Applied Soft Computing Journal 153 (2024) 111304

Contents lists available at ScienceDirect

Applied Soft Computing

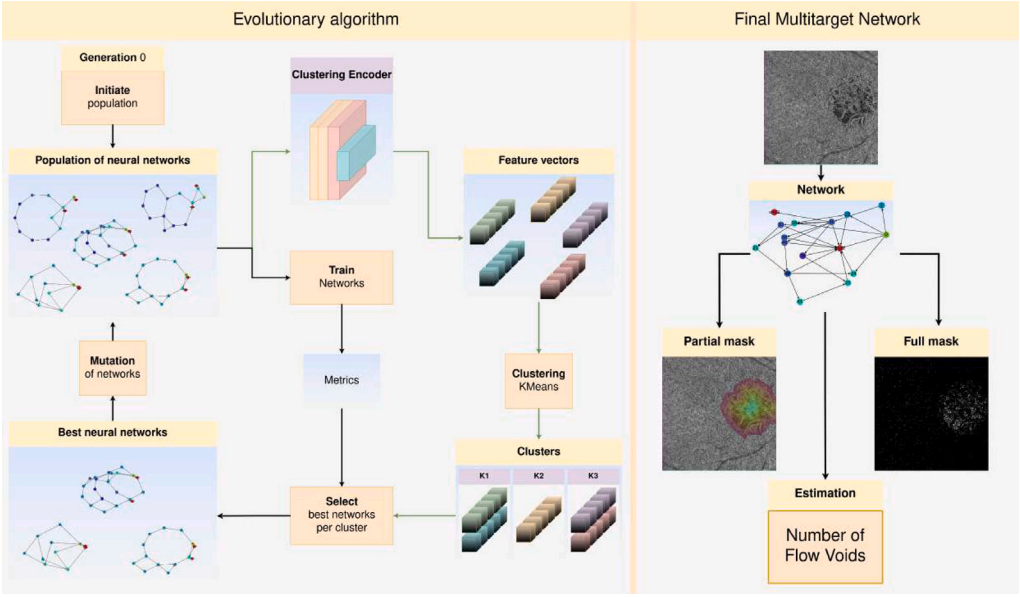
journal homepage: www.elsevier.com/locate/asoc

Evolutionary multi-target neural network architectures for flow void analysis in optical coherence tomography angiography

Emilio López-Varela [a,b], Joaquim de Moura [a,b,*], Jorge Novo [a,b], José Ignacio Fernández-Vigo [c,d], Francisco Javier Moreno-Morillo [d,f], Julián García-Feijóo [d,e], Marcos Ortega [a,b]

[a] VARPA Group, Biomedical Research Institute of A Coruña (INIBIC), University of A Coruña, A Coruña, Spain
[b] CITIC-Research Center of Information and Communication Technologies, University of A Coruña, A Coruña, Spain
[c] Centro Internacional de Oftalmología Avanzada, Madrid, Spain
[d] Departamento de Oftalmología, Hospital Clínico San Carlos, Instituto de Investigación Sanitaria (IdISSC), Madrid, Spain
[e] Ophthalmology Department, Facultad de Medicina, Universidad Complutense de Madrid, Madrid, Spain
[f] Infanta Sofía Hospital, San Sebastian de los Reyes, Madrid, Spain

GRAPHICAL ABSTRACT



ARTICLE INFO

Keywords:
Central serous chorioretinopathy
Evolutionary neural networks
Flow voids
Multi-target

ABSTRACT

Optical coherence tomography angiography (OCTA) is a non-invasive imaging modality used to evaluate the retinal microvasculature. Recent advances in OCTA allows to visualize the blood flow within the choriocapillaris region, where a granular image is obtained showing a pattern of small dark regions, called flow voids (FVs). Given its relevance, numerous clinical studies have linked the changes in FVs distribution to multiple diseases. The granular structure of these images makes accurate labeling and segmentation difficult,

* Corresponding author at: VARPA Group, Biomedical Research Institute of A Coruña (INIBIC), University of A Coruña, A Coruña, Spain.
E-mail addresses: e.lopezv@udc.es (E. López-Varela), joaquim.demoura@udc.es (J. de Moura), jnovo@udc.es (J. Novo), jfvigo@hotmail.com (J.I. Fernández-Vigo), javimorenomorillo@gmail.com (F.J. Moreno-Morillo), julianga@ucm.es (J. García-Feijóo), mortega@udc.es (M. Ortega).

https://doi.org/10.1016/j.asoc.2024.111304
Received 11 July 2023; Received in revised form 20 December 2023; Accepted 18 January 2024
Available online 20 January 2024

OCTA imaging

which can be overcome by using a multi-target perspective. However, manually designing a neural architecture that can accurately predict all targets in a balanced way is a major challenge. In this work, we propose a novel methodology based on evolutionary multi-target optimized networks that, through a set of evolutionary operators, traverses a search space of architectures in a deep but efficient way. This methodology allows us to discover efficient and accurate multi-target architectures tailored to our problem, but which are also adaptable to other tasks due to their robustness. To validate and analyze our methodology and the discovered network model, we performed extensive experimentation with cases from a real clinical study, achieving better results than the state of the art and manually designed architectures.

1. Introduction

In recent years, computer-aided diagnosis (CAD) systems have become very popular as auxiliary tools in the clinical field. These systems increase the robustness of diagnosis of different pathologies and can be applied in different medical imaging modalities widely used in clinical practice routine such as optical coherence tomography (OCT) [1] or optical coherence tomography angiography (OCTA) [2,3]. Specifically, OCTA is a non-invasive imaging modality that uses the time-varying signal intensity of OCT scans to generate images of the ocular vessels. Traditionally, this imaging modality has been used to visualize the blood flow in the upper layers of the retina, but recent advances in imaging devices have made it possible to image blood flow deeper in the choroid (CC), obtaining new information about choroidal physiology. These CC OCTA images extracted at the level of the choroid present a granular appearance with bright regions representing areas with blood flow and dark regions where there is no flow, called flow voids (FVs) [4,5].

The quantification and visualization of changes in the distribution of these FVs has recently become an area of great interest to the scientific community, as numerous clinical studies have linked these changes to the onset and progression of multiple ocular diseases [6–8], such as diabetic retinopathy, glaucoma or age-related macular degeneration. Therefore the particularities of OCTA images are critical in exploring clinically significant features that are paramount in understanding various ophthalmic conditions and exist the potential to use these changes in FVs distribution as a biomarker to monitor the progression of different pathologies. An example of these monitoring capabilities can be seen during photodynamic therapy treatment applied to patients suffering from central serous chorioretinopathy (CSC). The efficacy of this treatment is highly variable between patients, so it is essential to monitor the patient over the time. In clinical practice routines, this involves a long and extremely tedious process, in which clinical experts perform a careful visual inspection where they characterize changes in the distribution of the FVs. This process is prone to errors, so it is of great interest to have a system that allows to quantify changes in the FVs distribution in a repeatable and fully automatic way, in order to increase the accuracy and robustness of the process, while lightening the workload of the clinical staff and, therefore, the costs of health services [9,10]. The problem is that, due to the characteristics that constitute the CC OCTA imaging, it is not easy to create a methodology that allows an accurate analysis of changes in the distribution of the FVs. CC OCTA images present distinctive challenges setting them apart from typical images in general semantic segmentation. These images often manifest intricate local deformations and diffuse edges, deviating from the more structured boundaries prevalent in general datasets. Furthermore, in contrast to other imaging modalities such as computed tomography or magnetic resonance imaging, well-defined anatomical structures cannot be observed in CC OCTA imaging because the vessels present at this depth of the eye are too small to be observable. This causes in the images to acquire a grainy appearance with a complex and very rough texture, which makes it very difficult to perform an accurate segmentation of the FVs and, therefore, prevents an adequate patient monitoring.

2. Related work

At present, due to both the novelty and the drawbacks of working with this imaging modality, there are only a few studies focused on the segmentation and quantification of the FVs in CC OCTA imaging. These works are mostly semi-automated auxiliary methodologies used in clinical studies, or classical approaches used in other domains that do not take into account the specific problems of CC OCTA imaging. Moreover, there are no works on segmentation and quantification of the FVs in CC OCTA images based on deep learning, although it is the state of the art approach in general segmentation. For reference, in Sugano *el al.* [11] a morphometric analysis of choriocapillaris is performed using a semi-automated system called AngioTool [12], which was validated using images of explants, brains and retinas. A clinical study on the behavior of blood flow under the reticular pseudodrusen was presented in Alten et al. [13]. As a method to segment the FVs, the authors used a thresholding that sets the threshold value using the mean pixel value in the outer retinal layer as reference. In Al-Sheikh et al. [14], the authors used the classical Otsu global thresholding method to obtain the measurement of FVs in OCTA CC images obtained from healthy patients. Also, a method of segmentation of FVs based on local contrast adjustment followed by global thresholding was proposed in López-Varela et al. [15]. Finally, Phansalkar's local thresholding method [16] is the standard method commonly used in ophthalmic research studies related to the quantification of FVs [17–19]. This method was first introduced to segment cell nuclei in cytologic and histologic images. Although it works better than a global approach, this local approach may overestimate the amount of FVs in the image by labeling pixels as FVs that are not FVs in local areas with high values.

Furthermore, as previously mentioned, to the best of our knowledge, there are no state-of-the-art proposals based on deep learning to automatically segment and quantify FVs in CC OCTA images. Neural networks have given great results for similar tasks in other medical imaging modalities [20–22], so there is a great potential use for this quantification. The problem with this approach, consists in the difficulty of obtaining a ground truth that can be used to train this type of networks, since the CC OCTA image presents a very granular structure with multiple and diffuse FVs. One way to solve this is to use a multi-target perspective that allows the training of the network using several different targets that are easier to label, which strengthens the generalization capability of the network and allows performing the quantification task in an optimal and robust way. While this multi-target perspective can strengthen the robustness and accuracy of the network, it also destabilizes its training, due to the difficulty of establishing an appropriate balance between the multiple targets. This destabilization can greatly affect the accuracy of the model, losing efficiency in solving the different tasks and thus the positive effects produced by the multi-target perspective. Therefore, to avoid this effect and get the most out of the multi-target perspective, the network architecture used must be able to predict all targets equally. This maximizes the accuracy and robustness of the model and makes it more adaptable to solve any other type of task. Unfortunately, achieving this type of architecture is a major challenge. This is because the manual design of network architectures of this type, with excellent performance, usually requires a large amount of domain knowledge and a lot of development time. To avoid this enormous expenditure of time and resources, what

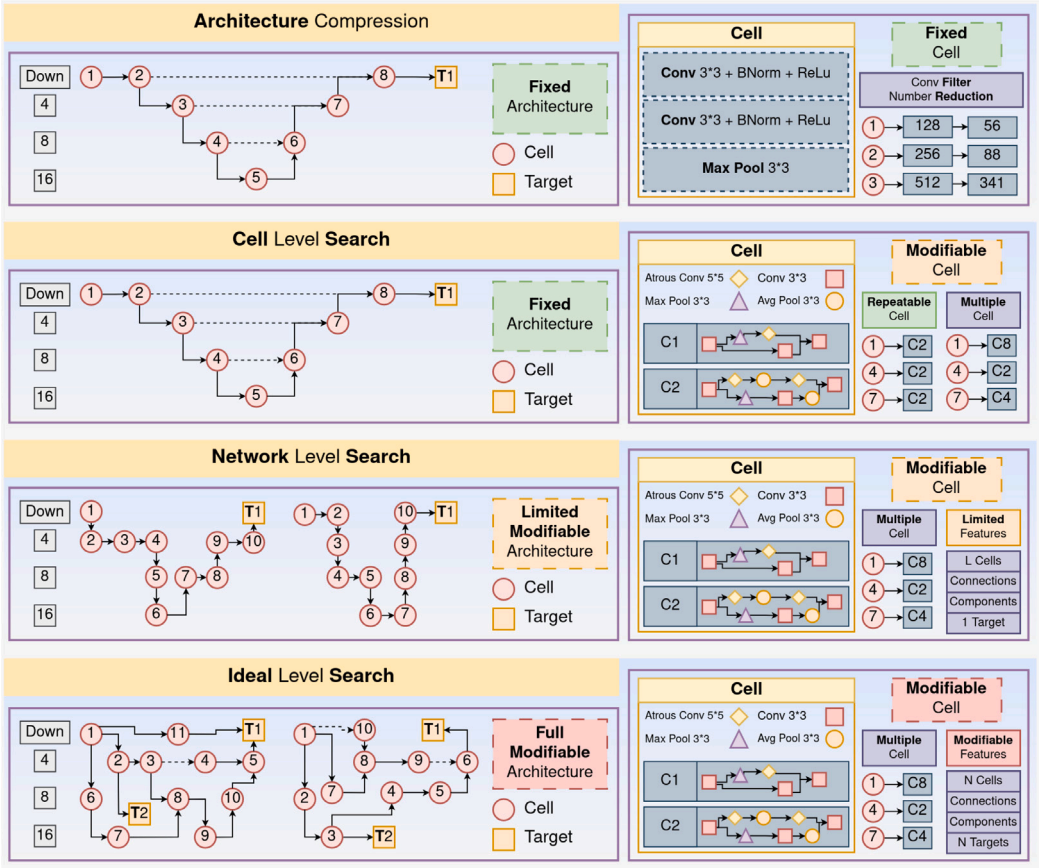


Fig. 1. Graphical representation of the different work categories in the state of the art, along with an example of ideal search.

is known as neural architecture search (NAS) has recently emerged as a subarea of automatic machine learning [23].

