

POLITECNICO DI MILANO

**School of Industrial and Information Engineering - Department of Electronics,
Information and Bioengineering**



**DEEP LEARNING IN FOREIGN BODY RESPONSE:
AUTOMATED SYSTEMS FOR MICROSCOPY
IMAGE ANALYSIS**

Supervisor:

Dr. Pietro Cerveri

Co-Supervisor:

Dr. Eleonora Dondossola

Co-Supervisor:

Dr. Stefano Casarin

Thesis by:

Mattia Sarti, 901402

Academic Year: 2018-2019

Acknowledgements

I am deeply thankful to my parents for allowing me to pursue my degree under the most favorable conditions.

I thank professor Pietro Cerveri for letting me learn about and specialize in the field I am interested in.

I thank Eleonora Dondossola and Stefano Casarin for the opportunity given and for the endless helpfulness always shown.

Ringraziamenti

Ringrazio profondamente i miei genitori per avermi permesso di continuare i miei studi nelle migliori delle condizioni.

Ringrazio il professor Pietro Cerveri per avermi permesso di apprendere e specializzarmi nell'ambito di mio interesse.

Ringrazio Eleonora Dondossola e Stefano Casarin per l'opportunità concessami e per l'infinita disponibilità sempre dimostratami.

Abstract

Introduction

The Foreign Body Response (FBR) is a mechanism of defense evolved by hosts to deal with the penetration of potentially harmful foreign materials and reduce consequent side effects [1]. In implant medicine, however, the FBR represents a clinical problem as it causes malfunction and failure of implanted devices. Thus, monitoring and analyzing the FBR dynamics in pre-clinical models is crucial [1]. To this purpose, mice implanted with polymeric scaffolds coupled to intravital non-linear microscopy acquisition have been implemented as a model to study the evolution of the FBR over time. From the latter, analysis consists of manual localization of both cellular and subcellular structures of interest and quantification of related features, which is time-consuming and subjected to human error. Consequently, automated software tools are needed to overcome these pitfalls, for which deep learning techniques represent a potential solution.

In this work we propose a versatile adaptation of neural-network-based systems to: **1)** determine the primary orientation of collagen fibrils around scaffold's framework, **2)** extract the fibroblasts' profile to quantify parameters of interest, and **3)** segment scaffold-associated and interstitial cells to quantify their amounts.

Materials and Methods

In each task, we used an adaptation of the U-Net neural network architecture to segment relevant structures [2] and integrated it into the standard image processing pipeline to extract significant features. For each task, the network was first trained with in-house microscopy images, labelled under the supervision of an expert biologist, and then validated on a distinct set of images.

1) Application 1: Quantification of Collagen Orientation. Quantification relative to scaffold framework was achieved by combining semantic segmentation to a customized

directional filtering. The segmentation isolated the scaffold structure, while directional filtering highlighted the orientation of both scaffold and collagen fibrils by filtering them with the defined kernels. Finally, collagen orientations, in the form of histogram counts, were referred to scaffold predominant directionality to quantify collagen overall trend. Input images were preprocessed to highlight information content for optimal machine perception, according to a cascade of normalization, gamma correction and histogram stretching based on each image statistics. The neural network for scaffold segmentation was trained and validated on independent datasets of respectively 256 and 120 samples created with the extensive use of data augmentation, following a customized training strategy. The directional filtering kernels were created by approximating linear edges oriented at the desired angular resolutions. Filtering results were combined to assign a predominant orientation to each pixel belonging to a boundary of the structure of interest. Finally, scaffold orientations were extracted from the skeletonized segmentation result, while collagen orientations were considered only outside the scaffold mask to avoid artifacts.

2) Application 2: Extraction of Fibroblast Parameters. First, the neural network accomplished semantic segmentation of fibroblasts. The segmentation was then refined through isolation of the cells based on pixel connectivity. Input images were preprocessed (as performed for collagen) to highlight relevant content, and the neural network was trained and validated on independent, augmented datasets of 200 and 88 samples, respectively. The training phase minimized a customized loss functions where each pixel was assigned a weight to segment with particular attention near or touching fibroblasts, allowing for the subsequent separation via connectivity. Finally, area, centroid coordinates and main orientation of each fibroblast and their mutual distances were extracted. Main orientation was quantified as for collagen.

3) Application 3: Segmentation of Scaffold-Associated Cells. Semantic segmentation of scaffold-associated cells, interstitial cells and acellular background was performed end-to-end to quantify area and growth of cellular species. Input images were individually normalized over their intensity values and the neural network was trained and validated on separate augmented datasets of respectively 200 and 88 samples with a multiclass training.

At inference time, area was eventually quantified for each class from the predicted segmentation.

Results and Discussion

The semantic segmentation methods resulted to be accurate for all applications yielding the following metrics on the respective validation sets:

1) Application 1: Quantification of Collagen Orientation. Collagen orientation was successfully extracted.

0.7955 mIoU (mean Intersection over Union), sufficient both to extract scaffold predominant directionality and to fully cover scaffold artifacts when evaluating collagen orientations.

2) Application 2: Extraction of Fibroblast Parameters. Fibroblasts were successfully segmented and area, centroid, main orientation of each and mutual distances retrieved.

0.8869 mIoU, reflecting the adhesion to the segmentation task. Moreover, a weighted probability likelihood metric was evaluated to also account for separation of near cells, with a considerable outcome. Contained borders of touching fibroblasts in predicted segmentation masks allowed for successful instance separation via connectivity indeed.

3) Application 3: Segmentation of Scaffold-Associated Cells. Scaffold-associated and interstitial cells were correctly identified and quantified.

0.5330, 0.5795 and 0.9019 mIoU respectively for scaffold-associated cells, interstitial cells and background, reasonable values given the data domain of application and the issues concerning training set labeling. A meaningful generalization capability allowed to question human performances in gold-standard label creation for some instances.

Finally, a qualitative evaluation of segmentation results was visually performed by biologists, leading to satisfying outcomes for all applications.

Conclusions

We herein present automated software tools for the detection of structures and associated features of interest within FBR phenomenon. For better outcomes, both quantity and quality of input images and labels will be increased. The first because training with minimal data variability limits the domain where a model can be meaningfully employed, the second since coherent examples have to be shown for an effective training.

Sommario

Introduzione

La risposta ad agenti estranei (Foreign Body Response, FBR) è un meccanismo di difesa sviluppato all'organismo in risposta alla penetrazione di materiale estraneo potenzialmente dannoso per ridurre i conseguenti effetti negativi [1]. In implantologia protesica, tuttavia, il FBR rappresenta un problema clinico dal momento che causa il malfunzionamento, nonché insuccesso, dei dispositivi impiantati. Dunque, il monitoraggio e l'analisi delle dinamiche coinvolte nel FBR in modelli pre-clinici è di cruciale importanza [1]. A questo fine, sono stati impiegati ratti con impianti di scaffold polimerici come modello per studiare l'evoluzione del FBR nel tempo tramite acquisizioni di microscopia non-lineare intra-vitale. Riguardo quest'ultima, l'analisi consiste nella localizzazione manuale delle strutture di interesse, sia cellulari che subcellulari, e nella quantificazione delle relative caratteristiche, procedura dispendiosa in termini di tempo e soggetta all'errore umano. Di conseguenza, strumenti software automatizzati si rendono necessari per superare questi svantaggi, a cui le tecniche di deep learning rappresentano una potenziale soluzione.

In questo lavoro viene proposto un versatile adattamento di sistemi basati su reti neurali per: **1)** determinare l'orientamento principale delle fibre di collagene intorno alla struttura dello scaffold, **2)** estrarre il profilo dei fibroblasti per quantificarne i parametri di interesse, e **3)** segmentare le cellule associate allo scaffold e le cellule interstiziali per stimarne le rispettive quantità.

Materiali e Metodi

In ognuna delle applicazioni, è stato utilizzato un adattamento dell'architettura della rete neurale U-Net per segmentare le strutture di interesse [2] ed integrato con le tradizionali procedure di elaborazione di immagini per estrarre le caratteristiche significanti. Per ogni applicazione, la rete è stata prima addestrata su immagini di microscopia in-house, etichettate sotto la supervisione di un biologo esperto, e successivamente validata su un distinto insieme di immagini

1) Applicazione 1: Quantificazione dell'Orientamento del Collagene. Tale quantificazione, riferita all'intelaiatura dello scaffold, è stata raggiunta combinando la segmentazione semantica ad un filtraggio direzionale ad-hoc. La segmentazione ha isolato la struttura dello scaffold, mentre il filtraggio direzionale ha evidenziato l'orientamento sia dello scaffold stesso che delle fibre di collagene tramite la convoluzione con i definiti kernel. Infine, gli orientamenti del collagene, sottoforma di istogramma, sono stati riferiti alla direzione predominante dello scaffold per quantificare la disposizione complessiva del collagene. Le immagini di ingresso sono state pre-elaborate in modo da evidenziarne il contenuto informativo per un'ottimale percezione artificiale, seguendo in cascata una normalizzazione, una correzione gamma ed uno stretching dell'istogramma basati sulle statistiche di ogni immagine. La rete neurale per la segmentazione dello scaffold è stata addestrata e validata su dataset indipendenti rispettivamente di 256 e 120 campioni creati tramite l'impiego estensivo di data augmentation, seguendo una strategia di training specifica. I kernel per il filtraggio direzionale sono stati create approssimando bordi lineari orientati alla risoluzione angolare desiderata. I risultati del filtraggio sono stati combinati per assegnare un orientamento predominante ad ogni pixel appartenente ad un contorno della struttura di interesse. Infine, gli orientamenti dello scaffold sono stati estratti dal risultato della segmentazione skeletonizzato, mentre gli orientamenti del collagene sono stati considerati solamente al di fuori della maschera dello scaffold per evitare artefatti.

2) Applicazione2: Estrazione dei Parametri dei Fibroblasti. Prima la rete neurale ha realizzato la segmentazione dei fibroblasti, poi le cellule sono state isolate in base alla connettività dei pixel. Le immagini in ingresso sono state pre-elaborate (come fatto per il collagene) per evidenziarne il contenuto rilevante, e la rete è stata addestrata e validata su dataset indipendenti ed incrementati tramite data augmentation di 200 e 88 campioni, rispettivamente. La fase di training ha minimizzato una funzione di perdita ah-hoc dove ad ogni pixel era stato assegnato un peso per segmentare con particolare attenzione le cellule vicine o a contatto, permettendo la seguente separazione via connettività. Infine, area, coordinate del centroide e orientamento principale di ogni fibroblasto e le loro mutue

distanze sono state estratte. L'orientamento principale è stato quantificato come per il collagene.

3) Applicazione 3: Segmentazione delle Cellule Associate allo Scaffold. La segmentazione semantica di cellule associate allo scaffold, cellule interstiziali e background acellulare è stata effettuata end to end per quantificare l'area e la crescita delle specie cellulari. Le immagini di ingresso sono state normalizzate individualmente sui propri valori di intensità e la rete neurale è stata addestrata e validata su dataset separati ed incrementati tramite data augmentation di rispettivamente 200 e 88 campioni con un training multi-classe. Applicate le inferenze, l'area è stata infine quantificata per ogni classe dalla segmentazione predetta.

Risultati e Discussioni

I metodi di segmentazione semantica sono risultati accurate per tutte le applicazioni dando le seguenti metriche sui rispettivi dataset di validazione:

1) Applicazione 1: Quantificazione dell'Orientamento del Collagene. L'orientamento del collagene è stato estratto con successo.

0.7955 di intersezione su unione media (mean Intersection over Union, mIoU), sufficiente sia ad estrarre la direzione predominante dello scaffold sia a ricoprire interamente gli artefatti dello scaffold in fase di valutazione degli orientamenti del collagene.

2) Applicazione 2: Estrazione dei Parametri dei Fibroblasti. I fibroblasti sono stati segmentati con successo e l'area, il centroide, l'orientamento principale di ognuno e le mutue distanze sono stati estratti.

0.8869 di mIoU, riflettendo l'adesione all'obiettivo di segmentazione. Inoltre, una metrica di similitudine di probabilità pesata è stata valutata per considerare anche la separazione di cellule vicine, con un considerevole risultato. I contorni contenuti delle cellule a contatto nelle maschere di segmentazione predette ha infatti permesso di separare le istanze con successo attraverso la connettività.

3) Applicazione 3: Segmentazione delle Cellule Associate allo Scaffold. Le cellule associate allo scaffold e le cellule interstiziali sono state correttamente identificate e quantificate.

0.5330, 0.5795 e 0.9019 di mIoU rispettivamente per cellule associate allo scaffold, cellule interstiziali e background, valori ragionevoli dati il dominio dei dati di applicazione e i problemi riguardanti l'etichettamento del dataset di training. Una significativa capacità di generalizzazione ha consentito di dubitare delle prestazioni umane nella generazione del gold-standard per alcuni campioni.

Infine, una valutazione qualitative dei risultati è stata effettuata visivamente dai biologi, portando a soddisfacenti esiti per tutte le applicazioni.

Conclusioni

Sono qui stati presentati strumenti software automatizzati per la detezione delle strutture e delle associate caratteristiche di interesse all'interno del fenomeno del FBR. Per risultati migliori, sia la quantità che la qualità delle immagini di ingresso e dei risultati ad essi manualmente etichettati sarà incrementata. La prima poiché addestrare una rete con variabilità minima nei dati limita il dominio su cui il modello può essere significativamente impiegato, la seconda dato che per un training efficace devono essere mostrati esempi coerenti.

