applied sciences

MDPI

Article

CAM-NAS: An Efficient and Interpretable Neural Architecture Search Model Based on Class Activation Mapping

Zhiyuan Zhang, Zhan Wang and Inwhee Joe *

Computer Science, Hanyang University, Seoul 04763, Republic of Korea; zhiyuan15@hanyang.ac.kr (Z.Z.); zhanbaobao6@gmail.com (Z.W.)
* Correspondence: iwjoe@hanyang.ac.kr

Abstract: Artificial intelligence (AI) has made rapid progress in recent years, but as the complexity of AI models and the need to deploy them on multiple platforms gradually increases, the design of network model structures for specific platforms becomes more difficult. A neural network architecture search (NAS) serves as a solution to help experts discover new network structures that are suitable for different tasks and platforms. However, traditional NAS algorithms often consume time and many computational resources, especially when dealing with complex tasks and large-scale models, and the search process can become exceptionally time-consuming and difficult to interpret. In this paper, we propose a class activation graph-based neural structure search method (CAM-NAS) to address these problems. Compared with traditional NAS algorithms, CAM-NAS does not require full training of submodels, which greatly improves the search efficiency. Meanwhile, CAM-NAS uses the class activation graph technique, which makes the searched models have better interpretability. In our experiments, we tested CAM-NAS on an NVIDIA RTX 3090 graphics card and showed that it can evaluate a submodel in only 0.08 seconds, which is much faster than traditional NAS methods. In this study, we experimentally evaluated CAM-NAS using the CIFAR-10 and CIFAR-100 datasets as benchmarks. The experimental results show that CAM-NAS achieves very good results. This not only proves the efficiency of CAM-NAS, but also demonstrates its powerful performance in image classification tasks.

Keywords: artificial intelligence; neural architecture search; class activation map; CAM-NAS

Citation: Zhang, Z.; Wang, Z.; Joe, I. CAM-NAS: An Efficient and Interpretable Neural Architecture Search Model Based on Class Activation Mapping. *Appl. Sci.* **2023**, *13*, 9686. https://doi.org/10.3390/app13179686

Academic Editor: Andrea Prati

Received: 28 July 2023
Revised: 21 August 2023
Accepted: 25 August 2023
Published: 27 August 2023

1. Introduction

In the early stages of deep learning development, the manual design of neural network structures was predominant. Nowadays, deep learning techniques have been applied to many research areas in computer vision and have achieved very good results in image classification [1,2], biometrics [3], and medical diagnosis [4]. These successes are attributed to the efforts of experts who have designed various powerful network architectures, such as VGG [5], ResNet [6], and DenseNet [7]. However, as the complexity and size of deep learning models continue to increase, manually designing neural networks becomes more difficult and time-consuming. Designing a high-performance neural network structure requires a great deal of trial and error and experiences, often costing a great deal of time and computational resources, and it is difficult for experts to design a network that satisfies specific conditions. In addition, traditional neural network structures may be limited by human creativity and experience in discovering superior structures. To address the tediousness and limitations of manually designing neural network structures, the need for an automated neural network architecture search has emerged. Therefore, some researchers have proposed that algorithms can be designed to assist in the search for candidate architectures, rather than the entire network directly. A neural architecture search (NAS) is an algorithm designed to identify the most effective deep learning network by automatically searching and optimizing the network structure, reducing the burden of

manual design efforts while searching for better network architectures to automatically find a better performing and efficient neural network structure for a given task and dataset for a specific problem. In early NAS methods, it was common to directly train the entire network using brute force methods to gauge its accuracy. These NAS methods require many computing resources, usually hundreds of GPUs, and many days to obtain the results. To save time and computational resources, newer methods have been proposed that do not rely on manual design efforts, but use reinforcement learning, genetic algorithms, or other optimization techniques to automatically find the best architecture from candidate neural network structures, such as chunking-based adaptive variational neural structure search algorithms [8].

However, existing NAS algorithms are still too slow in some cases due to internal training steps. The CAM (class activation mapping) family of algorithms is currently the most widely used method for interpretable artificial intelligence (XAI), and is capable of generating image-level heat maps that intuitively show the areas of the model's regions of interest, which helps to explain the underlying reasons for the model's predictions; the CAM family of algorithms is relatively simple and easy to implement, with a reasonably high degree of accuracy compared to other XAI methods. Because it relies heavily on the feature maps of convolutional layers, it is often more efficient than some methods that require repeated model runs or extensive sampling. Therefore, the CAM algorithm was chosen to be combined with NAS to provide greater opportunities for neural network interpretability and performance optimization. Compared with traditional NAS, which requires a training process, CAM-NAS can improve the search speed by several orders of magnitude, taking as little as 0.08 seconds to determine a structure. Numerous experiments have shown that CAM-NAS finds network structures with high accuracy on CIFAR-10, CIFAR-100, and ImageNet datasets.

In Section 2, we introduce the related works and illustrate the advantages and disadvantages of today's mainstream algorithms; in Section 3, we describe the implementation process and algorithmic flow of CAM-NAS in detail; in Section 4, we evaluate the experimental results and validate the practicality and accuracy of the model; and finally, in Section 5, we summarize the main ideas of CAM-NAS and discuss the strengths and weaknesses of the methodology, its application areas, and the future research directions.

