# Homogeneous Architecture Augmentation and Confidence Prediction for Evolutionary Neural Architecture Search

Pengcheng Jiang
*School of Software*
*Nanjing University of Information Science and Technology*
Nanjing, China
pcjiang@nuist.edu.cn

Yu Xue
*School of Software*
*Nanjing University of Information Science and Technology*
Nanjing, China
xueyu@nuist.edu.cn

Ferrante Neri
*School of Computer Science and Electronic Engineering*
*University of Surrey*
Guildford, United Kingdom
f.neri@surrey.ac.uk

2025 IEEE Congress on Evolutionary Computation (CEC) | 979-8-3315-3431-8/25/$31.00 ©2025 IEEE | DOI: 10.1109/CEC65147.2025.11043091

***Abstract*—Evolutionary neural architecture search (ENAS) automates the design of high-performing neural networks but is often hindered by the high computational cost of evaluating individual architectures. Surrogate models mitigate this issue by predicting performance, yet their accuracy depends on the quality of training data and their ability to utilise insights from real evaluations. This paper presents homogeneous encoding-based ENAS (HENAS), a novel method addressing these challenges through two key innovations: homogeneous architecture augmentation and confidence-based prediction. Through homogeneous architecture augmentation, HENAS exploits redundant encodings in the MobileNetV3 search space to generate multiple representations of the same architecture, enhancing the surrogate model's training data without additional cost. Confidence-based prediction introduces a mechanism to identify architectures with uncertain performance estimates, prioritising them for evaluation. Integrated into an evolutionary framework, these techniques improve search efficiency and exploration. Experiments on CIFAR-10, CIFAR-100, and ImageNet show that HENAS achieves state-of-the-art performance with reduced computational expense. Ablation studies confirm the contributions of its core components, highlighting the value of redundancy exploitation and uncertainty management in surrogate-assisted ENAS.**

***Index Terms*—Evolutionary computation, neural architecture search, surrogate model, data augmentation, confidence level.**

## I. Introduction

The (semi-)automated design of neural networks, framed as an optimisation problem—commonly referred to as neural architecture search (NAS)—has recently gained popularity as a strategy for creating machine learning models. This is particularly evident in convolutional neural networks (CNNs) for image classification. Among NAS approaches, a notable category treats architecture search as a global optimisation problem, leveraging evolutionary computation (EC)-based methods [1], [2]. This category, known as evolutionary neural architecture search (ENAS) [3], [4], applies principles of evolutionary algorithms to explore the search space.



ENAS distinguishes itself by employing population-based methods that iteratively manipulate and evaluate candidate architectures. However, the need to calculate the objective function for each individual in the population makes this process computationally expensive. To address this limitation, contemporary ENAS methods utilise proxy-based evaluation strategies, which, as reported in [5] broadly belong to the following categories.

- **Indirect proxy methods** use techniques such as weight inheritance, where knowledge from previously trained architectures is reused to accelerate real evaluations.
- **Surrogate models (direct proxy methods)**, on the other hand, employ machine learning models to predict architecture performance, entirely bypassing the need for computationally expensive real evaluations.

By integrating these proxy-based approaches, ENAS significantly reduces computational costs while maintaining effectiveness in searching for optimal neural architectures.

Current surrogate evaluation methods rely heavily on training data, and their performance is strongly influenced by data quantity. To enhance accuracy without increasing evaluation costs, some studies propose methods to expand training datasets. For example, Luo *et al.* [6] use semi-supervised learning to generate unlabelled architecture samples and assign pseudo-labels using the surrogate model. However, this approach depends heavily on the reliability of pseudo-labels. An alternative strategy involves analysing encoding to identify redundancies within the search space, enabling dataset expansion. Liu *et al.* [7] employ adjacency matrices to encode architectures, using isomorphic matrices to represent identical networks, thereby generating additional training data at no extra cost. Xie *et al.* [8] further extend this method by applying isomorphic generation to entire adjacency matrices. Despite its potential, this approach has primarily been explored in graph-based search spaces and requires further study in broader search spaces. Additionally, most research focuses on improving the surrogate model or training phase, often

overlooking the prediction phase. Jiang *et al.* [9] demonstrate that using isomorphic representations during prediction significantly enhances surrogate model performance. Thus, exploring improvements to the prediction phase and applying these techniques to diverse search spaces remain promising research directions.

Although surrogate-assisted ENAS methods effectively search for promising architectures, the surrogate model still underutilises real evaluations during the evolutionary process. Additionally, due to high computational costs in NAS, the number of evaluations typically does not exceed 500 [7], [9], [10]. In this context, selecting suitable architectures for training to expand the surrogate dataset is crucial, as it directly impacts the surrogate model's performance and the final architectures discovered.

In this paper, we propose a novel ENAS algorithm, termed Homogeneous Architecture Augmentation and Confidence Prediction (HENAS). We begin by analysing redundant representations in the MobileNetV3 search space and introduce a **homogeneous architecture augmentation method** that increases the amount of real data for surrogate model training, without incurring additional evaluation costs. Furthermore, we propose **a confidence-based prediction method** to identify architectures with high uncertainty in the population, leveraging real evaluations of these architectures to significantly improve the surrogate model's performance. Finally, we integrate these two innovations into an evolutionary framework to enhance the search for superior architectures. The key contributions of this work are as follows:

1) We design a homogeneous architecture augmentation method that generates multiple homogeneous representations of the same architecture using redundant encodings in the search space, enriching the surrogate dataset and improving model performance without adding computational overhead.
2) We develop a confidence prediction method that estimates the accuracy of pending architectures by predicting their performance based on homogeneous representations, thus providing the surrogate evaluation and confidence level for each architecture.
3) By combining homogeneous architecture augmentation with confidence prediction, we refine the traditional evolutionary algorithm framework, prioritising architectures with low confidence for real evaluation, which further enhances surrogate model training.

The remainder of this paper is organised as follows. Section II briefly discusses the computational challenges of NAS and reviews methods from the literature that address these issues. Section III introduces the proposed HENAS framework. Section IV-B presents the experiments conducted in this study. Finally, Section V concludes the paper and outlines directions for future work.