Contents

Acknowledgements	2
Ringraziamenti	3
Abstract.....	5
Introduction.....	5
Materials and Methods.....	5
Results and Discussion	7
Conclusions.....	8
Sommario	9
Introduzione	9
Materiali e Metodi	9
Risultati e Discussioni	11
Conclusioni	12
Contents	13
List of Figures.....	15
List of Abbreviations	18
1. Introduction	20
1.1 Foreign Body Response	20
1.1.1 Phases of the FBR	21
1.1.2 Intervention to Reduce the FBR.....	23
1.1.3 Intravital Longitudinal Multiphoton Detection of the FBR	24
1.2 Limitations of Current Analysis	25
1.3 Neural Networks	26
1.4 Scope of the Thesis	27
1.4.1 Application 1: Quantification of Collagen Orientation.....	27
1.4.2 Application 2: Extraction of Fibroblast Parameters	28
1.4.3 Application 3: Segmentation of Scaffold-Associated Cells.....	28

2. Materials and Methods	29
2.1 Experimental Data	29
2.2 Neural-Network-based Software	29
• Encoding	34
• Decoding	41
2.3 Applications	45
2.3.1 Application 1: Quantification of Collagen Orientation.....	45
2.3.2 Application 2: Extraction of Fibroblast Parameters.....	54
2.3.3 Application 3: Segmentation of Scaffold-Associated Cells.....	58
3. Results and Discussion	60
3.1 Application 1: Quantification of Collagen Orientation	60
3.2 Application 2: Extraction of Fibroblast Parameters.....	71
3.3 Application 3: Segmentation of Scaffold-Associated Cells.....	76
4. Conclusions	81
References	83

List of Figures

Figure 1: FBR elicited by a mammary implant. A, A fibrous capsule overlying a breast silicone implant; B, fibrotic capsule (cap); implant, (i); C & D, Fibrotic capsule, macroscopic overview and histological staining (Adapted from Quesada, 2019).	21
Figure 2: Sequential steps of the foreign body response process following a biomaterial implantation. A. Adsorption on the biomaterial surface of the proteins involved in the coagulation cascade, complement and adhesion pathways. B. Acute inflammatory	22
Figure 3: Level of intervention of this work.....	27
Figure 4: Example of NN architecture with single node magnification highlighted.....	30
Figure 5: Training phase schematic.....	32
Figure 6: UNet architecture (characteristic “U” shape) schematic showing layers and functions (blue: convolutional layer, magenta: batch-normalization layer, green: rectified linear unit activation function, red: max-pooling layer, yellow: transposed-convolutional layer, gray: concatenation through skip-connections dotted in light blue). Encoding part and constituent block highlighted in green, decoding part and constituent block in orange.	34
Figure 7: Encoding block schematic showing tensor (light blue) dimensions, considering a single sample, before and after each constituent layer (colors represented coherently with Fig. 6). Magnification highlighting several feature maps within the same tensor.	36
Figure 8: Decoding block schematic showing tensor (light blue) dimensions, considering a single sample, before and after each constituent layer (colors represented coherently with Fig. 6). Magnification highlighting feature maps added through skip-connection to the resulting concatenated tensor.....	42
Figure 9: Subsequent results of preprocessing steps applied to a raw sample: both images (I) and histograms (H) shown after normalization (i), after gamma-correction (ii) and after histogram stretching (iii).	47
Figure 10: Learning rate trend with epochs.....	50
Figure 11: Kernel creation steps exemplified for an orientation of 22.5° (counterclockwise). Coefficients cumulatively shown after: being assigned distance from central oriented line (A), ramp modulation and truncation of values (B), negative assignment of clockwise side values (C), truncation outside central radius (D) and normalization of both positive and negative values to unitary sum modules (E).	53

Figure 12: Intersection over Union (IoU) meaning exemplification.	61
Figure 13: Loss (red) and mIoU metric (blue) trends over training samples (T) and validation samples (V) with training epochs. Ideal mIoU limit value dotted in blue.....	62
Figure 14: Scaffold segmentation: preprocessed input image (left), predicted output mask (middle) and target manual label (right) shown for 4 different validation instances (A, B, C and D) - white showing foreground in masks.	64
Figure 15: Scaffold segmentation: preprocessed input image (left), predicted output mask (middle) and target manual label (right) shown for 4 different validation instances (A, B, C and D) - white showing foreground in masks.	65
Figure 16: Exemplification of stages for extracting collagen orientations relatively to scaffold for a given case test: raw input collagen image (A), extracted spatial map of orientations of collagen boundaries in the whole image (B) and in the subregion of interest where collagen directionalities are quantified (E) with associated colormap, resulting histogram of absolute collagen orientations in the subregion (F), raw input scaffold image (C), extracted spatial map of orientations of skeletonized scaffold mask (D), resulting histogram of absolute scaffold orientations (G) and collagen histogram of F shifted relatively to scaffold predominant orientation (H).	67
Figure 17: Exemplification of stages for extracting collagen orientations relatively to scaffold for a given case test: raw input collagen image (A), extracted spatial map of orientations of collagen boundaries in the whole image (B) and in the subregion of interest where collagen directionalities are quantified (E) with associated colormap, resulting histogram of absolute collagen orientations in the subregion (F), raw input scaffold image (C), extracted spatial map of orientations of skeletonized scaffold mask (D), resulting histogram of absolute scaffold orientations (G) and collagen histogram of F shifted relatively to scaffold predominant orientation (H).	68
Figure 18: : Exemplification of stages for extracting collagen orientations relatively to scaffold for a given case test: raw input collagen image (A), extracted spatial map of orientations of collagen boundaries in the whole image (B) and in the subregion of interest where collagen directionalities are quantified (E) with associated colormap, resulting histogram of absolute collagen orientations in the subregion (F), raw input scaffold image (C), extracted spatial map of orientations of skeletonized scaffold mask (D), resulting histogram of absolute scaffold orientations (G) and collagen histogram of F shifted relatively to scaffold predominant orientation (H).	69
Figure 19: Exemplification of stages for extracting collagen orientations relatively to scaffold for a given case test: raw input collagen image (A), extracted spatial map of orientations of collagen boundaries in the whole image (B) and in the subregion of interest where collagen directionalities are quantified (E) with associated colormap, resulting histogram of absolute collagen orientations in the subregion (F), raw input scaffold image (C), extracted spatial map of orientations of skeletonized scaffold mask (D), resulting	

histogram of absolute scaffold orientations (G) and collagen histogram of F shifted relatively to scaffold predominant orientation (H). 70

Figure 20: From left to right: input image, predicted segmentation mask (blue dotted circles highlighting touching cells correctly separated), target manual label and final predicted objects clustered based on 8-pixel connectivity (different colors for different objects) for different (A, B, C, D, E) validation instances. 72

Figure 21: Training loss (red) and mIoU (blue) trends with training epochs: same network with (BN) and without (NO BN) batch-normalization layers during different trials of 100 (T1) and 300 (T2) epochs. 73

Figure 22: Exemplification of extracted parameters: fibroblasts in each input images (A, B, C) counted (D) with their area distribution quantified (E, bar: mean, whisker: standard deviation); instanced fibroblasts (F) in the highlighted subregion of (A) shown with different colors and characterized (G) each in terms of main orientation, centroid coordinates and area, with mutual distances between centroids reported. 75

Figure 23: Segmentation of scaffold-associated cells (red), interstitial cells (blue) and background (black): preprocessed input image (left), predicted output mask (middle) and target manual label (right) shown for 4 different validation instances (A, B, C and D). 77

Figure 24: Segmentation of scaffold-associated cells (red), interstitial cells (blue) and background (black): preprocessed input image (left), predicted output mask (middle) and target manual label (right) shown for 4 different validation instances (A, B, C and D). 78

Figure 25: Example of area quantification in pixels both for SA class and IN class in terms of mean (bar) and standard deviation (whiskers) over five segmentation results (B, C, D, E, F), each illustrated as input image (left) and segmentation outcome (right) - red and blue always associated to SA and IN, respectively. 80

List of Abbreviations

FBR	Foreign Body Response
PMN	Polymorphonucleated
FBGC	Foreign Body Giant Cell
VEGF	Vascular Endothelial Growth Factor
PCL	Polycaprolactone
NN	Neural Network
CNN	Convolutional Neural Network
FCN	Fully-Convolutional Neural Network
CL	Convolutional Layer
BN	Batch-Normalization
ReLU	Rectified Linear Unit
IoU	Intersection Over Union
mIoU	Mean Intersection Over Union
SA	Scaffold-Associated Cells
IN	Interstitial Cells
BG	Background

1. Introduction

1.1 Foreign Body Response

The Foreign Body Response (FBR) denotes the physiological response of an organism to the penetration of a foreign material and is aimed to minimize the impact for the host through potential resorption and shielding by fibrotic encapsulation.

The FBR is a stepwise process, consisting of an initial inflammatory phase, followed by wound healing and eventual tissue fibrosis and scarring, triggered by the vascular damage following object penetration. Designed by nature to protect healthy tissue from foreign assault, the FBR is a clinical problem given that both inflammation and fibrosis can impair implanted medical devices' functionality, the former causing biomaterial degradation and the latter creating a physical barrier compromising its bioactivity. As result, sensors, pacemakers, prostheses, and scaffolds used in tissue engineering and regenerative medicine can experience malfunction and even failure. An overview of the components involved in the development of the FBR is offered in the next paragraphs.

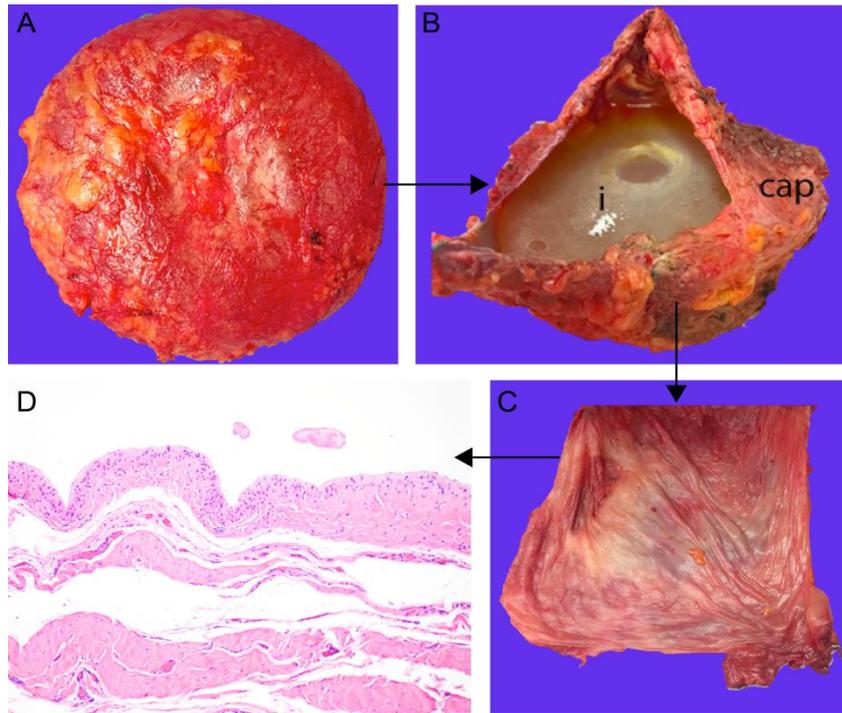


Figure 1: FBR elicited by a mammary implant. A, A fibrous capsule overlying a breast silicone implant; B, fibrotic capsule (cap); implant, (i); C & D, Fibrotic capsule, macroscopic overview and histological staining (Adapted from Quesada, 2019).

1.1.1 Phases of the FBR

As shown in Fig. 2, the FBR encompasses four main stages, here listed and further detailed below in dedicated sub-section:

1. Biofouling
2. Acute inflammation
3. Chronic inflammation
4. Granulation tissue and fibrous capsule formation

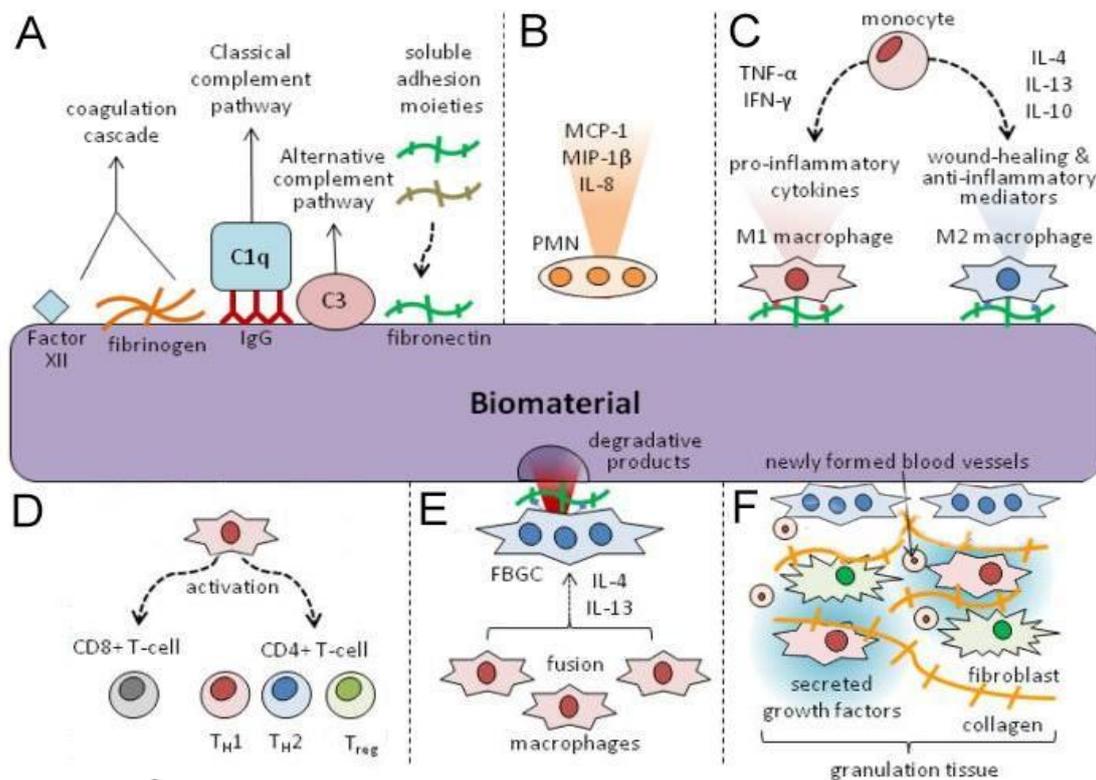


Figure 2: Sequential steps of the foreign body response process following a biomaterial implantation. A. Adsorption on the biomaterial surface of the proteins involved in the coagulation cascade, complement and adhesion pathways. B. Acute inflammatory

1) Biofouling (time: seconds-hours after implantation; Fig. 2, A)

The intrusion of a foreign body in a living tissue is associated to the disruption of local vessels, leading to the release of plasma proteins and thrombotic agents, which deposit on the foreign body surface [3], [4] and mediate platelet adhesion and activation [5], [6] and attraction of leukocytes, such as Polymorphonucleated (PMN).