2. Related Works

2.1. Neural Architecture Search

Many researchers have proposed different search strategies for finding neural architectures.

Zoph et al. first proposed a reinforcement learning NAS method [9], which uses a recurrent neural network as a controller to generate a network and uses the accuracy of the searched network on the dataset as feedback, which is then passed to the controller through reinforcement learning to enable the controller to generate a new network. However, this NAS approach requires significant computational resources and long network search times. To speed up the search process, Zoph et al. proposed NASNet [10], which searches for units on small datasets and then stacks the searched small units to form a complete network for larger datasets.

Pham et al. proposed an efficient neural architecture search (ENAS) [11], which transforms the process of searching for neural network architectures into a trainable reinforcement learning problem, where each candidate architecture shares the same weights to improve search efficiency; it is able to search for high-performance network architectures using relatively few computational resources. This approach uses reinforcement learning to provide an efficient and cost-effective search for high-performance neural network architectures, reducing the search time to less than 16 hours via a single NVIDIA® GTX1080Ti GPU. All of the above NAS methods treat the network structure in the search space as a discrete space.

Liu et al. proposed the Darts method [12], which transforms the network structure search into an optimization problem in a continuous space and uses gradient descent to find the optimal network structure. Some scholars proposed the concept of zero-shot NAS, which avoids the training process involving submodels, greatly speeding up the evaluation time of each submodel and improving the search efficiency. Chen et al. proposed TE-NAS [13], which predicts the accuracy of submodels on a dataset by analyzing the number of neural tangent kernels and linear regions. Mellor et al. proposed the NASWOT (neural architecture search without training) [14], which calculates the number of activations of the ReLU units in the network and then measures the similarity of the inputs using the Hamming distance; the lower the similarity, the higher the score, which indicates the accuracy of the submodel. This replaces the traditional training process and reduces the search time to less than 1 s. Lin et al. proposed another zero-shot NAS algorithm, Zen-NAS [15], which searches the network in 0.1 seconds and focuses only on the last layer of information in the network, calculating the Gaussian complexity as a Zen-score, with higher scores representing a better network structure. However, none of these methods can explain why the structures found are very different from those designed by experts.

ENAS is limited to specific tasks and uses reinforcement learning algorithms for specific tasks when searching for network structures, so the obtained structures may perform poorly on other tasks, requiring multiple reinforcement learning training, and the search process requires a large amount of computational resources and time. DARTS searches only one layer of the network structure in the search space and cannot search deeper complex network structures, which are prone to overfitting during the search process, resulting in poor performance of the searched structure on the validation set. TE-NAS uses genetic algorithms to perform the search; however, genetic algorithms can perform poorly in high-dimensional search spaces with low search efficiency, and there is a need to extract topologies from pre-trained models. The quality of these pre-trained models directly influences the search results. Zen-NAS uses the zero-shot NAS approach, which works better on certain tasks, but may not perform well on other tasks. The deficiencies of existing NAS methods are shown in Table 1. To solve the above problems, CAM-NAS is proposed.

Table 1. Deficiencies of existing NAS approaches.

Related Works	Deficiencies
NASNet	Requires expensive hardware facilities and time, is prone to overfitting, is usually optimized for specific tasks and datasets, finds optimal architectures that may not be applicable to other types of tasks, and needs to be researched or fine-tuned.
ENAS	Limited to specific tasks and requires multiple reinforcement learning training; the search process requires time and a large amount of computational resources.
DARTS	Searches only one layer of the network structure, cannot search deeper network structures, easy to overfit.
TE-NAS	Uses a genetic algorithm for search; has poor performance in high-dimensional search spaces and low search efficiency.
Zen-NAS	Can be affected by the limitations of the pre-trained model; the pre-trained architecture may not generalize well to the target task, thus affecting performance.

2.2. Class Activation Map

The class activation map (CAM) is a technique used to visualize the image classification process of a convolutional neural network (CNN) model, which is an interpretable artificial intelligence method. CAM can be used to explain to people what regions of an image the

CNN model focuses on, and it can help people understand to what extent the CNN model focuses on different regions of the image when classifying an image. The class activation map technique was first proposed by Zhou et al. [16]. In a network structure with global average pooling (GAP), CAM can extract the key regions of the network through the GAP layer. When CNN is used for image classification, the model performs several convolution and pooling operations on the image to extract the features of the image, and adds a global average pooling layer after the last fully connected layer of the model. The global average pooling layer averages the feature maps of each channel to obtain a channel weight value. The final CAM is then obtained by multiplying and summing these channel weights with the corresponding feature maps, upsampling the CAM to the size of the original image, and overlaying it with the original image to create the visualization. Darker regions indicate that the model pays more attention to the region and, thus, has a greater impact on the classification results.The disadvantage of CAM is that it requires modification to the network structure and retraining of the network.

