Evolutionary Neural Architecture Search for Bladder Cancer: A Multi-Objective Investigation

Haithem DAHIMI[1], Mhand HIFI[2⋆], Fabien SAINT[3]

***Abstract*— Artificial intelligence holds significant promise for enhancing diagnostic accuracy and mitigating subjectivity in medical imaging, particularly for high-stakes tasks such as cancer detection. However, model performance frequently degrades when transitioned to real-world clinical scenarios. Employing a proprietary clinical dataset consisting of white-light cystoscopy videos acquired from the Amiens University Hospital, this study investigates the design of robust Convolutional Neural Networks (CNNs) through the Non-dominated Sorting Genetic Algorithm (NSGA-II). The search process was configured to automatically generate architectures that maximize both mean sensitivity and mean specificity within a 5-fold cross-validation framework. Search efficiency was evaluated using the HI. The evolved architectures were subsequently benchmarked against VGG16, ResNet50, and DenseNet121 models pretrained on ImageNet. Experimental results show a gradual increase in the HI from 0.60 to 0.67 during the initial phase, followed by a decline upon the inclusion of the third fold, suggesting a "generalization shock" before a final marginal recovery. These fluctuations coincided with a substantial reduction in genotypic diversity from 0.87 to 0.28, leading to premature convergence in the NSGA-II optimization. Consequently, the resulting CNN architecture remained below the ResNet50 baseline, highlighting that further refinement of optimization strategies and training hyperparameters is essential to bridge the gap between automated design and expert-crafted models.**

I. INTRODUCTION

The use of deep learning in medical applications has gained significant traction in recent years. One of its most common applications is in the field of oncology, where diagnosis is conducted using models trained on medical images. Such tools can make diagnosis faster and more reliable, improving the overall treatment outcome. Yet, one of the least studied applications of AI is in the diagnosis of bladder cancer using white light cystoscopy (WLC). WLC performance varies significantly depending on the urologist's experience, due to the subjective nature of the medical assessment. Nevertheless, WLC remains one of the most common diagnostic methods due to its relatively low cost and availability. In such a context, deep learning can be used to build a supplementary assisting tool integrated into the WLC clinical diagnosis process. Such a tool would help urologists in the decision-making process and improve WLC performance without significantly increasing its cost. This motivation is consistent with recent multicenter and diagnostic studies showing the potential of AI-assisted cystoscopy for bladder cancer detection [16], [17].

One of the key steps in the deep learning process is selecting an appropriate model architecture and identifying the optimal hyperparameters to best fit the data. Since our task involves image data, convolutional neural networks (CNNs) are particularly well suited due to their ability to automatically extract meaningful features while preserving spatial relationships within images. Moreover, CNNs significantly reduce the number of trainable parameters compared to fully connected neural networks, making them more efficient and practical for handling high-dimensional inputs such as images. However, even with domain knowledge, it is time consuming to find a CNN architecture suited for the task at hand. To address this issue, multiple Neural Architecture Search (NAS) algorithms were developed to automate the design of CNN models.

In our study, we employ the non-dominated sorting genetic algorithm II (NSGA-II) as a population-based NAS method in which each individual represents a CNN architecture. Our purpose behind this process is the optimization of a CNN with the aim of maximizing two key conflicting performance metrics: sensitivity and specificity. Following NASNet, we design a cell-based search space using directed acyclic graphs (DAGs) where each edge corresponds to a specific neural operation. Through this process, each individual encodes the design of both a normal and a reduction cell that we stack three times to build the final architecture. To evaluate the NSGA-II process, we employ the Hypervolume Indicator (HI), a widely recognized metric in multi-objective optimization, which quantifies the dominance and spread of the potential architecture solutions in the objective space. Finally, we compare the resulting model performance to a set of standard pretrained models in a 5-fold cross validation setting.

II. BACKGROUND AND RELATED WORKS

Designing a convolutional neural architecture that optimizes a single performance metric is a laborious process, traditionally relying on extensive manual trial and error. In the presence of multiple conflicting objectives, identifying a balanced architecture becomes significantly more complex, necessitating the adoption of automated NAS frameworks.