2) Acute inflammation (time: hours-days after implantation; Fig. 2, B)

The second phase in the progression of the FBR is initiated by PMN cells, that activate and release molecules that attract the cells of chronic inflammation, including monocyte-derived macrophages and other leukocytes [7], [8], [9].

3) Chronic inflammation (time: days-weeks after implantation; Fig. 2, C-E)

Upon arrival, macrophages derived from circulating monocytes adhere to the matrix deposited on the foreign object [10]. Macrophages are myeloid immune cells with a

phagocytic function, strategically positioned throughout the body tissues, where they ingest and degrade dead cells, debris, foreign materials and orchestrate the inflammatory processes [11]. Based on the properties of the material, monocytes can differentiate in different subtypes and start producing specific cytokines (e.g. tumor necrosis factor, TNF α ; IL-4 and others, Fig. 2, C), resulting in a further attraction of immune cells at the wound site [12], such as lymphocytes (including CD8⁺, but mainly CD4⁺ T-cells).

In the majority of the cases, macrophages at the interface of an implanted material cannot resorb it and undergo “frustrated phagocytosis”. This phenomenon stimulates their fusion, which gives rise to the formation of the so called “foreign body giant cells” (FBGCs), large up to hundreds μm with dozens of nuclei that wrap the foreign material [3]. As result, this fusion seems to increase the effectiveness of resorption and to avoid apoptosis [13].

4) *Granulation tissue and fibrous capsule formation (time weeks-years; Fig. 2, F)*

The pro-fibrotic and -angiogenic factors secreted by macrophages, FBGCs and other immune cells further attract fibroblasts and endothelial cells to the implantation site, leading to the formation of a granulation tissue. This tissue is characterized by a net of collagen fibers, collagen-secreting fibroblasts, the newly formed capillary, together with the inflammatory cells [14].

Usually, a regular wound healing process ends with a resolution phase that implies a regression of both vessel development and collagen deposition and the apoptosis of fibroblasts and myofibroblasts. However, the presence of the foreign object represents a pro-inflammatory stimulus that prevents any resolution. The final stage of the biomaterial integration thus culminates with the formation of a fibrotic capsule that surrounds the implant and isolates it from the host tissue [3].

1.1.2 Intervention to Reduce the FBR

Given the impact of the FBR on medical devices, several strategies have been evaluated to reduce the resulting inflammatory and fibrotic response, including modulation of biomimetic properties or pharmacological interference [1].

Modifying the biomaterials properties. The size, shape, texture and functionalization of biomaterials can profoundly affect the adsorption of proteins and the recruitment of cells.

Thus, the modification of these intrinsic properties or the functionalization by specific groups followed by *in vivo* testing represent a key strategy to reduce the FBR that is currently exploited by many biomaterial labs.

Pharmacological interference. Pharmacological interference can be performed at either local or systemic level. The first approach may implicate the coupling of the implant with absorbable reservoirs (nanoparticles, hydrogels, smart hydrogels) designed to slowly release anti-inflammatory or anti-fibrotic drugs, [15], [16]. Systemic interference has been explored including local immunosuppression by administration of steroidal and non-steroidal anti-inflammatory drugs which inhibit the formation of inflammatory mediators [17]; dampening leukocyte and fibroblast activation by anti-transforming growth factor-antibody or halofunginone [18], [19]; stimulating vessel development to improve perfusion and functionality of the bioactive implants by pro-angiogenic vascular endothelial growth factor (VEGF) [17], [18], [20], [21].

Despite partially improving outcome, none of these approaches have resulted in a complete and long-lasting efficacy, with medical implants continuing to display inflammation, fibrosis and even graft failure as outcomes [17], [18].

Thus, in order to better understand the mechanisms at the basis of the FBR, and further identify which strategies are effective to reduce this phenomenon, preclinical experiments need to be properly monitored, analyzed and quantified.

1.1.3 Intravital Longitudinal Multiphoton Detection of the FBR

To longitudinally monitor the steps and fate of the FBR *in vivo*, we used fluorescent reporter mice harboring α SMA-RFP fibroblasts together with GFP-expressing bone marrow-derived cells. The chosen biomaterial was Polycaprolactone (PCL) fabricated by melt-electrospinning with fully interconnected honeycomb pore networks in the form of scaffolds, which were implanted in parallel to the deep dermis/subcutis interface in mice.

Non-linear multiphoton microscopy was applied to dissect the three-dimensional spatial organization and fate of scaffold integration. Compared to other types of fluorescent microscopy, this ensured better tissue penetration, improved detection of important

composites and lowered phototoxicity. To account for temporal dynamics and support longitudinal monitoring, an optical imaging window was implanted in mice. A custom-made multiphoton microscope with three different titanium-sapphire lasers and two optical parametric oscillators (yielding a tunable range of excitation wavelengths between 800 and 1300 nm) was employed. Multi-spectral detection of 3D stacks was performed using up to five photomultipliers and three excitation wavelengths in consecutive scans, to separate the following excitation and emission channels: GFP (920 nm; 525/50 nm); DsRed (1090 nm; 595/40 nm), SHG (1090 nm; 525/50 nm), THG (1180 nm; 387/15 nm) [1]. The volumes acquired were characterized by the same constant in-plane physical spatial resolutions of 360 x 360 μm , 1064 x 1064 px, while the depth physical resolution, in-between slices, was 5 μm .

Acquisitions showed immigrating myeloid cells engaged with and became immobilized at the scaffold-tissue interface and converted to multinucleated giant cells. In parallel, fibroblasts were recruited and activated to deposit a collagenous capsule with fibrils topologically aligned along or perpendicular to the scaffold interface.

1.2 Limitations of Current Analysis

Biologists mostly perform image analysis using Image J, an image software coupled to a rich set of standard built-in functions and plug-ins, which allows reconstruction and processing of images captured at the multiphoton microscope. However, given the complex nature of the parameters retrieved, no standard image processing toolbox could be applied to the complex analysis of the FBR. For instance, this software is not able to automatically discern a specific cellular type when other ones are present in the same context, such as scaffold-associated giant cells and interstitial leukocytes. In addition, even more advanced techniques as morphological operators cannot easily give a reliable segmentation result when exposed to highly intricate shapes, such as fibroblasts. In such a data domain, indeed, fibroblasts neither exhibited similar coherent intensities throughout different slices nor had a simple geometry to describe.

Thus, despite software functionalities were available for processing images, instances could not be singled out with built-in tools, and manual extraction of parameters of interest is

usually performed. This includes individual cell counting by eye inspection, manual measurement of distances, and qualitative identification of cellular subtypes. As an example, immune cell growth was quantified as overall area, without distinction among cell populations (e.g. scaffold-associated FBGCs vs other leukocytes), or the distance between fibroblasts and scaffold fibers was measured manually for each cell considered (for up to hundreds of cells). as consequence, this pipeline clearly implies several disadvantages. First of all, these manual procedures are extremely time consuming (up to weeks for specific analysis) while, by running an automated algorithm, few minutes would be necessary. Such a time saving would allow biologists to focus on their actual research activity instead of engaging in low-level mechanical procedures - much more suitable for a computer.

Second, produced results were not accurate. Despite a biologist was more likely to reach state-of-the-art performances than an algorithm in recognizing structures, handcrafted solutions can have a lower accuracy when performed serially for a huge amount of images. Third, the outcome is operator-dependent and not univocal. Asked to accomplish the same task again, a human user likely would not have produced exactly identical results and different users would produce different outcomes, leading to a lack of standardization.

For these reasons, the need for an automated approach is evident. Since the missing step in the workflow implies to emulate choices of human experts, who had learnt how to distinguish structures by seeing examples and not following detailed low-level rules, deep learning was identified as solution.

1.3 Neural Networks

Neural Networks (NNs) are computational techniques able to emulate biologists' performance seen their promising results in applications where learning from data was required. They are especially useful in image analysis since they can be instructed with a pool of images along with the expected results to extrapolate information of interest completely autonomously. The more data are used to train the NN, the more complex the task to accomplish can be, and in a more accurate fashion. As a downside, the more data are used, the more time-consuming the training of the NN gets, requiring a proper trade-off between size of data pool and time consumption for labeling data with results and training.

The NN-based applications developed in this study were implemented both to help researchers in their current activities and to give them general tools applicable to future analyses of the same biological structures.

1.4 Scope of the Thesis

This study aimed at implementing automated tools (Fig. 3) that allow biologists to investigate and quantify more efficiently the progression of the FBR, whose results were based on manual proceedings combined with standard image processing software functions for parameter extraction. Deep learning was employed where explicitly-programmed algorithms failed to represent a solution, exploiting models capable to learn how to recognize cellular and subcellular features in a human-like fashion, in a supervised training framework. These data-driven solutions were then integrated into image processing routines where human intervention was previously required, as fully automated solutions. Three customized applications were developed, as described.

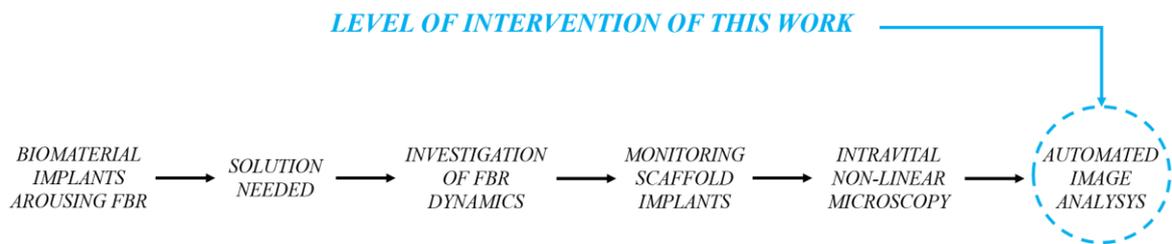


Figure 3: Level of intervention of this work.

1.4.1 Application 1: Quantification of Collagen Orientation

Collagen deposition, triggered by a foreign body, increases over time and forms a fibrotic capsule that can induce implant failure. In our model (porous PCL scaffolds), collagen bundles tended to dispose as perpendicular to or parallel to scaffold fibers with time, showing precise orientation alignments with progressive deposition. Thus, fibrous scarring and

encapsulation of the foreign body obey ordered mechanical dynamics: the initially thin and randomly ordered collagen matrix assumes well defined fiber orientations with the ongoing and severity of the response. In this work, we developed a system for quantifying collagen orientation relatively to scaffold framework by implementing an image processing tool for extracting collagen directionality, both as a spatial map and a histogram of pixel counts with tilting angle of attributed structures.

1.4.2 Application 2: Extraction of Fibroblast Parameters

Fibroblasts are the cell responsible for producing collagen and represent the most important component at the basis of the fibrotic response, hence understanding their biology, including shape, spatial distribution, reciprocal interplays and interactions with the surrounding environment is of fundamental importance. For these reasons, processing fibroblast images both for discerning entities and for extracting their parameters was required. In this work we developed a system to segment fibroblasts from the respective fibroblast-rich channel of input images and to measure their individual properties for preclinical analyses.

1.4.3 Application 3: Segmentation of Scaffold-Associated Cells

FBGCs are the key immune cells involved in the FBR, that besides adhering on the foreign materials to digest it, produce soluble mediators that induce the recruitment and activation of fibroblasts and foster fibrosis. Thus, monitoring their evolution over time is needed, also for testing potential therapeutic applications. In this work we implemented a system to automatically discern scaffold associated cells, interstitial cells and the remaining acellular background from the same image channel as input.

2. Materials and Methods

2.1 Experimental Data

The images employed in the proposed applications were acquired with modalities described in the Introduction.

2.2 Neural-Network-based Software

A Neural Network (NN) is a computational model inspired by human neurons' architecture and collective activity. Following an organized flow of information, each node receives numerical inputs from preceding nodes, processes them, and produces a unique numerical output, which feeds the following nodes within the hierarchical cascade (Fig. 4). Each node is dedicated to one purpose, can accept several inputs and can deliver the same output to several nodes. It operates by implementing a specific mathematical function defined over a domain whose dimension depends on the number of inputs the node receives. Considering the inputs as organized in a N-dimensional scalar vector $\vec{I} = (I_1, \dots, I_N)$, the node's characteristic function will map the vector such as $\mathfrak{R}^N \rightarrow \mathfrak{R}$.