Grad-CAM [17] proposed by Selvaraju et al. uses a gradient-based global average to compute the weights, thus overcoming the drawback of CAM, as it requires changes in the network structure and allows the extraction of features from any layer of the network. Grad-CAM++ [18], proposed by Chattopadhay et al., improves Grad-CAM by improving the computation of weights, allowing for better interpretation of the model's predictions and the ability to localize multiple targets in a single image. Wang et al. proposed Score-CAM [19], which computes the weights by forward-passing the scores of the target categories of the activation graph. The final result of Score-CAM is obtained by a linear combination of weights and the activation graphs obtained from a linear combination of the weights and the activation graphs without relying on the gradient for computation. Wang et al. proposed SS-CAM (softmax-weighted class activation mapping) [20], which combines class activation mapping and softmax weighting to generate more accurate and interpretable category activation maps. The softmax values of each class are used as weights, and the feature maps of all classes are weighted and averaged, which takes into account the feature maps at different levels in the network, thus providing more comprehensive information to explain the model's classification decisions. Naidu et al. proposed IS-CAM (integrated Score-CAM) [21], which introduces an integration operation in the Score-CAM pipeline that integrates the input mask and averages the score obtained from the normalized mask to achieve a quantitatively sharper saliency map. Omeiza et al. proposed smooth Grad-CAM++ [22], which improves the Grad-CAM++ method by introducing gradient smoothing and activation averaging to slightly perturb the input image, calculating the sensitivity of the model's output to perturbation. The category activation maps are computed on each convolutional layer and superimposed to form the final result, providing a more stable and accurate interpretation of the image classification. Axiom-based Grad-CAM [23] was proposed in 2020 to introduce a priori knowledge provided by human experts and combine it with the activation graph of CNN to generate more meaningful explanations. The introduction of axioms can help reduce the uncertainty and volatility of explanations, thus improving the stability of explanations.

The CAM family of techniques plays an important role in explaining and understanding the classification decision process of CNN models, allowing for a better understanding of operating deep learning models and helping to improve the interpretability and credibility of the models.

3. Method

3.1. Main Process

In this paper, a new NAS algorithm called CAM-NAS is proposed. Unlike traditional NAS algorithms, CAM-NAS reduces the training steps in the model search process and takes only 0.08 seconds to retrieve, compute, and evaluate the model. The main steps are shown in Figure 1.

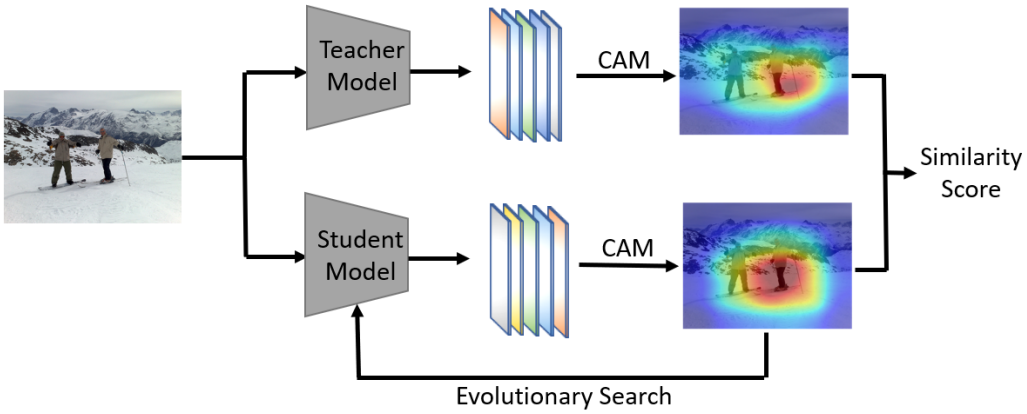

Figure 1. Main process of CAM-NAS. Similarity scores are calculated by comparing the CAM maps of the teacher model and the child model.

First, a SOTA model is selected as a training model and a small fraction of the images in the dataset is tested with this model. At the same time, the saliency map of the last layer of this model is extracted using the CAM technique. Next, the same image is tested using the model obtained from the search, and the last layer of the saliency map of the submodel is extracted using the same CAM technique. Then, the CAM image of the teacher's model is compared with the CAM image of the submodel and the similarity score is calculated. The CAM technique is able to visualize the model's focus on different features as heat maps by visualizing the activation mapping of the last layer of the neural network. This makes it possible to visualize the distribution of the model's attention on different images, which helps to understand the model's decision-making process.

Teacher models are typically larger, more complex models, such as deep neural networks, e.g., ResNet-152, BERT, etc. These models are trained on large datasets and are capable of extracting rich feature representations. Student models are typically relatively small, lightweight models, such as shallow neural networks. Teacher and student models are used to transfer knowledge from a complex, large-scale model to a simplified, small-scale model during training, capturing the knowledge of the teacher model in the student model for better generalization performance, higher efficiency, or lower computational costs.

By comparing the CAM maps of the teacher model and the submodel, the performance advantage of the submodel over the teacher model can be assessed. If the CAM maps of the submodel on the same images have a high similarity to the CAM maps of the teacher model, it means that the submodel is able to capture similar important features as the teacher model and it performs better. Conversely, if the similarity is low, it indicates that the submodel has some performance shortcomings. By evaluating the similarity scores of different submodels on key features, the better-performing submodels can be filtered out and used for the further neural network architecture search and optimization process. In this way, excellent network architectures can be found efficiently without the need to fully train the submodels, saving time and computational resources.