A. Single-Objective NAS

Early milestones in NAS focused primarily on the single-objective optimization of accuracy-related metrics. A pivotal

⋆Corresponding author

[1]Haithem DAHIMI, EPROAD UR 4669, Université de Picardie Jules Verne, 80000 Amiens, France (haithem.dahimi@u-picardie.fr)

[2]Mhand HIFI EPROAD UR 4669, Université de Picardie Jules Verne, 80000 Amiens, France (hifi@u-picardie.fr)

[3]Fabien SAINT, EPROAD UR 4669, Université de Picardie Jules Verne, 80000 Amiens, France (fabien.saint@u-picardie.fr)

shift occurred with the introduction of RNN-based controllers, where neural network topologies and connectivity patterns were expressed as variable-length strings. This formulation enabled the application of reinforcement learning (RL) to navigate the architecture space.

In the landmark NASNet framework [1], the complexity was managed by shifting from global architecture search to the optimization of modular building blocks. Specifically, the RNN controller was tasked with evolving two distinct types of cells: a *Normal Cell*, which preserves spatial resolution, and a *Reduction Cell*, which halves the feature map dimensions while doubling the channel depth. Each cell is represented as a Directed Acyclic Graph (DAG) defined by:

- *Nodes*: Represent intermediate feature maps resulting from the aggregation of preceding operations.
- *Edges*: Represent specific operations (e.g., convolutions, pooling) sampled from a predefined search space.

The final CNN macro-architecture is subsequently constructed by stacking these optimized cells in a predefined alternating sequence.

Alternative strategies have since emerged to enhance search efficiency. DARTS [2] adopts the NASNet cell-based search space but replaces RL with gradient-based optimization. By relaxing the discrete search space into a continuous, parameterized mixture of candidate operations, DARTS enables the use of gradient descent to minimize validation loss directly.

Parallel to these gradient-based methods, Evolutionary Algorithms (EA) have demonstrated significant robustness in NAS. These population-based techniques are inspired by biological processes; for instance, AmoebaNet [3] utilizes a tournament selection-based algorithm. In this approach, each individual—encoded as a pair of Normal and Reduction cell DAGs—undergoes a cycle of performance-based selection, mutation, and aging-based elimination. Through this evolutionary pressure, the population explores the vast architectural search space to identify increasingly optimal configurations.

B. Multi-objective NAS

In recent years, research has increasingly shifted toward multi-objective NAS, driven by the necessity of balancing performance with resource efficiency. Some approaches rely on RL algorithms that incorporate multi-faceted reward functions. For instance, MONAS [4] utilizes a linear combination of accuracy and power consumption as a reward signal for the LSTM controller to optimize CNN hyperparameters. However, a significant portion of multi-objective NAS methods leverage the inherent parallel search capabilities of EA-based strategies. A notable example is NSGAnet [5], which employs the NSGA-II algorithm to simultaneously minimize error rates and computational complexity (FLOPs). Recent surveys on evolutionary NAS and medical NAS further support the relevance of evolutionary search in resource-constrained medical-imaging scenarios [18], [19].

To tackle the prohibitive computational cost of evaluating numerous candidates, NSGANetV2 [6] integrates surrogate regression models to predict performance values without full training. Similarly, LEMONADE [7] streamlines the process by initializing child architectures with transformed weights inherited from their parents, facilitating rapid fine-tuning as an alternative to training from scratch.

Expanding beyond the traditional speed-accuracy trade-off, contemporary NAS methods are being adapted to address diverse task-specific objectives. For example, MORAS [8] proposes an EA-based framework designed to evolve models that are robust to adversarial attacks. Furthermore, NAS strategies can be tailored to optimize ethical or domain-specific constraints, such as in [9], where the search was directed toward finding CNN models that neutralize performance bias between demographic groups and ethnic subgroups in face recognition applications.

C. NSGA-II algorithm

The NSGA-II algorithm [10] is a population-based, elitist multi-objective evolutionary algorithm widely used to solve optimization problems. It draws inspiration from the biological process of natural selection and the principle of survival of the fittest. The algorithm proceeds iteratively, where each iteration corresponds to a generation.

