Automating UNet Architecture Search for Fetal Head Segmentation

Ayush Lokare
Dept. of Computer Science and Engineering
Manipal Institute of Technology, Manipal Academy of Higher Education
Manipal, India
lokareaayush@gmail.com

J. R. Harish Kumar
Dept. of Electrical and Electronics Engineering
BITS Pilani, WILP
Bengaluru, India
harish.kumar@pilani.bits-pilani.ac.in

***Abstract*—Accurate estimation of fetal head circumference (HC) is a critical biometric task in prenatal care used to monitor fetal growth and estimate gestational age. Traditionally, this process involves manual tracing by sonographers, which is prone to significant intra- and inter-operator variability. To address these limitations, this paper explores the automation of the fetal head segmentation process using advanced deep learning architectures. We specifically focus on optimizing the U-Net framework by automating the search for optimal hyperparameters and architectural configurations.**

Our methodology utilizes KerasTuner to systematically evaluate a variety of CNN components, including depth, activation functions, and filter multipliers, moving beyond traditional manual design. We also investigate the efficacy of transfer learning using an EfficientNet backbone and evaluate the performance of nnU-Net as a comprehensive end-to-end segmentation solution. Experiments conducted on the HC18 challenge dataset demonstrate that our Bayesian-optimized U-Net++ model achieves a mean Dice score of 97.87%, outperforming several standard architectures. The results indicate that automated architecture search can significantly enhance segmentation robustness across different trimesters, providing a more reliable tool for clinical ultrasound analysis and reducing the dependency on manual contouring expertise.

***Index Terms*—head circumference, ultrasound imaging, segmentation, convolutional neural networks, biomarker estimation, nnU-Net, neural architecture search**

I. Introduction

Prenatal care relies heavily on precise biometric measurements to ensure healthy fetal development. Among these, Head Circumference (HC) stands as a primary indicator for monitoring gestational age and detecting potential growth abnormalities. While ultrasound (US) is the preferred diagnostic modality due to its non-invasive nature and cost-effectiveness compared to CT or MRI [1], it presents significant challenges for automated analysis.

The primary hurdle in clinical settings remains the reliance on manual contouring. This process is inherently operator-dependent, leading to notable intra- and inter-operator variability, often reaching 95% limits of agreement of ±7 mm and ±12 mm, respectively [2]. Such discrepancies can lead to inconsistent growth tracking.

The HC18 challenge [3] was established to address this by providing a standardized dataset of 1,334 2D ultrasound images for automated computation. Despite the availability of data, accurate HC estimation remains difficult due to inherent speckle noise and low contrast. While recent advancements have integrated attention mechanisms [4] and residual connections [5], the search for an optimal, robust architecture remains a complex task that often requires significant manual trial and error.

II. Prior Work

Approaches to measuring the fetal head generally fall into three categories [6]. *Global intensity-based methods*, such as thresholding and region growth, utilize image contrast to identify structures. However, they lack robustness to the high speckle noise and acoustic shadows typical in ultrasound, often leading to over-segmentation or requiring extensive manual parameter tuning. *Deformable models* iteratively adjust shapes to match boundaries; while accurate, they are computationally expensive and highly sensitive to initialization. If the initial contour is not placed near the true boundary, these models risk converging to local minima, limiting their full automation. Recently, *learning-based methods*, specifically Convolutional Neural Networks (CNNs), such as U-Net, have become the standard for medical segmentation due to their ability to learn robust hierarchical features from annotated data without manual intervention.

III. Methodology

We explore several segmentation architectures that have achieved state-of-the-art results in medical image segmentation. These architectures were trained from scratch. We also explored transfer learning using EfficientNet pre-trained on the ImageNet dataset to leverage high-level features. Fig. 1 summarizes the complete pipeline, from the input ultrasound image through preprocessing, segmentation, and post-processing to the final head circumference measurement.

A. Data Preprocessing and Augmentation

The HC18 training dataset contains 999 ultrasound (US) images captured at different stages of pregnancy across multiple trimesters. Each image is paired with ground-truth data, including a skull contour map and associated head circumference (HC) values. Skilled sonographers annotated the fetal head contour as an elliptical shape. The dataset provides

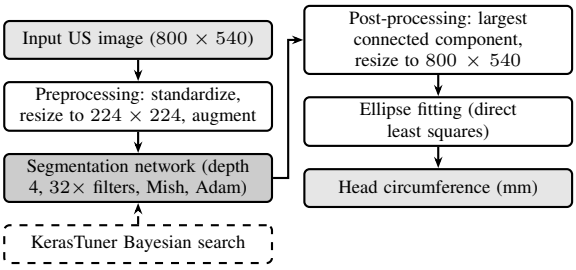

Fig. 1. End-to-end pipeline for automated head circumference measurement. The configuration of the segmentation network is selected by the KerasTuner Bayesian search (Table III).

