototyping AGI" is a theoretical proposal, not an empirical study. Its subjectively determined functional blocks include machine consciousness and an emotional control system. A "Universal Knowledge Model" presents major implementation and validation challenges. Overcome in our Design: AGI's universal knowledge model guides PulseID's spe cialized biometrics through data fusion (text, images, NIR, metadata) and metagraph structuring for its database and feature matching. Ethical considerations emphasize data privacy and security.

Efficient Multi-Objective Evolutionary Zero-Shot NAS [6]
Methodology: The authors present a framework for evaluating multi-objective evolutionary zero-shot NAS architects using training-free, randomly initialized models and zero-cost metrics. They also propose a general zero-cost metric design principle that unifies existing metrics and allows for efficient multi-metric calculation in a single pass. Evaluation metrics: Architecture quality is assessed by zero-cost metrics. Validation performance is benchmarked on NAS-Bench-201 with MedMNIST. High-throughput performance indicates search process efficiency. Limitations:

DOI: 10.17148/IJARCCE.2026.153107

Current large-dataset validation methods may misrepresent performance on small datasets, hindering optimal architecture evaluation for im- age classification. Overcome in our Design: Quickly prototype and eliminate weak models using a zero cost metric, accelerating model selection. Adapt this for the limited CASIA Palm Vein Dataset with a zero-shot approach. Utilize multi-objective evolution to balance accuracy and model complexity. Implement fast multi-zero-cost metric computation for rapid evaluation of PulseID system complexity.

MFENAS: Multifactorial Evolution for NAS [7]
Methodology: MFENAS, like MFEA, treats Neural Architecture Search (NAS) as a two-factor problem. It solves this by splitting the evolution population into two subgroups for knowledge transfer. Evaluation metrics: Key metrics include efficiency in finding optimal structures under resource constraints and performance against state-of-the-art methods.

Experimental validation uses the NATS-Bench benchmark to compare the quality of optimal structures. Limitations: Current NAS methods are computationally intensive. The algorithm's effectiveness depends on defining and managing two factors accurately. The complexity of handling two evolving populations and their interactions increases overhead. Validation is specific to the NATS-Bench search space. Overcome in our Design: MFENAS optimizes deep learning models for biometric systems like PULSEID by simultaneously considering accuracy and hardware constraints (inference speed, parameters). It uses knowledge transfer to quickly find optimal DL models, reducing NAS training time and GPU cost. Graph-in-Graph Evolutionary NAS for CNNs [8]

Methodology: The methodology uses a new multi-objective NAS algorithm with graph based solution representation for flexible automatic design. It introduces new ad-hoc crossover and mutation operators and speeds up candidate solution evaluation. Evaluation metrics: This multi-objective method optimizes network structure/size and accuracy. New evolutionary operators and the proposed acceleration mechanism for solution evaluation are also validated. Limitations: Traditional NAS methods simplify networks, assuming linearity and focusing on CNNs. They prioritize size reduction and assess performance by accuracy, ignoring complexity. Overcome in our Design: Optimize PulseID accuracy, complexity, and cost using multi-objective optimization. If sequential deep learning is chosen, adapt graph-based representation for modular RNNs. Design limited-size networks for a fast, affordable biometric system and investigate acceleration methods for faster model experimentation.

NSGANetV2: Evolutionary Multi-Objective Surrogate-Assisted NAS [9]
Methodology: This method enhances multi-objective neural architecture search (NAS) by integrating an evolutionary algorithm with two surrogate models. One model predicts performance, and the other utilizes super-net weight-sharing for faster evaluations. This allows optimizing objectives like accuracy, FLOPs, and parameters across architectures while reducing evaluation costs. Evaluation metrics: Machine learning models are evaluated based on classification accuracy, computational cost (parameters, FLOPs), search efficiency (architectures evaluated), and Pareto-front for objective trade-offs. Limitations: NAS is computationally expensive due to weight-sharing and surrogate models. Surrogate model inaccuracies can bias search and limit generalizability. FLOPs and parameter counts often don't reflect real-world performance (latency, energy, memory, quantization). Overcome in our Design: The design prioritizes hardware latency and energy using device-specific profiling or learned latency predictors. It reduces compute costs via weight-sharing and early stopping, limits searches to hardware-feasible architectures, and employs tailored surrogate models for domains like biometric/edge scenarios.