At initialization, a population of candidate solutions, called chromosomes, is randomly generated. At each subsequent generation, an offspring population of equal size is produced using genetic operators. Crossover recombines pairs of parent solutions to generate new individuals, while mutation introduces random perturbations to enhance exploration of the search space.

Algorithm 1: NSGA-II Based Method

Input : Population size, maximal generation number
Output: Pareto front CNN architectures, objective values, and HI values per generation

$P_0 \leftarrow$ Initialize population randomly based on search space and encoding strategy;
$t \leftarrow 0$;
while $\left(t < \textit{maximal generation number}\right)$ **do**
  Evaluate fitness of P_t using sensitivity and specificity metrics;
  $Q_t \leftarrow$ Generate offspring via tournament selection and crossover;
  $Q_t \leftarrow$ Apply the mutation operation to the newly generated offspring;
  $P_{t+1} \leftarrow$ Select from $P_t \cup Q_t$ using non-dominated sorting and crowding distance;
  Record the Pareto front solutions, objective values, and the hypervolume indicator;
  $t \leftarrow t + 1$;
return *The recorded lists of Pareto front solutions and performance indicators*;

The parent and offspring populations are subsequently merged, and a fast non-dominated sorting procedure is applied to classify solutions into a hierarchy of Pareto

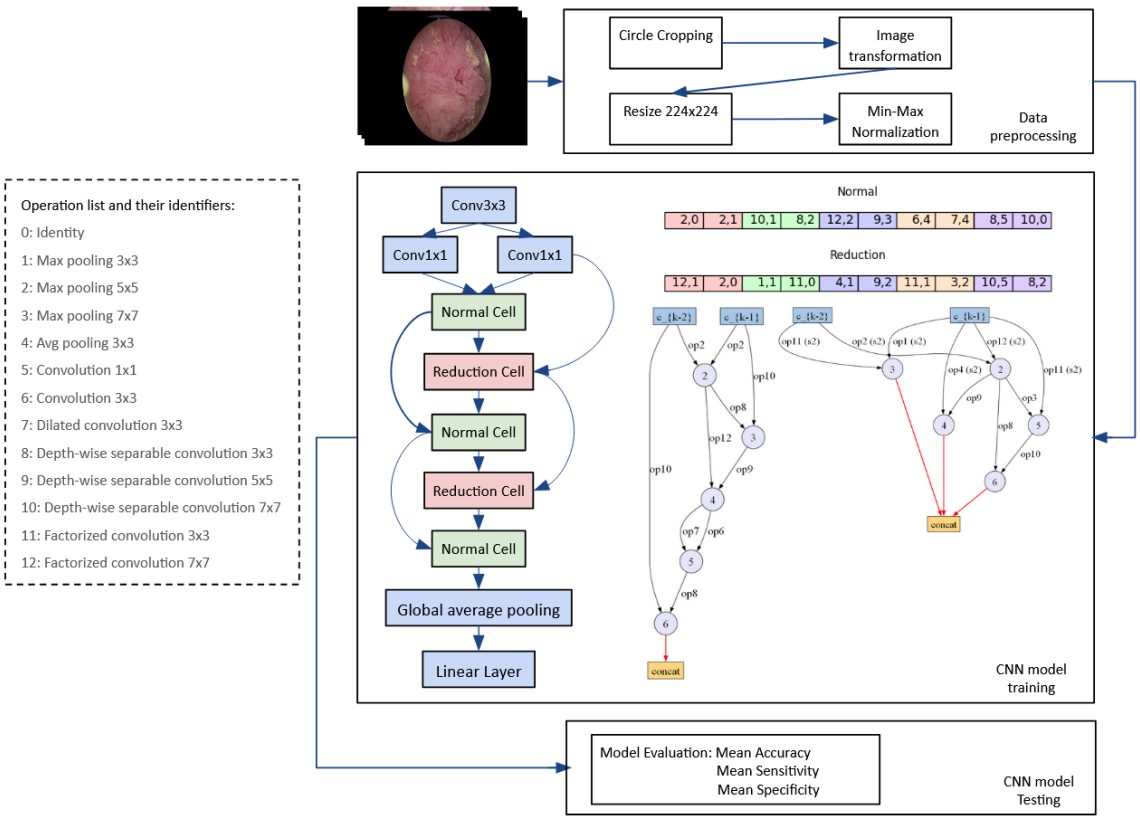

Fig. 1. The overall training and evaluation framework of the CNN architecture during the NSGA-II optimization process (Algo 1). This framework encompasses data preprocessing step, CNN model training with its corresponding architecture including a specific chromosome encoding mapped to its corresponding DAGs for normal and reduction cells, and the final model testing phase. On the left, a predefined operation set is tabulated with unique identifiers, serving as the search space for the architecture's structural components.

fronts based on dominance relations. To construct the next generation, individuals are selected according to their front rank and crowding distance. The latter serves as a diversity preservation mechanism by favoring solutions located in less crowded regions of the Pareto front (see Algo 1).

III. Methodology and Approach

Figure 1 illustrates the overall pipeline of the proposed methodology. First, raw cystoscopy images undergo a two-step preprocessing stage: circle cropping removes irrelevant black borders, while resizing to 224×224 pixels and Min-Max normalization prepare the images for training. Each preprocessed image is then fed into the CNN training block, which follows a NASnet-inspired macro-architecture: an initial Conv3$\times$3 layer and two parallel Conv1$\times$1 branches, followed by two stacked pairs of Normal and Reduction cells and a final Normal cell. A Global Average Pooling layer and a Linear Layer complete the classification head. On the right, a chromosome encoding example is shown: each row of