Neural Architecture Search and Knowledge Distillation for Semantic Image Segmentation on Big Wildfire Datasets

1st Evgenios Vlachos
School of Informatics
Aristotle University of Thessaloniki
Thessaloniki, Greece
evlachol@csd.auth.gr

2nd Christos Papaioannidis
School of Informatics
Aristotle University of Thessaloniki
Thessaloniki, Greece
cpapaionn@csd.auth.gr

3rd Ioannis Pitas
School of Informatics
Aristotle University of Thessaloniki
Thessaloniki, Greece
pitas@csd.auth.gr

***Abstract*—The increasing complexity of Deep Neural Network (DNN) models poses computational challenges for both DNN model development and their real-world deployment, particularly in the case of large training and test dataset scenarios. This is the case of forest fires, where huge UAV and synthetic image data have to be analyzed in real-time for efficient wildfire management. In this paper, we propose a novel combination of Neural Architecture Search (NAS) with Knowledge Distillation for burnt area image segmentation in the aftermath of a wildfire, by exploring a vast search space of DNN architectures and transferring learned DNN knowledge. We conducted our experiments on the BLAZE dataset depicting wildfires in Greece to evaluate the effectiveness of our approach on five different image segmentation DNN architectures. Our experiments demonstrated that for the best performing architecture, we have found a combination that can provide a 62.3% reduction of total trainable DNN parameters, alongside an increase of 1.02% in semantic image segmentation performance in terms of the mIoU metric.**

***Index Terms*—Neural Architecture Search, Knowledge Distillation, Semantic Image Segmentation, Wildfire Management, Deep Neural Networks**

I. Introduction

Wildfires are observed using UAV, other aerial and terrestrial methods for efficient wildfire management. To this end, huge amounts of visual data (images and videos) have to be analyzed in real-time. Flame/smoke and burnt area segmentation are essential both to predict wildfire evolution and to assess its impact. In this paper, we focus on burnt area segmentation, a key task that has attracted significant attention in the computer vision and machine learning communities [1].

Recent advances in segmentation favor increasingly complex DNNs with many parameters, offering high accuracy at the cost of computational efficiency. These models are hard to deploy on cloud, edge, or embedded platforms, such as UAVs used in wildfire monitoring.

Several DNN model compression techniques have been explored to mitigate these challenges. Knowledge Distillation (KD) [2] has emerged as a powerful tool for compressing DNN architectures by transferring the knowledge from a large, complex teacher model to a smaller, more efficient student model. This process involves training the student to mimic the teacher's outputs, using them as soft targets to guide learning. Another useful approach is Neural Architecture Search (NAS) which automates the search of smaller more efficient DNN architectures. NAS has significant implications for DNN development, particularly in domains where optimal performance is critical, e.g., in real-time on edge or in embedded computing or in big data analysis. By automating the DNN architecture design process, NAS not only accelerates the development on these tasks, but also reduces the dependency on human expertise, making advanced DNN technologies more accessible.

Previous efforts combining NAS with KD have shown promise in enhancing model efficiency and performance. However, these methods remain largely unexplored for segmentation tasks and are often designed for classification, limiting their adaptability. Many rely on fixed teacher-student architectures, reducing flexibility in preserving spatial consistency and feature representation. Unlike earlier NAS–KD studies that target image classification, our pipeline is designed for dense, pixel-wise prediction. The NAS process is optimized directly on mean IoU, which naturally steers the search toward decoder depths, skip links and up-sampling choices that preserve spatial detail and sharp object borders. During knowledge distillation we transfer the teacher's full-resolution soft segmentation maps, enabling the student to inherit rich per-pixel class probabilities rather than only coarse logits. We evaluate its effectiveness on five neural architectures: PIDNet Small, PIDNet Medium, PIDNet Large, UNet++, and CNN-I2I BiseNet. By integrating NAS and KD, our approach balances computational efficiency with high segmentation accuracy, ensuring better generalization across diverse tasks while optimizing inference speed and resource usage. By integrating NAS with KD, it becomes possible to create DNN models that are not only computationally efficient but also maintain high segmentation accuracy. NAS can automatically identify optimal architectures while KD ensures that the distilled models inherit robust performance characteristics from larger teacher networks. This combined approach can provide DNN models that generalize well across diverse tasks and datasets while being optimized for inference speed and resource usage.