Multi-Objective Evolutionary Design of Deep CNNs for Image Classification [10]
Methodology: This method employs an evolutionary algorithm to automatically design multi-objective convolutional neural network (CNN) architectures, optimizing for accuracy and complexity. It uses genetic programming to evolve CNNs and assesses trade-offs on the Pareto front. Evaluation metrics: Metrics include classification accuracy on image datasets, model complexity (number of parameters / operations), and trade-off fronts among the objectives. The work reports efficient architecture designs under multiple criteria. Limitations: NAS methods are computationally expensive due to the need to evolve and train numerous networks, leading to potential scalability issues with large datasets or hardware, extended search times, and the neglect of hardware-specific constraints. Overcome in our Design: The design optimizes compute via weight-sharing, early stopping, and smaller proxy evaluations. It also considers hardware latency/energy, constrains the search space for target devices, and uses transfer learning and domain adaptation for faster convergence.

NSGA-Net: Neural Architecture Search using Multi-Objective Genetic Algorithm [11]
Methodology: We propose a multi-objective genetic algorithm (NSGA-II) to evolve binary-encoded network architectures. The algorithm optimizes for classification error and computational complexity (FLOPs) using crossover and mutation. A Bayesian Net work built from search history will then be used for exploitation, sampling promising architectures. Candidate architectures are trained on CIFAR-10 to estimate objectives. Evaluation metrics: Neural

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 3, March 2026
DOI: 10.17148/IJARCCE.2026.153107

Architecture Search (NAS) evaluation metrics include ac curacy, computational cost, Pareto front (accuracy-FLOPs trade-off), and search efficiency. Limitations: Neural architecture search (NAS) faces challenges such as high computational cost, limited dataset diversity (mainly CIFAR-10), the inaccuracy of FLOPs as a performance metric, and increased complexity due to many hyperparameters. Overcome in our Design: Systems are optimized through five design adaptations: reducing compute (smaller proxy evaluations, early-stopping, parameter sharing), integrating real-device latency, simplifying search space, accelerating evaluation with transfer learning, and ensuring robustness (biometric/edge applications) via a specific objective.

Efficient Multi-objective Neural Architecture Search via Lamarckian Evolution [12]
Methodology: This proposal details an evolutionary multi-objective Neural Architecture Search (NAS) method using Lamarckian inheritance. It accelerates evaluation by having offspring networks inherit trained weights. The algorithm employs a steady-state NSGA-II framework to optimize accuracy and model complexity. Candidate architectures are mutated, trained briefly, and ranked by Pareto dominance to evolve a population achieving high accuracy and low cost. Evaluation metrics: Various architectures were evaluated on CIFAR-10/100, showing competitive accuracy, smaller model sizes, fewer training epochs, and better computational efficiency than reinforcement-learning NAS baselines. Limitations: Key challenges include expensive population training, limited exploration from incremental mutations, reliance on proxy datasets over large-scale tasks like Ima geNet, and neglecting real-time latency or hardware. Overcome in our Design: The document details model optimization via weight sharing, early stopping, extended objectives for edge deployment, a hybrid Lamarckian–Bayesian approach, and applying these to domain-specific datasets with customized constraints for efficient searching.

III. METHODOLOGY

The proposed Self-Evolving Cognitive Architecture (SECA) is designed as a dual-layered system that integrates structural optimization with cognitive modeling. The methodology consists of two tightly coupled components: (1) Structural Evolution Layer and (2) Cognitive Modeling Layer, operating within a closed-loop optimization framework.

A. Overall Framework

SECA operates through an iterative evolutionary cycle where neural architectures are continuously generated, evaluated, and refined. Simultaneously, a cognitive layer evaluates higher-order reasoning capabilities and feeds this information back into the optimization process. This dual feedback mechanism ensures both computational efficiency and cognitive adaptability.

B. Structural Evolution Layer

The Structural Evolution Layer is responsible for autonomous neural architecture search using Genetic Algorithms (GA). Each neural network architecture is encoded as a genome representing its structural configuration.

1) Genome Encoding

Each genome represents a candidate neural network and includes:

- Number of layers and blocks
- Type of layers (Convolutional, Dense, etc.)
- Kernel sizes and activation functions
- Connectivity patterns (skip connections, residual links)
- Hyperparameters such as learning rate and dropout

2) Population Initialization

An initial population of diverse architectures is generated randomly within predefined constraints such as maximum parameters and computational limits.