## II. Related Work

### A. Computational cost of neural architecture search

Traditional NAS methods are computationally expensive, requiring extensive training of numerous candidate architectures to evaluate performance [11]. This high cost has sparked significant efforts in both academia and industry to find more efficient alternatives. Reinforcement learning (RL)-based NAS, while popular, often suffers from poor search efficiency, resulting in long search times and limiting practical use [12]. Gradient-based methods, especially those incorporating attention mechanisms to guide architecture selection, have gained attention. However, they can get trapped in suboptimal solutions, often favouring architectures with imbalanced or sparse parameters [13], requiring further adjustments in the search strategy. In contrast, EC-based methods stand out by excelling in discrete optimization, offering robust global exploration capabilities and leveraging historical data, making them particularly well-suited for NAS [14].

### B. Proxy-based ENAS

In NAS, as mentioned above, two commonly employed proxy-based methods are direct proxy (surrogate models) and indirect proxy, both of which can complement each other. Direct proxy involves directly predicting the fitness of an individual. For instance, Liu *et al.* utilised random forests to predict the accuracy of architectures [7], while Wang *et al.* used support vector machines (SVM) to determine the superiority relationship between candidate architectures [4]. Other studies have expanded on the concept of superiority relationships, developing architectural scores derived from accuracy rankings [10], [15], [16]. These methods leverage machine learning models to reduce the need for extensive real evaluations, thus conserving resources.

In contrast, indirect proxy methods aim to reduce the training time of architectures, using the classification accuracy of under-trained models as an approximate fitness value. Techniques such as weight inheritance and low-fidelity training are commonly used. For example, Chu *et al.* employed a supernet with shared weights, allowing each individual to inherit weights from corresponding modules, thus shortening the training time [17]. Both direct and indirect proxy methods can be combined for greater efficiency. Lu *et al.* demonstrated this by using an ensemble of traditional machine learning models to predict classification accuracy, where the highest-performing individual inherited weights from the supernet and underwent rapid evaluation [18]. Similarly, Xue *et al.* utilised SVM to establish relationships between paired architectures, and through non-dominated sorting based on predicted relationships and computational complexity, accelerated real evaluations by using weight inheritance for certain individuals [19].

## III. Methodology: HENAS

This section presents a detailed description of the proposed homogeneous encoding-based evolutionary neural architecture search (HENAS) method. We begin by outlining the overall

**Algorithm 1:** Overall framework of HENAS

**Input:** The number of initial archive $N$, the number of generations $maxGen$, the generation for starting updating the surrogate model $S_{start}$, the interval between two updates $S_{step}$ and the total counts of updates $S_t$, the number of individuals adding to archive at each time $K$, the number of architecture augmentation for each individual $numAug$, the number of individuals which is added to low-confidence list $numConf$.

1 $\mathcal{D} \leftarrow \{\}$; $\mathcal{C} \leftarrow \{\}$; $G \leftarrow 0$; $S_{count} \leftarrow 0$;
2 Sample $N$ candidate architectures from the search space and get the initial population $P$;
3 Conduct real evaluations for $P$, and add the records to the archive $\mathcal{D}$;
4 $model \leftarrow$ Train the surrogate model with $\mathcal{D}$;
5 $P.fitness \leftarrow$ Evaluate $P$ with surrogate model;
6 **do**
7 $Q \leftarrow$ Generate offspring population using $P$;
8 $Q.fitness, Q.conf \leftarrow$ Evaluate $Q$ with surrogate model;
9 Select the individuals with higher fitness than the average with sorted confidence values;
10 Add the $numConf$ individuals with worst confidence into the $C$;
11 **if** $(G - S_{start})\%S_{step} = 0$ *and* $S_{count} < S_t$ **then**
12 $H \leftarrow$ Select $K$ individuals from $P$ and $Q$ with best fitness;
13 Select the $numConf$ individuals with worst confidence from $C$ and move them into $H$;
14 Conduct real evaluations for $H$, and add the records to the archive $\mathcal{D}$;
15 $model \leftarrow$ Update the surrogate model with $\mathcal{D}$;
16 Predict and update the confidence values for $C$;
17 $S_{count} \leftarrow S_{count} + 1$;
18 **end**
19 $P \leftarrow$ Environmental selection with fitness of $P$ and $Q$;
20 $G \leftarrow G + 1$;
21 **while** $G < maxGen$;
22 Get the best individual $Indi$ as the final architecture;
23 **return** $Indi$.

framework of the algorithm in Section III-A. Next, Section III-B discusses the search space and the encoding strategy used. In Section III-C, we introduce the homogeneous architecture augmentation method. Finally, Section III-D presents the confidence prediction method.

### A. Overall framework

The framework of the proposed method is detailed in Algorithm 1. Initially, two empty sets are created: the archive dataset $\mathcal{D}$ for training the surrogate model and the list $\mathcal{C}$

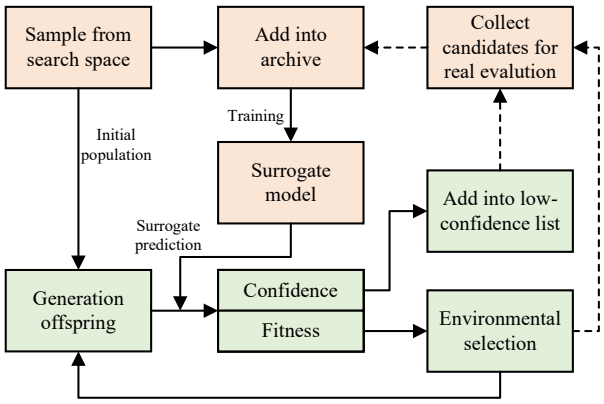


Fig. 1. Framework of the Homogeneous Encoding-based Evolutionary Neural Architecture Search (HENAS). The framework incorporates a periodic update of the surrogate model, indicated by the dashed line. This process includes both the evaluation of the offspring generation and the selection of individuals based on their fitness and confidence levels, as well as the real evaluations that periodically expand the training data for the surrogate model.

containing individuals with low confidence (line 1). At the beginning, $N$ individuals are randomly sampled from the search space, evaluated through real experiments, and recorded in the archive $\mathcal{D}$ (lines 2-3). The surrogate model is then trained using the data in $\mathcal{D}$ (line 4) and applied throughout the evolutionary process (lines 5 and 8). During the evolutionary phase, the parent population $P$ generates offspring population $Q$ through uniform crossover and mutation (line 7). The surrogate model evaluates their fitness, which is used for environmental selection. The overall framework is illustrated in Figure 1.