NAS represents a groundbreaking approach in enhancing the efficiency and effectiveness of deep learning networks. The process involves automating the manual tuning of neural networks, leading to the discovery of more intricate architectures. Reinforcement learning [24] and evolutionary algorithms [25] have been pivotal in addressing this challenge, showcasing superior performance in diverse applications such as image classification [26], object detection [27], and more recently in semantic segmentation [28]. So this technique presents the potential to improve the effectiveness of deep learning in a wide variety of applications in different imaging modalities. However, despite the success of NAS, several limitations persist, primarily stemming from the delicate balance required between exploring different architectures and exploiting their benefits. In a general sense, based on the depth of the search space, utility, and complexity of the addressed problems, we can categorize the works in the state of the art into three general categories: Architecture Compression, Cell Level or Block Level Search, and Network Level Search. In Fig. 1, a graphical representation of these three categories, along with an ideal search is shown:

- **Architecture Compression**: This category focuses on optimizing a predetermined architecture by reducing parameters [29,30], frequently by minimizing the number of filters in each convolutional layer. This approach utilizes a manually crafted single architecture, resulting in the smallest search space, limited utility, and lower difficulty.
- **Cell Level Search**: Building upon architecture compression, cell-level search involves altering the internal configuration of each block within a predetermined network structure [26,31–33]. While maintaining a constant external architecture, this approach expands the search space by modifying components at the cell or block level. Some works aim to find an optimal block configuration for the entire network [34,35], often reusing the same block, while more complex approaches optimize the internal configuration of each block individually [36]. Some works also employ multi-objective perspectives that enhance the learning of each task [37]. However, a significant limitation arises from the reliance on hand-designed architectures, restricting the outcomes to optimized versions rather than entirely new architectures.
- **Network Level Search**: Similar to the previous category, network-level search optimizes individual blocks while introducing changes to the external configuration of the architecture, overcoming limitations by expanding the search space [38,39]. However, this approach encounters challenges specific to segmentation tasks, where output dimensions must match input dimensions. To address this, methodologies in this category impose constraints such as a fixed number of blocks or a single branch of sequential blocks. Unfortunately, these constraints limit the incorporation of essential segmentation paradigms, such as skip-connections and multiscale features. Additionally, works in this category typically lack multi-target approaches, which are crucial for reinforcing and enhancing predictions.

All these approaches, while improving efficiency, often result in limited exploration, hindering the exploitation of the algorithm. In addition, the evolutionary approaches often suffer from early stagnation at a local minimum during the first few generations, as the population of networks converges to a single type of architecture that delivers the best results for that generation. To tackle this, a methodology maintaining population diversity while selecting optimal individuals becomes crucial for evolving network architectures effectively. Besides, existing NAS approaches tend to optimize architectures for specific

tasks, limiting their reusability across different tasks or imaging modalities. The adoption of a multi-target perspective emerges as a promising solution, enabling networks to encode more general knowledge and enhancing adaptability.

The ideal NAS would transcend current limitations by expanding the search space to create parallel branches, enabling the utilization of multiscale features. Each network block should have the ability to connect with various different blocks, and the number of blocks should be variable for optimal network usage. Crucially, the ideal architectures should embrace a multi-target approach, fostering adaptability and robustness across diverse tasks and imaging modalities [40].

3. Contributions

Considering the potential of FVs as a biomarker, the potential of architectural search and multi-target perspective, and the significant gap in the literature, in this work we propose a fully automatic methodology based on deep learning capable of robustly segmenting and quantifying FVs in CC OCTA images, which allows us to follow the progression of different pathologies accurately. For this purpose, an evolutionary dynamic algorithm was used to search for multi-target architectures to obtain three clinically relevant FV-associated biomarkers: (i) a location mask that indicates where the largest amounts of FVs accumulate, (ii) a segmentation mask that represents the pixel-wise classification, and (iii) a regression showing the total area of FVs per image. To obtain a suitable architecture, capable of carrying out these objectives accurately, we present an evolutionary algorithm, which through a set of evolutionary operators, allows us to find the optimal network architecture. Our evolutionary algorithm allows us to traverse the search space in a deep but efficient manner, allowing the convergence of the population of architectures without sacrificing exploration capacity by keeping the genetic variability of the population high. Diverging from the current state of the art, our approach significantly broadens the search space, facilitating the generation of parallel branches to harness multi-scale features. Every block within the network possesses the capability to connect with diverse blocks, and the architectural flexibility is enhanced by a variable number of blocks, promoting the development of more efficient networks. To maintain the diversity of architectures and avoid collapse, instead of selecting the best individuals from our population, we propose to classify our population into groups and select the best individuals from each group. For this, we extract a representative feature vector from each architecture using a graph encoder network and apply a clustering algorithm to classify these vectors. In contrast to generalist architectures that are designed manually for a single specific task, the architectures found by our evolutionary algorithm are specifically optimized to solve the different objectives as a whole in a precise way. The use of this multi-target perspective results in the selection of architectures with a higher generalization capability that produce more robust and accurate predictions in images with complex characteristics such as those of our problem. Besides, these features make the architectures more adaptable to other tasks as well as other imaging modalities gaining reusability. Last, when only one of the objectives needs to be achieved, these architectures can be subjected to a pruning process, which makes these networks more efficient too. To validate and analyze the proposed approaches, we performed extensive experimentation with cases from a real clinical study of patients affected by chronic CSC, demonstrating great potential for use in real clinical applications. Through an in-depth analysis of these images, our study contributes valuable insights to the continuous research efforts within this domain.

4. Materials

4.1. Dataset

In this work, we use a dataset consisting of a total of 620 CC OCTA (1024 × 1024 pixels) images acquired from the eyes of 10 healthy subjects (20 eyes) and from the pathological and contralateral eyes of 52 patients (104 eyes) affected by chronic CSC. These images were acquired during photodynamic therapy treatment of patients affected by chronic CSC during 5 different time instants corresponding to pre-treatment, 2 to 4 days after treatment, 1 month after treatment, 3 months after treatment, and 6 months after treatment. The acquisition of these images during the various post-treatment reviews of the patients enables precise follow-up of the evolution of the patients as it is done in the real clinical environment. This allows to validate our proposals in a very effective and realistic way, observing their usefulness in routine medical monitoring cases. This dataset was obtained with the Zeiss Plex Elite capture device, using a central wavelength between 1040 nm and 1060 nm with an axial resolution of 6,3 μm, a transverse resolution of 20 μm and a scanning speed of 100,000 A-scans per second. This device uses SS-OCT technology and only images of sufficient quality, as determined by a signal quality >7, were accepted. In addition, a 6×6 mm scan centered in the macula was performed and based on the default settings of the OCTA software device, the CC slab extends 29 to 49 μm below the RPE layer. The projection elimination algorithm of the device software was applied to all the images in order to eliminate false positives caused by deformations of the upper retinal layers. In addition, regarding the criteria for inclusion and exclusion of patients:

- **Inclusion criteria**: Persistent central serous chorioretinopathy with persistent subretinal fluid at or near the fovea (<500 μm) for at least 3 months without any previous treatments in the last 6 months. Patients had to be susceptible to undergo a standardized photodynamic therapy with a spot laser size of 4000 ± 500 μm.
- **Exclusion criteria**: Previous eye surgery, another maculopathy, active treatment with steroids, pregnancy, poor image quality due to motion artifacts or abundant vitreous floaters, and the presence of choroidal neovascularization.

The patients underwent a complete ophthalmologic examination including best corrected visual acuity (BCVA) using the Early Treatment Diabetic Retinopathy Study (ETDRS) scale, slit lamp biomicroscopy and posterior segment ophthalmoscopy. The age, gender and axial length (AL) (IOL master 700; Carl Zeiss Jena, Germany) were also recorded. The age of the patients enrolled in the study ranged from 30 to 60 years. 31 of the 52 patients were men and 21 were women. All explorations were performed within the same time frame (between 9:00 am and 14:00 pm) to avoid diurnal variations of the choroidal structures. Written informed consent was obtained from each participant and the study protocol adhered to the principles of the Declaration of Helsinki and was approved by the Ethics Committee of the Hospital Clínico San Carlos of Madrid.

This set of images was labeled in three different ways in order to train, improve and validate our networks. On the one hand, the methodology defined in [15] was used to extract a preliminary FVs mask. This is used because it is not feasible to mark these zones manually. Once extracted, an expert manually corrected small defects in the mask such as portions of the optic disc detected as FVs. On the other hand, the expert labeled each image creating a partial segmentation mask where the areas with a greater number of FVs are marked in an approximate manner. This mask is not only used for training, but also to provide the expert with a quick visualization of the area to be evaluated. Finally, the total area of FVs was extracted from the mask and a logarithm was applied to normalize its scale to a smaller range of values. In Fig. 2, an example of a CC OCTA image with its two associated segmentation masks can be seen. In addition to these labelings and in order to verify that our methodology is able to extract the FVs in a robust and accurate manner, each patient was labeled in a binary manner (recovered, not recovered) based on the efficacy of the photodynamic therapy applied. This efficacy is based on several factors involving the patient recovery over the next 3 months with the main factor being the reabsorption of persistent subretinal fluid.

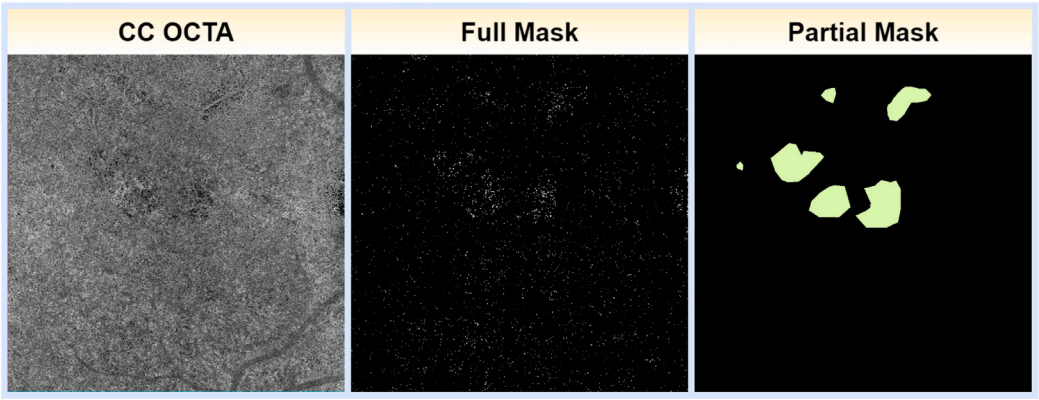


Fig. 2. CC OCTA image example with two labeled segmentation masks.

Table 1
Specifications of the equipment that was used during this work.