colored identifiers encodes the operations of a Normal (top) or Reduction (bottom) cell. The corresponding Directed Acyclic Graphs (DAGs) visualize the internal topology of each cell, where numbered nodes represent intermediate feature maps and labeled edges denote the applied operations drawn from the 13-operation search space listed on the left. Finally, the trained model is evaluated using three metrics: Mean Accuracy, Mean Sensitivity, and Mean Specificity.

A. Dataset Collection and Preprocessing

Between February 2022 and September 2025, 171 patients (265 videos) undergoing a TUR of an initial or recurrent suspected bladder tumor, whether unifocal or multifocal, were prospectively recorded. The trial was conducted in accordance with the International Conference on Harmonization guidelines for Good Clinical Practice and the Declaration of Helsinki (PI2022_843_0014). From this cohort, a total of 173 White Light Cystoscopy (WLC) videos were selected for this study. Histopathological analysis confirmed cancerous tumors in 121 videos, while the remaining 52 showed no signs of malignancy.

From each video, 400 RGB frames were extracted and cropped to remove non-informative black margins. These frames were manually curated to exclude low-quality images or those containing artifacts that could compromise model performance. For the final dataset, 20 frames were randomly selected from each video, forming a total of 3,460 images. To enhance model generalization, these frames were then augmented using several data transformation techniques: random flipping, rotation, width/height shifts, and zoom. Subsequently, the images were resized to 224×224 pixels

and Min-Max normalized before being fed to the model for training.

B. The Proposed Method

Diagnostic practice involves a trade-off between correctly identifying disease and minimizing false positives; this is referred to as the sensitivity–specificity trade-off. In bladder cancer diagnosis, high sensitivity is vital for ensuring timely intervention, while high specificity minimizes the risk of unnecessary invasive follow-up procedures. For this reason, this study utilizes a multi-objective NAS approach to optimize both metrics simultaneously.

1) Model Architecture: Inspired by NASNet, our architecture is constructed by stacking two cell types: Normal Cells, which maintain spatial and channel dimensions, and Reduction Cells, which halve spatial dimensions while doubling the feature channels.

The classification module is structured as follows:

- *Input Stage:* An initial 3×3 convolutional layer (24 channels) followed by two parallel 1×1 convolutional layers generating identical feature maps.
- *Core Backbone:* Two stacked pairs of Normal and Reduction cells, followed by a final Normal cell. Each cell receives input from the two preceding cells.
- *Output Stage:* An adaptive average pooling layer for compression, followed by a fully connected layer with a sigmoid activation function for cancer probability prediction.