3.2. CAM Techniques

A series of experiments were conducted among known CAM techniques to determine which CAM techniques are suitable for integration with NAS algorithms. The experiments used the pre-trained ResNet family of models, including ResNet-18, ResNet-34, ResNet-101, and ResNet-152. For all ResNet models, the class activation mapping of the last layer was extracted using the CAM technique. ResNet-18 and ResNet-34 are shallower ResNet variants with slightly lower performances on some complex tasks. ResNet-50, ResNet-101, and ResNet-152 are deeper ResNet variants that are capable of extracting rich feature representations for more complex tasks. Because ResNet-152 is deeper and can capture richer features, it has been trained on large image datasets and has learned many useful

feature representations. Using it as a teacher model allows the student model to benefit from these rich features. And the teacher model should be a stable model with high accuracy to ensure that the knowledge transferred to the student model is reliable. ResNet-152 has been shown to be a stable and high-performing model in many tasks. Therefore, ResNet-152 was selected as the teacher model.

Choosing the right CAM technique to combine with NAS algorithms is critical for an efficient submodel search and evaluation. Since the NAS algorithm requires a large number of model evaluations during the search process, it is important to extract the saliency map quickly and accurately. Based on the experimental results, we tend to choose the more computationally efficient and powerful CAM technique combined with the NAS algorithm to improve the search efficiency and performance.

The class activation maps of other models were compared with ResNet-152 and the similarities between them were calculated. As shown in Figure 2, the longest time was spent when using SS-CAM and IS-CAM. This may be due to the fact that they consider more spatially sensitive features in the computation process, resulting in a corresponding increase in the computational load. In contrast, the other CAM methods use a more simplified computational strategy and, therefore, require less time to extract the saliency maps.

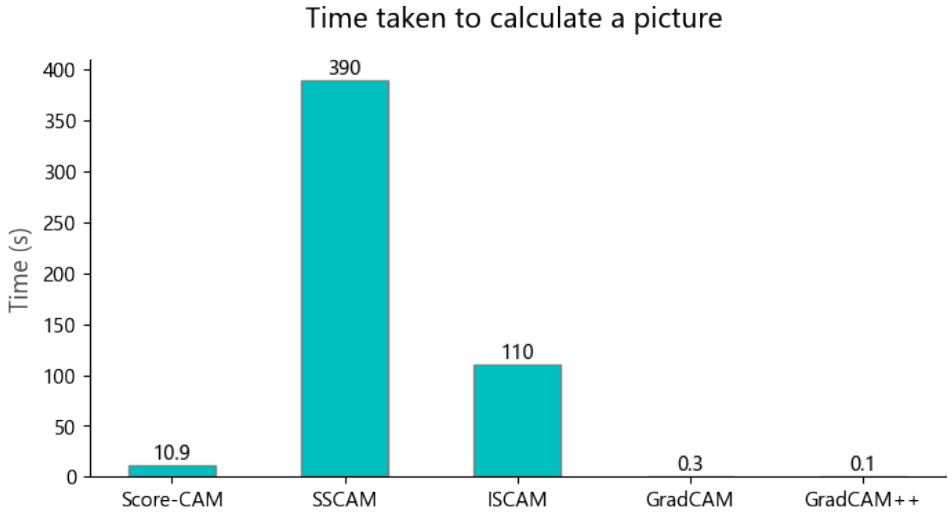

Figure 2. Calculation of the time it takes to scan an image using different CAM methods.

In the study, methods such as smooth Grad-CAM++ and axiom-based Grad-CAM (XGradCAM) were used to extract the activation mapping of the images in the model, and the effectiveness of these methods was evaluated by comparing the similarity scores and accuracies. As shown in Figure 3, the results indicate that when using the smooth Grad-CAM++ and XGradCAM methods, there is no linear correlation between similarity scores and model accuracy on the ImageNet validation dataset, i.e., the changes in these two metrics are not identical. However, under the same experimental conditions, there was a linear correlation between the similarity score and model accuracy for the Grad-CAM, Grad-CAM++, and Score-CAM methods. This implies that there is a correlation between the similarity scores of activation mappings and model accuracy in these methods, i.e., activation mappings with high similarity scores may be associated with high model accuracy. Although the Score-CAM method has a linear correlation between similarity scores and model accuracy, it is slower to compute, especially when the network structure is deep. On the other hand, the similarity scores obtained with the Grad-CAM++ method are not very clear in distinguishing between different network structures.

Therefore, the final choice is to use the Grad-CAM method in combination with NAS. This is due to the fact that the Grad-CAM method has a good linear correlation with model accuracy and high computational efficiency, which makes it the most suitable choice for activation mapping analysis during the neural network architecture search. By combining it

with the Grad-CAM method, researchers are able to better understand the feature mappings of network architectures, thus optimizing and improving the search process and increasing network performance.