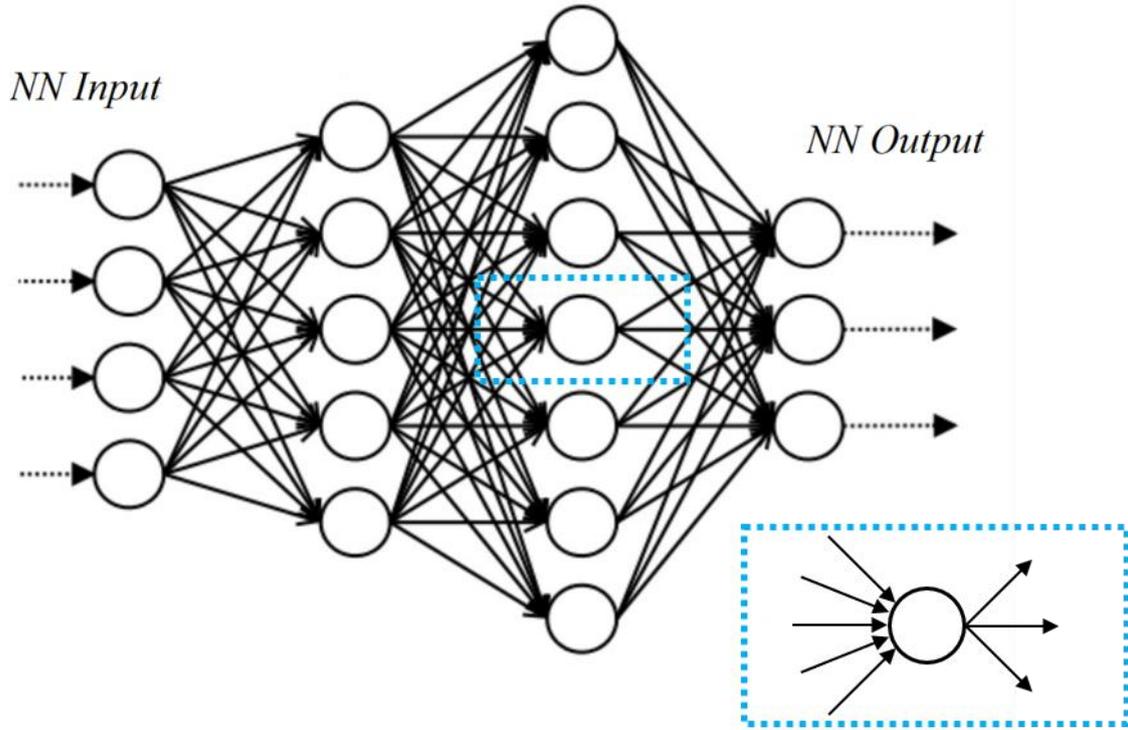


Figure 4: Example of NN architecture with single node magnification highlighted.

Each node resembles the mechanism of activation and firing of a neuron [22], meaning that an output is produced if and only if the proper input pattern is present in terms of intensity and sequence order. Accordingly, a node faces three sequential steps: i) weighted sum of the inputs, ii) offset addition, and iii) computation of the output.

The first step assigns a weight (w_i with $i = 1, \dots, N$) to each input on the base of their relevance for the current analysis, and computes the following weighted sum S :

$$S = \sum_{i=1}^N w_i * I_i \quad (1)$$

The second step adds an offset, commonly known as threshold (ξ), to the linear combination of Eq. (1), which is seen as an additional weight associated to a virtual input of value -1. The result is named action potential (P) and writes:

$$P = \left(\sum_{i=1}^N w_i * I_i \right) - \xi \quad (2)$$

The third step computes the output of the node, known as activation (A), by interpolating the value of P from Eq. (2) through an activation function chosen ad hoc for the application of interest, in formula:

$$A = f(P) = f \left(\left(\sum_{i=1}^N w_i * I_i \right) - \xi \right) \quad (3)$$

Through the described architecture, a node is more prone to yield high output values in presence of certain inputs rather than others, becoming specialized to privilege a certain class of stimuli which is determined by adjusting the weights ad hoc. This feature is extended to the whole network, allowing to guide the use of the NN towards an application rather than another. The NN structure is modular, i.e. the higher the number of nodes, the more complex the network is and the higher the complexity of detectable patterns can be. In summary, a NN is a function driven by peculiar coefficients set ad hoc (weights and threshold of each node) that turns a set of inputs into specific outputs.

A NN is organized in layers, each of them being the ensemble of nodes that process the same inputs in parallel. Considering feed-forward neural networks, single layer's inputs only come from preceding nodes, which are organized in layers as well.

Once chosen the nodes' architecture, the optimal weights' values are determined through a two-step procedure that sees a phase of: i) training and ii) validation.

(i) The training phase corresponds to the setup of the weights and it is carried out empirically. An error (or loss) function, which depends on the weights' value, is defined by comparing the NN and the desired output for a set of chosen training data. This target output is the gold-standard result reproduced by experts (e.g. biologists in this study). Such function is then minimized to retrieve the optimal weights for the proposed analysis (Fig. 5). Several iterative methods are available for such optimization problem and they are chosen based on empirical requirements. These are gradient-descent-based methods, which compute weight adjustments proportional to the gradient of the loss with respect to weights. Thus, loss gradient is propagated backwards from outputs to weights (backpropagation) to update them. Finally, the goodness of the training is evaluated via performance criterion chosen depending on the task (e.g. accuracy for classification, mean square error for regression, et cetera). The generalization capability of a NN is commensurate to the quantity and quality of data available for the training phase, which makes a network to be best trained with the most inclusive yet heterogeneous set of data as possible.

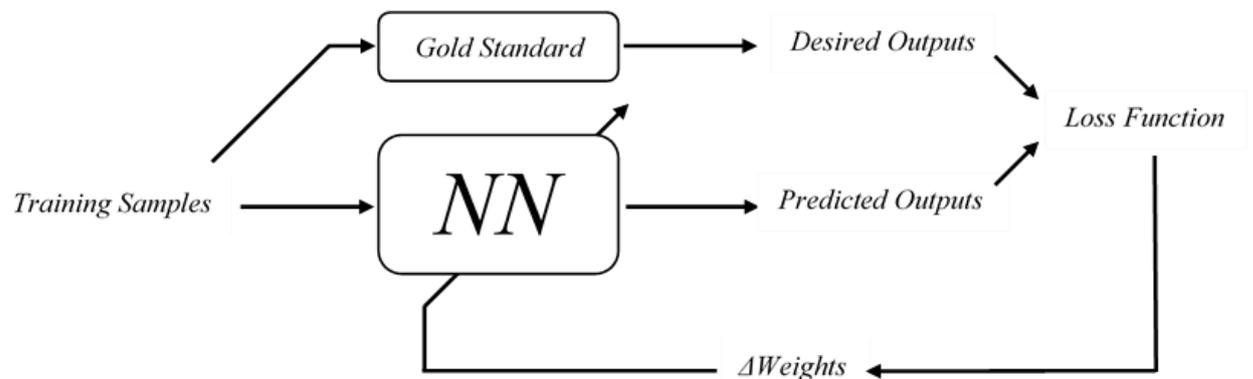


Figure 5: Training phase schematic.

(ii) The validation phase verifies the goodness of the weights chosen. It evaluates the discrepancy between the output of the NN, informed with the current weights, and the desired output for a new independent set of data, still “unseen” by the NN, that belongs to the same domain as the training one.

In this study, NNs were employed for semantic image segmentation, i.e. to recognize structures of interest within given images. The NN receives in input a 2D image and gives in output a 2D mask that associates a different class to each pixel. During training phase, each image is processed by the NN and the output compared with masks obtained by manual segmentation.

Finally, the versatility of NNs allowed us to apply the same framework here described to different tasks that will be further detailed in following sections. Among several architectures from semantic segmentation literature, the UNet was particularly suitable for our applications for three main reasons.

First, UNet requires fewer weights than other architectures to reach good performances. This translates into a small amount of training samples demanded (typically, the higher the weights to fit, the larger the training set needs to be) and a fast training (the less instances need to be analyzed, the less weights to be updated). Also, at inference time, the model requires limited time for predictions, since few operations are to be computed to yield an output due to the contained architecture complexity. Second, UNet was easily customizable to better fit the purposes of the current work in terms of number of encoding steps and layers-based structure of each block. Finally, it was originally designed to tackle segmentation of images similar to the ones analyzed in this work in terms of size and content [2].

UNet derives from Convolutional Neural Networks (CNNs), where features can be extracted directly from raw input images. Indeed, convolution allows to detect the presence of patterns in different points of an image as kernel-based filtering does. A peculiar structure is exploited, where all nodes within the same layer share the same weights and each node is associated to a different pixel and connected only to its neighboring pixels. This allows each layer to yield a convolution result, and the network to combine convolutions on more and more abstract features. UNet architecture is an extension of a Fully-Convolutional Neural network (FCN), which does not consider any fully-connected layer and associates features to corresponding locations. Indeed, to be able to reconstruct spatial localization of content, necessary for segmentation, only convolution can be employed.

UNet is sequentially composed of:

- an **encoding part** (FCN-based) that extracts features of interest (Fig. 6, in green)

- a **decoding part** that recreates the output mask (Fig. 6, in orange).

The same customized UNet variant, described in detail below, was employed in all tasks proposed.

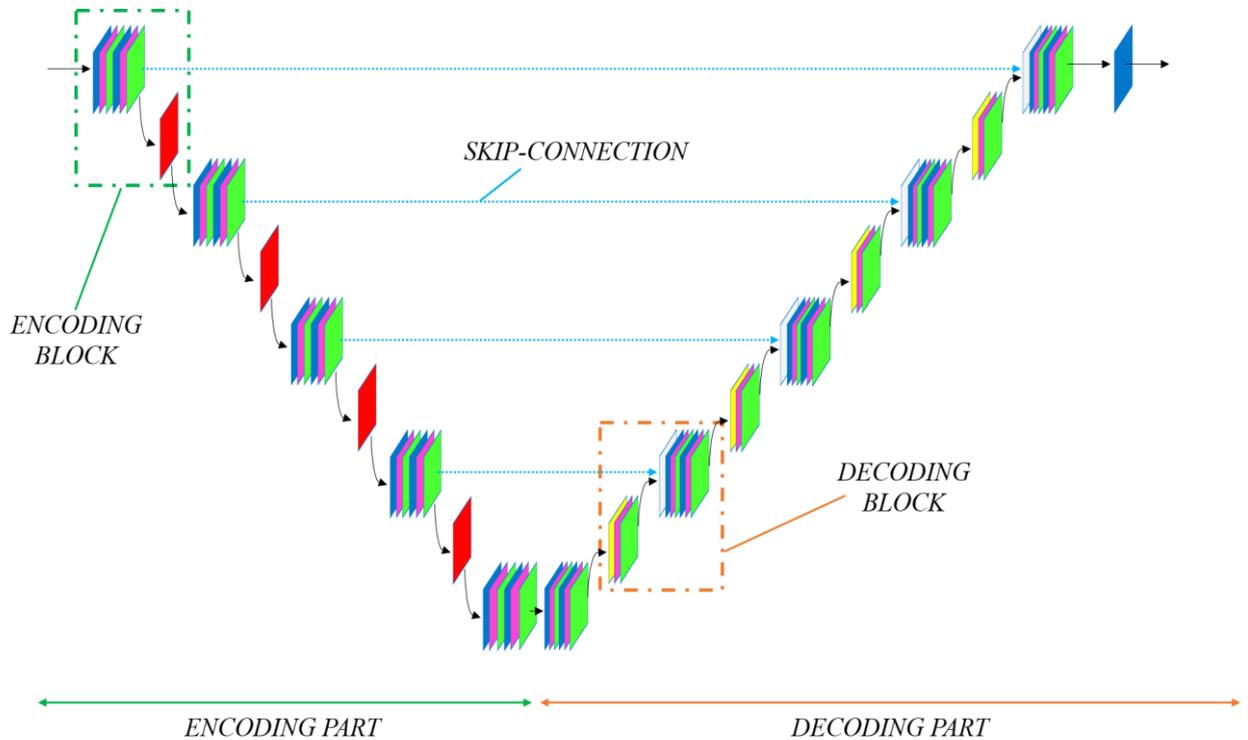


Figure 6: UNet architecture (characteristic “U” shape) schematic showing layers and functions (blue: convolutional layer, magenta: batch-normalization layer, green: rectified linear unit activation function, red: max-pooling layer, yellow: transposed-convolutional layer, gray: concatenation through skip-connections dotted in light blue). Encoding part and constituent block highlighted in green, decoding part and constituent block in orange.

• Encoding

The encoding part extracts relevant features from the raw input image through an FCN-based approach that deconvolutes the input patterns’ complexity. In this way, the input information is easily combined for a general and meaningful interpretation of the structures, which feeds the decoding block to reach the final pixel-wise classification.

The encoding part is made of a cascade of blocks of layers, where each block contains the same sequence of layers and accomplishes a single encoding step. Hence, the number of blocks determines the severity of the encoding process and the complexity of the NN.

However, a higher complexity (identifiable with the NN depth, i.e. the number of layers) leads to a higher number of weights, a slower training and inference, more memory storage required, worse loss shape (i.e. trainability), more training samples and manual labels required. Accordingly, the number of blocks was empirically set to 5 in all the applications to favor a proper trade-off, also considering that the available training set sizes were similar.

Each encoding block is composed of a rigorous sequence of layers and functions, see Fig. 7, outlined below and described in detail in dedicated sub-section.