Each time the offspring generation $P$ is evaluated, the surrogate model not only computes the fitness value but also estimates the confidence level of its predictions (line 8). Individuals whose fitness values exceed the average fitness value of the offspring generation are then ranked based on their confidence levels. The individuals with the lowest confidence levels are added to the list $\mathcal{C}$ (lines 9-10).

The surrogate model is periodically updated throughout the process (lines 11-18). Using the fitness values of both the parent generation $P$ and the offspring generation $Q$, the top $K$ individuals, along with the $numConf$ individuals with the lowest confidence levels from list $\mathcal{C}$, are selected for real evaluation and added to the archive $\mathcal{D}$ (lines 12-14). Once evaluated, the corresponding individuals are removed from $\mathcal{C}$. Following this, the surrogate model is retrained using the updated archive $\mathcal{D}$, and the model is then used to re-predict the fitness and update the confidence levels of each individual in $\mathcal{C}$ (lines 15-16).

### B. Search space & encoding strategy

The search space of MobileNetV3, a lightweight backbone network, is employed in this study. A macroscopic representation of the architecture is shown in Figure 2. Each candidate network consists of 5 blocks, with each block containing up

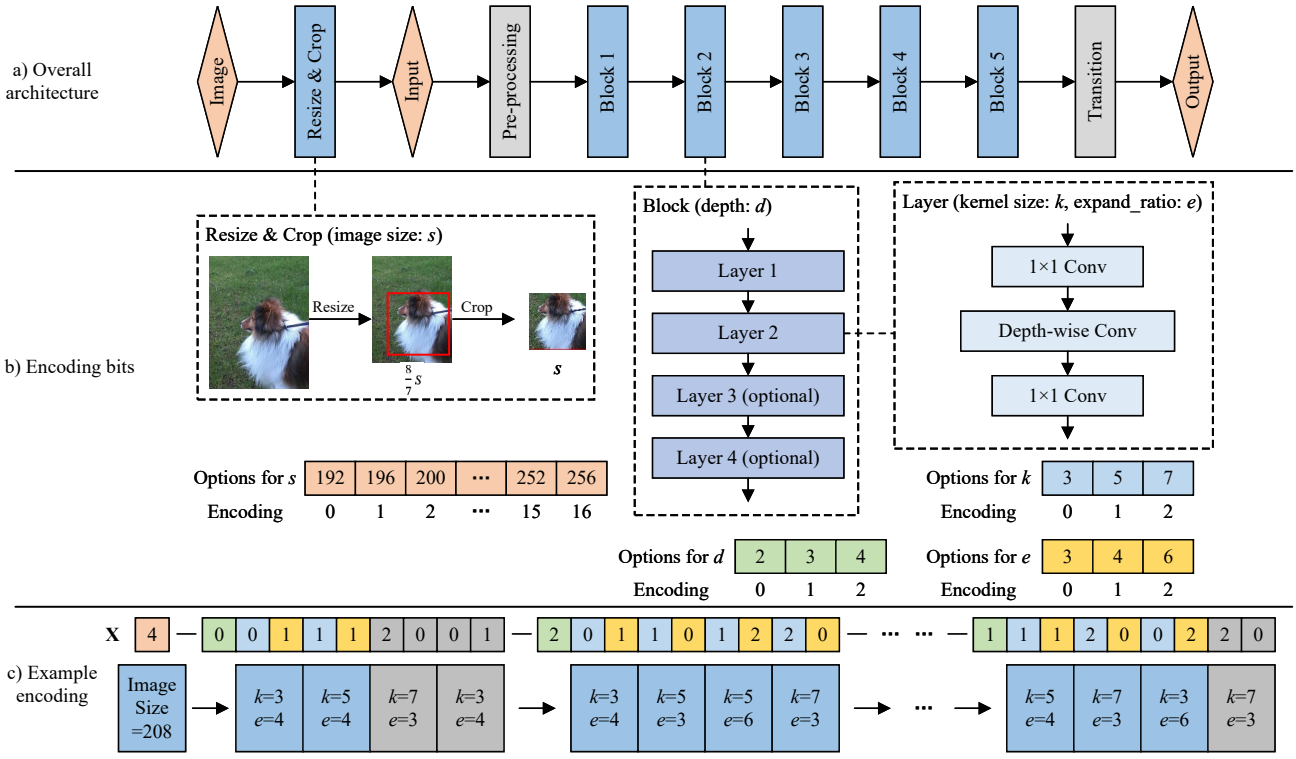


Fig. 2. The MobileNetV3 search space. a) The overall architecture. It contains the image processing, pre-processing layer, transition layer and five blocks. The blue parts is searched by the proposed methods. b) Options in encoding bits and their meanings. c) A sample coding with gray areas indicating unused layers.

to four possible layers. The structure of each layer follows an inverted residual design, which includes:

1) A 1x1 convolution operation for preprocessing.
2) A depth-wise separable convolution.
3) A final 1x1 convolution operation as a linear layer.

Each layer has two configurable parameters: the expansion ratio $e$ and the kernel size $k$, which control the internal convolution kernels. The expansion ratio $e$ can be selected from the set $\{3, 4, 6\}$, while the kernel size $k$ can be chosen from $\{3, 5, 7\}$. Additionally, the depth of each block, denoted by $d$, is also a tunable parameter, with $d \in \{2, 3, 4\}$. A depth of 3 means that one of the layers is skipped, and therefore, the last expansion ratio and kernel size in the block's encoding are not used. When searching for the architecture, the input image size $s$, which influences the classification accuracy of the network, is also considered. The input size $s$ is represented as one of 17 possible resolutions, ranging from 192 to 256 with an interval of 4. In total, the encoding of the search space requires $1 + 5 + 5 \times 4 \times 2 = 46$ bits of integer encoding. These bits represent the various architectural configurations, with integer values replacing continuous values.