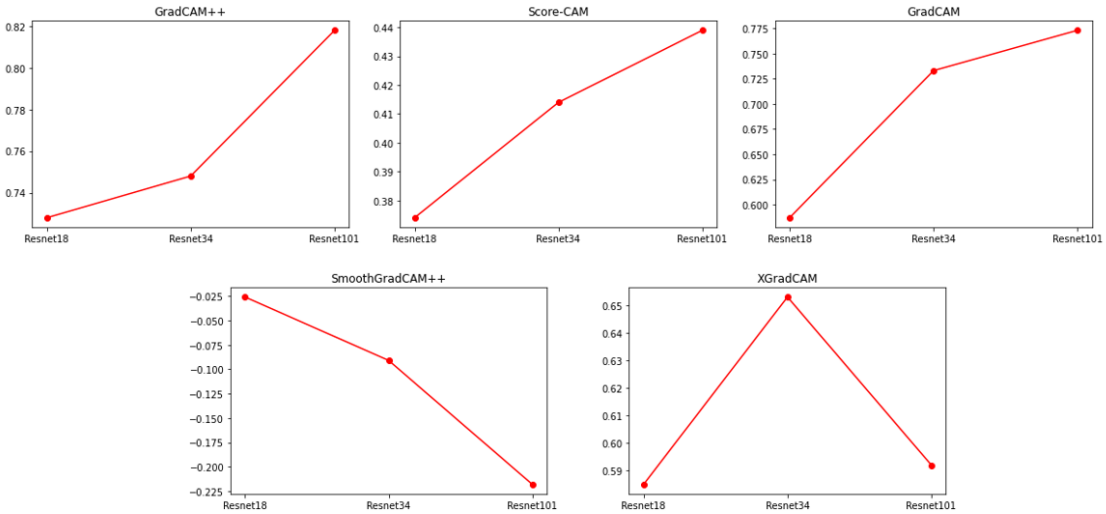

Figure 3. Similarity scores obtained by applying various CAM techniques to the ResNet model. The similarity score unit is a unit-less proportional value (with a maximum similarity of 1) that measures the degree of similarity between two saliency maps, with larger values indicating greater similarity.

3.3. Mathematical Equation for the Main Processes

In this section, we describe the main process of CAM-NAS in terms of mathematical equations and we define the method for calculating the similarity score. Since we combine Grad-CAM and NAS methods in this paper, some of the formulas are derived from Grad-CAM [15].

$$\alpha_k^c = GP(\frac{\partial y^c}{\partial A_l^k}) \tag{1}$$

For a convolutional layer, l denotes the layer, and A_l^k denotes the activation map of the kth channel. $GP()$ denotes the global pooling operation. First, we compute the gradient of the score for the class c, y_c, with respect to the feature maps A^k by (1). Then we use the global average pooling operation to obtain the importance weights of the neurons α_k^c. α_k^c denotes a partial linearization of the deep network downstream of A, and represents the confidence of a feature map k for a target class c.

$$L_{Grad-CAM}^c = ReLU(\sum_k \alpha_k^c A_l^k) \tag{2}$$

To obtain the class activation map (CAM), we perform a weighted combination of activation maps. Then, we applied the ReLU function to focus only on the positive part of the activation map, as in (2). The positive part better reflects the model's ability to discriminate objects.

$$A_{Target} = \sum_i^N L_{Grad-CAM(i)}^{ci} \tag{3}$$

$$A_{Search} = \sum_i^N L_{Grad-CAM(i)}^{ci} \tag{4}$$

In the NAS algorithm, we first randomly select N images from the dataset and input them into the teacher model, and then use (3) to obtain A_{Target}. For the search submodel,

we input the same N images into the search submodel and then use (4) again to obtain A_{Search}.

$$Score = \frac{1}{\sqrt{(A_{Target} - A_{Search})^2}} \tag{5}$$

Formula (5) defines how the similarity score is calculated. The calculation used is the inverse of the Euclidean distance. The higher the similarity score, the closer the CAM of the teacher model is to the CAM of the searched submodel. It is assumed that this means that the submodel and the teacher model focus on the same images in a similar way, and that the submodel will eventually perform more similar to the teacher model on the dataset. The CAM-NAS process is shown in Algorithm 1.

Algorithm 1: CAM-NAS

Require: Search space S, inference budget B, maximal depth L, total number of iterations T, evolutionary population size N, initial structure F_0.

Initialize population $P = \{F_0\}$
for $i = 1; i < T; i++$ **do**
  Randomly select $F_t \in P$
  Uniformly select a block h in F_t
  Uniformly alternate the block type, kernel size, width and depth of h witnin some range.
  Obtain the mutated structure F_m
  if F_m exceeds inference budget or has more than L layers **then**
    Do nothing
  else
    Obtain Similarity-Score, score = $Score(F_m)$
    Append F_m to P
  end if
  if the length of $P > N$ **then**
    Delete the submodel of the smallest Similarity-Score
  end if
end for
Return the submodel of the highest Similarity-Score in P

3.4. CAM-NAS Algorithm

The proposed similarity score can be combined with various existing methods to generate submodels for NAS. In this paper, evolutionary algorithms are used to generate submodels.