II. Related Work

This section reviews three key areas relevant to the KD-NAS DNN design methodology for burnt area segmentation: semantic image segmentation, knowledge distillation, and neural architecture search. In semantic image segmentation, each pixel of an image is assigned a unique class label (e.g., burnt region), producing a segmentation mask that identifies different objects and areas. DNNs have been extensively used in semantic image segmentation, with applications spanning autonomous vehicles [3], remote sensing [4], and medical image diagnostics [5]. These models extract spatial features that allow for distinguishing objects, isolating foregrounds from backgrounds, and automating various tasks.

Detailed research on DNN segmentation methods has focused on applications such as smoke, flame, and burnt area segmentation, which are vital for accurate fire detection in disaster scenarios. Innovative approaches have been proposed to segment smoke and flames from images and videos [6], enhancing early detection capabilities and reducing the risk of large-scale damage. For example, [7] not only explores fire detection but also demonstrates how DNN-based segmentation can extract critical features like flames and smoke, enabling precise monitoring and timely intervention.

In the context of burnt area segmentation, five state-of-the-art models including: PIDNet Small, PIDNet Medium, PIDNet Large [8], UNet++ [9], and CNN-I2I BiseNet [10] have shown distinct strengths in addressing segmentation challenges. These models differ in architectural complexity, efficiency, and accuracy; however, deploying them on resource-constrained platforms like wildfire monitoring UAVs remains challenging. To overcome these limitations, this study integrates NAS and KD to enhance and compress these models. NAS is used to explore and optimize architectural designs, while KD transfers knowledge from larger teacher models to efficient student models, aiming to identify architectures that deliver real-time, high-accuracy segmentation of burnt, half-burnt, and non-burnt regions, as well as fire and smoke detection, which are crucial for timely wildfire management and mitigation.

A. Knowledge Distillation

KD trains a compact "student" DNN to match the performance of a larger "teacher" DNN [11]. The student learns from both ground truth labels and the teacher's "soft" predictions (logits), which provide richer information than labels alone [2], [12].

Let $T(x;\theta_T)$ be the teacher model and $S(x;\theta_S)$ the student model, producing logits z_T and z_S. The KD loss function [2] combines cross-entropy loss with the Kullback-Leibler (KL) divergence of the softened teacher–student logits:

$$\begin{aligned}\mathbf{L}_{KD} = \alpha\mathcal{L}_{CE}(y, S(x)) \\ +(1-\alpha)\mathcal{L}_{KL}\big(\text{Softmax}(z_T/T), \text{Softmax}(z_S/T)\big)\end{aligned} \quad (1)$$

where α weighs the two losses, and T is a temperature parameter that softens the logits. By adjusting T and α, the student effectively learns from both ground truth labels and teacher outputs.

B. Neural Architecture Search

NAS automates the design of DNNs by searching a space S of candidate architectures. For any architecture $A \in S$, its performance $P(A)$ is evaluated by training and validating on a given task. NAS seeks the optimal A^* maximizing $P(A)$:

$$A^* = \arg\max_{A \in S} P(A). \quad (2)$$

Simultaneously, NAS can optimize the architecture and its weights W by minimizing a task-specific loss $\mathcal{L}(A, W)$. Efficient methodologies like EfficientNet [13], DARTS [14], and ENAS [15] have made NAS more practical. By incorporating KD with NAS, we aim to discover architectures that achieve high accuracy while remaining computationally efficient, making them well-suited for tasks like burnt area segmentation.

C. Combination of NAS and KD

Several studies have explored combining NAS with KD to enhance model efficiency and performance. For instance, researchers proposed a block-wise architecture search guided by distilled knowledge from a teacher model [16]. Another work introduced a framework integrating oracle knowledge distillation with NAS to optimize memory and improve oracle prediction emulation from ensemble models [17]. Additionally, NAS and KD have been leveraged for medical image segmentation to reduce inference time and computational costs [18].

III. KD-NAS DNN Design Methodology

The proposed KD-NAS DNN design method combines NAS and KD to iteratively identify and optimize DNN architectures. The goal is to develop efficient and high-performing models for burnt area segmentation tasks, which are critical for real-time wildfire management.

A. Neural Architecture Search (NAS)

In our pipeline, NAS automates the design and optimization of DNNs, enabling systematic exploration of architectural possibilities to achieve optimal performance for specific tasks. NAS is employed to design DNN architectures that are efficient and effective for burnt area segmentation, a critical task in real-time wildfire management.