Name	Description
OS	Ubuntu 20.04.1 LTS (Focal Fossa)
Kernel	Linux 5.4.0-81-generic
Architecture	x86-64
CPU	Intel(R) Xeon(R) CPU E5-2650 v4 @ 2.20 GHz
GPU	NVIDIA Corporation Tesla M60
Driver version	460.91.03
CUDA version	11.2

4.2. Software and hardware resources

Regarding the software resources, Python 3.9.5 with PyTorch 1.9.1, PyTorch Geometric [41] and CUDA v.11.2 were used for the implementation of the proposed methodology. OpenCV (4.5.3) [42], scikit-learn [43] and Numpy (1.20.3) were used to perform image processing operations. We also adapted a UNet [44] architecture with a DenseNet-169 encoder pre-trained on the ImageNet, adding a branch to perform the regression of the total area of FVs per image. We chose UNet as a baseline for our comparative analysis, recognizing its pivotal role in medical image segmentation. In addition, we used the Auto-Deep [39] and ESEI [37] methodologies for a broader perspective. In the case of ESEI, we replace the classification block with a block to perform the regression. While UNet offers a comparison with a fundamental convolutional model, these additional models provide contrast against other advanced NAS methodologies. Furthermore, we also used the TransUNet model [45] into our analysis. Regarding hardware resources, we include in Table 1 the full disclosure of the components used.

5. Methodology

In this section, we describe all the methodological details of our proposal for the segmentation and quantification of FVs. To accomplish this, we propose to use convolutional network architectures composed of a set of blocks based on the inverted residuals [46], which we create using an evolutionary algorithm. This evolutionary algorithm creates and improves the initial architectures generated through a set of evolutionary operators, while preserving the diversity of the population of architectures using a graph embedding network and a clustering algorithm. In addition to the segmented FVs, these networks give us the areas where the largest numbers of FVs accumulate, allowing us to find at a glance the main affected areas. These networks also give us a direct estimation of the total area of FVs in the image. This estimation allows us by pruning the network to calculate very efficiently these values without having to use the segmentation masks. In Fig. 3, a diagram of our methodology can be seen where the three outputs of a neural network created by our evolutionary algorithm are shown.

5.1. Network architectures

In this work, we use two different types of neural networks: (i) a Population of Evolutionary Networks and (ii) a Graph Network. In Fig. 4, a scheme with the main elements and parts of the two types of networks is shown. The Population of Evolutionary Networks comprise a broad set of architectures initialized randomly. These architectures evolve generation by generation, with the purpose of predicting various objectives, the two segmentation masks and the estimation of the number of flow voids. All architectures in the population are represented as a directed graph that does not form closed loops and consist of three elements: base nodes, special node 1, and special node 2. Each architecture may have a different number of base nodes, and these nodes can be connected in various ways, as long as no closed loops are formed, and the final resolution reaching special node 1 matches the resolution of the input image. These characteristics ensure a wide search space at the network level. All the base nodes of the networks are formed by a common layer block, which is a derivation of the inverted residuals [46], a type of block that allows to extract the features of the images in a precise and efficient way. This block is represented by a vector of seven base features that define and provide adaptability to the block structure. These features include:

- **Number of filters**: The number of filters used in the last convolutional layer of the block and thus the number of feature maps coming out of the block. The range can be from 12 to 1024.
- **Pooling layer**: Defines whether a pooling layer is used at the beginning of the block to reduce the input size by half or not. The options for the pooling layer are MaxPool and AvgPool.
- **Expansion Level**: Defines the level of expansion that is performed to the feature maps within the block at the initial 1×1 convolution. If it is set to one it does not expand, if it is set to two the number of features expands to double and then shrinks to the original size in the final convolution.
- **Squeeze and Excitation block**: Defines whether a Squeeze and Excitation block is used or not.
- **Final nonlinear layer**: Defines which type of nonlinear layer is used before the block output. It can be ReLU, SiLU, Tanh, Mish.
- **Normalization layer**: Defines which normalization layer is used in the block. It can be BatchNorm or InstanceNorm.
- **Dilation level**: Defines the dilation level of the 3×3 convolutional layer of the block. It can be 1, 2 or 4.

Fig. 4 illustrates a representative example of one of the population architectures consisting of 19 base nodes. Each node shows a darker blue color the more downsampled its input features are compared to the original image resolution. The initial node is marked in green with a zero, where the input image first passes through. The output of this node (or block of layers) is connected to nodes 1, 2, 3, and so on, until

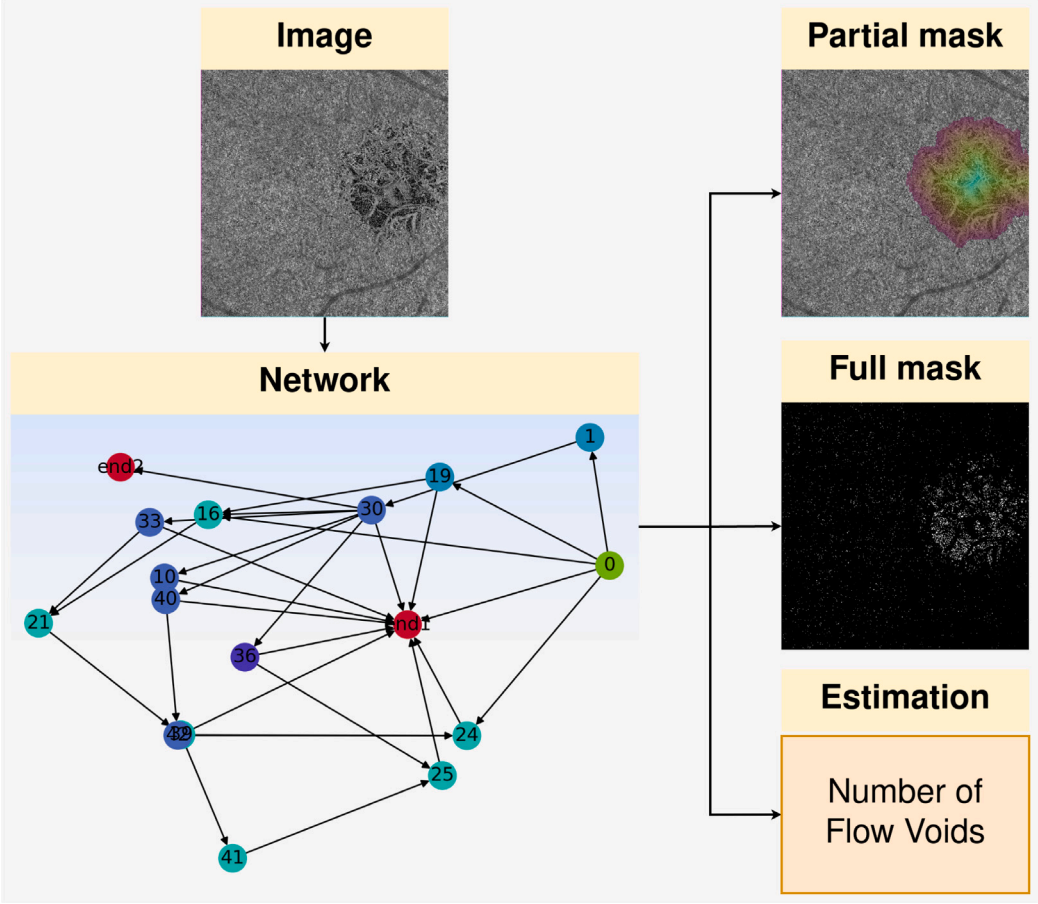


Fig. 3. Diagram of our methodology showing the three outputs of a network architecture created by our evolutionary algorithm.

reaching the output of the network composed of the two final special nodes. Each of the 19 base nodes forming the network is unique and consists of a series of consecutive layers created based on the features of the node. For example, node 3 may have 60 filters, no pooling layer, and use InstanceNorm as the normalization layer, while node 11 may have 245 filters, a MaxPooling layer, and use BatchNorm. This ensures a broad search space at the block level. There are certain nodes, such as node 8, which may have several parent nodes and therefore may have several inputs. In these cases, if the feature maps are of different sizes, we upsample to the size of the largest one and concatenate the feature maps to form a single input. The two special end nodes are connected to a series of layers that produce the outputs of the network. On the one hand, special node 1 (end1) is connected to a 1×1 convolutional layer followed by a sigmoid that gives rise to the two segmentation maps. On the other hand, special node 2 (end2) consists of an adaptive MaxPool layer that reduces the size to 256, a 1×1 convolutional layer that reduces the feature maps to one and a linear layer that outputs the final numerical value of the estimated FVs (total area of FVs).

In addition, we also use a graph network with a complementary function during the evolutionary process. The role of this network is to convert the graph representing each of the networks in our population into a feature vector. These feature vectors represent more compactly each architecture and allow us to perform clustering to group the population based on their similarity. This is crucial in the selection process, as it enables us to select the best individuals within the most diverse groups. If, instead, we only select the best individuals from the total population, we would quickly fall into a local minimum, as the lack of diversity in structures and node types would often lead to too similar structures. Therefore, it is necessary to use this network to promote diversity within the population. This network accepts as input the set of nodes and edges that form a graph and is formed by a set of graphical attention layers that encode the information of each node based on its nearby nodes. First, the network has two continuous blocks of a graph attention layer followed by a ReLU. Next, comes another graphical attention layer but this time followed by an Average Pooling that reduces and conjugates the number of nodes. Finally we have a linear function that encodes the final feature vector.

5.2. Evolution of neural architectures

In this section, we show how our evolutionary dynamic multi-target algorithm works. This algorithm is repeated in a loop for a given number of generations to improve the set of neural architectures (graphs) that form the population and can be divided into three basic operations defined as; initialization (generation of new individuals), selection and mutation. In addition, there is a special operation called pruning that although it is not part of the cycle, it can be applied to our final optimized architectures. In Fig. 5, a comprehensive diagram illustrating the completion of a generation is presented. Each of the basic operations is detailed in the following subsections, but a basic overview of the initialization, selection, and mutation processes can be gleaned from Fig. 6.

5.2.1. Initialization

We start our networks in such a way that they can have multiple branches connected in any way, as long as they have no loops or unconnected nodes. This ensures a wide search space without major restrictions that limit the potential of the architectures that are found. To create each of these networks there are two main steps, the creation

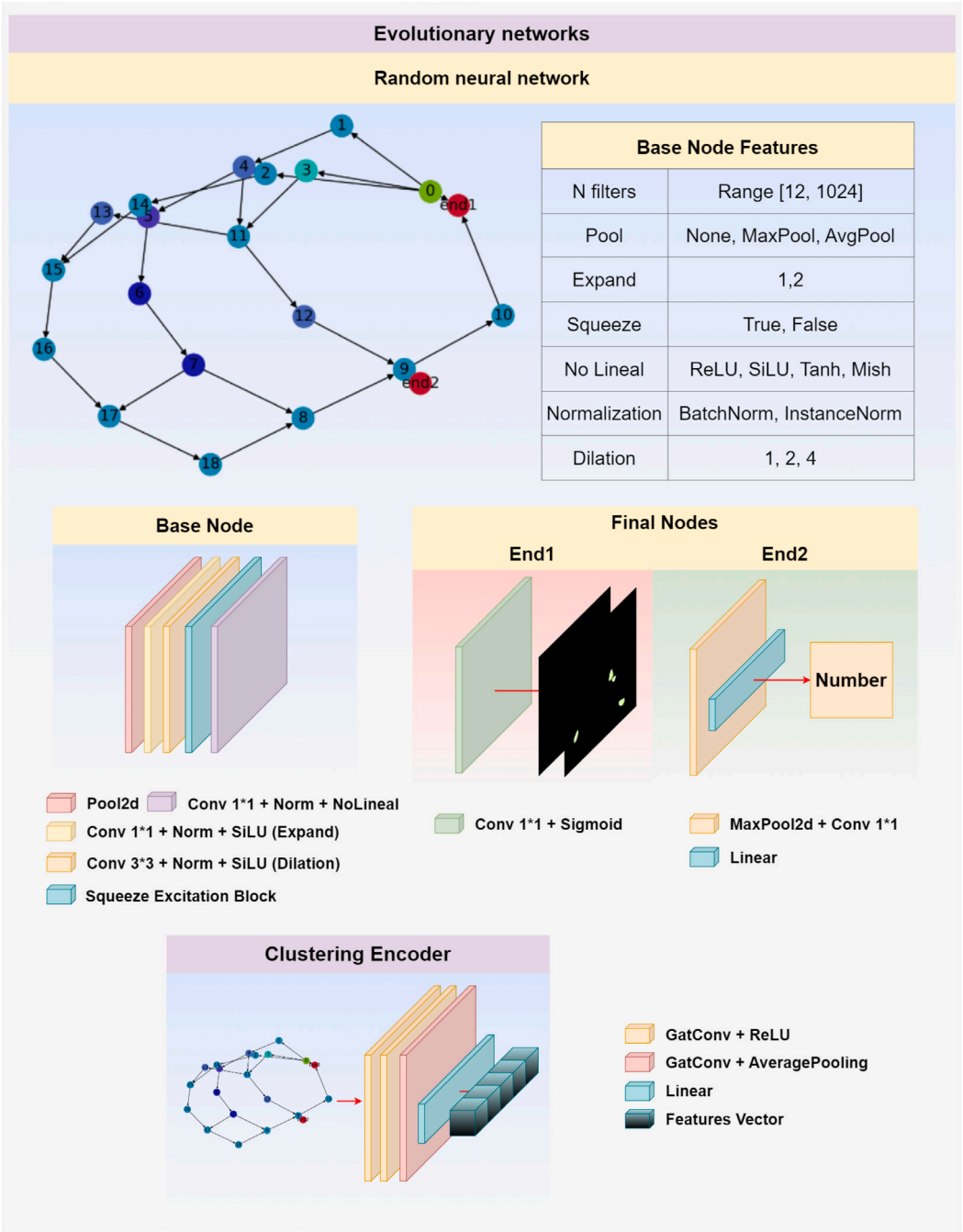


Fig. 4. Diagram showing the main elements and parts of the two types of networks used: evolutionary networks and graph encoder.