- a) **Convolutional Layer (CL)** with identity activation function to process preceding features and extract abstract and complex ones
- b) **Batch-Normalization (BN) Layer** to normalize the input distributions
- c) **Rectified Linear Unit (ReLU) Activation Function** to make features always positive while avoiding gradient issues during backpropagation
- d) **CL** (same as point 1)
- e) **BN Layer** (same as point 2)
- f) **ReLU Activation Function** (same as point 3)
- g) **Max-Pooling Layer** to downsample features only to most relevant ones.

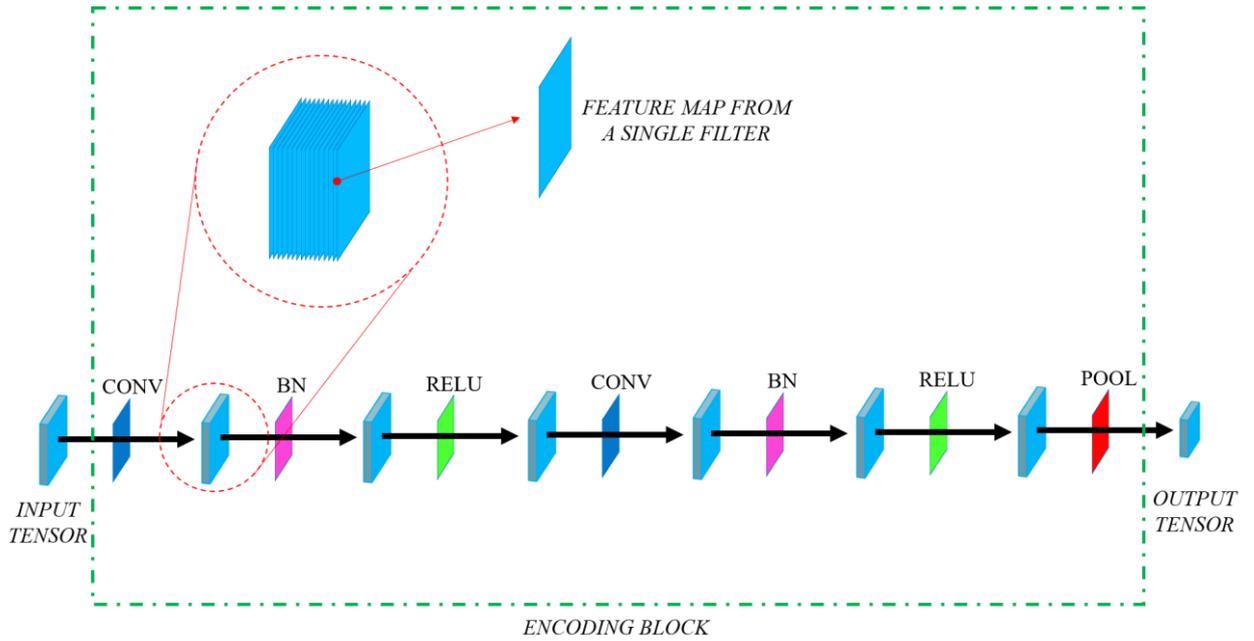


Figure 7: Encoding block schematic showing tensor (light blue) dimensions, considering a single sample, before and after each constituent layer (colors represented coherently with Fig. 6). Magnification highlighting several feature maps within the same tensor.

As a note, in the last encoding block, the max-pooling layer is omitted to avoid that a meaningless downsampling, followed by an upsampling, worsens resolution without adding any processing.

a) Convolutional Layer (CL)

The CL consists of several filters receiving the same input and producing an output map each (Fig. 7). The output of the layer is the parallel stacking of these maps along the additional dimension of filtering results (considered as channels). Each filter represents a convolution based on a different kernel, whose parameters, together with the threshold, are shared only by the nodes within the same filter.

We limited the number of filters, since the number of weights is proportional to them. When the input to the CL exhibits more channels or maps, convolution is accomplished on all channels at the same time for each neighborhood of pixels the kernel is multiplied with. The actual kernel size is thus extended along a third dimension in addition to the two spatial ones.

All kernel weights are learnt independently during training, to allow the CL to combine differently information from different feature maps.

The CLs within the same block were assigned the same number of filters, and after each encoding step it was doubled, maintaining the original block ratios [2]. Defining only the number of filters of the first block layers as an hyperparameter, its value was tuned depending on the application.

All remaining convolutional hyperparameters (i.e. parameters defining architectural choices) were shared among all layers and applications. Kernels were assigned a 3x3 size (9 coefficients per input feature map each) and convolutional strides were maintained of 1 pixel along both dimensions, as in [2]. Instead, zero-padding was performed to ensure that feature map size was maintained. Chosen kernels ensured final receptive fields large enough to capture significant patterns at a large scale. As $n = 10$ convolutions were performed throughout the encoding part, the last encoder layer covers receptive fields of size $1 + 2 \cdot n = 1 + 2 \cdot 10 = 21$ original pixels along each spatial dimension, quite comprehensive of dynamics.

Each CL was endowed with an identity activation function, which corresponds to a fictitious activation function. Accordingly, the output was simply the result of the convolution with the threshold offset added.

b) Batch-Normalization (BN) Layer

The BN Layer avoids the input distribution to a single layer to change considerably during the training phase while weights are continuously updated. Indeed, the inputs to each layer are affected by the parameters of the preceding layers, and even small weight changes can hugely impact deeper layers. BN increases the efficiency of training by preventing the layers to have to adapt to their new input distributions. In addition, it makes parameter initialization and learning rate magnitude less critical, as even a less careful optimization yields satisfactory results [23].

To batch-normalize a layer, first each k-th node activation ($x^{(k)}$) of the previous layer was normalized to zero mean and unit variance over the values assumed for different training instances, to reach the following:

$$\hat{x}^{(k)} = \frac{x^{(k)} - E[x^{(k)}]}{\sqrt{\text{var}[x^{(k)}]}} \quad (4)$$

Then, an additional linear transformation was applied to the whitened activation ($\hat{x}^{(k)}$) to shift and scale the distribution to more proper mean $\beta^{(k)}$ and variance $\gamma^{(k)}$ values:

$$y^{(k)} = \gamma^{(k)} \hat{x}^{(k)} + \beta^{(k)} \quad (5)$$

The parameters $\beta^{(k)}$ and $\gamma^{(k)}$ of such a transformation are seen as further weights and were adjusted during training. This allows the NN to learn how to shift and scale each activation distribution in an optimal way, preserving the capability of representing also a simple normalization when $\beta^{(k)} = 0$ and $\gamma^{(k)} = 1$.

At each iteration of weight update, the activations were normalized only over the values they assume for the samples in the current mini-batch (“batch”-normalization) and momentums in Eq. (4) were estimated only from the current mini-batch activations, avoiding the choice of an overly restrained mini-batch size.

At inference time, the statistics used for normalization in Eq. (4) were the same computed as a moving average over the mini-batches seen during training.

BN allowed for backpropagation of gradient independently for each mini-batch when updating weights. This normalization is indeed differentiable, and loss gradient with respect to BN weights can be computed as well for their update.

Unlike in a traditional fully-connected layer, the activations of a CL are not normalized independently. Rather, the whole output feature map of each layer filter is jointly normalized. Therefore, the first and second momentum in Eq. (4) were computed over the activations of all nodes of the considered filter map and over all mini-batch samples. Also, the parameters in Eq. (5) were shared among the feature map. This preserved the

convolutional principle that elements at different locations but in the same feature map are processed, and so normalized, by the same shared weights.

As herein applied, BN only added 2 weights per feature map. Thus, it is less critical in terms of added optimization parameters compared to CLs.

The normalization dependence on more training samples is advantageous to the generalization of the network, adding a slight, side regularization effect [23]. For this reason, the employment of other regularization techniques was reduced, as illustrated in following sections.

BN is particularly relevant if applied not directly to inputs but rather before the activation function, which is generally non-linear and whose codomain is considerably sensitive to the input distribution. Such normalization is more meaningful after the weighted summation of inputs as the resulting values exhibit a more gaussian distribution, which is not the case if mere inputs, with a more heterogeneous distribution, and being the direct output of other non-linearities, were to be normalized [23].

c) Rectified Linear Unit (ReLU) Activation Function

A ReLU activation function is defined as a linear identity function rectified to 0 for negative input P values, yielding an activation A_{RELU} :

$$A_{RELU} = \max(0, P) \quad (6)$$

Thus, it simply restores positive sign of input values, allowing following CLs to process non-negative feature maps.

Advantageously, when compared to traditional saturating activation functions (sigmoid, hyperbolic tangent, *et cetera*), ReLU allows for an effective computation of gradient of loss with respect to all weights. It is determinant for optimization methods based on gradient-descent, where weight adjustments are proportional to such gradient. This aspect is particularly determinant in deep architectures as the one here proposed. Indeed, when computing weight variations of a layer, the more the following layers, the more terms

proportional to the derivative of the used activation function are multiplied together. These products can lead to weight variations either close to 0 (phenomenon known as “vanishing gradient”) or approaching huge magnitudes (known as “exploding gradient”), depending on whether this derivative is < 1 or > 1 , respectively. ReLU solves this issue as $\frac{\partial A_{RELU}}{\partial P} = 1$ for all positive input values [24]. $\frac{\partial A_{RELU}}{\partial P} = 0$ for negative inputs implies that some nodes “die” during training, since they are not updated any longer because of its derivative. Despite not always desirable, this also means that ReLU introduces sparsity, for which only few useful node connections are maintained and computational burden decreases [24].

An additional advantage is that ReLU allows for fast computations.

ReLU, being the actual activation function suitable for CLs, was applied only after BN to take full advantage of such normalization, for aforementioned rationales. Thus, ReLU finalized the processing step of feature extraction initiated by the preceding CL and BN.

g) Max-Pooling Layer

The max-pooling layer downsamples the features of the layers yet maintaining highly representative ones, reducing computational complexity. Since features are organized in stacked maps, each resulting from a different convolutional filter, downsampling was applied along both original spatial dimensions but independently for each filter.

Max-pooling is a non-linear downsampling method that allows not to lose spatial correspondence. Original values are not sampled at regular rates as in simple downsampling, but rather each resulting value is obtained from a non-linear routine on the original features falling within a corresponding window of consecutive values, where fixed windows are evenly-spaced on the original map. The non-linear routine is the maximum value in such case, which is ideal to achieve translational invariance regarding features [25]. It allows to obtain the same outcome when maximum values are slightly translated on the feature space, provided that they remain under the respective windows.

The pooling hyperparameters were maintained as in the original work [2] for all the layers of the proposed applications, i.e. a window size of 2x2 and a stride of 2 along both directions.

- **Decoding**

Once the encoding part generates maps capturing few representative features of structures of interest, a mask that classifies each pixel as belonging to the most likely class can be reconstructed. The decoding part interprets such features and recreates target content at the original spatial resolution, i.e. on the input image space.

A drawback of the encoding process is a weakening of the spatial location, with resulting maps being affected by poor spatial resolution because of the pooling layers. To restore precise localization, the decoder is composed by a cascade of layer blocks alike the encoder (Fig. 8), with skip-connections between corresponding blocks (Fig. 6). Skip-connections make intermediate encoder outputs available for the decoding processing.

Every decoding block has a different corresponding encoding block it is connected to by skip-connections. Each decoding block has the same number of parallel feature maps, output size and hyperparameters, for layers of the same kind, as its corresponding one, as in the original work [2].

Spatial localization, despite at lower resolution, was maintained by the FCN-based encoder, allowing to remodel content by maintaining information localized. Hence, convolution is suitable again for processing features locally while preserving spatial correspondences. For this rationale, each block of the decoding cascade was based on CLs, BN and ReLU for feature processing (i.e. interpretation), similarly to encoder blocks but with max-pooling replaced by a peculiar upsampling to restore spatial resolution. Also, an additional, concatenation operation was added to allow for skip-connections. Each block is thus composed of the following rigorous sequence of layers/functions:

- a) **Transposed-Convolutional Layer** with identity activation function to increase spatial resolution by upsampling features in an optimizable way - seen as a CL
- b) **BN Layer**
- c) **ReLU Activation Function**
- d) **Concatenation through Skip-Connections** merging current features with same-resolution features from the corresponding encoder block to improve spatial details

- e) CL with identity activation function
- f) BN Layer
- g) ReLU Activation Function
- h) CL with identity activation function
- i) BN Layer
- j) ReLU Activation Function

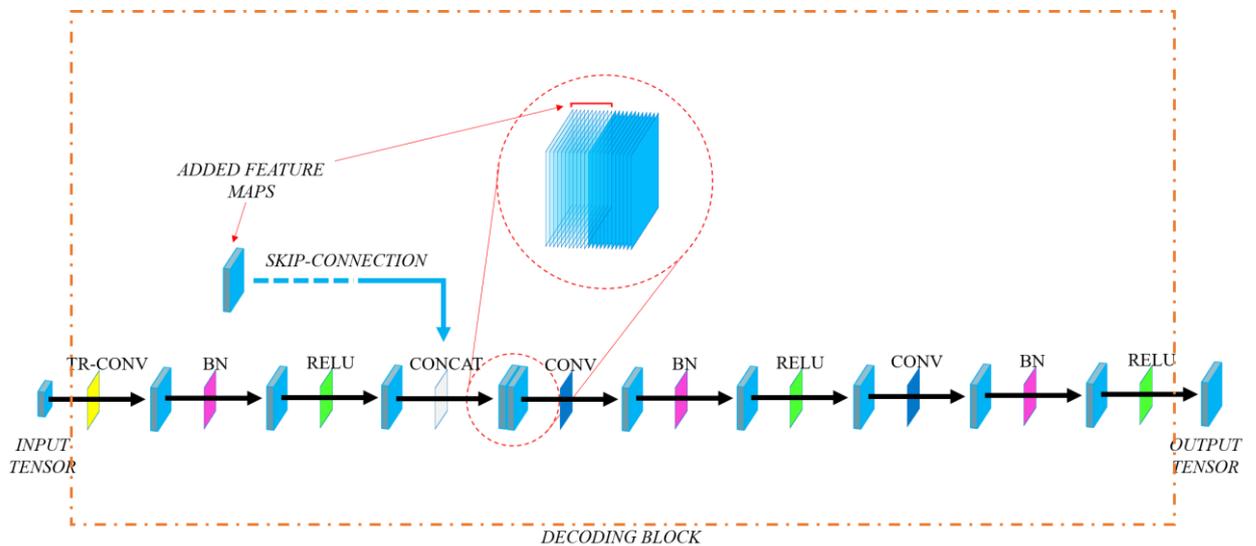


Figure 8: Decoding block schematic showing tensor (light blue) dimensions, considering a single sample, before and after each constituent layer (colors represented coherently with Fig. 6). Magnification highlighting feature maps added through skip-connection to the resulting concatenated tensor.