2) Cell Structure and Encoding: Each cell is modeled as a Directed Acyclic Graph (DAG) containing seven nodes. The first two nodes act as input nodes, receiving feature maps from the outputs of the two preceding cells. Each of the five remaining intermediate nodes is connected to two previous nodes via directed edges.

Each edge corresponds to an operation selected from a search space of 13 possible convolution and pooling operations defined in the NASNet paper. Convolution edges are composed of a succession of ReLU, convolution and batch normalization sub-operations. At each node, the outputs of its two incoming operations are combined using element-wise summation. The final output of a cell is formed by concatenating the feature maps of all nodes that were not used as inputs to any subsequent operations.

In order to encode the two cell types and the underlying architectural operations for the NSGA-II process, each chromosome is encoded as a vector of 40 genes. The first 20 genes define the normal cell, while the remaining 20 describe the reduction cell. Since each cell consists of 5 intermediate nodes and every node receives 2 incoming edges, each node is represented by 4 genes (2 pairs of [operation, source node]).

3) Methodology Settings: For NSGA-II, we set the total number of generations at 20 with an initial population size of 20 randomly generated models. To ensure high performance while avoiding overfitting, we employ a dynamic grouped by video 5-fold cross-validation setting: only the first 2 folds are used for the first 10 generations, with the third fold added for the remaining generations. We used a 2-individual tournament selection algorithm, one-point crossover, and a polynomial mutation algorithm.

During the NSGA-II optimization process, each model is trained for five epochs using the AdamW[11] optimizer ($\beta_1 = 0.9, \beta_2 = 0.99$, weight decay 1×10^{-4}) and a CyclicLR[12] scheduler (learning rate: 1×10^{-5} to 1×10^{-4}).

For the comparative study, pretrained models (VGG16[13], ResNet50[14], and DenseNet121[15]) were fine-tuned for 100 epochs. A lower learning rate range (1×10^{-6} to 1×10^{-5}) was used, as these models undergo fine-tuning rather than training from scratch.

4) Assessment Metrics: The optimization objectives are mean sensitivity (Eq. 1) and mean specificity (Eq. 2). To evaluate the quality of the Pareto front, the HI was incorporated, representing the area covered by non-dominated solutions relative to a reference point at (0,0). This choice follows standard set-based multi-objective evaluation practice, where hypervolume jointly reflects convergence and diversity [20].

$$\text{Sensitivity} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{1}$$

$$\text{Specificity} = \frac{\text{True Negatives}}{\text{True Negatives} + \text{False Positives}} \tag{2}$$

Convergence behavior is assessed using genotypic diversity (H^*), defined as the average gene-wise Shannon entropy across the population at each generation:

$$H^*(g) = \frac{1}{L} \sum_{j=1}^{L} \frac{- \sum_{a \in \mathcal{A}_j} p_j(a \mid g) \ln p_j(a \mid g)}{\ln(k_j)} \tag{3}$$

Where L is the chromosome length and k_j represents the number of possible values for gene j (13 for operations, or index-dependent for source nodes).

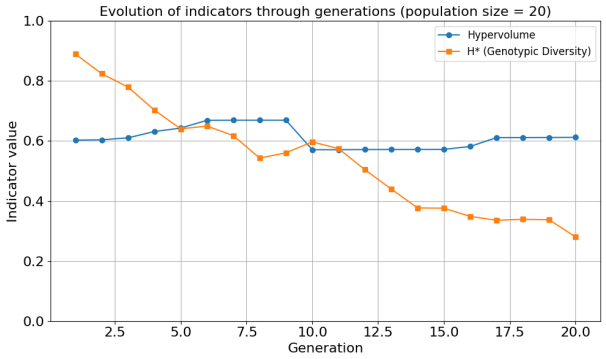

Fig. 2. The evolution of the HI and genotypic diversity across generations.