3) Fitness Evaluation

Each genome is converted into a neural network model and trained using TensorFlow/Keras. The fitness score is computed based on:

- Validation accuracy
- Model complexity (parameters, FLOPs)
- Training stability

A composite fitness function is defined as:

$$F = Acc - \alpha \log(Params) - \beta \log(FLOPs)$$

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 3, March 2026

DOI: 10.17148/IJARCCE.2026.153107

4) Genetic Operators

The evolution process applies the following operators:

- **Selection:** Tournament or rank-based selection of high-performing architectures
- **Crossover:** Exchange of architectural components between parent genomes
- **Mutation:** Random modification of layers, parameters, or connections

Elitism is used to retain top-performing architectures across generations.

C. Cognitive Modeling Layer

The Cognitive Modeling Layer introduces consciousness-inspired mechanisms using Theory of Mind (ToM) and recursive self-modeling.

1) Self-Modeling

Each AI agent maintains an internal representation of its performance, uncertainty, and decision-making process. This enables introspection and self-evaluation.

2) Theory of Mind (ToM)

The system models the internal states of other agents or environments, enabling:

- Prediction of external behaviors
- Adaptive decision-making
- Social reasoning capabilities

3) Recursive Reasoning

The system performs iterative reasoning such as:

"I think that the system thinks that I will act..."

This recursive modeling improves decision-making in multi-agent environments.

D. Integration Mechanism

The integration of both layers forms a feedback loop:

1. The Structural Layer generates candidate architectures
2. The Cognitive Layer evaluates reasoning and awareness
3. Cognitive metrics are incorporated into the fitness function
4. The system evolves both structure and cognition simultaneously

The enhanced fitness function becomes:

$$F = Acc - \alpha \log(Params) - \beta \log(FLOPs) + \gamma(ToM)$$

E. Optimization Strategy

To improve efficiency and scalability, the following techniques are used:

- Low-fidelity training for early-stage evaluation
- Weight inheritance to reuse learned parameters
- Parallel evaluation of multiple architectures
- Caching of evaluated genomes to avoid redundancy

F. Algorithmic Workflow

The overall methodology can be summarized as:

1. Initialize population of neural architectures
2. Train and evaluate each candidate
3. Compute fitness scores
4. Select top-performing architectures
5. Apply crossover and mutation
6. Integrate cognitive evaluation
7. Repeat until convergence

This iterative process enables SECA to evolve architectures that are both computationally efficient and cognitively adaptive.

IV. WORKING MECHANISM

The working mechanism of the Self-Evolving Cognitive Architecture (SECA) is based on a continuous closed-loop interaction between structural evolution and cognitive modeling. The system operates iteratively, enabling simultaneous

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 3, March 2026
DOI: 10.17148/IJARCCE.2026.153107

optimization of neural architectures and development of higher-order cognitive abilities.

A. Overall Workflow

SECA follows an evolutionary loop consisting of initialization, evaluation, evolution, cognitive reasoning, and feedback integration. The process ensures that both structural and cognitive improvements occur progressively over multiple generations.

B. Step 1: Initialization

The system begins by generating an initial population of neural network architectures. Each architecture is encoded as a genome representing its structural configuration, including layers, parameters, and connectivity.

A diverse population is created to ensure broad exploration of the search space.

C. Step 2: Model Construction and Training

Each genome is decoded into a neural network model using TensorFlow/Keras. The model is then trained on the given dataset for a limited number of epochs (low-fidelity training) to estimate its performance efficiently.

D. Step 3: Performance Evaluation

After training, each model is evaluated using validation data. The evaluation includes:

- Accuracy or task-specific performance metric
- Model complexity (number of parameters and computational cost)
- Training stability and convergence behavior

These metrics are combined to compute the fitness score of each architecture.

E. Step 4: Cognitive Evaluation

The Cognitive Modeling Layer is applied to each candidate architecture. This step introduces intelligence beyond performance metrics.

1) Self-Assessment

The system evaluates its own predictions, uncertainty, and performance limitations.

2) Theory of Mind Reasoning

The model simulates external agents or environments and predicts their behavior.

3) Decision Analysis

The system generates reasoning traces explaining why certain outputs were produced.

These cognitive outputs are quantified and integrated into the overall fitness.