First, an initial model F_0 and a list of submodels P are created to store the models to be subjected to the mutation operation. The initial model is placed in P and variations are performed on the initial model, including changing the network structure, adding or removing layers, adjusting layer parameters, and so on. The mutation operation is intended to create a variant model that is different from the initial model, but has the potential for better performance. If the depth of the mutated model F_m is less than L, we compute the similarity score of the mutated model and put this model into P. If the number of submodels in P exceeds N, the submodel with the lowest similarity score is removed from P. Finally, CAM-NAS returns the submodel with the highest similarity score in P as the search result.

4. Experiment

4.1. Model Selection Process

This section compares different NAS models that do not need to be trained on CIFAR-10 and CIFAR-100 using the same search space, search strategy, and training settings. The

experiments use the same search space as Zen-NAS. In the search space, the maximum depth of the neural network structure is limited to 6 layers; choices include convolutional kernels of sizes 1×1, 3×3, and 5×5 sizes, with channel options of 8, 32, 64, 128, 192, and 256 channels, and with the flexibility to choose whether or not to use jump joins, ReLU activation functions, the Adam optimizer, and batch normalization. Different permutations of the search space affect the diversity, performance, interpretability, and computational efficiency of the neural network structures found by the search algorithm. By designing the search space appropriately, the search algorithm can better explore the possibilities of network structures and find a neural network model with a superior performance suitable for a specific task. The initial structure is a small randomly selected network that is guaranteed to satisfy the inference budget. To evaluate the performance of the NAS model, we will measure its accuracy and efficiency. Accuracy will be calculated based on the model's classification performance on the CIFAR-10 and CIFAR-100 validation datasets. Meanwhile, efficiency will be measured in terms of computational cost, including inference time and the number of model parameters. This is crucial for practical applications seeking efficient architectures. Through a comparative analysis, we aim to understand the strengths and limitations of each NAS model under the same search spaces and settings.

Table 2 shows the different NAS proxies and their accuracies on the CIFAR-10 and CIFAR-100 datasets. Using different NAS agents and CAM-NAS involves finding high-performance neural network structures that are suitable for specific tasks through search algorithms. These proxies are the criteria used to evaluate the quality and performances of different network structures. For example, Zen-score denotes a quality evaluation metric used by Zen-NAS, FLOPs denotes the computation amount of the network structure, Syn-Flow refers to the mobility of the network structure, TE-score denotes the evaluation metric of TE-NAS, and random denotes the random search method. Table 2 lists these proxies, and their performances are evaluated using the same dataset and settings. The accuracy of CAM-NAS on the CIFAR-10 and CIFAR-100 datasets is acceptable and significantly better than a random search. This result shows that comparing the similarities of class activation maps generated by different models can be used as a new criterion to evaluate the quality of submodels. CAM-NAS has the highest accuracy on the CIFAR-10 and CIFAR-100 datasets compared to other zero-shot proxies.

Table 2. Different NAS proxy accuracies for the CIFAR-10 and CIFAR-100 datasets. The use of different NAS agents and CAM-NAS involves finding high-performance neural network structures that are suitable for specific tasks through search algorithms. These proxies are the criteria used to evaluate the quality and performances of different network structures.

Proxy	CIFAR-10	CIFAR-100
Zen-Score	96.2%	80.1%
FLOPs	93.1%	64.7%
Grad	92.8%	65.4%
Synflow	95.1%	75.9%
TE-Score	96.1%	77.2%
NASWOT	96.0%	77.5%
Random	$93.5 \pm 0.7\%$	$71.1 \pm 3.1\%$
CAM-NAS	95.68%	75.94%

Figure 4 shows the time required by different agents to process five images under the same conditions. Among the four zero-shot NAS methods, TE-NAS is the slowest, while the proposed CAM-NAS is comparable to the Zen-score and NASWOT. CAM-NAS is many orders of magnitude faster than traditional NAS methods.

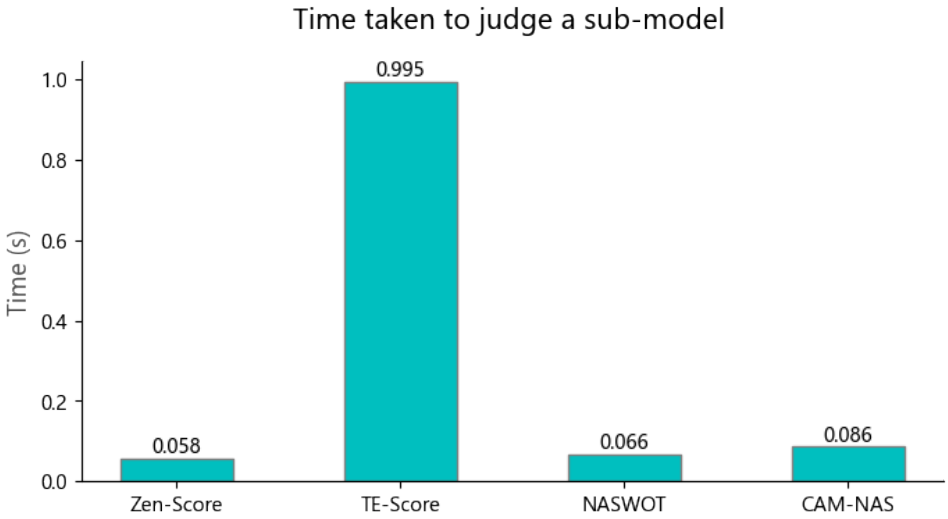

Figure 4. The time needed for different zero-shot NAS methods to judge a single submodel.