### C. Homogeneous architecture augmentation

The encoding strategy for MobileNetV3 reveals that the absence of layers in any block leads to redundant encodings. Furthermore, each additional missing layer exponentially increases the number of distinct representations for the same architecture, resulting in a vast number of redundant encodings in the search space. These redundancies can be leveraged to expand the surrogate model's training dataset without requiring additional real evaluations.

To address this, we propose a homogeneous architecture augmentation method that exploits the redundant encodings in the search space to generate augmented architectures. This method enhances the dataset for training the surrogate model without incurring extra real evaluation costs. For each candidate architecture, the 5 bits representing the number of layers in each block are examined to identify any block that is not fully populated with layers. If a block contains fewer than the maximum number of layers, the last invalid encodings for that block are randomly modified to other values, thereby generating a homogeneous representation of the architecture. This process is repeated for each block, producing a new architecture that is substantially different in terms of its encoding, yet still valid within the search space. Both the newly generated architecture and its original label are then added to the surrogate dataset, contributing additional training data for the surrogate model. Figure 3 illustrates an example of a network with missing layers and the corresponding homogeneous encoding generated through this augmentation process.

### D. Surrogate training & confidence-based prediction

Due to the introduction of the architecture augmentation method, adjustments are required in the surrogate model's

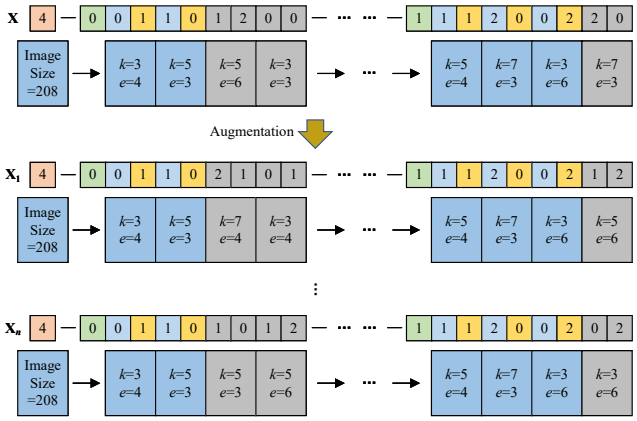


Fig. 3. An example of homogeneous architecture augmentation. The figure demonstrates how the missing layers in a block (shown as invalid encodings) are randomly modified to create new, valid architecture representations. This process generates a new network configuration that is significantly different from the original architecture in its encoding, but retains the same overall structure. The augmented architecture is then added to the surrogate dataset to improve training without requiring additional real evaluations.

training process. Additionally, a confidence-based prediction approach is proposed to further enhance the model's performance.

*Training Phase:* In the training phase of the surrogate model, we begin by collecting the encoding representations of all individuals that have been evaluated in real experiments from the architecture archive. These evaluations form the surrogate training dataset, denoted as $D_{train}$. For each individual in the dataset, we apply the method described in Section III-B to generate $numAug$ homogeneous encoding representations. These augmented encodings are then added to $D_{train}$, enriching the dataset without additional real evaluations.

To ensure the surrogate model performs optimally, we convert each encoding in $D_{train}$ into a binary one-hot representation. This step is necessary to mitigate any performance loss that may arise due to scale differences between the various feature values.

*Prediction Phase:* During the prediction phase, the surrogate model generates $numAug$ homogeneous encoding representations for each architecture, as described in Section III-B. These generated encodings, along with the original encoding, are used to make predictions with the surrogate model. Each architecture to be evaluated thus receives $N + 1$ predicted values.

To obtain a robust fitness value, we calculate the average of these predictions. The standard deviation of the predictions is used to estimate the confidence level of the architecture. A smaller confidence value indicates greater certainty in the predicted result, thus enabling the model to prioritize architectures with higher confidence for further evaluation.

## IV. Experimental Results

To validate the efficacy of the proposed HENAS method, we conducted experiments on various datasets. In Section IV-A, we describe the experimental setup and the datasets used. In Section IV-B, we present the experimental results on standard datasets and compare them with state-of-the-art methods. In Section IV-C, we provide an ablation study to evaluate the contribution of the key modules proposed.

### A. Hyper-parameter settings & datasets

The experiments were conducted within a genetic algorithm framework, with a uniform mutation probability set to 0.1 and a uniform crossover probability set to 0.5. During initialization, we randomly generated and evaluated 100 individuals. The search process spanned 50 generations, with a population size of 300 individuals per generation. In each generation, the 5 individuals with the lowest confidence were added to a separate list.

Starting from the 5th generation, the surrogate model was updated every 5 generations, resulting in 5 updates throughout the experiment. Prior to each update, the 5 individuals with the lowest confidence from the list were evaluated and added to the surrogate dataset. Additionally, the top 64 individuals from the current generation were selected through environmental selection, evaluated, and added to the surrogate dataset. After these steps, the surrogate model was updated.

The surrogate model employed in this study is AdaBoost, using 3000 decision trees as base learners. Three standard datasets were used: CIFAR-10, CIFAR-100, and ImageNet. For real-world evaluation, we leveraged pre-trained weights provided by the OFA [20]. Specifically, we fine-tuned the model for 5 epochs on CIFAR for real-world evaluation, while for ImageNet, we directly used the inherited weights.

### B. Experimental Results on Standard Datasets

We conducted experiments on the CIFAR-10, CIFAR-100, and ImageNet datasets. For each experiment, we present detailed results, including classification accuracy (Acc), multiply-add operations (Madds), and search time. To highlight the advantages of the proposed method, we also provide comparisons with state-of-the-art algorithms, which are presented in the tables.