F. Step 5: Fitness Calculation

A combined fitness score is computed by incorporating both structural and cognitive metrics:

$$F = \text{Acc} - \alpha \log(\text{Params}) - \beta \log(\text{FLOPs}) + \gamma(\text{Cognitive Score})$$

This ensures that the system prioritizes not only accuracy but also efficiency and intelligence.

G. Step 6: Selection of Parent Architectures

High-performing architectures are selected based on their fitness scores using selection strategies such as:

- Tournament selection
- Rank-based selection

The best-performing individuals are preserved using elitism.

H. Step 7: Genetic Evolution

New architectures are generated using genetic operators:

- **Crossover:** Combines structural components from two parent architectures
- **Mutation:** Introduces random variations such as adding/removing layers or modifying parameters

These operations enable exploration of new architectures while retaining beneficial traits.

I. Step 8: Integration of Cognitive Feedback

The Cognitive Layer provides feedback on reasoning quality, uncertainty handling, and adaptability. This feedback influences:

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 3, March 2026

DOI: 10.17148/IJARCCE.2026.153107

- Fitness evaluation
- Mutation strategies
- Selection priorities

This creates a feedback loop where cognition directly impacts structural evolution.

J. Step 9: Iterative Optimization Loop

Steps 2 to 8 are repeated over multiple generations. With each iteration:

- Architectures become more efficient
- Cognitive reasoning improves
- The system adapts to task complexity

K. Step 10: Convergence and Output

The process continues until:

- A maximum number of generations is reached
- Performance improvement

stabilizes The final output includes:

- Optimized neural architecture
- Trained model weights
- Cognitive reasoning outputs (self-awareness, ToM insights)

L. Summary of Working Mechanism

The SECA system operates as a dual optimization loop:

- Structural Loop: Evolves neural architectures for efficiency and performance
- Cognitive Loop: Enhances reasoning, awareness, and adaptability

Together, these loops enable the system to evolve both its computational structure and its cognitive intelligence, forming a foundation for autonomous and adaptive AI systems.

V. INPUTS AND OUTPUTS

The Self-Evolving Cognitive Architecture (SECA) requires a combination of data-driven inputs and configuration parameters to operate effectively. The system produces both structural and cognitive outputs, reflecting its dual-layered design.

A. Inputs

The inputs to SECA can be categorized into three primary components:

1) Task Data

Task-specific datasets are essential for training and evaluating candidate neural architectures. These include:

- Training dataset (e.g., images, text, or sensor data)
- Validation dataset for performance evaluation
- Optional test dataset for final assessment

The nature of the dataset determines the type of neural architectures evolved by the system.

2) Evolutionary Configuration Parameters

These parameters define the behavior of the Structural Evolution Layer:

- Population size (number of candidate architectures)
- Number of generations (iterations of evolution)
- Mutation rate and crossover probability
- Selection strategy (tournament, rank-based)
- Constraints (maximum parameters, FLOPs, latency)

These parameters control the exploration and optimization of the architecture search space.

3) Cognitive Layer Configuration

The Cognitive Modeling Layer requires initialization parameters for reasoning and awareness:

- Definition of agents for Theory of Mind (ToM)
- Self-modeling rules and recursive reasoning depth
- Cognitive evaluation metrics (e.g., confidence estimation, ToM accuracy)

These inputs enable the system to simulate cognitive processes and introspective reasoning.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 3, March 2026

DOI: 10.17148/IJARCCE.2026.153107

B. Outputs

The outputs of SECA are divided into two main categories corresponding to its structural and cognitive components.

1) Structural Outputs

The Structural Evolution Layer produces optimized neural architectures:

- Final neural network architecture (layer types, depth, connectivity)
- Optimized hyperparameters (learning rate, dropout, activation functions)
- Trained model weights

These outputs represent the best-performing models discovered through the evolutionary process.

2) Cognitive Outputs

The Cognitive Modeling Layer produces higher-level intelligence outputs:

- Self-awareness metrics (confidence, uncertainty estimation)
- Theory of Mind (ToM) reasoning outputs
- Decision traces explaining model behavior
- Emergent behaviors in multi-agent environments

These outputs demonstrate the system's ability to reason about itself and other agents.

C. Input–Output Mapping

The relationship between inputs and outputs in SECA can be summarized as follows:

- Task data influences model training and performance evaluation
- Evolutionary parameters guide architecture optimization
- Cognitive configurations enable reasoning and awareness

These inputs are processed through iterative evolution and cognitive evaluation to generate optimized architectures and intelligent behaviors.