C. Resources

We used PyTorch to implement and train the models. NSGA-II optimization was accelerated on a Tesla V100-32G GPU, while the comparative evaluation of pretrained models was conducted on an Intel® Xeon® E5-2680 v4 CPU, both provided by the Matrics platform (`https://www.matrics.u-picardie.fr/`).

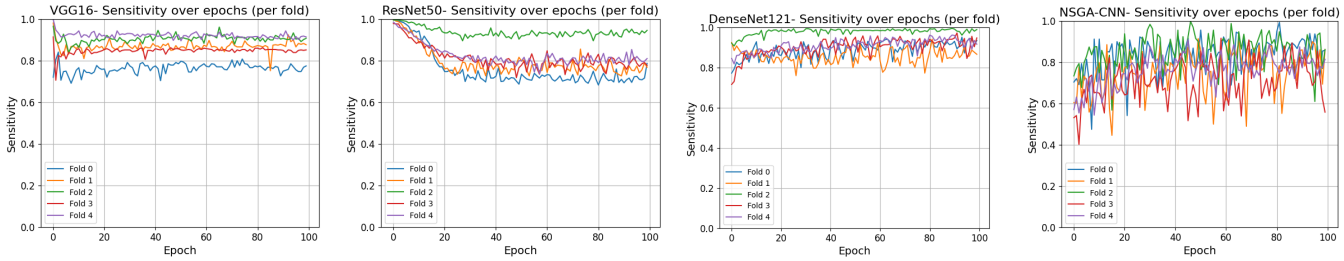

Fig. 3. The 5-fold cross-validation evolution of testing *"sensitivit"* for VGG16, ResNet50, DenseNet121, and the selected NSGA-II-CNN architecture.

IV. Experimental Results

A. Evolutionary Performance of NSGA-II

The optimization process displays two distinct phases. During the first 10 generations (using 2 folds), we observe a steady improvement in the Pareto front quality, with the HI increasing from 0.60 to 0.67. However, the introduction of the third fold at generation 11 induces a significant performance shock, dropping the HI value to 0.57. This suggests that the architectures optimized on the initial folds lacked generalization across the broader dataset variance. Further iterations achieved a partial recovery, slowly increasing the indicator value to 0.61 by generation 20.

Analysis of the genotypic diversity indicator (H^*) reveals a rapid loss of diversity in the early stages, as H^* decreases from 0.87 to 0.64. This is followed by a temporary stabilization, after which diversity declines again with the introduction of the third fold, converging to approximately 0.28 by the end of the optimization process (Fig. 2). This indicates that the population converged toward a narrow region of the search space, potentially missing more robust global optima.

B. Comparative Analysis with Pretrained Networks

The cross-validation dynamics (shown in Fig. 3 and Fig. 4) highlight fundamental differences in how architectures handle the sensitivity–specificity trade-off in bladder cancer imaging:

- *ResNet50:* Shows the most robust behavior. Both metrics stabilize around the 25th epoch with minimal inter-fold variability. Sensitivity ranges between 0.7 and 0.9, while specificity remains consistent between 0.5 and 0.7.
- *VGG16:* Initially provides high sensitivity coupled with low specificity; however, it rapidly achieves a more balanced performance by the second epoch. At this stage, sensitivity ranges from 0.7 to 0.9, while specificity improves to between 0.4 and 0.7.
- *DenseNet121:* Displays a pronounced bias toward the majority class, indicative of overfitting. This behavior is reflected in consistently high sensitivity (Fig. 3) and near-zero specificity in several folds (Fig. 4), suggesting that the model fails to learn discriminative features and instead overfits to cancer prevalence.
- *NSGA-II-CNN:* Demonstrates a competitive balance initially (0.6–0.8 for both metrics). However, unlike ResNet50, it shows signs of late-stage overfitting, as evidenced by the degradation in performance stability across later epochs (specifically in the specificity plots in Fig. 4 after epoch 60, where fold curves begin to diverge significantly).

V. Discussion

This study explored the application of NSGA-II for NAS in the context of bladder cancer diagnosis. While the evolutionary process identified functional architectures, a deep analysis of the results reveals several critical insights into the behavior of both evolved and standard models.