In the first decoding block, transposed-convolution, with respective BN and ReLU, is omitted to efficiently connect to the last encoding block, as mentioned above.

A final CL was added after the last decoding block. It was based on 1x1 kernels to process all final feature map values independently for each pixel. Having these final maps a spatial resolution comparable to the one of the original image, each pixel had its representative features for class assignment. The number of filters, and thus of values finally assigned to each pixel, of this final layer was chosen differently for the applications. For binary segmentation (only 2 classes), only 1 filter was necessary as each pixel only needed 1 class

probability value associated, seeing the other class probability as the complementary one. In such a case, the layer was assigned a sigmoid activation function to map probabilities from 0 to 1. For multiclass segmentation (more than 2 classes), the number of filters was set as the number of classes, and each pixel was associated a value of likelihood of belonging to each possible class. To turn these values into complementary probabilities, softmax normalization was performed independently for each pixel, after a fictitious identity activation function.

The detailed description of the new layers/functions follows. The layers shared with the encoding routine were not instead re-described.

a) Transposed-Convolutional Layer

This layer increases spatial resolution through a transposed-convolution, which combines upsampling with convolution. It allows the network to upsample optimally since it has learnable convolutional parameters adjusted together with other weights during training.

Transposed-convolution has the reverse effect of a simple convolution in terms of resulting size, increasing it still by maintaining spatial localization. Simple convolution can be seen as a multiplication of the whole input vector by a convolution matrix. This is a sparse matrix composed only by kernel weights, repeated where respective input pixels are to be multiplied for each row. Transposed-convolution can be seen also as a matrix multiplication [26]. The applied matrix has the transposed shape of the direct convolution matrix applied in the opposite direction. Kernel coefficients are not related to the ones of any direct convolution, rather it just regards sizes. Equivalently, for a more efficient implementation, transposed-convolution can be defined as an operation which: i) dilates the input by adding new zero-valued pixels in-between original ones and ii) performs a regular convolution on the dilated input, with given padding and stride. The applied dilation, padding and stride are designed to obtain the desired output dimension. The weights of the transposed-convolution are the ones of the kernel employed for this latter convolution. As in CLs, more convolutional filters are applied to the same input in parallel, and outputs are stacked together.

Hyperparameters were set to restore an output size equal to the respective encoder block one for each layer. Considering hyperparameters of the convolution to be transposed, kernels

were assigned a 2x2 shape, stride was set to 2 pixels along both spatial dimensions, no dilation was applied and padding was maintained of 1 pixel along both spatial dimensions. The same values were employed for all the layers of this kind throughout all the applications proposed.

d) Concatenation through Skip-Connections

This step concatenates new feature maps to the current tensor ones along the dimension representing different filter results. In this way, more feature maps are available for the convolutional processing of the following layers. This is advantageous as far as the added feature maps can convey richer spatial information for locating target structures. Therefore, they are always made available, by means of “skip-connections”, from the tensor feeding the max-pooling layer in the corresponding block within the encoding part (Fig. 6). This way, they have higher information about original image primitive details. This is a consequence of the lower number of encoding steps they were passed through. Hence, skip-connections are suited for precisely restoring spatial structures, yielding better localization accuracy, unreasonable only from the extremely low spatial resolution of the decoder output.

Thanks to the connection choices and to the chosen padding in CLs, the added feature maps have the same dimensions as the current ones and can be concatenated without cropping.

In addition, skip-connections improve training and contribute to a smooth loss function surface which is easy to minimize with standard gradient-descent based methods. In general, the deeper the network, the more chaotic the loss landscape without skip-connections [27]. This makes them beneficial for our case that relies on a deep architecture.

Lastly, such long skip-connections allow to better update weights in initial layers during training [28]. This is an issue for deep networks because loss gradient with respect to such weights can get numerically negligible, and assigned weight changes as well (as mentioned for ReLU issues above). As a result, weights are never actually adjusted. Instead, skip-connections mathematically relate outputs of final layers to outputs of initial layers directly. This way, gradient formulation varies, and computations yield much more significant weight changes even for former layers.

2.3 Applications

The same network architecture described above was employed in all the applications proposed where segmentation was required. For each of them, the NN was first trained following customized strategies depending on the task and then embedded within the image processing workflow, along with a proper data preprocessing pipeline and followed by explicitly-programmed (non-machine learning-based) post-processing algorithms to extract measurements of interest.

A detailed description for each application follows. Only the new methodology was described, while the above-mentioned one was not.

2.3.1 Application 1: Quantification of Collagen Orientation

The task sees two integrated subroutines. An ad hoc scaffold segmentation was combined with a customized directional filtering to detect orientations of scaffold and collagen fibrils from their boundaries. The scaffold was segmented both to detect its main orientation and to eliminate its interference with collagen boundaries. Finally, collagen directionalities were referred to scaffold main orientation to obtain relative collagen direction.

- **Scaffold Segmentation**

The scaffold was segmented from images showing its backbone. Due to the scaffold's jagged morphology and variable intensity and texture of the image, deep learning techniques better suited the segmentation routine than morphological operators. Indeed, the chosen method required a limited number of training samples (32 images) to reach satisfactory performances, along with a fast scaffold manual labeling.

Training and validation data were manually sampled, without an a-priori subdivision, from different available volumes to ensure a wide intra-diversity, as variability is beneficial to network generalization. Collected images were also chosen within the same volumes, but never as consecutive slices to ensure their content was completely different and they could be considered as independent samples. A total of 47 images was selected. To decrease the

computational burden, samples were resized from the original resolution of 1064x1064 pixels to 512x512 pixels by bicubic interpolation before feeding the segmentation network. In this way, the input size was lowered by a factor higher than 2 along both spatial dimensions, and overall by a factor higher than 4. A small computational complexity allows fast training and inference. In addition, samples resizing allowed for a reduced (by the same factor) tensor size when training with mini-batches and mostly with several feature maps, as RAM memory can easily be saturated. Moreover, this allowed for structure observation on a more global scale than with the original input resolution, as input spatial patterns were compressed by factors higher than 2. As a result, the receptive fields covered by the last encoding layer captured spatial patterns wider more than twice the original ones. Furthermore, with the input pixel sizes as powers of 2, an arbitrary number of encoding blocks could be added. Each block had indeed a pooling layer requiring an even input size along both spatial dimensions and halving sizes.

Each sample was individually preprocessed based on its own intensities to cope with the heterogeneity of originally acquired volumes. Depending on the volume images were sampled from, they could exhibit different intensity ranges and exposures, consequently requiring a pre-processing routine, sequentially including: i) **Normalization**, ii) **Gamma-correction**, and iii) **Histogram-stretching**.

(i) **Normalization** was performed by linearly re-mapping intensity values within [0; 1] range, so that image maximum was set to 1 and minimum to 0, as resulting from Fig. 9, (i). In this way, the input distribution to the network was the same for each sample, and extracted features were coherent. Additionally, normalization was necessary to apply gamma-correction, as its input intensity domain is [0;1].

(ii) **Gamma-correction** re-mapped intensities by expanding the range of lower intensity values and compressing the range of higher ones. Thus, it highlighted underexposed structures by increasing their contrast, as shown in Fig. 9, (ii). Given an input intensity (u), it was transformed into the corrected value $o = u^\Gamma$, where Γ was set to 0.5 for optimal results after empirical trials.

(iii) **Histogram stretching** was independently performed for each image to make its intensity values distributed over the whole available levels within the normalized range. The lowest and highest levels with histogram counts above respective thresholds of 50 and 25

pixels were re-mapped to 0 and 1, respectively. The other levels followed the linear stretching imposed by these extremes, and levels mapped to values outside the $[0; 1]$ range were saturated to extremities. In this way, all the available intensity values were exploited for machine perception, allowing to distinguish patterns with similar intensities. The overall result is exemplified in Fig. 9, (iii).

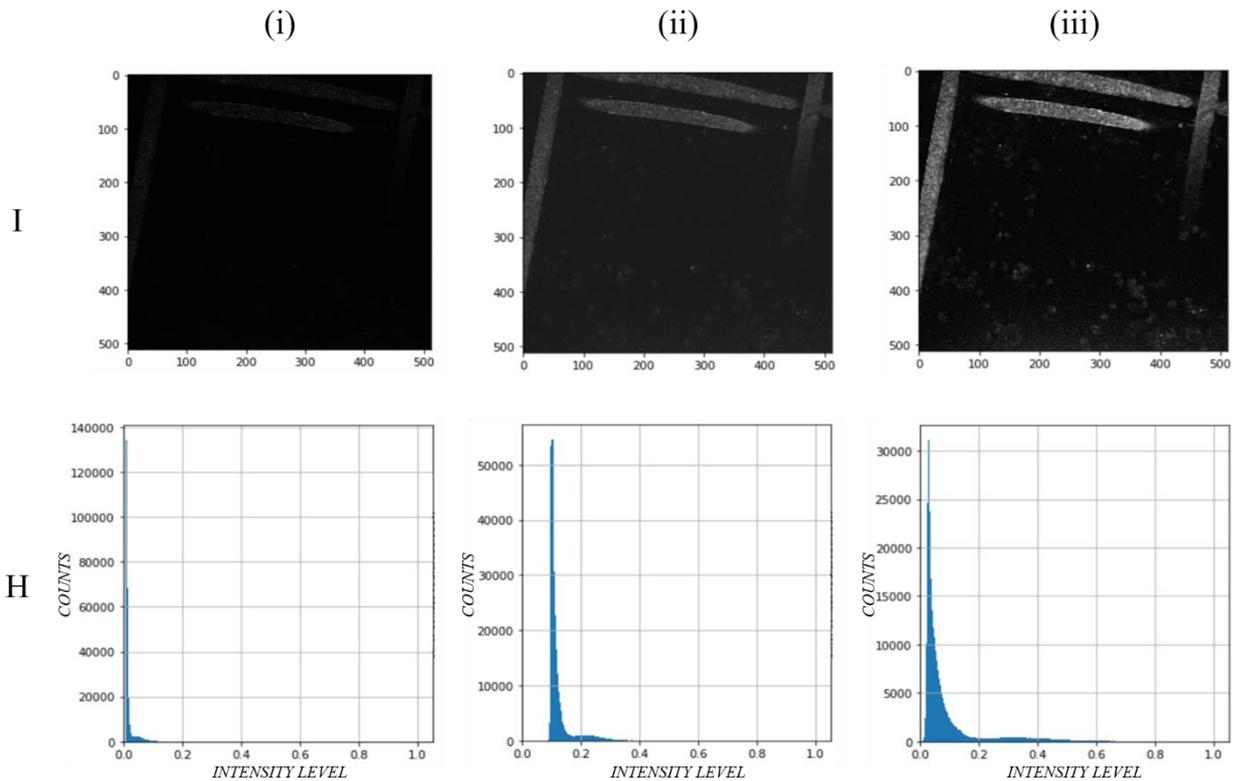


Figure 9: Subsequent results of preprocessing steps applied to a raw sample: both images (I) and histograms (H) shown after normalization (i), after gamma-correction (ii) and after histogram stretching (iii).

After resizing and preprocessing, samples were manually labeled to avoid label degradation with further interpolations and to highlight content for users, respectively. Then, samples, together with respective labels, were randomly split between training and validation sets. About 0.7 of the total amount of data was assigned to the training set (32 samples), while the remaining amount (15 samples) was collected to validate the network.

Both datasets were separately augmented, i.e. the number of samples (and labels) was increased by combining the original instances from the original datasets, to teach the network the desired invariance and robustness. The new samples (and labels) were created by

mirroring and/or rotating of $\frac{\pi}{2}$ multiples the original images in all combinations (total of 8) avoiding redundant results. Thus, the actual training and validation sizes were of $32 * 8 = 256$ samples and of $15 * 8 = 120$ samples, respectively. Correspondences in applied transformations were maintained with respective labels, while other transformations for data augmentation were not considered to maintain shape and scale of scaffolds as determinant for recognition. Data augmentation was performed after training and validation split to ensure that validation samples were independent of training ones, avoiding a biased validation process.

Finally, the network was trained on the augmented training set for 300 epochs. i.e. each training sample contributed to weight adjustments 300 times. Samples were propagated through the network in simultaneous mini-batches of size 8 for loss computation and weights' update. Such a low mini-batch size improves training stability and generalization performances [29], in addition to decreasing memory requirements. The mini-batch size could not be restrained further to allow for an efficient employment of batch-normalization, as mentioned above. The loss function was defined as a Dice loss and minimized by ADAM optimization algorithm, as detailed next.