HC measurements and pixel dimensions for each image in a corresponding text file and spans gestational ages from 10 to 40 weeks [3]. The images were resized from 800×540 to 224×224 and normalized. Although resizing changes the aspect ratio during feature extraction, the spatial integrity of the measurements is preserved during post-processing by projecting predictions back to the original image space. The training dataset was split into a training set (800 images) and a validation set (199 images). Submissions included ellipse parameters and head circumference, which were evaluated on 335 test images via the HC18 challenge website. We used the Albumentations [7] library to perform complex augmentations. We applied flips, contrast variations, rotations, and brightness changes. CLAHE [8] was used to improve low-contrast medical images. Gamma correction was used to correct images with biased grayscale values and to enhance contrast. Noise-adding and blurring augmentations may be less relevant for ultrasound images, which are already low-contrast and noisy.

B. Training Models

1) KerasTuner: Manual architecture search is unreliable and depends on domain expertise. Therefore, there is a need to automate architecture and hyperparameter search. While manual design is common, the trend is shifting toward Medical Neural Architecture Search (MedNAS) [9]. Our approach utilizes KerasTuner to bridge the gap between manually designed U-Nets and fully automated search spaces. KerasTuner [10] is a hyperparameter-optimization framework designed to simplify the process of finding an effective set of hyperparameters. Hyperparameter optimization involves fine-tuning parameter values to maximize model performance. KerasTuner provides an API to define and explore different hyperparameter configurations. With built-in search algorithms such as Hyperband [11] and the Bayesian Optimization tuner [12], KerasTuner searches the hyperparameter space to identify promising values. For Hyperband, the number of epochs must be specified; for Bayesian Optimization, the maximum number of trials must be specified. We evaluated settings of 100 and 250 for each method to study whether increasing these values improves results. The results are shown in Tables I and VI. The model trained with the configuration achieving the best mean Dice score is used for later comparisons with other methods.

TABLE I
MODEL EVALUATION METRICS ON TRAINING SET

Model	Prec.	Spec.	Sens.	Jacc.	Dice
Hyperband250	0.9651	0.9862	0.9926	0.9594	0.9782
Hyperband100	0.9760	0.9901	0.9760	0.9527	0.9755
Bayesian250	0.9916	0.9971	0.9916	0.9820	0.9903
Bayesian100	0.9958	0.9982	0.9954	0.9885	0.9942

TABLE II
HYPERPARAMETER RANGES

Parameter	Options
Depth of the UNet	2, 3, 4, 5
Activation Function	ELU, Mish, LeakyReLU
Dropout	0, 0.1, 0.2, 0.3, 0.4, 0.5
Filter Multiplier	16x, 32x, 48x, 64x
Optimizer	Adam, SGD, Adagrad
Learning Rate	10^{-3} to 10^{-5}

TABLE III
SELECTED HYPERPARAMETER CONFIGURATION (BAYESIAN TUNER)

Parameter	Selected Value
Depth of the UNet	4
Activation Function	Mish
Dropout	0.2
Filter Multiplier	32x
Optimizer	Adam
Learning Rate	2.1×10^{-4}

a) Hyperband: Hyperband is a trial-scheduling algorithm that improves hyperparameter-optimization efficiency by allocating more resources to promising configurations based on initial performance. It tests many configurations using successive halving and early stopping. Hyperband does not use the results of previous configurations.

b) Bayesian Optimization: Bayesian optimization uses probabilistic models and acquisition functions to guide the search for optimal hyperparameters. It leverages results from previous trials to explore the search space efficiently and identify configurations with high potential for improvement. However, this sequential nature restricts parallelization while testing different configurations. We defined the search space as shown in Table II.

Due to limited resources, we encountered out-of-memory errors when the U-Net depth exceeded 4 and the filter multiplier exceeded 32. With the selected hyperparameter configuration, each model was trained from scratch. The configuration selected by the Bayesian tuner is given in Table III. The ten highest-scoring trials show strong agreement in architecture. All ten selected a network depth of 4 with Mish activation and the Adam optimizer, and nine of ten selected a filter multiplier of 32 with dropout 0.2; the single exception used 16 filters and dropout 0.1. Learning rate was the only parameter varying appreciably, spanning 1.6×10^{-4} to 4.7×10^{-4} across these trials, while validation Dice varied by less than 0.004. The search therefore converged on a consistent architecture, with performance largely insensitive to learning rate within that band. We note that this consensus reflects the configurations

the search favored rather than an isolated measurement of each component's marginal contribution, which would require a controlled component-wise ablation.