of the network, and the specification of the feature vector of each of the base nodes. For the creation of the network, we use a three-step sequence. The first step consists of creating a single branch which we call trunk. For this, we add sequentially and with a given probability nodes to the network until we add the final node 1. Since the final node 1 predicts the segmentation maps, we avoid the nodes that compose the trunk to have a pooling layer. In the second step, we create new branches that can have pooling layers and therefore can perform convolutions and other operations much more efficiently. This allows the branches to be much deeper, to use many more filters and to obtain image features at different scales. To do this, with a given probability, we create the source nodes of these new branches as children of the base nodes in the trunk. Then, with another given probability, we expand these new branches sequentially and we attach them to a non ancestor random node. We repeat this cycle recursively, and each time the depth is reduced, we decrease the probabilities of creating and expanding new branches, to avoid infinite graphs. In the third step, once we have created the entire network with end node 1, we add the rest of the special end nodes by joining them to a random base node. At the end of this process we have a graph representing our network where each base node is a block of layers. To finish the creation of our architecture, we establish for each of the base nodes the seven characteristics that compose the block (number of filters, pooling, etc.). All the characteristics are set randomly except the number of filters. To optimize the initiation of the number of filters, we sample the number of filters from all network nodes of a specific distribution and sort them by node depth (pooling level) and order of occurrence. A network architecture created following this initiation methodology is shown in Fig. 7.

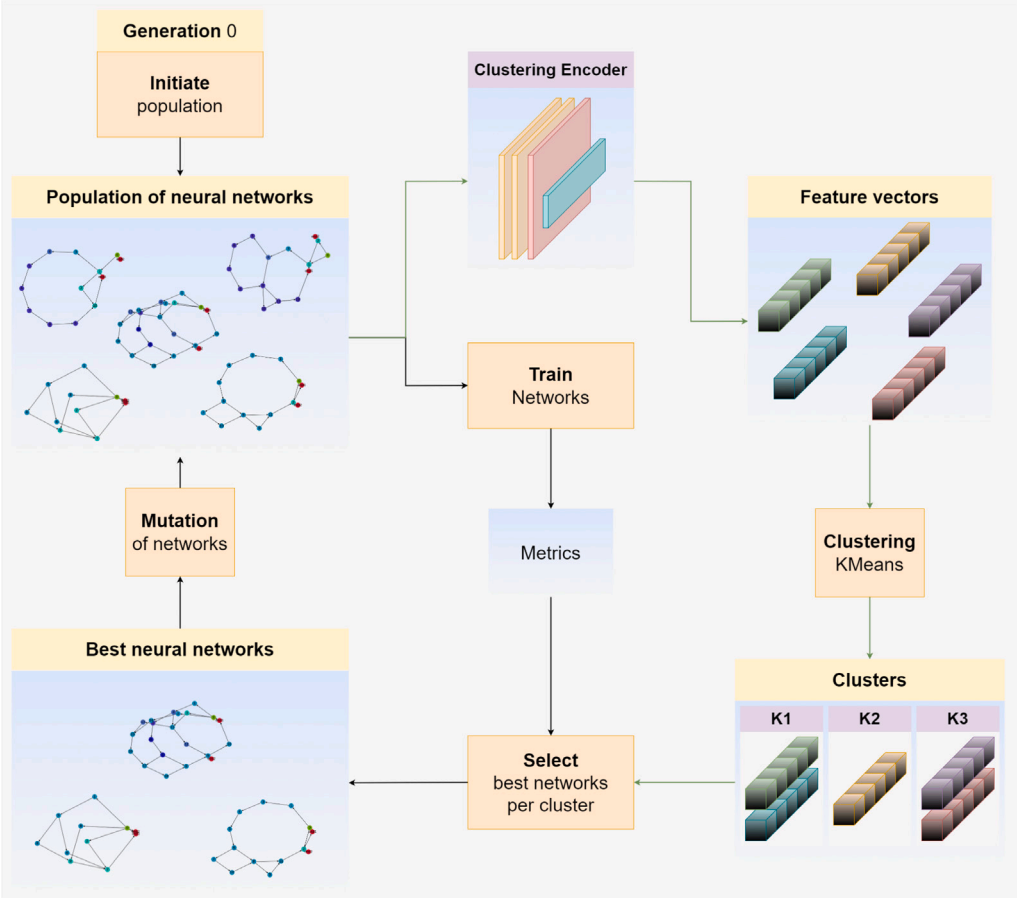


Fig. 5. General diagram of the cycle followed by the evolutionary algorithm over the generations. First, the population of evolutionary architectures is initialized in generation zero. Subsequently, each architecture undergoes training for a set number of epochs, resulting in a final metric. Concurrently, all networks in the population pass through the graph network, and the resulting vectors are grouped using a clustering algorithm. The obtained metrics are then used to select the best architectures from each group, and these are mutated to form the population of the next generation.

5.2.2. Selection process

Once the population is created, the process of training the different architectures begins. All the networks in the population are trained using the same configuration and a final test metric is obtained for each architecture. This final test metric is the fitness function Eq. (1), being ultimately used to select individuals for the next generation. In our case our three objectives present a similar range of values, therefore a sum of the three objectives is used and can be represented as:

$$f = X_1 + X_2 + X_3 \tag{1}$$

where X_1 is the value obtained for the FVs accumulation segmentation metric, X_2 is the value obtained for the pixel-wise segmentation metric, and X_3 is the regression value obtained of the area of the FVs metric. Next, instead of performing the selection of the best performing individuals, we perform a clustering of the population, grouping the different neural architectures based on their similarity. This allows us to select the best individuals from each cluster by evolving the population while maintaining diversity, which allows us to traverse the search space efficiently. To do this, we first pass all the graphs that compose the population through our graph network (clustering encoder) which transforms each graph into a feature vector that represents more compactly the structure, composition and shape of each architecture. Then, we perform a clustering of these vectors with which we subgroup our population into several groups and we select the best performing individuals within each group. This results in the population converging over the generations to a set of different individuals that are very effective at solving the different objectives set, preventing the population from converging to a single local minimum.

5.2.3. Mutation process

In this step, we generate the next generation of architectures by mutating the set of individuals within each group. In this work, we use four basic evolutionary operators that we define as node deletion, node addition, edge change and node features change. This set of operations allows the different architectures of our population to mutate into new variants with their own characteristics, although similar to their close predecessors, gradually evolving the population until the convergence of multiple architectures capable of effectively solving the different objectives. Each of these operations are applied with a given probability to each of the base nodes of the network and are defined as:

- **Node deletion**: The node is deleted and its parents are randomly reconnected to its direct descendants. If there is any unconnected direct descendant, it is randomly connected upwards to a non-descendant node.
- **Node creation**: A new child node is created for the selected node. With a 50% probability this new child node is connected to any other random node that is not its ancestor, otherwise this new child node is connected as an intermediate node to some of the child nodes of the selected node (changing parent–child to parent-intermediate-child).
- **Edge change**: The edges of the node are changed. The node randomly adds new links to other nodes that become its children and removes links from its current children that are reconnected to other random nodes that are not its descendants.
- **Change of node features**: Several node characteristics are randomly changed. In the case of the number of filters, the number is

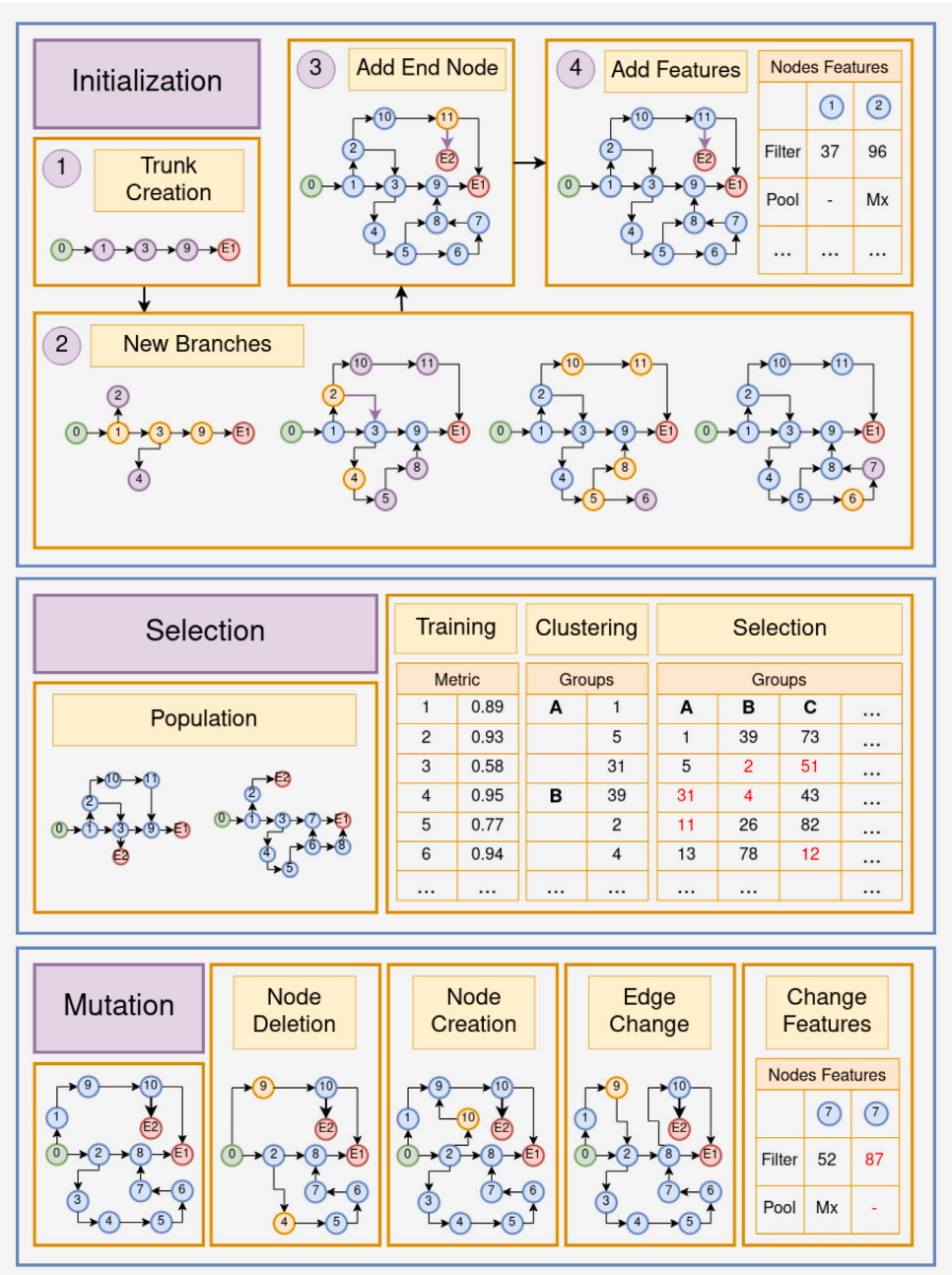


Fig. 6. Basic overview illustrating the functioning of the initialization, selection, and mutation operations of the genetic algorithm.