D. Example Input–Output Flow

Input:

- Dataset: CIFAR-10 (image classification)
- Population size: 10 architectures
- Mutation rate: 0.2, Crossover rate: 0.7
- Cognitive module enabled with ToM reasoning

Output:

- Optimized model: 20-layer convolutional neural network
- Validation accuracy: 94%
- Cognitive output: Improved confidence estimation and agent behavior prediction

E. Summary

In summary, SECA transforms raw data and configuration inputs into both optimized neural architectures and advanced cognitive outputs. This dual-output capability distinguishes SECA from traditional AI systems, enabling it to evolve not only efficient models but also intelligent, self-aware behaviors.

VI. APPLICATIONS

The Self-Evolving Cognitive Architecture (SECA) has wide-ranging applications across multiple domains due to its ability to combine autonomous structural optimization with cognitive intelligence. Its adaptability and self-awareness enable it to function effectively in complex and dynamic environments.

A. Adaptive Artificial General Intelligence (AGI)

SECA provides a foundation for developing AGI systems that can evolve both their architecture and reasoning capabilities.

B. Robotics and Autonomous Systems

SECA enables robots to dynamically optimize their control architectures and improve decision-making in real-time environments.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 3, March 2026

DOI: 10.17148/IJARCCE.2026.153107

Figure 1: Applications

C. Human-AI Interaction

SECA enhances human-AI collaboration using Theory of Mind (ToM), allowing systems to understand human intentions and behavior.

D. Autonomous Research Systems

SECA enables AI systems to generate hypotheses, design experiments, and refine models autonomously.

E. Complex Decision-Making Systems

SECA is suitable for solving large-scale problems such as financial forecasting, policy analysis, and multi-agent coordination.

F. Smart Healthcare Systems

SECA can assist in diagnosis, treatment planning, and intelligent patient monitoring systems.

float ... Applications of SECA across domains including Adaptive AGI, Robotics, Human-AI Interaction, Autonomous Research, Smart Healthcare, and Complex Decision-Making Systems

VII. IMPACT

The proposed Self-Evolving Cognitive Architecture (SECA) introduces a transformative paradigm in artificial intelligence by integrating autonomous structural optimization with cognitive awareness. This dual capability has significant implications across technological, scientific, and societal domains.

A. Acceleration of Artificial General Intelligence (AGI)

SECA contributes directly to the advancement of Artificial General Intelligence by enabling systems that can evolve both their neural architectures and cognitive reasoning capabilities. Unlike traditional AI systems limited to predefined structures, SECA dynamically adapts to new tasks, thereby reducing the gap between narrow AI and general intelligence.

B. Reduction of Human Dependency

The automation of neural architecture design eliminates the need for extensive human expertise in model development. By enabling AI systems to design and optimize themselves, SECA reduces development time, cost, and reliance on domain experts, thereby democratizing access to advanced AI technologies.

C. Enhanced Adaptability and Robustness

SECA enhances system robustness by allowing continuous adaptation to changing environments and tasks. The integration of cognitive modeling enables the system to assess its own performance and adjust accordingly, making it resilient in uncertain and dynamic scenarios.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 3, March 2026
DOI: 10.17148/IJARCCE.2026.153107

D. Improved Human-AI Collaboration

Through the incorporation of Theory of Mind (ToM) and self-modeling, SECA enables AI systems to understand human intentions and behavior. This leads to more natural, effective, and context-aware interactions in applications such as virtual assistants, healthcare, and decision support systems.

E. Advancement in Autonomous Systems

SECA enables the development of highly autonomous systems capable of self-optimization and intelligent decision-making. This has applications in robotics, autonomous vehicles, and industrial automation, where systems must operate independently with minimal human intervention.

F. Acceleration of Scientific Discovery

By enabling AI systems to generate hypotheses, evolve models, and refine reasoning processes autonomously, SECA can significantly accelerate research and innovation. It has the potential to transform domains such as medicine, physics, and environmental science.

G. Economic and Societal Impact

The widespread adoption of SECA can lead to increased productivity, reduced operational costs, and the creation of new technological opportunities. However, it also necessitates careful consideration of ethical challenges, including transparency, accountability, and control over autonomous systems.