2) nnU-Net: nnU-Net [13], an extension of the U-Net architecture, was used in our experiments. This framework, known for its adaptability and robustness, was trained using a combination of Dice and cross-entropy loss with five-fold cross-validation. In the Medical Segmentation Decathlon challenge, nnU-Net achieved the highest mean Dice scores across most tasks, demonstrating strong performance without requiring manual, task-specific adjustments. These results, along with the framework's ability to self-adapt, motivated our use of this model. We trained nnU-Net on Google Colab for 100 epochs. This training process is extremely resource-intensive for both CPU and GPU. The 2D variant of nnU-Net was trained for one fold for 150 epochs.

3) Transfer learning: Transfer learning leverages a pre-trained model's weights and architecture for a different but related task. In U-Net, it can improve performance by using a pre-trained encoder and training the decoder (and, optionally, the final layers) from scratch. We used EfficientNet [14] as the backbone for U-Net, replacing the encoder. Outputs from intermediate EfficientNet blocks were used as skip connections, while the decoder was trained from scratch. We evaluated EfficientNet variants from B4 to B8. Because the EfficientNet input size acts as a scaling parameter for its convolutional blocks, input resolution influences the choice of variant.

4) Segment Anything Model: The Segment Anything Model (SAM) [15] is designed to transfer its knowledge to new image distributions and tasks. Its diverse training data enable it to generate masks for objects in images or videos, including those not seen during training. SAM often delivers strong zero-shot performance, sometimes matching or outperforming prior fully supervised approaches. Points and boxes can be provided as prompts. We used U-Net++ as a prompting model and provided the ellipse center as a prompt. The results were inconsistent, and the model did not segment elliptical shapes reliably. Consequently, OpenCV's ellipse-fitting routine could not fit an ellipse to many predicted masks. Therefore, this approach did not produce usable results.

C. Post-processing

The segmentation results may contain structural inaccuracies, such as holes or spurious noise. To prevent these artifacts from affecting ellipse fitting, we apply post-processing. We detect contours in the predicted segmentation and select the largest connected component. To ensure that the head circumference (HC) calculation remains geometrically accurate, we apply an inverse transformation to the predicted masks. The 224×224 segmentation output is resized back to the original image dimensions (800×540) before ellipse fitting, ensuring that physical measurements (mm) correspond to the original anatomical proportions. We then fit an ellipse to the segmentation mask to obtain ellipse parameters. The semi-major and semi-minor axes are used to compute head circumference using (1), in which the auxiliary term h is defined by (2).

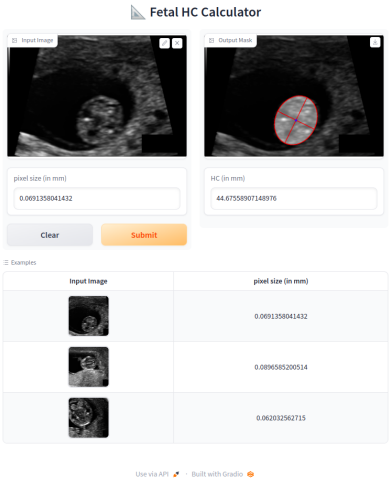

Fig. 2. User Interface made using Gradio hosted as a HuggingFace space

TABLE IV
COMPARISON OF MODEL PERFORMANCE ON TEST SET

Model	Dice (%)	MAE (mm)
UNet++ (Keras Tuner)	97.87 ± 1.12	1.95 ± 1.86
Efficient-UNet	97.39 ± 3.17	2.45 ± 3.32
nnUNet	95.46 ± 6.45	2.58 ± 5.71

$$HC = \pi(a+b)\left(1 + \frac{3h}{10 + \sqrt{4-3h}}\right). \tag{1}$$

$$h = \frac{(a-b)^2}{(a+b)^2} \tag{2}$$

IV. RESULTS

We report precision, specificity, sensitivity and the Jaccard index, all derived from the true positive, true negative, false positive and false negative counts, together with the Dice coefficient, which is equivalent to the F-score in binary segmentation. The mean absolute error (MAE), defined in (3), is the average absolute difference between the predicted and ground-truth head circumference values (in mm), and is used as an evaluation metric in the HC18 challenge.

$$MAE_{HC} = \frac{1}{N}\sum_{i=1}^{N}\left|\hat{HC}_i - HC_i\right| \tag{3}$$

We created a user interface using the Gradio [16] library and hosted it on Hugging Face, as shown in Fig. 2. Users upload an ultrasound image and enter its pixel size; example images and pixel sizes are provided. The uploaded image can also be cropped in the browser. The live demo link is https://aaylmao-hc-prediction.hf.space/.