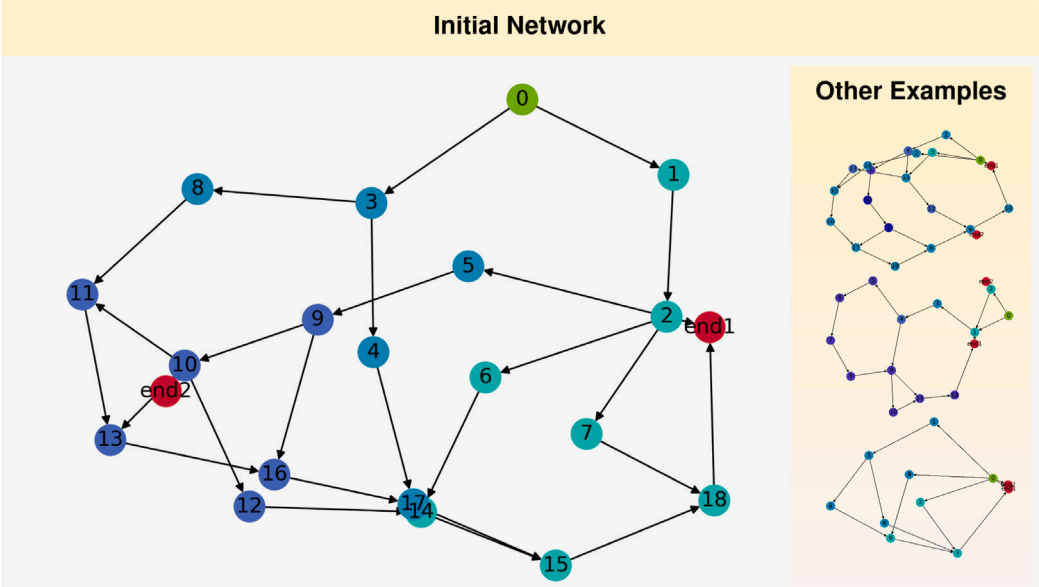
increased or decreased based on a percentage of change sampled from a specific distribution. For example, if there are 100 filters in the node and the percentage is 0.4, the node will have 140 filters.

5.2.4. Pruning

Pruning is a special operation that we can apply to any of our architectures since they are built as a graph. Within the different architectures of our population, several output nodes have multiple base nodes in common that provide robustness and stability when encoding the input image information, but there are also nodes that are only used in one of the outputs. In cases where we only want to obtain one of the network outputs, such as when a doctor only needs a numerical biomarker and not a visualization of the segmentation, our architectures allow us to prune the nodes that are not necessary. This greatly reduces the time and number of calculations required, making the use of our networks much more efficient. In Fig. 8, we can observe a network architecture of our population before and after being pruned to obtain only the total area of FVs.

5.3. Evolution and training details

Before entering the cyclic phase of our evolutionary algorithm, we initialized the population of neural architectures. For the creation of each architecture we used 0.3, 0.4 and 0.6 as probabilities of trunk expansion, creation of new branches and expansion of new branches. These probability values that we achieved in a previous experiment allow us to generate networks with small trunks and very large and diverse branches, thus initializing the population to a set of efficient and effective architectures. The probabilities of expansion and creation of new branches decreased by half each time that we descend one unit of depth to avoid infinite recursion. This also brings the initial population closer to more optimal structures by avoiding inefficient



	N filters	Redu	T-Redu	Expand	Squee	No lineal	Normal	Dila
0	15	-	-	2	-	SiLU	Batch	-
1	17	-	-	-	-	SiLU	Batch	2
2	30	-	-	2	True	Mish	Batch	-
end1	98	-	-	-	-	SiLU	Insta	-
3	102	-1	AvgPool	-	-	SiLU	Batch	-
4	120	-1	-	-	True	Tanh	Batch	2
5	133	-1	AvgPool	-	-	SiLU	Batch	-
6	48	-	-	-	-	Mish	Batch	-
7	48	-	-	-	-	Tanh	Insta	2
8	122	-1	-	-	True	Tanh	Batch	-
9	236	-2	MaxPool	2	-	ReLU	Batch	-
10	271	-2	-	-	True	Mish	Batch	2
11	278	-2	AvgPool	-	-	SiLU	Batch	4
12	293	-2	-	-	-	Tanh	Insta	-
13	300	-2	-	-	True	Mish	Insta	-
14	72	-	-	-	-	Tanh	Insta	4
15	81	-	-	-	True	SiLU	Batch	-
16	325	-2	-	-	-	Mish	Batch	2
17	138	-1	-	-	-	Tanh	Insta	-
18	87	-	-	2	True	Mish	Insta	-
end2	275	-2	-	-	True	SiLU	Batch	2

Fig. 7. Example of a network architecture created following the initialization methodology.

networks that are too small or too large. The number of node filters for each architecture was obtained by sampling a random continuous gamma variable set using a probability density function (2), which can be represented as follows:

$$f(x,a) = \frac{x^{a-1}e^{-x}}{\gamma(a)} \tag{2}$$

where $\gamma(a)$ refers to the gamma function. When a is an integer, γ reduces to the Erlang distribution, and when $a = 1$ to the exponential distribution which promotes the existence of an important percentage of nodes with a relatively low number of filters. We set a to be 1.2 and limit the minimum and maximum of the value to between 12 and 1024 filters. Thus, we obtain a total of 100 distinct architectures that compose our population. We set 10 as the number of clusters and cluster the architectures using the K-Means algorithm and the feature vector extracted by our graph encoder. Our graph encoder was trained previously using three architectures in each batch as input. Basically two random architectures are created using our initialization algorithm and one of them is mutated. After the network encodes the three feature vectors, we implement a process that encourages similarity between the feature vectors of the source and mutated architecture. Simultaneously, we ensure that these vectors are distinctly differentiated from the feature vector of the remaining architecture. For this purpose the TripletMarginLoss (3) was used as loss function in the training of the model.

$$L(a,p,n) = max(d(a_i,p_i) - d(a_i,n_i) + margin, 0) \tag{3}$$

where a, p, n represent an anchor vector, a positive example and a negative example, and d represents the euclidean distance. As an optimizer for this training we used Adam and set an initial learning rate of 0.001 that decreases exponentially at each epoch using a gamma equal to 0.90. On the other hand, our population of 100 networks was trained at each evolution cycle using the same training configuration. Our dataset was divided into training (50%), validation (25%) and test (25%). In this case in each cycle we use the training and validation to obtain the metrics to select the architectures while reserving the test for the final evaluation. Each architecture was trained a total of 60

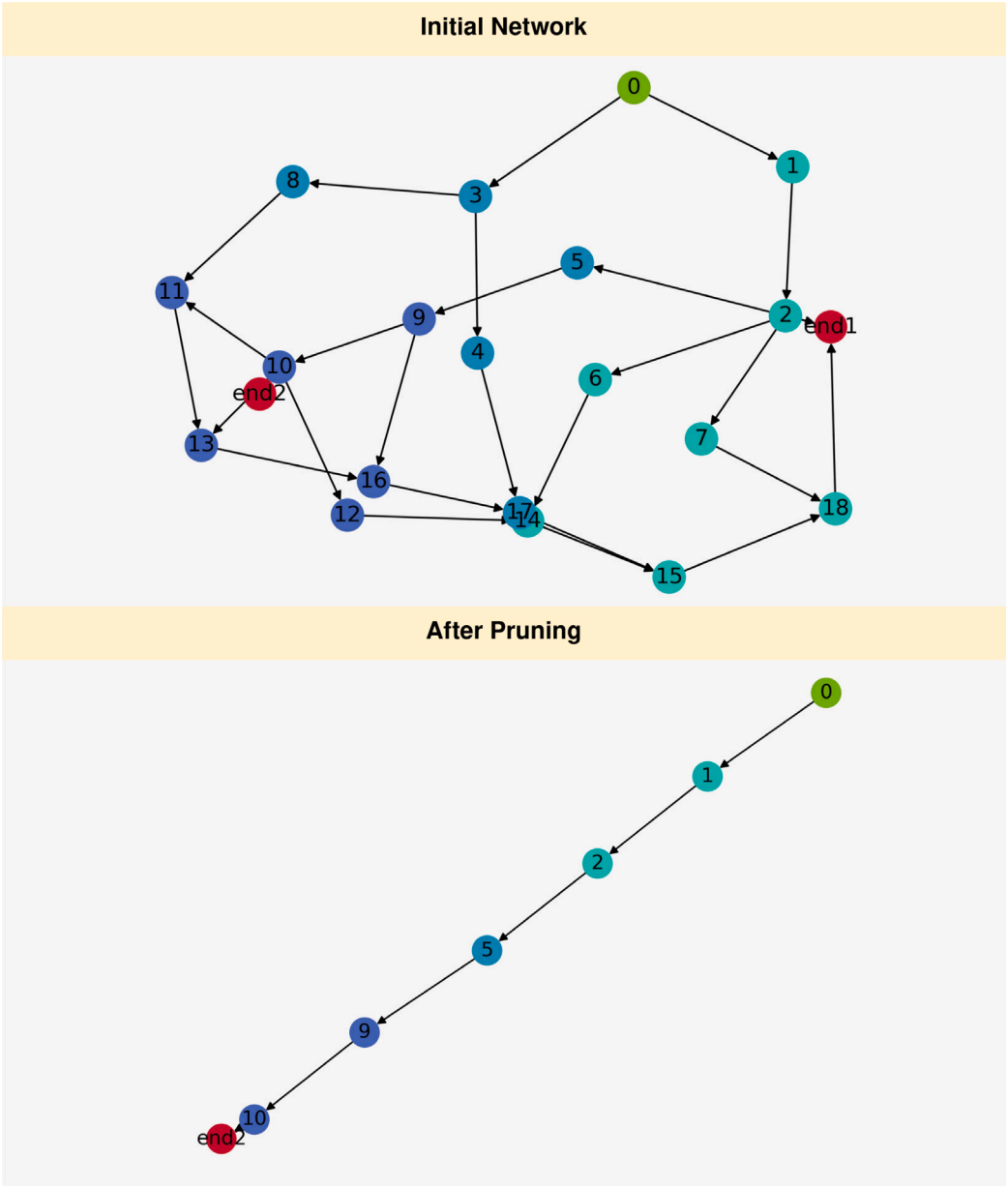


Fig. 8. Network architecture of our population before and after being pruned.

epochs and a batch size of 1. In addition, a data augmentation process was used in which the images were flipped vertically and horizontally and 256 × 256 crops were performed. This set of parameters was used in order to speed up the training of each network to make the costly evolution process more efficient. As loss function, a combination of two losses was used, the Dice loss defined in (4) for the segmentations, where X represents the prediction region and Y the ground truth region, and the MSE loss defined in (5) for the regression, where m and n are the number of rows and columns in the cover image. To combine the losses of each target in a balanced way we used the methodology described in [47] letting the model learn how to balance each specific loss as given in the total loss (L) defined in (6), where w_1 and w_2 are learnable parameters. These parameters update their values using the gradient which reduces the prioritization of the network by a single target during training.

$$DSC = 2\frac{(X \cap Y)}{X + Y} \tag{4}$$

$$MSE = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}(x_{ij} - y_{ij})^2 \tag{5}$$

$$L = (w_1 + (1 - DSC) \cdot e^{-w_1}) + (w_2 + MSE \cdot e^{-w_2}) \tag{6}$$

For each cluster, we select 3 individuals and mutate all of them to generate a new population. For the mutation of each architecture, we randomly sampled in the range (0.1, 0.3) the probabilities of deleting the node, creating a new node, changing the edges or changing the node features. Further with a probability of 0.25 we mutate only the node characteristics without changing the network structure. This last type of mutation serves to precisely refine the characteristics of already efficient architectures. Finally, to evaluate the efficiency of our models we use the MSE defined in (5), the Dice coefficient (DSC) defined in (4) and the Jaccard index (IoU) defined in (7).