The evolution of the HI and genotypic diversity (Fig. 2) provides clear evidence of the search dynamics. The initial 10-generation phase shows a successful, albeit slow, optimization on a limited data subset. However, the sharp drop in HI at generation 11 upon introducing the third fold highlights a lack of *generalization*. The simultaneous collapse of genotypic diversity (H^*) to 0.28 confirms a premature convergence: the population lost its exploratory power too early, becoming trapped in a local optimum that was only "expert" on the first two folds. This is a direct consequence of the limited population size (20 individuals), which is statistically insufficient to navigate a search space estimated at 1×10^{34} configurations.

Comparing the cross-validation curves in Fig. 3 and Fig. 4 allows for a more granular understanding of model robustness. *"ResNet50"* stands out as the gold standard due to its low inter-fold dispersion, meaning the model learns features that are consistent across different patients. In contrast, *"DenseNet121"* shows a *"severe performance collapse"* in specificity. As seen in Fig. 4, several folds drop to zero specificity, indicative of a pronounced *"systematic bias toward the majority class"*. This suggests that DenseNet's dense connectivity may exacerbate overfitting on small, imbalanced clinical datasets.

The NSGA-II-CNN architecture initially maintains a balanced profile, but its specificity curves (Fig. 4, far right) show increasing instability and divergence after epoch 60. This suggests that while NAS can find "balanced" architectures, they may lack the inherent regularization properties (such as residual connections) necessary to maintain stability during long training cycles.

The challenge of morphological mimicry between cancerous and non-cancerous lesions in cystoscopy further com-

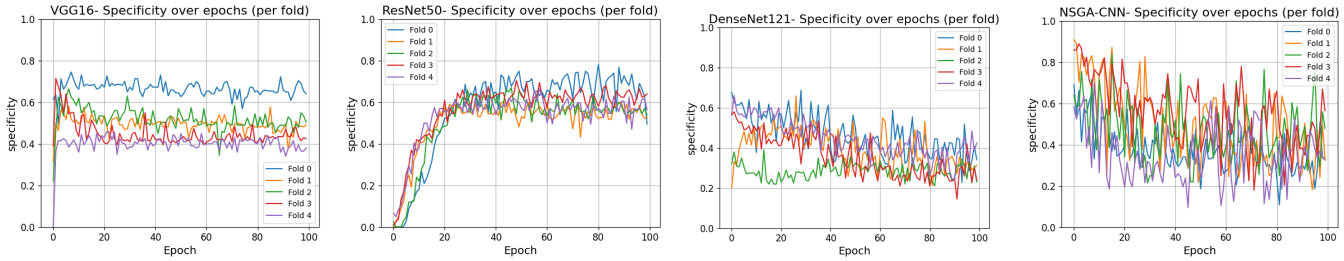

Fig. 4. The 5-fold cross-validation evolution of testing *"specificity"* for VGG16, ResNet50, DenseNet121, and the selected NSGA-II-CNN architecture.

plicates the landscape. The sensitivity fluctuations seen in *"VGG16"* (Fig. 3) suggest that simpler architectures struggle to find stable discriminative boundaries. To overcome the observed premature convergence, future work must prioritize exploration by increasing the mutation probability beyond the current 5% or adopting a *"uniform crossover"*. Furthermore, using surrogate models or weight-sharing mechanisms could allow for more evaluation epochs without increasing the computational burden, leading to architectures that are inherently more robust to the "shock" of new clinical data.

VI. CONCLUSIONS AND FUTURE WORK

The proposed preliminary study investigated an evolutionary-based methodology for designing task-specific CNN architectures for the binary classification of bladder cancer in white-light cystoscopy. By defining a cell-based search space and treating sensitivity and specificity as conflicting optimization objectives, the discovery of balanced diagnostic models was automated. During this investigation, the implementation of the NSGA-II algorithm faced significant challenges. As evidenced by the stagnation of the HI and the collapse of genotypic diversity to 0.28 (Fig. 2), the search process was limited by its exploratory capacity. Consequently, the evolved architectures underperformed relative to established pretrained models like ResNet50. However, this study successfully established a baseline and identified a "generalization shock" when expanding the training data, providing a valuable diagnostic of the current NAS limits in clinical settings.