A comparison of the trained models is shown in Table IV. The mean absolute difference (mm) determines submission quality.

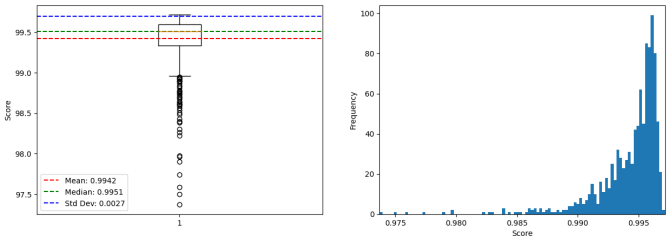

Fig. 3. Distribution of Dice scores for U-Net++ tuned with the Bayesian optimization tuner

TABLE V
COMPARISON OF TRAINED MODELS ON THE HC18 TEST DATASET

Model	Dice (test)
SegNet	95.02
nn-UNet (proposed)	95.46
U-Net	96.51
Res-U-Net	97.21
Dilated SE-Unet	97.24
Efficient-U-Net (proposed)	97.39
U-Net++	97.46
Fusion U-Net++	97.49
CE-Net	97.52
U-Net++ (tuned, proposed)	97.87
GCA Net	98.21

The metrics in Table I were computed on the training dataset. Fig. 3 shows the distribution of the obtained Dice scores.

Table V compares our proposed approaches with other models reported on the HC18 test set.

Table VI and Fig. 4 show the distribution of Dice scores across trimesters. Performance in the first trimester is lower than in the other two trimesters, and this trend is also visible in Figs. 5 and 6. Three factors contribute. First, the fetal skull is not yet visible as a bright structure in the first trimester [3]; because it is still relatively soft, it does not reliably appear brighter than the interior of the fetal head, making the outer edge difficult to localize, particularly where the head lies close to the uterine wall. The boundary is therefore diffuse, and speckle is more likely to dominate a low-contrast edge. Our lowest-scoring cases show exactly this failure: the predicted contour extends past the annotated boundary into adjacent tissue. Second, the head is smaller at early gestational ages, and because Dice normalizes the intersection by the summed areas, a boundary error of fixed width incurs a larger penalty on a small region than on a large one. Third, first-trimester images are under-represented in HC18, and our tuner objective is the aggregate validation Dice, computed without stratification by gestational age, so the search receives no signal to penalize degradation on the minority group. Three remedies follow: a boundary-aware loss, a gestational-age-stratified search objective, and addressing the imbalance directly through stratified sampling or GAN-based synthesis of first-trimester examples [4].

Figs. 5 and 6 show sample images, predicted masks, ground-truth masks, and the overlap between predictions and ground

TABLE VI
MODEL EVALUATION METRICS ON TEST SET. T1–T3 DENOTE THE FIRST, SECOND AND THIRD TRIMESTER.

Model	T1	T2	T3	Mean
Bayesian250	96.73	97.96	97.94	97.75
Hyperband250	94.30	97.94	97.90	97.34
Hyperband100	96.14	97.90	97.99	97.62
Bayesian100	96.84	98.07	98.05	97.87