$$IoU = \frac{X \cap Y}{X \cup Y} \tag{7}$$

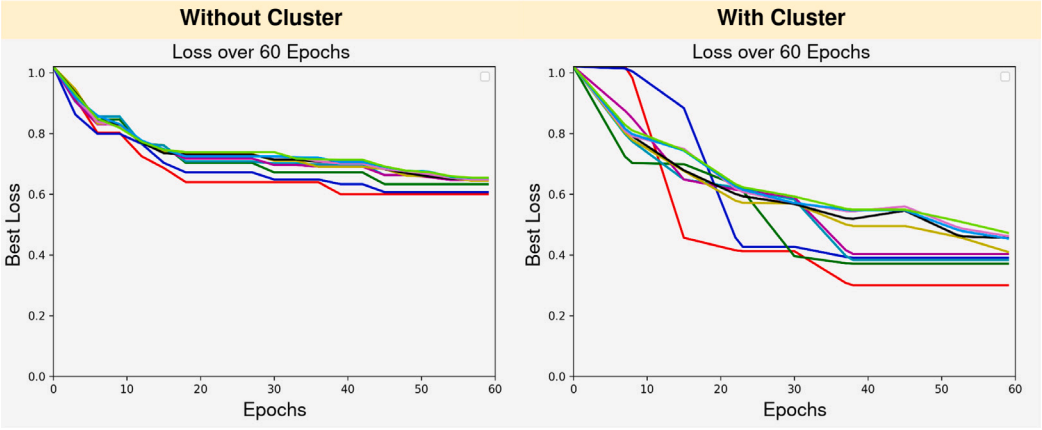


Fig. 9. Evolution of the total loss (L) achieved by the 10 best architectures of the population along the generations of our evolutionary algorithm without using and using the clustering methodology.

6. Results and discussion

6.1. Evaluation of neural architectures

In this section, we present the results obtained from the evaluation of our evolutionary algorithm and the evaluation of the best generated network architecture. To test the effectiveness of the application of this network together with the clustering strategy to maintain the diversity of the population, we have run our evolutionary algorithm with and without the clustering strategy. When this strategy was not used, the best individuals from the entire population were selected, while when it was used, the best individuals within each group (in our case 10 clusters) were selected. In Fig. 9, we can see the evolution of the total loss (L) achieved by the 10 best architectures of the population along the generations of our evolutionary algorithm. By maintaining a diversity of architectures within the population, we enable the survival and evolution of multiple individuals. While these individuals may not currently perform tasks optimally, their unique structural features hold the potential for significant improvements. This diversity ensures a broad exploration of the architectural space, preventing the algorithm from converging prematurely on local optima. Consequently, it enhances our ability to discover ever more effective architectures over successive generations. We can observe how the 10 best individuals of the population have managed to converge satisfactorily. These 10 individuals represent diverse structures, although some convergence to a series of common structures or parameters can be observed, such as the little use of dilation in the convolutional layers, or the expansion of the number of filters only where the input feature maps are of smaller size.

In Fig. 10, we can observe the graph structure, and the characteristics of each node, of the best performing architecture in our population. The Redu column indicates the size of the feature maps in that node, for example with an initial image of size 256×256, a Redu equal to -1 would indicate that in that node the input has a size of 128×128. We can see how this architecture has some similarity with a Feature Pyramid Network (FPN) [48] in the sense that multiple feature maps obtained at different depths are concatenated to establish the final segmentation maps. From the structural point of view, the architecture also presents many skip-connections that arise naturally during the evolutionary process, which makes sense from the point of view of network stability. We also see how this architecture complies with the previous comments, since the feature maps only undergo an expansion in deep nodes, while the dilation is only present in a single node that uses the original size of the image. Regarding the nonlinear functions that are used, we can see that there is a clear bias not to use the Tanh except at special nodes. This makes sense knowing the special behavior of Tanh with respect to other functions such as ReLU or SiLU. Finally we can see how in general using a Squeeze and Excitation block is very beneficial, especially in those nodes that use the original size of the image. This is because this block helps to concentrate the information on the most important feature maps, while diluting the noise present in others.

Table 2
Different metrics showing the performance of the evolutionary network and the performance of the baseline methods for the three different objectives.

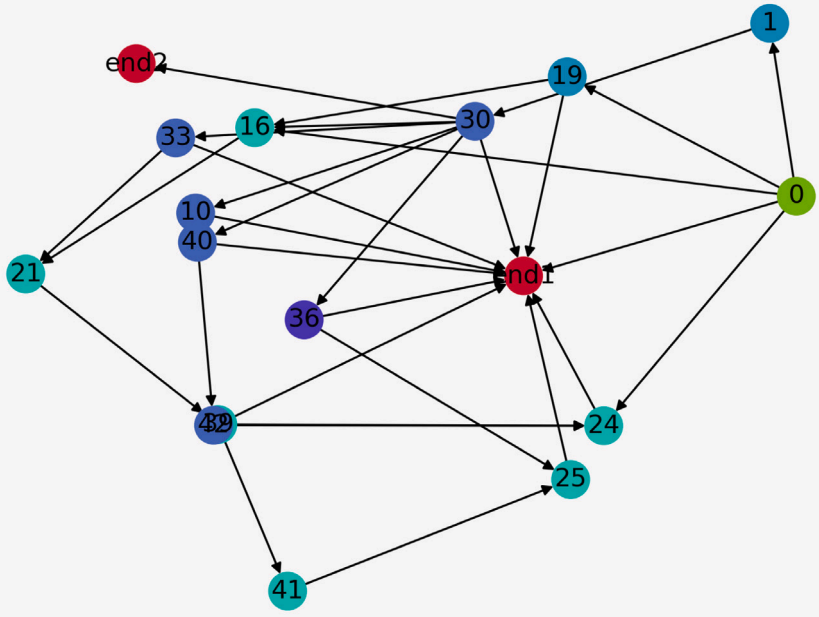
	Full mask		Partial mask		Total area
	DSC ↑	IoU ↑	DSC ↑	IoU ↑	MSE ↓
UNet (Baseline)	0.882	0.875	0.05	0.01	1.3×10^4
TransUNet	0.891	0.879	0.086	0.073	–
Auto-Deep	0.879	0.873	0.583	0.561	–
ESEI	0.895	0.891	0.752	0.748	0.93
Our proposal	**0.903**	**0.896**	**0.805**	**0.798**	**0.02**

To evaluate the learning capacity and the effectiveness in solving the different tasks of the architectures discovered using our evolutionary algorithm, we compare these results with those achieved by a state of the art baseline architecture (UNet). In Fig. 11, we can observe the evolution of the training and validation loss for the baseline and for our best evolutionary network. As we can observe, the baseline network is unable to solve the multiple objectives effectively, which causes one of them to be prioritized while the others remain unsolved. This lack of balance also causes a clear overfitting during the training of the network resulting in worse results in general. In contrast, our evolutionary network is able to maintain this balance by solving the different tasks in a precise and balanced way, which also avoids overfitting.

In Table 2, we present the results for the different evaluation metrics, separated by objectives, obtained by the baseline networks and by our evolutionary network on the test set. We can clearly observe that UNet and TransUNet networks predominantly focus on learning the segmentation of the total FVs mask, disregarding the segmentation of the partial mask entirely. In contrast, networks trained using NAS methodologies exhibit superior adaptability to the established multi-objectives. Specifically, our evolutionary network manages to fulfill the three objectives accurately, improving the results of the baseline networks to a great extent.

In Fig. 12, we can see the difference for the two segmentation masks obtained with respect to the ground truth. The baseline network is unable to generate the partial mask at the same time as it generates the full mask and attempts to predict the total area of FVs. Instead, our network predicts the partial masks in much the same way as the expert does. In addition, we can observe in more detail several zooms of the segmentation masks of the FVs predicted by both networks. Looking at these details, we can see how our evolutionary network is much more

Best Network



	N filters	Redu	T-Redu	Expand	Squee	No lineal	Normal	Dila
0	29	-	-	-	True	ReLU	Batch	-
1	45	-1	AvgPool	-	-	SiLU	Insta	-
end1	42	-	-	-	True	SiLU	Insta	-
10	106	-2	-	2	-	ReLU	Insta	-
16	197	-	-	-	-	Tanh	Batch	-
19	49	-1	MaxPool	2	True	Mish	Batch	-
21	202	-	-	-	True	Mish	Insta	-
24	32	-	-	-	True	SiLU	Insta	-
25	78	-	-	-	-	ReLU	Batch	-
end2	60	-2	-	2	True	Tanh	Batch	-
30	97	-2	MaxPool	2	True	SiLU	Insta	-
33	38	-2	-	-	True	SiLU	Batch	-
36	465	-3	AvgPool	-	-	SiLU	Batch	-
39	297	-	-	-	True	Tanh	Insta	2
40	126	-2	-	-	-	SiLU	Insta	-
41	297	-	-	-	True	SiLU	Insta	-
42	126	-2	-	-	-	Mish	Batch	-

Fig. 10. Diagram of the graph structure, and the features of each node, of the best performing architecture in our population.

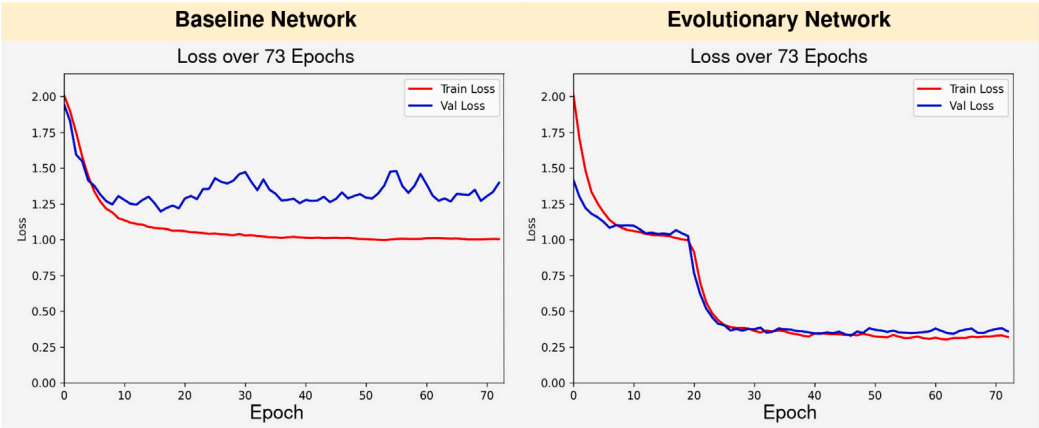


Fig. 11. Evolution of the training and validation loss for the baseline and for our best evolutionary network.