To extend the impact of this study, several future research perspectives will be explored. Premature convergence observed in the early generations will be addressed by incorporating dynamic mutation rates and uniform crossover operators to sustain higher genotypic entropy. Furthermore, to overcome the 5-epoch training bottleneck, future iterations will integrate surrogate models or weight-sharing mechanisms, allowing for high-fidelity performance estimation without prohibitive computational costs. A promising direction also involves a "warm-start" strategy, where the evolutionary process begins with building blocks from pretrained networks to combine ImageNet-derived robustness with task-specific optimization. Finally, the evaluation process will be refined to prioritize architectural stability across diverse clinical data, ensuring robustness against significant inter-patient variability.

Finally, the identified failure modes offer a definitive roadmap for more sophisticated Neural Architecture Search strategies. Calibrating the balance between exploration and exploitation will be paramount to developing autonomous tools capable of matching or exceeding the reliability of expert-designed backbones in urological oncology.

REFERENCES

[1] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning Transferable Architectures for Scalable Image Recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2018.

[2] H. Liu, K. Simonyan, and Y. Yang, "DARTS: Differentiable Architecture Search," in *Int. Conf. Learn. Represent. (ICLR)*, 2019.

[3] E. Real, A. Aggarwal, Y. Huang, and Q. V. Le, "Regularized Evolution for Image Classifier Architecture Search," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, 2019.

[4] C.-H. Hsu, S.-H. Chang, D.-C. Juan, Y.-C. Pan, and W. Chen, "MONAS: Multi-Objective Neural Architecture Search using Reinforcement Learning," *arXiv preprint arXiv:1806.10332*, 2018.

[5] Z. Lu, I. Whalen, Y. Dhebar, K. Deb, E. Goodman, W. Banzhaf, and V. N. Boddeti, "NSGA-Net: Neural Architecture Search using Multi-Objective Genetic Algorithm," in *Proc. Genet. Evol. Comput. Conf. (GECCO)*, 2019.

[6] Z. Lu *et al.*, "NSGANetV2: Evolutionary Multi-Objective Surrogate-Assisted Neural Architecture Search," *arXiv preprint arXiv:2007.10396*, 2020.

[7] T. Elsken, J. H. Metzen, and F. Hutter, "Efficient Multi-objective Neural Architecture Search via Lamarckian Evolution," in *Int. Conf. Learn. Represent. (ICLR Workshop)*, 2019.

[8] X. Dong and Y. Yang, "Multi-objective Search of Robust Neural Architectures against Multiple Types of Adversarial Attacks," in *Eur. Conf. Comput. Vis. (ECCV)*, 2020.

[9] P. Terhorst, J. Kolf, N. Damer, and A. Kuijper, "Rethinking Bias Mitigation: Fairer Architectures Make for Fairer Face Recognition," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, 2021.

[10] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, 2002.

[11] I. Loshchilov and F. Hutter, "Decoupled Weight Decay Regularization," *arXiv preprint arXiv:1711.05101*, 2017.

[12] L. N. Smith, "Cyclical Learning Rates for Training Neural Networks," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, pp. 464–472, 2017.

[13] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 770–778, 2016.

[15] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely Connected Convolutional Networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 4700–4708, 2017.

[16] S. Wu et al., "AI system for bladder cancer detection via cystoscopy," *J. Natl. Cancer Inst.*, 114(2):220–227, 2022.

[17] Y. Chang et al., "Real-time intelligent bladder tumor detection during cystoscopy," *Ann. Surg. Oncol.*, 32:3220–3226, 2025.

[18] Y. Liu et al., "A survey on evolutionary neural architecture search," *IEEE TNNLS*, 34(2):550–570, 2023.

[19] H. Benmeziane et al., "Medical neural architecture search: Survey and taxonomy," in *Proc. IJCAI*, pp. 7932–7940, 2024.

[20] E. Zitzler and L. Thiele, "MOEAs," *IEEE TEC*, 3(4):257–271, 1999.