In conclusion, CAM-NAS performs well on the CIFAR-10 and CIFAR-100 datasets, outperforms other proxies in terms of accuracy, and is more efficient in terms of computational time compared to traditional NAS methods. By comparing the similarity of activation mappings, CAM-NAS provides an efficient way to evaluate the submodel quality and achieve a significant advantage in zero-shot NAS. CAM-NAS has lower complexity in the neural network architecture search compared to existing methods: instead of fully training the submodels during the search process, CAM-NAS evaluates the quality of the submodels by comparing the similarity of the saliency graphs. This makes CAM-NAS more efficient in performing the search and evaluation process. Traditional neural network architecture search methods may require complete training on each submodel, which consumes significant time and computational resources. However, CAM-NAS avoids this time-consuming training process by directly calculating similarity scores to evaluate submodels. As a result, CAM-NAS is more efficient in terms of performance. In addition, the search space setup and methodology of CAM-NAS are relatively simple and do not require extensive parameter tuning and complex search strategies. Instead, CAM-NAS uses CAM techniques to compare the regions of interest of models to guide the search direction, making the approach easier to understand and implement. Taken together, CAM-NAS can efficiently find neural network architectures with high performance on a given task at relatively low complexity; thus, it has great potential in the field of neural network architecture search.

4.2. Illustration of an Example

When applying the CAM-NAS neural network architecture search algorithm, a SOTA model is first selected as the teacher model, e.g., ResNet-152. N images are randomly selected from the dataset and fed into the teacher model, and then (3) is used to obtain A_{Target}. At the same time, CAM is used to extract the saliency maps of the last layer of the teacher model, which will show the distribution of the teacher model's attention on different images. Next, CAM-NAS is used to obtain a new submodel. For the search submodel, we input the same N images into the search submodel and use the same CAM technique to extract the saliency maps of the last layer of the submodel, these saliency maps will show the distribution of attention of the submodel on the same images. Then (4) is used again to obtain A_{Search}. Finally, (5) is used to compute the similarity score. If the saliency map of the submodel has a high similarity with the saliency map of the teacher model on the same image, it means that the submodel is able to capture similar important features as the teacher model and may have a better performance. Conversely, if the similarity is low, it indicates that the submodel may have performance shortcomings. By comparing the

similarity scores of different submodels on key features, better-performing submodels can be filtered out and used in further neural network architecture searches and optimization processes. This approach can efficiently find excellent network architectures without the need to fully train all submodels, saving time and computational resources.

5. Conclusions

By combining neural network architecture search (NAS) and class activation mapping (CAM) techniques, we propose the CAM-NAS algorithm, which provides better interpretability and efficiency for the search process of neural network architectures. NAS is able to improve the model performance by automatically searching and optimizing the structure of neural networks, while the CAM technique helps us understand how different network architectures pay attention to different regions in the task Combining CAM techniques with NAS can provide a more in-depth explanation and guidance during the search process.

Using CAM techniques, we can visualize regions of interest for different candidate network structures in an image classification task to better understand how NAS-searched network structures function in a given task. By observing the regions of interest displayed by CAM, researchers can adjust the design of the network structures to increase the focus on important regions and, thus, improve model performance. In addition, the CAM-NAS algorithm evaluates the quality of submodels by directly comparing the similarity between the teacher model and the last layer of the class activation graph of the submodels, avoiding the complex process of training submodels in the traditional NAS approach, thus greatly reducing the computational cost, and being able to complete the search and evaluation within 0.08 seconds.

Experiments on the CIFAR-10 and CIFAR-100 datasets validate the effectiveness of CAM-NAS. Our results confirm that the accuracy of models retrieved via CAM-NAS outperforms the current state-of-the-art zero-shot NAS technology. This provides an empirical validation of our approach and highlights the potential of CAM-NAS to improve the neural network architecture search.

In conclusion, the CAM-NAS algorithm is an important step toward a more efficient and interpretable neural network architecture search. By combining the focus of CAM techniques with the search process, we demonstrate the potential for building models with better performance and lower complexity. These results open up new directions for future research and encourage the exploration of more advanced architectures and optimization strategies based on CAM-NAS. In future work, we hope to explore the structure of the submodules themselves and discover the relationship between the model structure itself and the performance of the models on the dataset.

Author Contributions: Conceptualization, Z.Z.; methodology, Z.Z. and Z.W.; software, Z.Z.; validation, Z.Z., Z.W. and I.J.; writing—original draft preparation, Z.Z. and Z.W.; writing—review and editing, Z.Z., Z.W. and I.J.; visualization, Z.Z. and Z.W.; supervision, I.J.; project administration, Z.Z., Z.W. and I.J. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korean government (MSIT) (no. 2020-0-00107, development of technology to automate the recommendations for big data analytic models that define data characteristics and problems).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets generated during and/or analyzed during the current study are available from the corresponding author upon reasonable request.