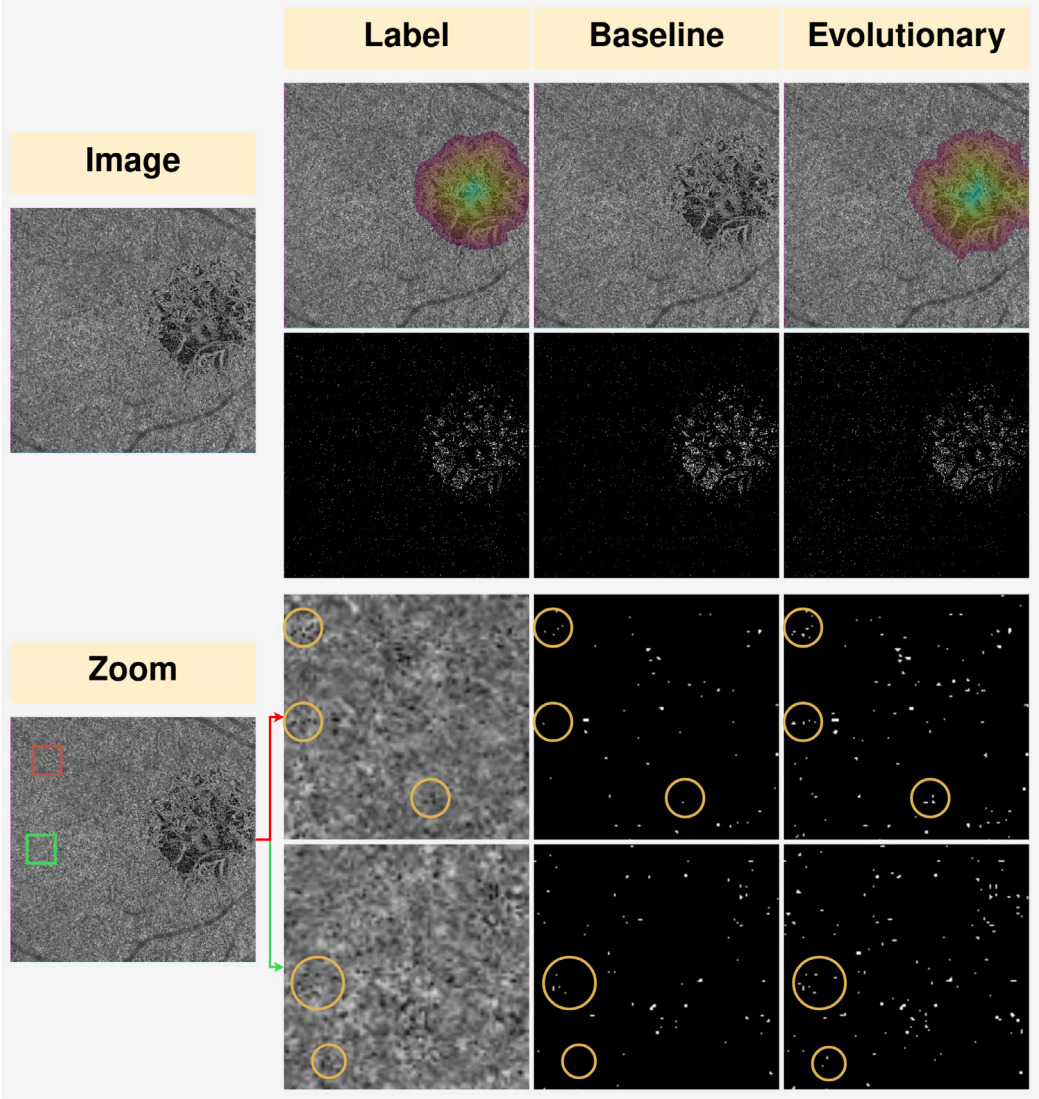


Fig. 12. Original and zoomed example of an image and the segmentation masks predicted by the networks. Green and red squares indicate zoomed areas, while yellow circles indicate significant differences between masks.

robust and accurate in extracting the FVs from the CC OCTA images. This robustness and accuracy is due to the higher generalization capability of the neural architecture and the multi-target training that forces to solve all the objectives in a balanced way. In addition, our network architecture has a much lower number of parameters (2.7M) than the baseline architecture (7.7M), this reduction of parameters increases the generalization capacity and avoids network overfitting helping to further improve network efficiency. This is achieved thanks to our evolutionary algorithm that performs a precise tuning of the required components instead of using a sequence of layers of a general untuned architecture.

6.2. Prediction of the evolution of the FVs changes

To test the usefulness and robustness of our evolutionary network, as well as to compare it with the state-of-the-art methodologies, we performed an experiment using an imaging dataset composed of CC OCTA images that were acquired during photodynamic therapy treatment of patients affected by chronic CSC (described in Section 4.1). The long-term efficacy of photodynamic therapy treatment can be estimated by observing the change in the distribution of FVs before and after treatment. This involves segmenting the FVs in both images, calculating the total area of FVs in each image, and measuring the percentage change from one to the other. It is worth mentioning that our network allows us to obtain this biomarker in two ways, indirectly through the segmentation mask or directly through the regression node. This allows that in cases like this, in which we only want to obtain the biomarker, we can prune our network leaving only the necessary nodes to perform this direct estimation. This makes the calculation process much more efficient by reducing greatly the number of layers and the number of network parameters required from 2.7M to 0.1M parameters. With this, our network is able to perform the estimation much faster, taking on average for 100 images 0.16 s instead of 0.61 s to perform the estimation.

We measured the correlation between the biomarker and the recovered and non-recovered classes using the point biserial correlation, where a better correlation indicates a more robust and accurate extraction of FVs. In Table 3, we can observe the correlation coefficient obtained by each of the approaches. We can see how this coefficient is higher when we use our evolutionary network than when we use the state of the art. This suggests that our evolutionary network is capable to extract the biomarker in a more accurate and robust way, which improves our predictions.

Finally, in Fig. 13 we show a visual comparison of the segmentation masks extracted using the methodologies proposed in the state of the

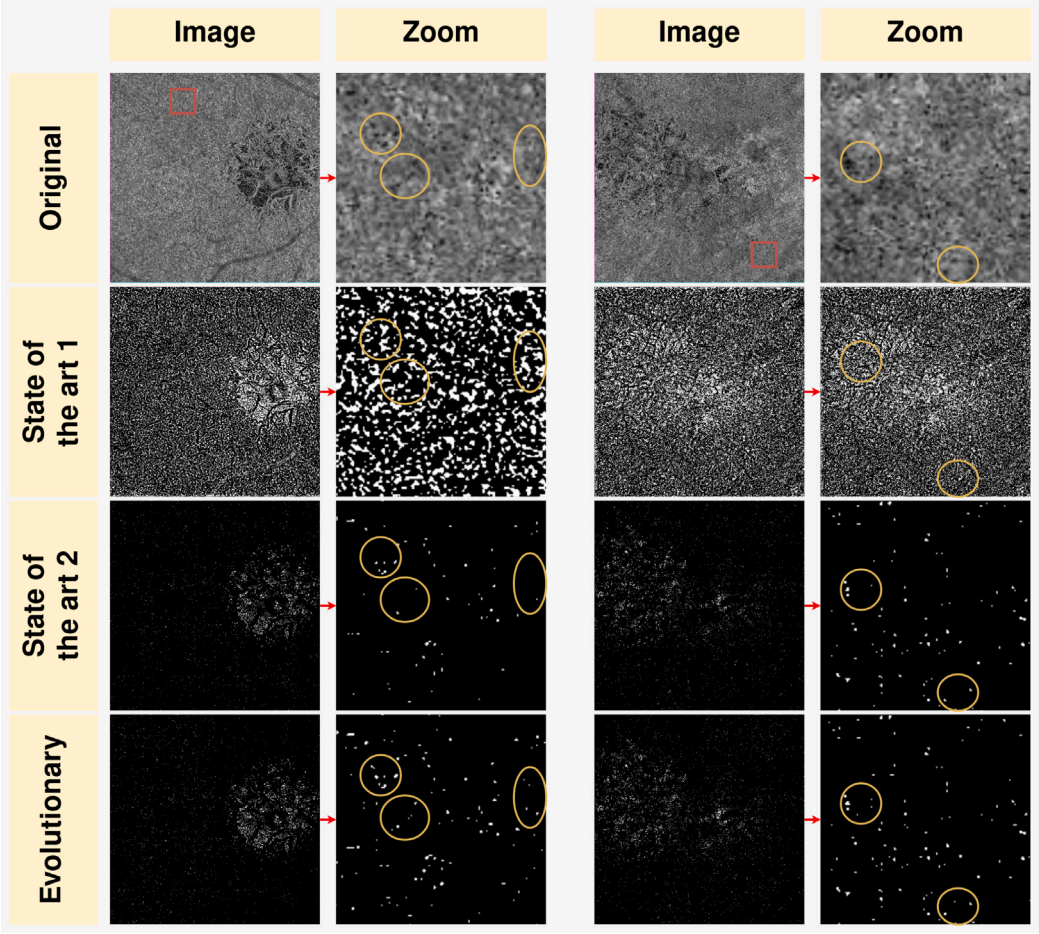


Fig. 13. Comparison of the segmentation masks extracted using the methodologies proposed in the state of the art and our evolutionary network.

Table 3
Comparison of the correlation coefficient obtained using the state of the art and our evolutionary network indirectly and directly.

	State of the art [16]	State of the art [15]	Evolution direct	Evolution indirect
Coefficient	0.2649	0.5016	0.5179	**0.5187**

art and our evolutionary network. As when compared to the baseline network, we can see how our evolutionary network is much more robust and accurate in extracting the FVs than the state of the art methodologies. This accurate extraction together with the partial segmentation mask helps the clinician when monitoring different patients, which facilitates and removes subjectivity from this medical task.

7. Conclusions

In this work, we have described the enormous potential and the significant gap that exists in the literature with respect to segmentation and quantification of FVs. This gap is due to, on the one hand, the novelty of the imaging modality, and on the other hand, to the complex characteristics that make it difficult to perform accurate manual labeling of these regions. In order to overcome these challenges and extract the maximum potential of this biomarker, we have proposed a optimized neural network architecture trained with a multi-target paradigm. This paradigm allows us to train the network in a much more generalized way, making our model more robust, accurate and adaptable while allowing us to obtain multiple objectives that together contribute to facilitating medical work. However, it is extremely difficult to manually find a suitable architecture to obtain all the proposed objectives in a balanced and accurate way. Therefore, in this work, we have proposed an evolutionary algorithm, based on a set of evolutionary operators, that allows us to create and progressively improve a population of network architectures. This population of architectures, in contrast to a generalist architecture, is optimized and tuned to most efficiently and accurately obtain a set of biomarkers used in the CC OCTA imaging. Moreover, our evolutionary algorithm allows us to create very diverse architectures in a large search space. To traverse this space efficiently and avoid falling into local minima, we propose a selection mechanism based on clustering. In particular, this clustering allows us to select the best individuals from each group within the population, which makes the population progressively better while maintaining the diversity of architectures.

Our evolutionary algorithm progressively improves the architectures using this multi-target perspective. This has numerous advantages from all points of view. On the one hand, the architectures become increasingly robust and accurate for multiple tasks. This makes them more adaptable for new tasks as well as for other imaging modalities. Therefore our models have great potential from the perspective of transfer learning and fine tuning. On the other hand, this type of learning confers much more utility to the models by directly extracting various targets. In our case, our best evolutionary network manages to accurately and robustly segment the FVs, while segmenting the areas with the highest accumulation of FVs and obtaining a direct estimate of their total area. This makes it possible to show a visualization to the clinical staff where the affected area is identified at a glance, to

show the specific FVs areas, or to extract relevant biomarkers for the diagnosis of multiple pathologies.

Even taking all this into account, there are several applications to test and some limitations to improve as future work. From an application point of view, it is interesting to test the potential of the best architectures discovered to be used in other imaging tasks and modalities. The adaptability of these models can be exploited to perform other complex tasks on other image types that present difficulty in labeling. In addition, our evolutionary algorithm can be tested on other datasets by adding new features or changing the hyper-parameters of its use. From the point of view of limitations, although our clustering methodology manages to maintain the diversity of the population, some convergence can be observed in some individuals. Based on this, our graph encoder network could be improved by adding more depth and changing the pooling. Besides, our evolutionary algorithm is still quite computationally expensive. To improve this, transfer learning can be adapted so that it is applied to nodes that share parents and children and thus reduce the training time of each network.

CRediT authorship contribution statement

Emilio López-Varela: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Software, Validation, Writing – original draft, Writing – review & editing. **Joaquim de Moura:** Conceptualization, Data curation, Investigation, Project administration, Supervision, Validation, Writing – review & editing. **Jorge Novo:** Conceptualization, Data curation, Funding acquisition, Investigation, Project administration, Supervision, Validation, Writing – review & editing. **José Ignacio Fernández-Vigo:** Data curation, Funding acquisition, Investigation, Project administration, Supervision, Validation. **Francisco Javier Moreno-Morillo:** Data curation, Funding acquisition, Investigation, Project administration, Supervision, Validation. **Julián García-Feijóo:** Data curation, Funding acquisition, Investigation, Project administration, Supervision, Validation. **Marcos Ortega:** Conceptualization, Data curation, Funding acquisition, Investigation, Project administration, Supervision, Validation, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The authors do not have permission to share data.