*1) **Results on CIFAR-10**:* Table I presents the search results on CIFAR-10. The final architecture searched by HENAS achieves a classification accuracy of 98.38%, which marks a significant improvement over most existing methods. Additionally, the resulting network architecture has only 328M Madds, making it computationally efficient. In comparison with other methods that achieve classification accuracy above 97%, our approach offers clear advantages in terms of computational complexity. Despite HENAS requiring 1.8 GPU days for the search, this duration remains practical and acceptable for real-world applications.

*2) **Results on CIFAR-100**:* The experimental results of HENAS on CIFAR-100 are presented in Table II. HENAS outperforms all state-of-the-art methods in terms of classification accuracy, achieving an impressive 87.36%. In terms of computational complexity, HENAS ranks second only to SLE-NAS [35], outperforming all other methods. This suggests that the network architecture found by HENAS not only achieves

TABLE I
COMPARISON WITH STATE-OF-THE-ART METHODS ON THE CIFAR-10 DATASET.

| Method | Acc (%) | Madds (M) | Search Cost (GPU Days) | Search Method |
|---|---|---|---|---|
| MobileNetV2 [21] | 95.74 | 300 | - | manual |
| EfficientNet-B0 [22] | 98.1 | 387 | - | manual |
| ENAS [23] | 97.11 | - | 0.5 | RL |
| BlockQNN [24] | 97.20 | - | 32 | RL |
| NASNet-A [12] | 97.35 | 608 | 1800 | RL |
| sharpDARTS [25] | 97.71 | 357 | 1.8 | GD |
| MixNet-M [26] | 97.9 | 359 | - | GD |
| MixPath-A [27] | 98.1 | 348 | 10 | GD |
| MixPath-B [27] | 98.2 | 377 | 10 | GD |
| DARTS [28] | 97.00 | 547 | 1.5 | GD |
| ADARTS [29] | 97.54 | - | 0.2 | GD |
| PA-DARTS [13] | 97.42 | - | 0.36 | GD |
| AmoebaNet-B [30] | 97.5 | 555 | 3150 | EA |
| CARS [31] | 97.43 | 728 | 0.4 | EA |
| FairNAS-A [17] | 98.2 | 391 | 12 | EA |
| FairNAS-B [17] | 98.1 | 348 | 12 | EA |
| FairNAS-C [17] | 98.0 | 324 | 12 | EA |
| RWE-S (Micro) [32] | 95.95 | 203 | 0.05 | EA |
| RWE-M (Micro) [32] | 96.63 | 249 | 0.05 | EA |
| RWE-L (Micro) [32] | 97.02 | 340 | 0.05 | EA |
| RWE-L (Macro) [32] | 95.73 | 1074 | 0.14 | EA |
| NSGA-Net [33] | 97.98 | 817 | 4 | EA |
| FPSO [34] | 95.16 | 393 | 1.25 | EA |
| SLE-NAS-A [35] | 96.01 | 58 | 0.35 | EA |
| SLE-NAS-B [35] | 96.53 | 248 | 0.4 | EA |
| AE-CNN [1] | 95.3 | - | 27 | EA |
| AE-CNN+E2EPP [36] | 94.7 | - | 7 | EA |
| CNN-GA [3] | 96.78 | - | 35 | EA |
| Evo-OSNet [37] | 97.44 | - | 0.5 | EA |
| SI-EvoNAS [38] | 97.31 | - | 0.458 | EA |
| EffPnet [4] | 96.51 | - | $< 3$ | EA |
| RelativeNAS [39] | 97.66 | - | 0.4 | EA |
| SPNAS [16] | 98.20 | 319 | 1.4 | EA |
| HENAS | 98.38 | 328 | 1.8 | EA |

TABLE II
COMPARISON WITH STATE-OF-THE-ART METHODS ON THE CIFAR-100 DATASET.

| Method | Acc (%) | Madds (M) | Search Cost (GPU Days) | Search Method |
|---|---|---|---|---|
| MobileNetV2 [21] | 80.8 | 300 | - | manual |
| NASNet-A (L) [12] | 86.7 | 12031 | 1800 | RL |
| NASNet-A (M)) [12] | 83.9 | 600 | 1800 | RL |
| ENAS [23] | 81.09 | - | 0.5 | RL |
| MixNet-M [26] | 87.1 | 359 | - | GD |
| ADARTS [29] | 82.94 | - | 0.2 | GD |
| PA-DARTS [13] | 83.03 | - | 0.36 | GD |
| MUXNet-M [40] | 86.11 | 400 | 11 | EA |
| AE-CNN [1] | 77.60 | - | 36 | EA |
| AE-CNN+E2EPP [36] | 77.98 | - | 10 | EA |
| CNN-GA [3] | 79.47 | - | 40 | EA |
| ZenNet [41] | 84.4 | 487 | 0.5 | EA |
| FairNAS-A [17] | 87.3 | 391 | 12 | EA |
| FairNAS-B [17] | 87 | 348 | 12 | EA |
| FairNAS-C [17] | 86.7 | 324 | 12 | EA |
| NSGA-Net [33] | 85.58 | 817 | 4 | EA |
| SI-EvoNAS [38] | 84.30 | - | 0.813 | EA |
| EffPnet [4] | 81.51 | - | $< 3$ | EA |
| Evo-OSNet [37] | 84.16 | - | 0.5 | EA |
| SLE-NAS-A [35] | 78.76 | 58 | 0.4 | EA |
| SLE-NAS-B [35] | 81.93 | 208 | 0.4 | EA |
| RelativeNAS [39] | 84.14 | - | 0.4 | EA |
| SPNAS [16] | 87.26 | 351 | 1.6 | EA |
| HENAS (ours) | 87.36 | 297 | 1.68 | EA |

TABLE III
COMPARISON WITH STATE-OF-THE-ART METHODS ON THE IMAGENET DATASET.