The NAS process explores a unified search space encompassing a wide range of design parameters:

- **Hyperparameters:** Learning rate, Batch size, Weight decay, Momentum, Dropout rate.
- **Architectural Parameters:** Number of layers, Type of layers, Number of neurons or filters per layer, Convolution filter sizes, Strides, Padding, Activation functions, Normalization layers.

This comprehensive search space ensures the discovery of architectures tailored to the unique challenges of burnt

area segmentation while balancing segmentation accuracy and computational efficiency. We use the Tree-structured Parzen Estimator (TPE) method, which builds a simple model to predict which hyperparameter settings are likely to perform well based on previous trials. This helps focus the search on promising areas of the space rather than trying all combinations. The process continues until the mIoU score does not improve over specified consecutive trials.

The NAS procedure is applied individually to five DNN families: PIDNet Small, PIDNet Medium, PIDNet Large, UNet++, and CNN-I2I BiseNet. For each family, NAS is performed to extract multiple candidate architectures that explore a range of trade-offs between computational cost and segmentation performance. These candidate architectures are then evaluated to identify those that excel in balancing performance and efficiency.

To quantify segmentation accuracy, NAS uses the mean Intersection-over-Union (mIoU) as the primary evaluation metric. The mIoU measures the overlap between predicted and ground truth regions across all semantic classes and is defined as:

$$\text{mIoU} = \frac{1}{N} \sum_{i=1}^{N} \frac{|P_i \cap G_i|}{|P_i \cup G_i|}. \tag{3}$$

where: N is the total number of classes, P_i is the set of pixels predicted for class i, G_i is the ground truth set for class i, $|P_i \cap G_i|$ is the number of correctly predicted pixels (intersection), and $|P_i \cup G_i|$ is the total number of pixels from both prediction and ground truth (union). A higher mIoU indicates better alignment between the model's predictions and the ground truth, which is critical for accurately delineating burnt, half-burnt, and non-burnt regions.

B. Knowledge Distillation Framework

The KD framework described in [19] improves how knowledge is transferred and how DNN models (agents) work together. Agents can use their own rules to train their networks, work with other agents, and check how well they understand a dataset. This framework also sets guidelines for how agents should communicate, allowing them to share data, weights, feature maps, and soft targets. By following these rules, agents can easily exchange knowledge in a collaborative environment. Agents can either already have knowledge or learn it, allowing them to be both students and teachers. In addition, the framework helps make experiments easier to repeat by enabling agents to share data, layer output, architectures, and weights. This helps researchers to consistently repeat and confirm their results, creating a stronger and more cooperative research environment for neural networks.

This structured approach ensures efficient knowledge transfer while maintaining reproducibility across experiments, enabling consistent evaluation of the student models.

C. Integration of KD and NAS

The KD-NAS methodology is implemented in three structured stages. NAS is applied independently to each DNN family, generating multiple candidate architectures optimized for performance and efficiency. Each architecture is evaluated based on mIoU, parameter count, and computational cost. The top-performing architectures within each DNN family are designated as teacher models. Using these teacher models, knowledge is distilled into smaller, efficient student models, further optimizing performance. The best-performing student models from each family undergo an additional round of KD, where knowledge is transferred across families. This final step integrates the strengths of multiple DNN families to identify the ultimate winning architecture. The reason we choose the model with the largest number of parameters as the final teacher is that such models typically encode richer intermediate representations and spatial details, which are valuable in dense prediction tasks like segmentation. While an ensemble or highest-mIoU teacher could also be considered, we found that large-capacity models transfer smoother and more informative soft predictions across all pixels, especially around boundaries. These soft targets include not only the highest class probabilities but also relative confidences across neighboring classes, helping the student model better resolve ambiguous or mixed-label pixels (e.g., at smoke–burnt region transitions). This is particularly useful in segmentation, where spatial structure and local context matter. The overview of our pipeline is depicted in Fig. 2.

IV. Experimental KD-NAS Evaluation

A. Dataset and Annotations

The BLAZE dataset, which was specifically designed for burnt image region segmentation, forest fires in Greece, was utilized in this evaluation study. The Blaze dataset comprises 5,408 UAV images for wildfire classification, sourced from 56 videos and supplemented with 829 images from the D-Fire public dataset and 34 from the Burnt Area UAV public dataset. The dataset is categorized into five classes: 'Burnt', 'Half-Burnt', 'Non-Burnt', 'Fire', and 'Smoke'. The BLAZE dataset annotation was carried out using Segments.ai, a sophisticated data labeling platform tailored primarily for computer vision applications. It facilitated precise and efficient burnt image region annotation. For the purpose of training the DNN models, 75% and 25% of the data images were allocated to the disjoint training and the test sets respectively. The KD-NAS methodology was applied to burnt area segmentation, leveraging the BLAZE dataset. The methodology's ability to optimize DNN architectures while addressing the unique challenges of wildfire segmentation demonstrates its effectiveness for real-world applications. A sample of the dataset is presented in Fig. 1.