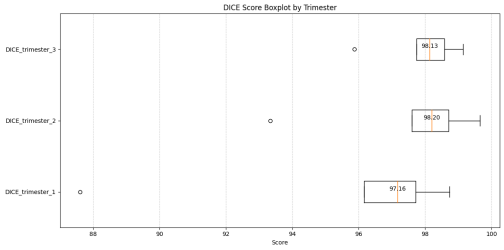

Fig. 4. Trimester-wise distribution of Dice scores on the test set

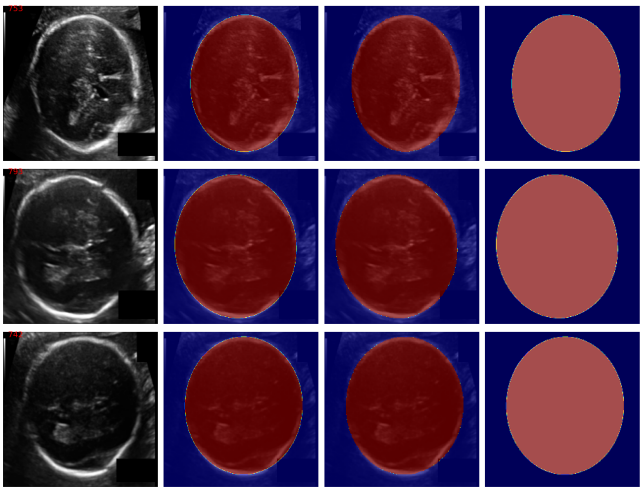

Fig. 5. Images with the highest Dice scores in the training set

truth. Misclassified pixels appear as a distinct color near the boundaries in the fourth column of each row.

A. Ablation of the Search Procedure

Tables I and VI permit an ablation over the two components of our search procedure: the optimization strategy and the trial budget. At both budgets, Bayesian optimization outperforms Hyperband on the test set (97.87% against 97.62% at the lower budget, and 97.75% against 97.34% at the higher), consistent with its use of information from prior trials. Increasing the trial budget did not improve test performance for either strategy, indicating that the search saturates well within the smaller budget. Comparing training and test metrics additionally shows that the Bayesian runs attain markedly higher training Dice (0.9942 and 0.9903, against 0.9755 and 0.9782 for Hyperband) without a proportionate gain on the test set, indicating mild overfitting to the search objective. Table IV extends this comparison across approaches, showing

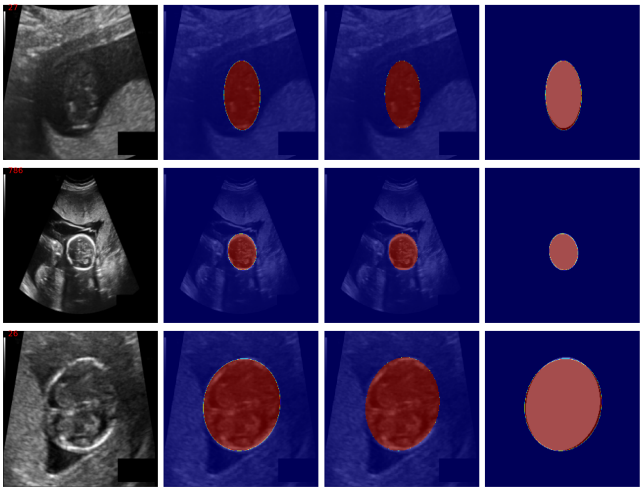

Fig. 6. Images with the lowest Dice scores in the training set

that the tuned model outperforms both EfficientNet-based transfer learning and nnU-Net under our compute budget.

V. Limitations of This Study

Dataset and Generalization: Our results derive from a single benchmark. HC18 was collected at one center using two devices from a single vendor. Performance is therefore unverified under distribution shifts like different vendors, populations, or acquisition settings. The publicly released training annotations are also single-observer. External validation on an independently annotated, multi-center dataset including growth-restricted cases remains necessary.

Search Space and Compute: Our architecture search was constrained by the compute available on a single standard Google Colab GPU instance (16 GB device memory). Configurations combining depth above 4 with filter multipliers above 32 exceeded device memory, so the deeper and wider region of the search space was not explored.

Clinical implications: The system is decision support, not autonomous measurement. It presumes a correctly acquired standard plane; obtaining the 2D standard plane remains a requirement of the measurement protocol and is itself a major source of clinical error, which the model neither verifies nor flags.

VI. Conclusion

Table V compares models from the HC18 competition with our proposed methods. Fine-tuning U-Net++ outperforms most specialized models and provides competitive performance with state-of-the-art methods while requiring less training effort. With limited resources, approaches such as nnU-Net may underperform. Because hyperparameter selection is automated, a notable performance improvement can be achieved with minimal effort using hyperparameter-optimization techniques provided by libraries such as KerasTuner.

References

[1] S. Rueda, S. Fathima, C. L. Knight, M. Yaqub, A. T. Papageorghiou, B. Rahmatullah, A. Foi, M. Maggioni, A. Pepe, J. Tohka, R. V. Stebbing, J. E. McManigle, A. Ciurte, X. Bresson, M. B. Cuadra, C. Sun, G. V. Ponomarev, M. S. Gelfand, M. D. Kazanov, C. W. Wang, H. C. Chen, C. W. Peng, C. M. Hung, and J. A. Noble, "Evaluation and comparison of current fetal ultrasound image segmentation methods for biometric measurements: a grand challenge," *IEEE Trans. Med. Imag.*, vol. 33, no. 4, pp. 797–813, Apr. 2014.