| Method | Top-1 Acc (%) | Top-5 Acc (%) | Madds (M) | Search Cost (GPU Days) | Search Method |
|---|---|---|---|---|---|
| MobileNetV2 [21] | 72.0 | 91.0 | 300 | - | manual |
| EfficientNet-B0 [22] | 76.3 | 93.2 | 390 | - | manual |
| EfficientNet-B1 [22] | 78.8 | 94.4 | 700 | - | manual |
| NASNet-A [12] | 74.0 | 91.6 | 564 | 1800 | RL |
| NASNet-B [12] | 72.8 | 91.3 | 488 | 1800 | RL |
| NASNet-C [12] | 72.5 | 91.0 | 558 | 1800 | RL |
| MnasNet [42] | 76.13 | 92.85 | 391 | - | RL |
| Shapely-NAS [43] | 76.1 | - | 582 | 4.2 | GD |
| MixPath-A [27] | 76.9 | 93.4 | 349 | 10.3 | GD |
| MixPath-B [27] | 77.2 | 93.5 | 378 | 10.3 | GD |
| PDARTS-AER [44] | 75.7 | 92.8 | 587 | 0.3 | GD |
| PDARTS-AER [44] | 76.0 | 92.8 | 578 | 2.0 | GD |
| PA-DARTS [13] | 75.3 | 92.25 | - | 0.4 | GD |
| Once-For-All [20] | 76.9 | 93.2 | 230 | 2 | EA |
| MUXNet [40] | 76.6 | 93.2 | 318 | 11 | EA |
| CARS [31] | 75.2 | 92.5 | 591 | 0.4 | EA |
| NSGANetV1 [33] | 76.9 | 93.0 | 292.5 | 12 | EA |
| FairNAS [17] | 77.5 | - | 392 | 12 | EA |
| SI-EvoNAS [38] | 75.8 | 92.59 | - | 0.458 | EA |
| MFENAS [45] | 73.94 | 91.82 | - | 0.6 | EA |
| Evo-OSNet [37] | 77.48 | 93.53 | - | 8.6 | EA |
| EffPnet [4] | 72.99 | 90.75 | - | $< 3$ | EA |
| CENAS-A [46] | 77.4 | 92.8 | 276 | 1.91 | EA |
| CENAS-B [46] | 78.9 | 93.6 | 396 | 1.91 | EA |
| CENAS-C [46] | 79.6 | 94.1 | 482 | 1.91 | EA |
| EPCNAS [47] | 72.9 | 91.5 | - | 1.17 | EA |
| RelativeNAS [39] | 75.12 | 92.30 | 563 | 0.4 | EA |
| SLE-NAS-A [35] | 74.4 | 91.8 | 293 | 3 | EA |
| SLE-NAS-B [35] | 75.7 | 92.5 | 412 | 3 | EA |
| SPNAS [16] | 78.62 | 94.07 | 687 | 0.37 | EA |
| HENAS (ours) | 78.69 | 94.01 | 580 | 0.22 | EA |

superior inference performance but also offers a good balance with computational efficiency.

*3) **Results on ImageNet**:* Table III shows the results of the proposed method on the ImageNet dataset. After final training on the ImageNet training set, the architecture achieves a classification accuracy of 78.69%, outperforming most state-of-the-art methods. Compared to SPNAS [16], our method reduces Madds by approximately 100M while maintaining similar classification accuracy. Furthermore, due to the use of weight inheritance, the search process is completed in just 0.22 days, which is significantly faster than most current methods.

### C. Ablation study

In this section, we analyse the surrogate model component of the proposed HENAS method and conduct ablation experiments to assess the effectiveness of each module. The experimental results are presented in Tables IV and V.

For all experiments in Table IV, we maintained a consistent experimental setup, including the sampling process. Specifically, a total of 400 architectures were used for real evaluation and to train the surrogate model. The architectures for real evaluation inherited weights from the supernet provided by OFA [20] and were validated on a random sample of 10,000 images from the ImageNet training set. Validation accuracy was used as the evaluation metric. During the surrogate model validation, we sampled 500 new architectures that were distinct from the training set. These architectures underwent the same real evaluation process to obtain their labels. After training the surrogate model, predictions were made on the

TABLE IV
ABLATION STUDY ON OPERATIONS FOR THE SURROGATE MODEL. (AUG: THE HOMOGENEOUS ARCHITECTURE AUGMENTATION. CONF: THE CONFIDENCE-BASED PREDICTION.)

| Ablation setting | Kendall's Tau ($KTau$) correlation |
|---|---|
| w/o Onehot | 0.6612 |
| w/o Aug & w/o Conf | 0.6239 |
| w/o Conf | 0.6501 |
| Proposed | 0.6786 |

TABLE V
ABLATION STUDY ON CONFIDENCE-BASED PREDICTION DURING SEARCHING PROGRESS. (CONF: THE CONFIDENCE-BASED PREDICTION.)

| Ablation setting | CIFAR-10 | | CIFAR-100 | | ImageNet | |
|---|---|---|---|---|---|---|
| | Acc (%) | Madds (M) | Acc (%) | Madds (M) | Acc (%) | Madds (M) |
| w/o Conf | 98.24 | 432 | 86.35 | 280 | 78.01 | 616 |
| Proposed | 98.38 | 328 | 87.36 | 297 | 78.69 | 580 |

surrogate validation set, and the Kendall's Tau ($KTau$) correlation coefficient between the predicted and real evaluation results was computed.

Confidence prediction relies on the architecture augmentation method, and as such, is not applicable when this method is not employed. From the results in Table IV, it can be observed that using one-hot encoding slightly improved $KTau$, with an increase of 0.0174. However, the homogeneous architecture augmentation method significantly enhanced the predictive performance of the surrogate model, achieving an improvement of 0.0262. For the experiments using architecture augmentation, we divided the 400 architectures into four sampling sessions. For each session, except the first, 300 new architectures were randomly sampled, and the 10 architectures with the worst confidence and prediction accuracy greater than the average were selected to update the surrogate model, alongside 90 new architectures. With this approach, confidence prediction showed a notable improvement of 0.0285 when using the full set of 400 architectures.