The Dice loss is suitable for segmentation tasks as it represents the distance between segmentation result and labeled target. Advantageously, it evaluates the superposition between predicted and expected masks, giving a value which is absolute despite the amount of background pixels might vary. It is further defined as the discrepancy from the perfect superposition, for which all pixels belonging to foreground are assigned probability 1 and outside foreground mask are assigned 0. It modulates each pixel contribution by considering the network output probabilities before discretizing to most-likely class, yielding a differentiable loss whose gradient with respect to weights can be computed. For each pixel position (x), the predicted probability (\hat{y}) was compared to the expected probability (y). The employed loss was implemented by averaging this value for each image (S) of the N_S ones in the considered mini-batch:

$$L_{DICE} = \frac{1}{N_S} \sum_S \left(1 - \frac{2 \sum_{x \in S} y(x) \hat{y}(x)}{\sum_{x \in S} y(x) + \sum_{x \in S} \hat{y}(x)} \right) \quad (7)$$

The actually minimized loss (L_{TOT}) was eventually added a regularization term to take weights' magnitudes into account. This translated into a trade-off between minimizing Dice loss, which reflected the adherence to segmentation labels for training data, and constraining weights. This helped the network to generalize the same performances at inference time, i.e. to prevent overfitting the training samples. Reducing weights' absolute values - in particular the squared L2 norm ($\|\omega\|_2^2$) - prevents indeed weights from growing indefinitely. This is beneficial to training as weights are maintained close to 0 and cannot be adjusted to perfectly fit training examples but general features. The added regularization term ($\frac{\lambda}{2} \|\omega\|_2^2$) was multiplied by a $\lambda = 0.01$ factor to give it a relative importance of 1% with regard to correctly accomplishing segmentation of training examples.

$$L_{TOT} = L_{DICE} + \frac{\lambda}{2} \|\omega\|_2^2 \quad (8)$$

The introduced loss was iteratively minimized by ADAM optimization algorithm [30]. This optimizer relies on loss gradient with respect to weights for adjusting weights following a customized gradient descent. It is well suited for a highly dimensional parameter space, relying only on first order derivatives, and has little memory requirements. Within each iteration of the training process, each weight is assigned an independent learning rate (i.e. a factor regulating the magnitude of the performed adjustment). The base learning rate, for each weight, is adaptively modulated based on the first (mean) and second (uncentered variance) moments of the corresponding loss gradient term. It adapts actual learning rates considering moving averages of first and second moments of gradient terms over previous iterations, reducing actual learning rates when updates are noisy and incoherent. It also corrects the initial bias due to null gradient initialization. The algorithm was employed following the original [30] implementation.

Yet, the base learning rate was not maintained constant but varied with training iterations. A custom learning rate trend with epochs was defined, emulating the shape of a sigmoid to bound extreme values. Learning rate ranged from the initial value of 0.001 (l_{r0}) at the

beginning of training (at the 1st epoch $E_0 = 0$) to the final value of 0.0001 ($l_{r\infty}$) at the 200th epoch ($E_\infty = 200$), after which saturation was reached. This allowed steps in the weight domain to be a factor of 10 wider at the beginning of training, when far from a loss minimum point, than at the end of it, when small steps toward a near minimum were required. The learning rate trend (l_r) with epochs (E), shown in Fig. 10, was defined as:

$$l_r(E) = (l_{r0} - l_{r\infty}) \cdot \frac{1}{1 + e^{(s(E))}} + l_{r\infty} , \quad (9)$$

where the exponent

$$s(E) = \frac{E - \left(\frac{E_\infty - E_0}{2}\right)}{\frac{E_\infty - E_0}{10}} \quad (10)$$

stretched and translated the sigmoidal trend properly.

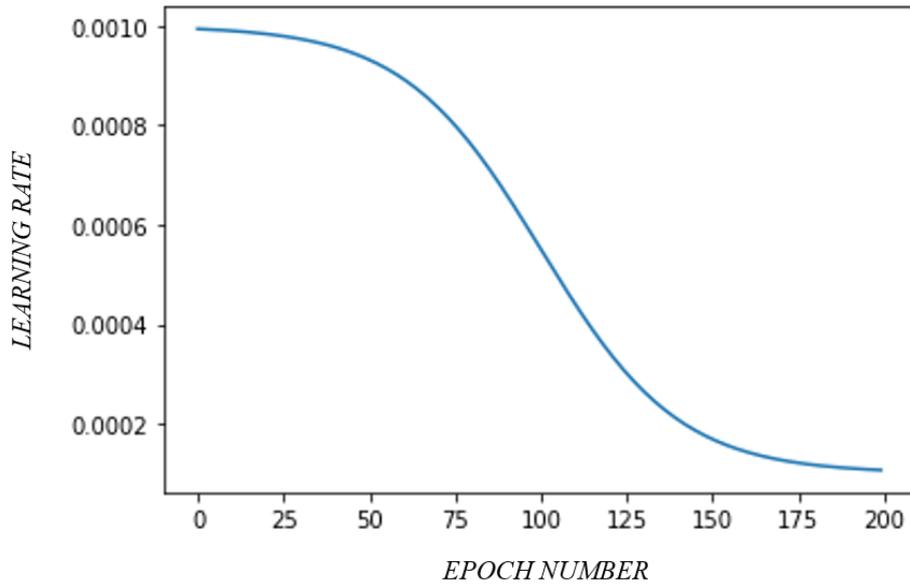


Figure 10: Learning rate trend with epochs.

The hyperparameter representing the number of filters in the first encoding block was set to 16 after empirical trials. The NN was implemented in Python through Keras deep learning library [31] running on TensorFlow library [32] as backend. Training was accelerated by GPU employment relying on Google Colaboratory [33] services.

Once trained, the network was validated on the validation samples. Provided satisfactory performances generalized outside the training dataset, the network was employed for inference. The same preprocessing applied to test cases. The output mask, representing pixel-wise classes, was re-interpolated to the original size of 1064x1064 pixels. Interpolation relied on nearest neighbor method, assigning each new pixel the class the closest pixel on the original mask belonged to. It avoided categorical classes turning into numerical variables, in order not to imply a meaningless hierarchical order among them. This upsampling was necessary to superimpose the scaffold mask to the original image channel showing collagen, required in the following steps of the processing workflow.

- **Collagen/Scaffold Orientation Detection**

A customized directional filtering was proposed to associate an orientation to each boundary pixel, as intensity edges were assumed to express shape and thus orientation of structures. The input image was filtered by different kernels, each one most sensitive to a particular boundary directionality. Then, each pixel was classified as oriented as the kernel associated with the highest value for that pixel, meaning that the boundary the pixel belonged to was oriented preferentially along its corresponding direction. Pixels associated with output values below a certain threshold for all kernels were not considered as boundaries and were assigned no orientation.

Thus, a different kernel was implemented for each orientation angle. Each kernel was defined as a matrix of coefficients giving a convolution output highest for input pixels belonging to edges with same kernel's orientation. The approach extended compass masks to a higher angular resolution, not limiting to the 8 compass directions: angles with an angular resolution of $\frac{\pi}{24}$ between 0 and 2π were employed as possible orientations. After filtering, edges were considered as oriented in the same direction for angles differed by π ,

despite opposite intensity variations occurred. The implemented approach was generalizable to any angular resolution when extending kernel dimensions, but a trade-off between kernel size and boundary localization exhibited. The higher the kernel size, the higher angular resolutions could be successfully adapted, but the coarser boundaries were reconstructed.

Each kernel was defined as a 15x15 matrix, and its coefficients were first assigned the respective distances (in pixels) between their position in the matrix and the line passing through the center of the kernel and with the angular coefficient corresponding to angle the kernel was tuned to (Fig. 11, A). Next, each kernel coefficient was re-mapped so that values between 0 and 1 were maintained the same, values between 1 and 2 were linearly re-mapped respectively between 1 and 0, and values above 2 were set to 0 (Fig. 11, B). Then, coefficients of each kernel on the clockwise side of the respective central line were multiplied by -1 (Fig. 11, C). Afterwards, in each kernel only the central coefficients within a radius of 6 pixels were maintained, while the other values were set to 0 (Fig. 11, D). Lastly, all the positive coefficients were normalized so as to sum up to 1 independently for each kernel, and all the negative coefficients of each kernel were normalized to always sum up to -1 (Fig. 11, E). This was fundamental to avoid some filters yielded higher outputs than other ones because their positive and negative coefficients had prevalent sums. As a result, each kernel was tuned to yield highest outputs for boundaries oriented as the respective linear edge its coefficients represented. Also, the coefficients in each kernel summed up to 0, which was desirable to maintain the mean value of the filtered images.

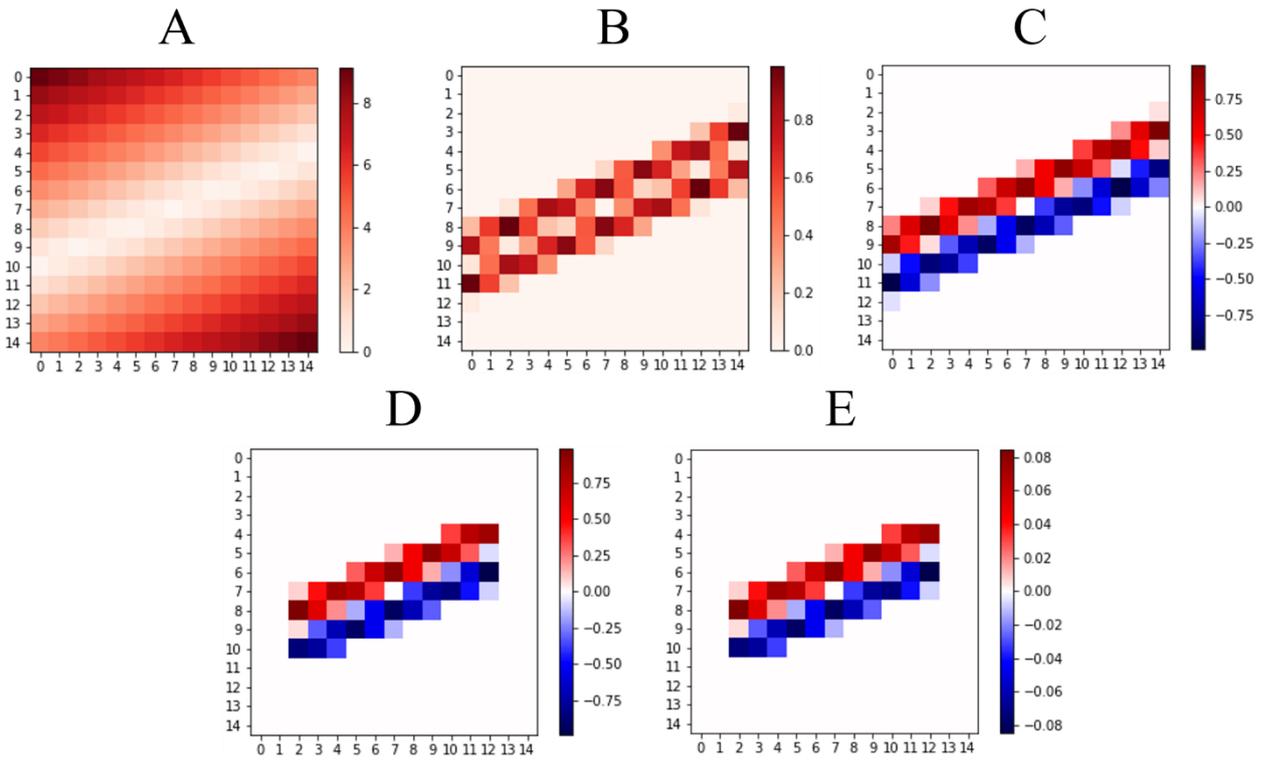


Figure 11: Kernel creation steps exemplified for an orientation of 22.5° (counterclockwise). Coefficients cumulatively shown after: being assigned distance from central oriented line (A), ramp modulation and truncation of values (B), negative assignment of clockwise side values (C), truncation outside central radius (D) and normalization of both positive and negative values to unitary sum modules (E).

The directionality was estimated both for scaffold mask and for collagen boundaries. Small objects were removed from the predicted scaffold mask by applying binary opening, an algorithm which sequentially applies binary erosion first and then binary dilation. A 20x20 spherical structuring element was employed to smooth boundaries and round indented borders of inaccurate predictions. In this way, isolated pixel clusters whose diameters were lower than the structuring element one were erased, while the scaffold mask shape was homogenized.

Then, the filtered mask was expanded by applying 15 iterations of binary dilation, with a 3x3 spherical structuring element to preserve boundary smoothness. In this way, the dilated mask ensured to cover scaffold borders completely, necessary to avoid that its boundaries were detected as collagen ones later.

The resulting mask was skeletonized to reduce the scaffold structure to its skeleton segments. This allowed to better exploit the defined kernels for extracting its predominant orientation, as they were suitable for detecting directionality of lines.

Hence, the skeletonized scaffold mask was filtered by each kernel, yielding a different resulting image each time. Each pixel of the mask was considered a boundary if the maximum among the values it assumed over all the resulting images was above an arbitrary threshold of 0.5, and it was assigned the orientation of the kernel that produced such a maximum. After these assignments, a histogram of orientations was built for pixels considered as boundaries. Scaffold predominant orientation was computed as the one maximizing histogram counts and pixels on the image's outermost frame were not counted to discard artifacts.