H. Summary

Overall, SECA represents a foundational step toward scalable, autonomous, and intelligent AI systems. Its ability to unify structural optimization with cognitive awareness positions it as a key enabler for the next generation of artificial intelligence technologies.

VIII. CONCLUSION

This paper presented the design and conceptual framework of a Self-Evolving Cognitive Architecture (SECA), a novel approach that integrates autonomous neural architecture optimization with consciousness-inspired cognitive modeling. By combining evolutionary algorithms with Theory of Mind (ToM) and recursive self-modeling, SECA enables the simultaneous evolution of both computational structure and cognitive capabilities.

Unlike traditional artificial intelligence systems that rely on fixed architectures or task-specific optimization, SECA introduces a dual-layered paradigm in which neural networks continuously adapt their structure while developing higher-order reasoning abilities. This integration allows the system to achieve improved performance, adaptability, and interpretability in dynamic and complex environments.

The proposed framework demonstrates significant potential in advancing Artificial General Intelligence (AGI) by bridging the gap between efficiency-driven optimization and cognition-driven intelligence. Furthermore, SECA reduces dependency on manual model design and opens new possibilities for autonomous systems capable of self-improvement and introspection.

Future work will focus on implementing large-scale experimental validation, integrating multi-modal learning capabilities, and developing robust evaluation metrics for cognitive intelligence. Additionally, ethical considerations and control mechanisms will be explored to ensure safe and responsible deployment of such autonomous systems.

In conclusion, SECA represents a foundational step toward the development of scalable, adaptive, and self-aware artificial intelligence systems, with far-reaching implications across scientific, industrial, and societal domains.

ACKNOWLEDGMENT

The authors would like to express their sincere gratitude to the faculty and management of their institution for providing the necessary support and resources to carry out this research.

We extend our heartfelt thanks to our project guide and mentors for their continuous guidance, valuable suggestions, and encouragement throughout the development of this work.

We also acknowledge the contributions of the open-source community, particularly the developers of TensorFlow and Keras, whose tools and frameworks played a crucial role in the implementation of this project.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 3, March 2026
DOI: 10.17148/IJARCCE.2026.153107

Finally, we thank our peers and colleagues for their constructive feedback and support, which helped improve the quality of this research.

REFERENCES

[1]. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, ”Communication-Efficient Learning of Deep Networks from Decentralized Data,” in *Proc. Artificial Intelligence and Statistics (AISTATS)*, 2017.
[2]. B. Zoph and Q. V. Le, ”Neural Architecture Search with Reinforcement Learning,” in *Proc. International Confer- ence on Learning Representations (ICLR)*, 2017.
[3]. H. Pham, M. Guan, B. Zoph, Q. Le, and J. Dean, ”Efficient Neural Architecture Search via Parameter Sharing,” in *Proc. ICML*, 2018.
[4]. E. Real et al., ”Large-Scale Evolution of Image Classifiers,” in *Proc. ICML*, 2017.
[5]. T. Elsken, J. H. Metzen, and F. Hutter, ”Neural Architecture Search: A Survey,” *Journal of Machine Learning Research*, vol. 20, no. 55, pp. 1–21, 2019.
[6]. J. Schmidhuber, ”Deep Learning in Neural Networks: An Overview,” *Neural Networks*, vol. 61, pp. 85–117, 2015.
[7]. I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.
[8]. D. Silver et al., ”Mastering the Game of Go with Deep Neural Networks and Tree Search,” *Nature*, vol. 529, pp. 484–489, 2016.
[9]. A. Vaswani et al., ”Attention Is All You Need,” in *Proc. NeurIPS*, 2017.
[10]. T. Brown et al., ”Language Models are Few-Shot Learners,” in *Proc. NeurIPS*, 2020.
[11]. S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 4th ed. Pearson, 2020.
[12]. M. Wooldridge, *An Introduction to MultiAgent Systems*, 2nd ed. Wiley, 2009.
[13]. C. Frith and U. Frith, ”Theory of Mind,” *Current Biology*, vol. 15, no. 17, pp. R644–R645, 2005.
[14]. K. Simonyan and A. Zisserman, ”Very Deep Convolutional Networks for Large-Scale Image Recognition,” in *Proc. ICLR*, 2015.
[15]. M. Tan and Q. Le, ”EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks,” in *Proc. ICML*, 2019.