To further investigate the role of confidence prediction in the search phase, we conducted an ablation study by removing this process. To ensure fairness, in the ablation experiment presented in Table V, we included an additional step: each time the surrogate model was updated, we also added 5 architectures with the best fitness values to the surrogate dataset. This step ensures that the surrogate models in the ablation study are trained on the same sample size at each stage. The results from this experiment are shown in Table V. The findings highlight that confidence prediction is effective across the three standard datasets. On CIFAR-100 and ImageNet, the final networks obtained with confidence prediction achieve slightly higher classification accuracy, while maintaining similar computational complexity. On CIFAR-10, confidence prediction leads to a savings of nearly 100M Madds, along with a 0.14% improvement in classification accuracy.

## V. CONCLUSION AND FUTURE WORK

In this paper, we introduced HENAS, a novel method designed to identify superior architectures within the MobileNetV3 search space. HENAS leverages a unique homogeneous architecture augmentation technique, which allows for the generation of additional evaluated architecture samples without increasing search or evaluation time. This enables the training of a more accurate surrogate model, enhancing the overall search process.

Building on the architecture-augmented samples, we proposed a confidence prediction method that further refines the surrogate model's performance. This method predicts multiple potential outcomes for a given architecture, derived from different representations of the same architecture. By computing the standard deviation of these predicted values, we obtain the surrogate model's confidence in its predictions. Using this confidence, we strategically incorporate the most uncertain samples into the surrogate dataset, boosting the model's accuracy.

Additionally, we introduced an improvement to the genetic algorithm framework by incorporating a candidate pool. This pool stores individuals with low prediction confidence, allowing us to focus the surrogate model updates on these uncertain samples. This targeted approach improves the effectiveness of the model and the overall architecture search.

For future work, we aim to extend the current approach by developing a more generalized confidence prediction method that can be applied across a wider range of architectures. Furthermore, we plan to conduct additional experiments to explore an optimal balance between the number of samples selected from the candidate pool and the number of top-performing architectures chosen during each surrogate model update. This will enable us to fine-tune the genetic algorithm's efficiency and performance in diverse settings.

## REFERENCES


[1] Y. Sun, B. Xue, M. Zhang, and G. G. Yen, "Completely automated CNN architecture design based on blocks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 4, pp. 1242–1254, 2020.

[2] Y.-W. Wen, S.-H. Peng, and C.-K. Ting, "Two-stage evolutionary neural architecture search for transfer learning," *IEEE Transactions on Evolutionary Computation*, vol. 25, no. 5, pp. 928–940, 2021.

[3] Y. Sun, B. Xue, M. Zhang, G. G. Yen, and J. Lv, "Automatically designing CNN architectures using the genetic algorithm for image classification," *IEEE Transactions on Cybernetics*, vol. 50, no. 9, pp. 3840–3854, 2020.

[4] B. Wang, B. Xue, and M. Zhang, "Surrogate-assisted particle swarm optimization for evolving variable-length transferable blocks for image classification," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 8, pp. 3727–3740, 2022.

[5] S. Liu, H. Zhang, and Y. Jin, "A survey on computationally efficient neural architecture search," *Journal of Automation and Intelligence*, vol. 1, no. 1, p. 100002, 2022.

[6] R. Luo, X. Tan, R. Wang, T. Qin, E. Chen, and T.-Y. Liu, "Semi-supervised neural architecture search," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 10 547–10 557.

[7] Y. Liu, Y. Tang, and Y. Sun, "Homogeneous architecture augmentation for neural predictor," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, Montreal, QC, Canada, 2021, pp. 12 229–12 238.

[8] X. Xie, Y. Sun, Y. Liu, M. Zhang, and K. C. Tan, "Architecture augmentation for performance predictor via graph isomorphism," *IEEE Transactions on Cybernetics*, vol. 54, no. 3, pp. 1828–1840, 2024.

[9] P. Jiang, Y. Xue, F. Neri, and M. Wahib, "Surrogate-assisted evolutionary neural architecture search with isomorphic training and prediction," in *2024 International Conference on Intelligent Computing (ICIC)*, vol. 14863, Tianjing, China, 2024, pp. 191–203.

[10] B. Guo, T. Chen, S. He, H. Liu, L. Xu, P. Ye, and J. Chen, "Generalized global ranking-aware neural architecture ranker for efficient image classifier search," in *Proceedings of the 30th ACM International Conference on Multimedia*, Lisboa Portugal, 2022, pp. 3730–3741.

[11] X. Zhou, A. K. Qin, Y. Sun, and K. C. Tan, "A Survey of Advances in Evolutionary Neural Architecture Search," in *2021 IEEE Congress on Evolutionary Computation (CEC)*, Kraków, Poland, 2021, pp. 950–957.

[12] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8697–8710.

[13] Y. Xue, C. Lu, F. Neri, and J. Qin, "Improved differentiable architecture search with multi-stage progressive partial channel connections," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 8, no. 1, pp. 32–43, 2024.

[14] Y. Liu, Y. Sun, B. Xue, M. Zhang, G. G. Yen, and K. C. Tan, "A survey on evolutionary neural architecture search," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 2, pp. 550–570, 2023.

[15] Y. Xu, Y. Wang, K. Han, Y. Tang, S. Jui, C. Xu, and C. Xu, "ReNAS: Relativistic evaluation of neural architecture search," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Nashville, TN, USA, 2021, pp. 4409–4418.

[16] P. Jiang, Y. Xue, and F. Neri, "Score predictor-assisted evolutionary neural architecture search," *IEEE Transactions on Emerging Topics in Computational Intelligence*, pp. 1–15, 2025.

[17] X. Chu, B. Zhang, and R. Xu, "FairNAS: Rethinking evaluation fairness of weight sharing neural architecture search," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 12 219–12 228.

[18] Z. Lu, K. Deb, E. Goodman, W. Banzhaf, and V. N. Boddeti, "NSGANetV2: Evolutionary multi-objective surrogate-assisted neural architecture search," in *Proceedings of the European Conference on Computer Vision*, Cham, 2020, pp. 35–51.