Acknowledgments

This research was funded by Government of Spain, Ministerio de Ciencia e Innovación y Universidades, Government of Spain, RTI2018-095894-B-I00 research project; Ministerio de Ciencia e Innovación, Government of Spain through the research projects with reference PID2019-108435RB-I00, reference PDC2022-133132-I00 and TED2021-131201B-I00; Consellería de Cultura, Educación e Universidade, Xunta de Galicia through the Grupos de Referencia Competitiva, grant ref. ED431C 2020/24; CITIC, as Research Center accredited by Galician University System, is funded by "Consellería de Cultura, Educación e Universidade from Xunta de Galicia", supported in an 80% through ERDF Funds, ERDF Operational Programme Galicia 2014–2020, and the remaining 20% by "Secretaría Xeral de Universidades", grant ref. ED431G 2019/01. Emilio López Varela acknowledges its support under FPI Grant Program through PID2019-108435RB-I00 project. Funding for open access charge: Universidade da Coruña/CISUG.

References

[1] E. López-Varela, P.L. Vidal, N.O. Pascual, J. Novo, M. Ortega, Fully-automatic 3D intuitive visualization of age-related macular degeneration fluid accumulations in OCT cubes, J. Digit. Imaging (2022) 1–12.

[2] P.L. Vidal, J. De Moura, J. Novo, M.G. Penedo, M. Ortega, Intraretinal fluid identification via enhanced maps using optical coherence tomography images, Biomed. Opt. Express 9 (10) (2018) 4730–4754.

[3] E. López-Varela, J. Novo, J.I. Fernández-Vigo, F.J. Moreno-Morillo, M. Ortega, Unsupervised deformable image registration in a landmark scarcity scenario: Choroid OCTA, in: International Conference on Image Analysis and Processing, Springer, 2022, pp. 89–99.

[4] R.F. Spaide, J.G. Fujimoto, N.K. Waheed, Image artifacts in optical coherence angiography, Retina (Philadelphia, PA) 35 (11) (2015) 2163.

[5] T.E. De Carlo, A. Romano, N.K. Waheed, J.S. Duker, A review of optical coherence tomography angiography (OCTA), Int. J. Retina Vitreous 1 (1) (2015) 5.

[6] J. Cao, D.S. McLeod, C.A. Merges, G.A. Lutty, Choriocapillaris degeneration and related pathologic changes in human diabetic eyes, Arch. Ophthalmol. 116 (5) (1998) 589–597.

[7] G. Lutty, J. Grunwald, A.B. Majji, M. Uyama, S. Yoneya, Changes in choriocapillaris and retinal pigment epithelium in age-related macular degeneration, Mol. Vis. 5 (35) (1999) 35.

[8] I. Bhutto, G. Lutty, Understanding age-related macular degeneration (AMD): relationships between the photoreceptor/retinal pigment epithelium/bruch's membrane/choriocapillaris complex, Mol. Aspects Med. 33 (4) (2012) 295–317.

[9] J.I. Fernández-Vigo, F.J. Moreno-Morillo, M. Ortega-Hortas, E. López-Varela, J. Novo-Bujan, B. Burgos-Blasco, L. López-Guajardo, J. García-Feijóo, J. Donate-López, Early changes in choriocapillaris flow voids as an efficacy biomarker of photodynamic therapy in central serous chorioretinopathy, Photodiagn. Photodyn. Therapy 38 (2022) 102862.

[10] M. Ho, F.H.P. Lai, D.S.C. Ng, L.P.L. Iu, L.J. Chen, A.C.Y. Mak, Y. Yip, C. Cheung, A.L. Young, M. Brelen, Analysis of choriocapillaris perfusion and choroidal layer changes in patients with chronic central serous chorioretinopathy randomised to micropulse laser or photodynamic therapy, Br. J. Ophthalmol. (2020).

[11] Y. Sugano, T. Sekiryu, M. Furuta, R. Tomita, H. Shintake, H. Maehara, A. Ojima, Morphometrical evaluation of the choriocapillaris imaged by swept-source optical coherence tomography angiography, Clin. Ophthal. (Auckland, NZ) 12 (2018) 2267.

[12] E. Zudaire, L. Gambardella, C. Kurcz, S. Vermeren, A computational tool for quantitative analysis of vascular networks, PLoS One 6 (11) (2011) e27385.

[13] F. Alten, P. Heiduschka, C.R. Clemens, N. Eter, Exploring choriocapillaris under reticular pseudodrusen using OCT-angiography, Graefe's Arch. Clin. Exp. Ophthalmol. 254 (11) (2016) 2165–2173.

[14] M. Al-Sheikh, K.G. Falavarjani, M. Pfau, A. Uji, P.P. Le, S.R. Sadda, Quantitative features of the choriocapillaris in healthy individuals using swept-source optical coherence tomography angiography, Ophthalmic Surg. Lasers Imaging Retina 48 (8) (2017) 623–631.

[15] E. López-Varela, J. De Moura, J. Novo, J.I. Fernández-Vigo, F.J. Moreno-Morillo, M. Ortega, Fully automatic segmentation and monitoring of choriocapillaris flow voids in OCTA images, Comput. Med. Imaging Graph. (2022).

[16] N. Phansalkar, S. More, A. Sabale, M. Joshi, Adaptive local thresholding for detection of nuclei in diversity stained cytology images, in: 2011 International Conference on Communications and Signal Processing, IEEE, 2011, pp. 218–220.

[17] C. Rochepeau, L. Kodjikian, M.-A. Garcia, C. Coulon, C. Burillon, P. Denis, B. Delaunay, T. Mathis, Optical coherence tomography angiography quantitative assessment of choriocapillaris blood flow in central serous chorioretinopathy, Am. J. Ophthalmol. 194 (2018) 26–34.

[18] H.S. Yang, T.G. Kang, H. Park, J.S. Heo, J. Park, K.S. Lee, S. Choi, Quantitative evaluation of choriocapillaris using optical coherence tomography and optical coherence tomography angiography in patients with central serous chorioretinopathy after half-dose photodynamic therapy, PLoS One 15 (1) (2020) e0227718.

[19] M. Nassisi, Y. Shi, W. Fan, E. Borrelli, A. Uji, M.S. Ip, S.R. Sadda, Choriocapillaris impairment around the atrophic lesions in patients with geographic atrophy: a swept-source optical coherence tomography angiography study, Br. J. Ophthalmol. (ISSN: 0007-1161) 103 (7) (2019) 911–917, http://dx.doi.org/10.1136/bjophthalmol-2018-312643, arXiv:https://bjo.bmj.com/content/103/7/911.full.pdf. URL https://bjo.bmj.com/content/103/7/911.

[20] A. Shah, L. Zhou, M.D. Abrámoff, X. Wu, Multiple surface segmentation using convolution neural nets: application to retinal layer segmentation in OCT images, Biomed. Opt. Express 9 (9) (2018) 4509–4526.

[21] J. Kugelman, D. Alonso-Caneiro, S.A. Read, J. Hamwood, S.J. Vincent, F.K. Chen, M.J. Collins, Automatic choroidal segmentation in OCT images using supervised deep learning methods, Sci. Rep. 9 (1) (2019) 1–13.

[22] V. Badrinarayanan, A. Kendall, R. Cipolla, Segnet: A deep convolutional encoder-decoder architecture for image segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 39 (12) (2017) 2481–2495.

[23] A. Xiao, B. Shen, X. Shi, Z. Zhang, Z. Zhang, J. Tian, N. Ji, Z. Hu, Intraoperative glioma grading using neural architecture search and multi-modal imaging, IEEE Trans. Med. Imaging 41 (10) (2022) 2570–2581.
[24] Z. Zhong, J. Yan, C.-L. Liu, Practical network blocks design with q-learning, 2017, arXiv preprint arXiv:1708.05552 6.
[25] L. Xie, A. Yuille, Genetic cnn, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 1379–1388.
[26] H. Liu, K. Simonyan, Y. Yang, Darts: Differentiable architecture search, 2018, arXiv preprint arXiv:1806.09055.
[27] M. Tan, R. Pang, Q.V. Le, Efficientdet: Scalable and efficient object detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 10781–10790.
[28] Y. Weng, T. Zhou, Y. Li, X. Qiu, Nas-unet: Neural architecture search for medical image segmentation, IEEE Access 7 (2019) 44247–44257.
[29] Y. Zhou, G.G. Yen, Z. Yi, Evolutionary compression of deep neural networks for biomedical image segmentation, IEEE Trans. Neural Netw. Learn. Syst. 31 (8) (2019) 2916–2929.
[30] Y. Zhou, G.G. Yen, Z. Yi, Evolutionary shallowing deep neural networks at block levels, IEEE Trans. Neural Netw. Learn. Syst. 33 (9) (2021) 4635–4647.
[31] E. Real, A. Aggarwal, Y. Huang, Q.V. Le, Regularized evolution for image classifier architecture search, in: Proceedings of the Aaai Conference on Artificial Intelligence, Vol. 33, 2019, pp. 4780–4789.
[32] H. Pham, M. Guan, B. Zoph, Q. Le, J. Dean, Efficient neural architecture search via parameters sharing, in: International Conference on Machine Learning, PMLR, 2018, pp. 4095–4104.
[33] C. Liu, B. Zoph, M. Neumann, J. Shlens, W. Hua, L.-J. Li, L. Fei-Fei, A. Yuille, J. Huang, K. Murphy, Progressive neural architecture search, in: Proceedings of the European Conference on Computer Vision, ECCV, 2018, pp. 19–34.
[34] A. Singh, S. Saha, R. Sarkhel, M. Kundu, M. Nasipuri, N. Das, A genetic algorithm based kernel-size selection approach for a multi-column convolutional neural network, 2019, arXiv preprint arXiv:1912.12405.
[35] H. Shu, Y. Wang, Automatically searching for U-net image translator architecture, 2020, arXiv preprint arXiv:2002.11581.
[36] B. Zoph, V. Vasudevan, J. Shlens, Q.V. Le, Learning transferable architectures for scalable image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 8697–8710.
[37] Y. Zhou, X. Yuan, X. Zhang, W. Liu, Y. Wu, G.G. Yen, B. Hu, Z. Yi, Evolutionary neural architecture search for automatic esophageal lesion identification and segmentation, IEEE Trans. Artif. Intell. 3 (3) (2021) 436–450.
[38] Z. Huang, Z. Wang, Z. Yang, L. Gu, AdwU-Net: adaptive depth and width U-net for medical image segmentation by differentiable neural architecture search, in: International Conference on Medical Imaging with Deep Learning, PMLR, 2022, pp. 576–589.
[39] C. Liu, L.-C. Chen, F. Schroff, H. Adam, W. Hua, A.L. Yuille, L. Fei-Fei, Auto-deeplab: Hierarchical neural architecture search for semantic image segmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 82–92.
[40] Z. Lu, I. Whalen, Y. Dhebar, K. Deb, E.D. Goodman, W. Banzhaf, V.N. Boddeti, Multiobjective evolutionary design of deep convolutional neural networks for image classification, IEEE Trans. Evol. Comput. 25 (2) (2020) 277–291.
[41] M. Fey, J.E. Lenssen, Fast graph representation learning with PyTorch Geometric, in: ICLR Workshop on Representation Learning on Graphs and Manifolds, 2019.
[42] Itseez, Open source computer vision library, 2015, https://github.com/itseez/opencv.
[43] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in python, J. Mach. Learn. Res. 12 (2011) 2825–2830.
[44] P. Yakubovskiy, Segmentation models pytorch, 2020, https://github.com/qubvel/segmentation_models.pytorch.
[45] J. Chen, Y. Lu, Q. Yu, X. Luo, E. Adeli, Y. Wang, L. Lu, A.L. Yuille, Y. Zhou, Transunet: Transformers make strong encoders for medical image segmentation, 2021, arXiv preprint arXiv:2102.04306.
[46] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L.-C. Chen, Mobilenetv2: Inverted residuals and linear bottlenecks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 4510–4520.
[47] A. Kendall, Y. Gal, R. Cipolla, Multi-task learning using uncertainty to weigh losses for scene geometry and semantics, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 7482–7491.
[48] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, S. Belongie, Feature pyramid networks for object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 2117–2125.