[2] I. Sarris, C. Ioannou, P. Chamberlain, E. Ohuma, F. Roseman, L. Hoch, D. G. Altman, A. T. Papageorghiou, and International Fetal and Newborn Growth Consortium for the 21st Century (INTERGROWTH-21st), "Intra-and interobserver variability in fetal ultrasound measurements," *Ultrasound Obstet. Gynecol.*, vol. 39, no. 3, pp. 266–273, Mar. 2012.

[3] T. L. A. van den Heuvel, D. de Bruijn, C. L. de Korte, and B. van Ginneken, "Automated measurement of fetal head circumference using 2D ultrasound images," *PLoS ONE*, vol. 13, no. 8, p. e0200412, Aug. 2018. [Online]. Available: https://doi.org/10.1371/journal.pone.0200412

[4] N. A. E. Joudi, M. Lazaar, and F. Delmotte, "U-net-based fetal head circumference segmentation with synthetic-driven generation data augmentation," *International Journal of Imaging Systems and Technology*, vol. 35, no. 6, 2025.

[5] L. Halder, S. Mohanto, M. Islam, and M. T. Jawad, "Fetal head segmentation and head circumference measurement using residual u-net," in *2024 2nd International Conference on Information and Communication Technology (ICICT)*. IEEE, 2024, pp. 145–149.

[6] H. R. Torres, P. Morais, B. Oliveira, C. Birdir, M. Rüdiger, J. C. Fonseca, and J. L. Vilaça, "A review of image processing methods for fetal head and brain analysis in ultrasound images," *Comput. Methods Programs Biomed.*, vol. 215, p. 106629, Mar. 2022.

[7] A. Buslaev, V. I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin, and A. A. Kalinin, "Albumentations: Fast and flexible image augmentations," *Information*, vol. 11, no. 2, p. 125, Feb. 2020. [Online]. Available: https://www.mdpi.com/2078-2489/11/2/125

[8] A. M. Reza, "Realization of the contrast limited adaptive histogram equalization (CLAHE) for real-time image enhancement," *J. VLSI Signal Process. Syst. Signal Image Video Technol.*, vol. 38, pp. 35–44, Aug. 2004.

[9] L. Meng, F. Nie, Y. Zhang, and C. Han, "Optimizing neural network architecture for medical image segmentation using monte carlo tree search," *arXiv preprint arXiv:2602.22361*, 2026. [Online]. Available: https://arxiv.org/abs/2602.22361

[10] T. O'Malley, E. Bursztein, J. Long, F. Chollet, H. Jin, L. Invernizzi, G. de Marmiesse, Y. Fu, J. Podivìn, and F. Schäfer. (2023) Keras tuner. Accessed: Apr. 2, 2022. [Online]. Available: https://github.com/keras-team/kerastuner

[11] L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh, and A. Talwalkar, "Hyperband: A novel bandit-based approach to hyperparameter optimization," *J. Mach. Learn. Res.*, vol. 18, no. 1, pp. 6765–6816, Jan. 2017.

[12] K. Kandasamy, W. Neiswanger, J. Schneider, B. Poczos, and E. P. Xing, "Neural architecture search with Bayesian optimisation and optimal transport," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 31, Montreal, QC, Canada, Dec. 2018, pp. 5671–5681.

[13] F. Isensee, J. Petersen, A. Klein, D. Zimmerer, P. F. Jaeger, S. Kohl, J. Wasserthal, G. Koehler, T. Norajitra, and S. Wirkert. (2018) nnU-Net: Self-adapting framework for U-Net-based medical image segmentation. [Online]. Available: https://arxiv.org/abs/1809.10486

[14] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. 36th Int. Conf. Mach. Learn. (ICML)*, Long Beach, CA, USA, Jun. 2019, pp. 6105–6114.

[15] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, and W.-Y. Lo. (2023) Segment anything. [Online]. Available: https://arxiv.org/abs/2304.02643

[16] A. Abid, A. Abdalla, A. Abid, D. Khan, A. Alfozan, and J. Zou. (2019) Gradio: Hassle-free sharing and testing of ML models in the wild. [Online]. Available: https://arxiv.org/abs/1906.02569