B. KD-NAS Searchable DNNs

The following DNN families were used in neural architecture search to feed the Knowledge Distillation framework and finally conclude to the ultimate winning DNN architecture.

UNet++: This DNN features a contracting path that repeatedly applies convolutions, ReLUs, and max-pooling for downsampling (doubling the number of feature channels each

Fig. 1: Blaze Dataset: Images showing fire impact, from burnt to non-burnt conditions.

time), and an expansive path that upsamples feature maps, halves their channels, and concatenates them with cropped maps from the contracting path. Cropping is necessary to address border pixel loss caused by convolutions. A final convolution layer produces the desired number of classes, and the network comprises 23 convolutional layers [9].

PIDNET: This DNN model [8] is based on a conceptual link between Convolutional Neural Networks (CNNs) and Proportional-Integral-Derivative (PID) controllers. A two-branch network is equivalent to a Proportional-Integral (PI) controller, which essentially experiences analogous overshoot problems. To solve this issue, PIDNET is a three-branch network architecture for parsing detailed, contextual, and boundary information, respectively, it utilizes boundary attention to guide the integration of detailed and contextual branches. PIDNets achieve a very good balance between inference speed and accuracy, outperforming all existing models with comparable inference speed in terms of accuracy on the Cityscapes and CamVid datasets.

BISENET: BiseNet [20] is a well-known image segmentation architecture. CNN-I2I BiseNet [10] is a BiseNet variant, which integrates a real-time semantic image segmentation network based on CNNs with an image-to-image translation neural branch. Both neural pathways share the same feature extraction CNN and are simultaneously trained using a novel multi-task loss function that takes into account both tasks. Furthermore, skip connections were added between neurons of the two branches, enabling semantic information to transfer from the image-to-image translation neural branch, to the segmentation branch during inference. The complete search space of hyperparameters and architectural parameters for these five architectures is explained in Section III, enabling a thorough exploration and optimization of the architectures.

C. Experimental Results

We have employed five different DNN families: CNN-I2I BiseNet, PIDNET small, medium, and large and UNet++ for our experiments. The first part of Table I illustrates the initial DNN parameters, mIoU and the total number of trainable parameters respectively. Our goal was to reduce the number of parameters and at the same time to increase performance. We can observe that in the beginning, the best performing DNN was CNN-I2I BiseNet with mIoU of 74.82% and a total number of 18,408,758 parameters. The evaluations were performed on an NVidia GeForce GTX 1080 Ti graphics card, using a batch size of 4.

Then, by applying Neural Architecture Search on the expanded search space we created for each DNN families architecture, we achieve similar performance, but with fewer parameters and half-training epochs, as illustrated in the second part of Table I. The optimized DNN models also run faster and use less energy, making them better for real-time applications. This approach proves that by optimizing the search space and hyperparameters, we can create powerful DNN models that are efficient and suitable for a wide range of tasks and devices.

We then use our Knowledge Distillation framework to extract the best performing DNN from DNN families. As illustrated in the third part of Table I, KD leads to improved performance and reduced number of DNN parameters.

Finally, we select the best winning architecture from each DNN family for a final round of Knowledge Distillation. In this round, we choose the DNN with the largest number of parameters as the teacher DNN model. Its extensive knowledge and performance capabilities are then distilled into the student DNN models. This final step ensures that the student DNN models, which are more efficient and have fewer parameters, inherit the high performance and accuracy of the larger teacher DNN model. The result of the overall KD-NAS method illustrated in the fourth part of Table I is a set of optimized DNNs that maintain top-tier performance, while being resource efficient and suitable for deployment across various platforms and applications. The final winning architecture is PidNet-Small with 6,942,117 trainable parameters and an mIoU of 75.84%. This represents a reduction of 62.3% in the total trainable parameters and a 1.02% improvement in mIoU, along with a reduction in training epochs from 120 to 50. While the absolute increase of +1.02% in mIoU may appear modest, it is meaningful in dense segmentation tasks, especially for wildfire mapping where accurate boundary detection of burnt, half-burnt, and smoke regions can guide real-time emergency decisions. The main advantage of our pipeline is the large efficiency gain: the final PidNet-Small model is 62.3% lighter than the CNN-I2I baseline, making it deployable on edge devices such as UAVs or embedded boards. We further note that this mIoU improvement is consistent across different wildfire scenes in the BLAZE dataset. Across five subsets covering urban–rural edges, dense forests, and synthetic smoke overlays, the optimized model outperformed the baseline. This suggests that the gain is not only statistically reliable but also practically useful in varied real-world conditions.