[19] Y. Xue, C. Zhu, M. Zhou, M. Wahib, and M. Gabbouj, "A pairwise comparison relation-assisted multi-objective evolutionary neural architecture search method with multi-population mechanism," *arXiv preprint arXiv:2407.15600*, 2024.

[20] H. Cai, C. Gan, T. Wang, Z. Zhang, and S. Han, "Once-for-All: Train one network and specialize it for efficient deployment," in *International Conference on Learning Representations*, 2019.

[21] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4510–4520.

[22] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *International Conference on Machine Learning*, 2019, pp. 6105–6114.

[23] H. Pham, M. Guan, B. Zoph, Q. Le, and J. Dean, "Efficient neural architecture search via parameter sharing," in *International Conference on Machine Learning*, vol. 80, 2018, pp. 4095–4104.

[24] Z. Zhong, Z. Yang, B. Deng, J. Yan, W. Wu, J. Shao, and C.-L. Liu, "BlockQNN: Efficient block-wise neural network architecture generation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 7, pp. 2314–2328, 2021.

[25] A. Hundt, V. Jain, and G. D. Hager, "sharpDARTS: Faster and more accurate differentiable architecture search," *arXiv preprint arXiv:1903.09900*, 2019.

[26] M. Tan and Q. V. Le, "MixConv: Mixed depthwise convolutional kernels," in *Proceedings of the British Machine Vision Conference*, 2019, pp. 116.1–116.13.

[27] X. Chu, S. Lu, X. Li, and B. Zhang, "MixPath: A unified approach for one-shot neural architecture search," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 5972–5981.

[28] H. Liu, K. Simonyan, and Y. Yang, "DARTS: Differentiable architecture search," in *International Conference on Learning Representations*, 2018.

[29] Y. Xue and J. Qin, "Partial connection based on channel attention for differentiable neural architecture search," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 5, pp. 6804–6813, 2023.

[30] E. Real, A. Aggarwal, Y. Huang, and Q. V. Le, "Regularized evolution for image classifier architecture search," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 4780–4789.

[31] Z. Yang, Y. Wang, X. Chen, B. Shi, C. Xu, C. Xu, Q. Tian, and C. Xu, "CARS: Continuous evolution for efficient neural architecture search," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 1826–1835.

[32] S. Hu, R. Cheng, C. He, Z. Lu, J. Wang, and M. Zhang, "Accelerating multi-objective neural architecture search by random-weight evaluation," *Complex & Intelligent Systems*, vol. 9, no. 2, pp. 1183–1192, 2023.

[33] Z. Lu, I. Whalen, Y. Dhebar, K. Deb, E. D. Goodman, W. Banzhaf, and V. N. Boddeti, "Multiobjective evolutionary design of deep convolutional neural networks for image classification," *IEEE Transactions on Evolutionary Computation*, vol. 25, no. 2, pp. 277–291, 2021.

[34] J. Huang, B. Xue, Y. Sun, and M. Zhang, "A flexible variable-length particle swarm optimization approach to convolutional neural network architecture design," in *IEEE Congress on Evolutionary Computation*, 2021, pp. 934–941.

[35] J. Huang, B. Xue, Y. Sun, M. Zhang, and G. G. Yen, "Split-level evolutionary neural architecture search with elite weight inheritance," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 10, pp. 13 523–13 537, 2024.

[36] Y. Sun, H. Wang, B. Xue, Y. Jin, G. G. Yen, and M. Zhang, "Surrogate-assisted evolutionary deep learning using an end-to-end random forest-based performance predictor," *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 2, pp. 350–364, 2020.

[37] H. Zhang, Y. Jin, and K. Hao, "Evolutionary search for complete neural network architectures with partial weight sharing," *IEEE Transactions on Evolutionary Computation*, vol. 26, no. 5, pp. 1072–1086, 2022.

[38] H. Zhang, Y. Jin, R. Cheng, and K. Hao, "Efficient evolutionary search of attention convolutional networks via sampled training and node inheritance," *IEEE Transactions on Evolutionary Computation*, vol. 25, no. 2, pp. 371–385, 2021.

[39] H. Tan, R. Cheng, S. Huang, C. He, C. Qiu, F. Yang, and P. Luo, "RelativeNAS: Relative neural architecture search via slow-fast learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 1, pp. 475–489, 2023.

[40] Z. Lu, K. Deb, and V. N. Boddeti, "MUXConv: Information multiplexing in convolutional neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 12 041–12 050.

[41] M. Lin, P. Wang, Z. Sun, H. Chen, X. Sun, Q. Qian, H. Li, and R. Jin, "Zen-NAS: A zero-shot nas for high-performance image recognition," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 337–346.

[42] M. Tan, B. Chen, R. Pang, V. Vasudevan, M. Sandler, A. Howard, and Q. V. Le, "Mnasnet: Platform-aware neural architecture search for mobile," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2815–2823.

[43] H. Xiao, Z. Wang, Z. Zhu, J. Zhou, and J. Lu, "Shapley-NAS: Discovering operation contribution for neural architecture search," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 11 892–11 901.

[44] K. Jing, L. Chen, and J. Xu, "An architecture entropy regularizer for differentiable neural architecture search," *Neural Networks*, vol. 158, pp. 111–120, 2023.

[45] S. Yang, Y. Tian, X. Xiang, S. Peng, and X. Zhang, "Accelerating evolutionary neural architecture search via multifidelity evaluation," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 14, no. 4, pp. 1778–1792, 2022.

[46] L. Ma, N. Li, G. Yu, X. Geng, S. Cheng, X. Wang, M. Huang, and Y. Jin, "Pareto-wise ranking classifier for multi-objective evolutionary neural architecture search," *IEEE Transactions on Evolutionary Computation*, vol. 28, no. 3, pp. 570–581, 2024.

[47] J. Huang, B. Xue, Y. Sun, M. Zhang, and G. G. Yen, "Particle swarm optimization for compact neural architecture search for image classification," *IEEE Transactions on Evolutionary Computation*, vol. 27, no. 5, pp. 1298–1312, 2023.