Conflicts of Interest: The authors declare that they have no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AI	artificial intelligence
CNN	convolutional neural network
NAS	neural architecture search
ENAS	efficient neural architecture search
CAM	class activation map
CAM-NAS	class activation map-based neural architecture search
GAP	global average pooling
XGradCAM	axiom-based Grad-CAM

References

1. Sun, Y.; Xue, B.; Zhang, M.; Yen, G.G. Evolving deep convolutional neural networks for image classification. *IEEE Trans. Evol. Comput.* **2019**, *24*, 394–407. [CrossRef]
2. Ang, K.M.; El-kenawy, E.S.M.; Abdelhamid, A.A.; Ibrahim, A.; Alharbi, A.H.; Khafaga, D.S.; Tiang, S.S.; Lim, W.H. Optimal Design of Convolutional Neural Network Architectures Using Teaching–Learning-Based Optimization for Image Classification. *Symmetry* **2022**, *14*, 2323. [CrossRef]
3. Ashraf, S.; Ahmed, T. Dual-nature biometric recognition epitome. *Trends Comput. Sci. Inf. Technol.* **2020**, *5*, 8–14.
4. Fernandes, F.E.; Yen, G.G. Automatic searching and pruning of deep neural networks for medical imaging diagnostic. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *32*, 5664–5674. [CrossRef] [PubMed]
5. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
6. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
7. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.
8. Xue, Y.; Wang, Y.; Liang, J.; Slowik, A. A self-adaptive mutation neural architecture search algorithm based on blocks. *IEEE Comput. Intell. Mag.* **2021**, *16*, 67–78. [CrossRef]
9. Zoph, B.; Le, Q.V. Neural architecture search with reinforcement learning. In Proceedings of the 5th International Conference on Learning Representations, Toulon, France, 24–26 April 2017.
10. Zoph, B.; Vasudevan, V.; Shlens, J.; Len, Q.V. Learning transferable architectures for scalable image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8697–8710.
11. Pham, H.; Guan, M.; Zoph, B.; Le, Q.; Dean, J. Efficient neural architecture search via parameters sharing. In Proceedings of the International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; pp. 4095–4104.
12. Liu, H.; Simonyan, K.; Yang, Y. Darts: Differentiable architecture search. *arXiv* **2018**, arXiv:1806.09055.
13. Chen, W.; Gong, X.; Wang, Z. Neural Architecture Search on ImageNet in Four GPU Hours: A Theoretically Inspired Perspective. In Proceedings of the International Conference on Learning Representations, Addis Ababa, Ethiopia, 26–30 April 2020.
14. Mellor, J.; Turner, J.; Storkey, A.; Crowley, E.J. Neural architecture search without training. In Proceedings of the International Conference on Machine Learning, Virtual, 18–24 July 2021; pp. 7588–7598.
15. Lin, M.; Wang, P.; Sun, Z.; Chen, H.; Sun, X.; Qian, Q.; Li, H.; Jin, R. Zen-NAS: A Zero-Shot NAS for High-Performance Image Recognition. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 347–356.
16. Zhou, B.; Khosla, A.; Lapedriza, A.; Oliva, A.; Torralba, A. Learning deep features for discriminative localization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2921–2929.
17. Selvaraju, R.R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D. Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization. In Proceedings of the Computer Vision (ICCV), IEEE International Conference, Venice, Italy, 22–29 October 2017; pp. 618–626.
18. Chattopadhay, A.; Sarkar, A.; Howlader, P.; Balasubramanian, V.N. Grad-CAM++: Improved Visual Explanations for Deep Convolutional Networks. In Proceedings of the 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), Lake Tahoe, NA, USA, 12–15 March 2018; pp. 839–847.
19. Wang, H.; Wang, Z.; Du, M.; Yang, F.; Zhang, Z.; Ding, S.; Mardziel, P.; Hu, X. Score-CAM: Score-weighted visual explanations for convolutional neural networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 14–19 June 2020; pp. 24–25.
20. Wang, H.; Naidu, R.; Michael, J.; Kundu, S.S. SS-CAM: Smoothed Score-CAM for Sharper Visual Feature Localization. *arXiv* **2020**, arXiv:2006.14255.
21. Naidu, R.; Ghosh, A.; Maurya, Y.; Kundu, S.S. IS-CAM: Integrated Score-CAM for axiomatic-based explanations. *arXiv* **2020**, arXiv:2010.03023.

22. Omeiza, D.; Speakman, S.; Cintas, C.; Weldermariam, K. Smooth Grad-CAM++: An Enhanced Inference Level Visualization Technique for Deep Convolutional Neural Network Models. *arXiv* **2019**, arXiv:1908.01224.
23. Fu, R.; Hu, Q.; Dong, X.; Guo, Y.; Gao, Y.; Li, B. Axiom-based Grad-CAM: Towards Accurate Visualization and Explanation of CNNs. *arXiv* **2020**, arXiv:2008.02312.