V. Conclusion

In this study, we proposed the novel KD-NAS method for DNN model design that combines NAS with KD to iteratively select optimal DNN architectures for specific tasks. Through experimentation and evaluation on the BLAZE image dataset, we demonstrated the effectiveness of our method in identifying superior architectures across five distinct state-of-the-art DNN families. Our results indicate that the cascade combination of NAS and KD allows for the systematic exploration of

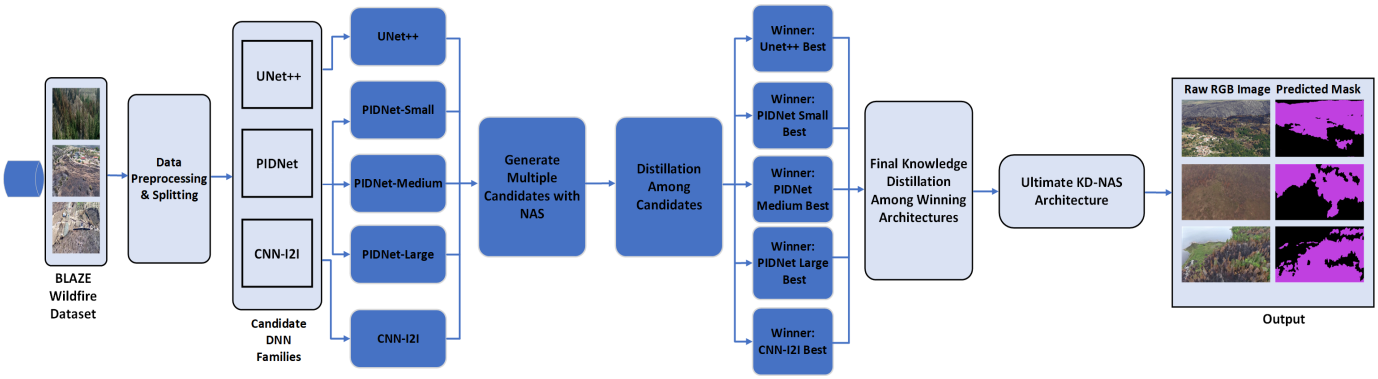

Fig. 2: The KD-NAS Pipeline.

TABLE I: Comparison of segmentation performance across all pipeline stages.

Model	Baseline			NAS			NAS + KD			Final KD-NAS		
	mIoU %	Epochs	Params	mIoU %	Epochs	Params	mIoU %	Epochs	Params	mIoU %	Epochs	Params
CNN-I2I BiseNet	74.82	120	18.4M	74.18	50	17.8M	74.18	50	17.8M	73.92	50	17.8M
PIDNet-Small	73.79	120	7.72M	74.22	50	6.97M	75.04	50	6.94M	**75.84**	50	**6.94M**
PIDNet-Medium	71.27	120	28.8M	71.07	50	19.6M	71.93	50	14.5M	72.94	50	14.5M
PIDNet-Large	70.13	120	37.3M	69.66	50	28.0M	71.08	50	19.3M	71.08	50	19.3M
UNet++	64.11	120	7.76M	65.47	50	1.93M	67.14	50	1.86M	68.21	50	1.86M

architectural DNN variations, leading to the identification of DNN architectures that strike a balance between performance and computational efficiency. By automating DNN architecture selection, we accelerate DNN development and deployment for real-world big data and embedded/edge computing applications.

Acknowledgment

The research leading to these results has received funding from the European Commission - European Union (under HORIZON EUROPE (HORIZON Research and Innovation Actions) under grant agreement 101093003 (TEMA) HORIZON-CL4-2022-DATA-01-01). Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the European Union - European Commission. Neither the European Commission nor the European Union can be held responsible for them.

References

[1] Knopp, L., Wieland, M., Rättich, M., Martinis, S.: A Deep Learning Approach for Burned Area Segmentation with Sentinel-2 Data. Remote Sensing **12**(15), 2422 (2020)

[2] Hinton, G., Vinyals, O., Dean, J.: Distilling the Knowledge in a Neural Network. arXiv preprint arXiv:1503.02531 (2015)

[3] Chen, C., Wang, C., Liu, B., He, C., Cong, L., Wan, S.: Edge Intelligence Empowered Vehicle Detection and Image Segmentation for Autonomous Vehicles. IEEE Transactions on Intelligent Transportation Systems **24**(11), 13023–13034 (2023).

[4] Kotaridis, I., Lazaridou, M.: Remote sensing image segmentation advances: A meta-analysis. ISPRS Journal of Photogrammetry and Remote Sensing **173**, 309–322 (2021)

[5] Siddique, N., Paheding, S., Elkin, C. P., Devabhaktuni, V.: U-Net and Its Variants for Medical Image Segmentation: A Review of Theory and Applications. IEEE Access **9**, 82031–82057 (2021).

[6] Khan, S., Muhammad, K., Hussain, T., Del Ser, J., Cuzzolin, F., Bhattacharyya, S., Akhtar, Z., de Albuquerque, V. H. C.: Deepsmoke: Deep Learning Model for Smoke Detection and Segmentation in Outdoor Environments. Expert Systems with Applications **182**, 115125 (2021)

[7] Frizzi, S., Bouchouicha, M., Ginoux, J.-M., Moreau, E., Sayadi, M.: Convolutional Neural Network for Smoke and Fire Semantic Segmentation. IET Image Processing **15**(3), 634–647 (2021)

[8] Xu, J., Xiong, Z., Bhattacharyya, S. P.: PIDNet: A Real-time Semantic Segmentation Network Inspired by PID Controllers. arXiv preprint arXiv:2206.02066 (2023)

[9] Ronneberger, O., Fischer, P., Brox, T.: U-Net: Convolutional Networks for Biomedical Image Segmentation. arXiv preprint arXiv:1505.04597 (2015)

[10] Papaioannidis, C., Mademlis, I., Pitas, I.: Autonomous UAV Safety by Visual Human Crowd Detection Using Multi-Task Deep Neural Networks. In: 2021 IEEE International Conference on Robotics and Automation (ICRA), pp. 11074–11080. IEEE (2021).

[11] Wang, L., Yoon, K.-J.: Knowledge Distillation and Student-Teacher Learning for Visual Intelligence: A Review and New Outlooks. IEEE Transactions on Pattern Analysis and Machine Intelligence **44**(6), 3048–3068 (2022).

[12] Cho, J. H., Hariharan, B.: On the Efficacy of Knowledge Distillation. arXiv preprint arXiv:1910.01348 (2019)

[13] Tan, M., Le, Q.: EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In: Proceedings of the 36th International Conference on Machine Learning, vol. 97, pp. 6105–6114. PMLR (2019)

[14] Liu, H., Simonyan, K., Yang, Y.: DARTS: Differentiable Architecture Search. arXiv preprint arXiv:1806.09055 (2019)

[15] Pham, H., Guan, M. Y., Zoph, B., Le, Q. V., Dean, J.: Efficient Neural Architecture Search via Parameter Sharing. arXiv preprint arXiv:1802.03268 (2018)

[16] Li, C., Peng, J., Yuan, L., Wang, G., Liang, X., Lin, L., Chang, X. (2020). Block-wisely supervised neural architecture search with knowledge distillation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 1989-1998).

[17] Kang, M., Mun, J., Han, B.. Towards oracle knowledge distillation with neural architecture search. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 34, No. 04, pp. 4404-4411).

[18] Zheng, Z., Kang, G.. Model compression with nas and knowledge distillation for medical image segmentation. In 2021 4th International Conference on Data Science and Information Technology (pp. 173-176).

[19] Kaimakamidis, A., Pitas, I.: Facilitating Experimental Reproducibility in Neural Network Research with a Unified Framework. In: Proceedings of the IEEE/ACM 10th International Conference on Big Data Computing, Applications and Technologies, pp. 1–5. ACM (2024).

[20] Yu, C., Wang, J., Peng, C., Gao, C., Yu, G., Sang, N.: Bisenet: Bilateral Segmentation Network for Real-Time Semantic Segmentation. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 325–341 (2018)