Collagen orientation was detected similarly. First, the input collagen channel image was preprocessed as mentioned above for segmentation, i.e. normalized, gamma-corrected and histogram-stretched, to highlight content. Then, the image was filtered as done for scaffold mask, obtaining an image where each pixel was classified as belonging to boundary (if its maximum filtered value was above 0.075) or not and, in case it was, associated its predominant orientation. As the collagen channel contained both collagen fibril and scaffold, a histogram of orientations was built by only considering pixels falling outside the dilated scaffold mask in order not to associate boundaries generated by scaffold artifacts to collagen ones. Also, pixels were discarded from the histogram counts if they belonged to clusters with an area lower than 15 pixels to maintain only actual fibril boundaries and discard artifacts. The main absolute orientation of collagen was estimated as the one of the kernel that generated most counts among the pixels of collagen boundaries. Finally, the resulting histogram and the main orientation of collagen were referred to scaffold predominant orientation.

2.3.2 Application 2: Extraction of Fibroblast Parameters

Our customized UNet was trained to segment a mask individuating fibroblasts with particular attention to touching cells. Instances were individuated based on connectivity, and parameters of interest were extracted for each of them.

Fibroblasts-rich images were employed as input. Both training and validation data were selected, interpolated, preprocessed and augmented as done for scaffold segmentation (Application 1), obtaining 200 training and 88 validation samples. Labels were manually created following an intensity threshold of 0.25 where borders of fading structures were hard to discern. Transformations alike the ones used in Application 1 were employed for data augmentation preventing fibroblasts' shape and size from being distorted.

We chose a NN to segment fibroblasts' structures as refined clusters of pixels and to properly separate them. A NN-based approach was preferred to a simple thresholding, whose results were too coarse and inaccurate, and also to morphological operators that could not ensure a proper separation of touching fibroblasts.

A particular attention was dedicated to instructing the network to correctly separate close or adjacent fibroblasts through an accurate definition of their borders. The latter were optimized by defining a customized loss function, which was minimized while adjusting the weights. As suggested in [2], a map representing the importance of pixels on touching boundaries was exploited to define the loss value for each sample.

Therefore, after manual labeling, each sample was associated with a map containing values to weight each pixel based on its importance for a correct segmentation of touching cells. The procedure aimed at avoiding that near cells were connected if they resulted as separated by background. Hence, background pixels were assigned boundary weights higher than 0, being relevant to avoid wrong connections, while pixels belonging to fibroblasts were attributed null factors. Each background pixel was assigned a value dependent on its distance to its two nearest cells, so that the more the pixel was in-between touching cells, the more it had relevance for a precise segmentation and had to be correctly classified. This value was computed as a gaussian of the sum of the two shortest distances between the considered pixel and the separate objects in the image shown by manual labels. In this way, the maximal amplitude reached for pixels touching both the two nearest cells was bounded. Thus, each pixel position (x) was assigned a boundary weight $w_B(x)$:

$$w_B(x) = \begin{cases} 0, & x \notin \text{background} \\ ae^{-\frac{(d_1(x)+d_2(x))^2}{2\sigma^2}}, & x \in \text{background} \end{cases} \quad (11)$$

where $d_1(x)$ and $d_2(x)$ represent the Euclidean distance between the pixel position x and the nearest and the second nearest object (i.e. fibroblast), respectively. The amplitude (a) and the standard deviation (σ) of the gaussian were respectively set to 100 (adimensional), after empirical trials, and to 5 pixels, according to the original work [2] value. Objects within a labeled example were distinguished based on connectivity, as explained later for clustering fibroblasts from network output. “class importance” factors were defined for balancing the two classes. Background importance (i_B) was defined as the mean value of $(w_B(x) + 1)$ assumed by background pixels over all the training set. Similarly, foreground importance (i_F) was defined as the mean value of $(w_B(x) + 1)$ for foreground pixels over all the training set. In this way, $i_F = 1$ by definition, allowing the foreground offset to be unitary and to multiply pixels on the foreground by their respective weights without effects. As a result, $i_B = 1$. Hence, previous boundary weights ($w_B(x)$) were corrected by adding this different offset depending on the importance of the class each pixel belonged to, yielding class-boundary weights $w_{C-B}(x)$:

$$w_{C-B}(x) = \begin{cases} w_B(x) + i_F = 1, & x \notin \text{background} \\ w_B(x) + i_B, & x \in \text{background} \end{cases} \quad (12)$$

Class-boundary weight maps were associated to samples during the data augmentation process and experienced their same respective transformations to be available for every sample in the final datasets. Maps were precomputed and not derived while computing the loss function because of the time-consuming procedure, not affordable in terms of multi-epoch training.

Binary Cross-Entropy (BCE) was chosen as loss function to allow all pixels to be weighted by their respective class-boundary relevance, unlike Dice loss. The traditional BCE loss (L_{BCE}) is defined for a single pixel (x) as the deviation between the predicted probability

($\hat{y} \in [0; 1]$) and the true probability ($y \in \{0; 1\}$) of belonging to a considered class (foreground):

$$L_{BCE}(x) = -y \log(\hat{y}) - (1 - y) \log(1 - \hat{y}) \quad (13)$$

The employed loss (L_{w-BCE}) was thus a pixel-weighted version of the traditional BCE function. The pixel-wise loss values were multiplied by the respective weights and averaged on all pixels (x) of all the N_S samples (S) in the considered mini-batch:

$$L_{w-BCE} = \frac{1}{N_S} \sum_S \frac{\sum_{x \in S} w_{C-B}(x) L_{BCE}(x)}{\sum_{x \in S} w_{C-B}(x)} \quad (14)$$

Training was performed by minimizing this customized loss function by ADAM optimizer as described for Application 1, still after adding a term for L2 regularization. Given the large size of input images, labels, and associated maps, only necessary variables were maintained in working memory. Training was performed on-the-fly, i.e. by loading only the current mini-batch into RAM. A custom data-generator was implemented for retrieving each sample with the associated class-boundary map and label.

After training, performances were assessed on the validation set and, when satisfactory, the network was employed for inference. Otherwise, the training was re-started by differently tuning hyperparameters and iterative arrangements. The final values employed were a mini-batch size of 8, 300 training epochs and 16 filters in the first encoding block. Same considerations about mini-batch size held as in Application 1.

Once the output mask was upsampled to the original pixel resolution as in Application 1, single instances (i.e. fibroblasts) were separated. Foreground pixels connected together were assigned to the same entity, whereas the ones not grouped together were defined as belonging to different objects. In this way, different clusters were assembled, each representing a fibroblast. The connectivity between two pixels was defined as whether they were 8-

connected, i.e. they exhibited touching either edges or corners. Clusters with an area lower than 200 pixels were not considered objects and discarded.

Finally, the number of instances was counted and following parameters were extracted for each: area, centroid coordinates and main orientation. The main orientation of a fibroblast was estimated by the same procedure employed for Application 1, based on the predominant directionality of its boundaries. Also, inter-fibroblast distances were computed between the centroids of each couple of fibroblasts.

2.3.3 Application 3: Segmentation of Scaffold-Associated Cells

Our UNet variant was trained to accomplish a multiclass segmentation end-to-end directly from input images, classifying each pixel as belonging to one of the following classes:

- **scaffold-associated cells**, i.e. giant cells and other cellular bodies adhering to scaffold surface
- **interstitial cells**, i.e. other biological structures not belonging to the first class
- **background**, i.e. all the remaining acellular content (both scaffold framework and non-visible substrate).

Samples were selected as described for Application 1 and, for the same issues, each image was normalized relatively to its maximum and minimum values, finally ranging within [0; 1] intensity domain. Labels were manually created sample-by-sample by exploiting the full resolution of original images. Samples and labels were indeed resized only after training and validation split, when augmenting data, following the same procedures as Application 1. Data augmentation relied on the same transformations because shape and scale were relevant to classify cells. Respecting the same proportions, 0.7 of the available data was assigned to the training set (200 images), while the remaining 88 were used for validation.

The loss function was derived by extending the Dice loss employed in Application 1. Dice loss is employed in binary segmentation, where background is not relevant, referring to foreground. Hence, the Dice loss term representing predicted mask goodness was computed for each class (C), by considering all remaining classes as background and evaluating the pixel-wise predicted (\hat{y}_C) and expected (y_C) probability values associated to the

corresponding class. Next, these terms were weighted by the respective class balance coefficients (α_c), to avoid classes with more pixels were favored in the overall loss. These coefficients were computed by counting the occurrences of all pixels of respective classes in the training set, and normalized so as to sum to 1, representing weighted averaging coefficients. Weighted terms of each class were indeed finally summed, and the result was subtracted to 1, generalizing the loss to the multiclass case. The resulting value referred to a single image, and was averaged over all N_S mini-batch samples (S) to yield the actually optimized multiclass class-weighted Dice loss L_{M-DICE} :

$$L_{M-DICE} = \frac{1}{N_S} \sum_S \left(1 - \sum_C \alpha_C \left(\frac{2 \sum_{x \in S} y_C(x) \hat{y}_C(x)}{\sum_{x \in S} y_C(x) + \sum_{x \in S} \hat{y}_C(x)} \right) \right) \quad (15)$$

Such a loss extension was derived by similar approaches [34], [35], [36].

The training phase was performed on-the-fly, minimizing the regularized loss by means of ADAM as for Application 2. 300 epochs, with mini-batches of size 4, were run, and the number of filters in the first encoding block was tuned to 16. The validation phase followed with the same modalities appreciated in Application 1 and 2.

Once the mask was upsampled to the original resolution, still via nearest-neighbor interpolation, the area of each class was provided, yielding a tool capable of quantifying the amounts of the cellular species in the input image.

3. Results and Discussion

Here the results are presented and discussed separately for each application.

3.1 Application 1: Quantification of Collagen Orientation

Collagen is a key parameter in the formation of the fibrotic capsule. In order to quantify its orientation relatively to scaffold framework, an image processing tool was implemented that extracted collagen directionality, generating a spatial map and a histogram of pixel counts with tilting angle of attributed structures. The input consisted of a second harmonic generation image displaying both scaffold and collagen fibers, and a third harmonic generation image displaying clearly only scaffold, as reported in Section 1.1.3.

First, the scaffold framework was extracted from the input third harmonic generation images.

Semantic segmentation performances were evaluated in terms of Intersection over Union (IoU). The latter gives an absolute metric value as the ratio between intersection area and union area of predicted and target foreground masks (Fig. 12), so that it is independent of the ratios between whole image size and mask sizes, unlike pixel-wise accuracy. The choice of IoU overcomes the need of a correction for class balance in metrics comparing prediction errors pixel by pixel when background is predominant relatively to foreground mask size, which is the case of the current application.

The IoU value was averaged either over validation samples, always required to validate the network on an independent set before employment, or over training samples, when necessary to evaluate the network capability to properly fit training instances or to assess overfitting did not occur by comparison with generalization performances. Thus, training IoU was assessed only when validation IoU was not sufficiently satisfactory, to investigate training issues.

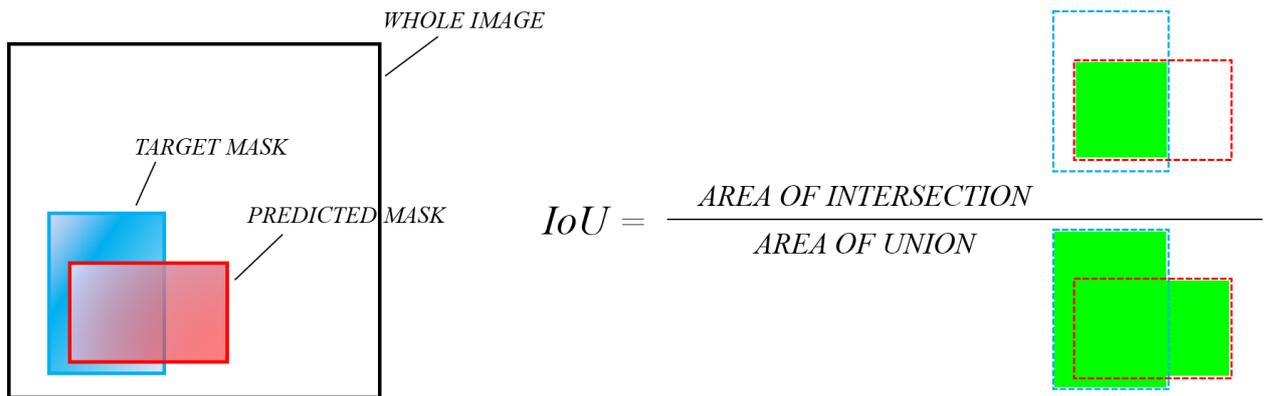


Figure 12: Intersection over Union (IoU) meaning exemplification.

Being a binary segmentation task, scaffold segmentation by our UNet variant was evaluated exclusively in terms of scaffold IoU. Evaluating and averaging the metric value over the available validation samples, unbiased from the training ones as mentioned above, yielded an overall mean IoU (mIoU) of 0.7955. The preprocessing pipeline applied to input images proved to be fundamental, as during empirical trials different preprocessing methods (normalization only, normalization only and gamma-correction) consistently failed to reach such metric value.

The discrepancy from the ideal IoU value of 1.0 highlighted that the task was meaningfully accomplished but with some limitations. The average IoU over training samples was 0.8927. Despite higher than mIoU on the validation set, it indicated the lack of capability to perfectly fit training examples. Therefore, the validation metric gap from the ideal value was not attributable to overfitting occurrence, instead the model reasonably generalized over validation data. The model complexity in terms of weights and layer blocks could not be increased to better fit training samples, as the training size (limited to the available instances) required to be increased as well to prevent overfitting. Longer training did not represent a solution either, as the loss already exhibited a saturating trend with the iterated number of epochs, shown in Fig. 13.

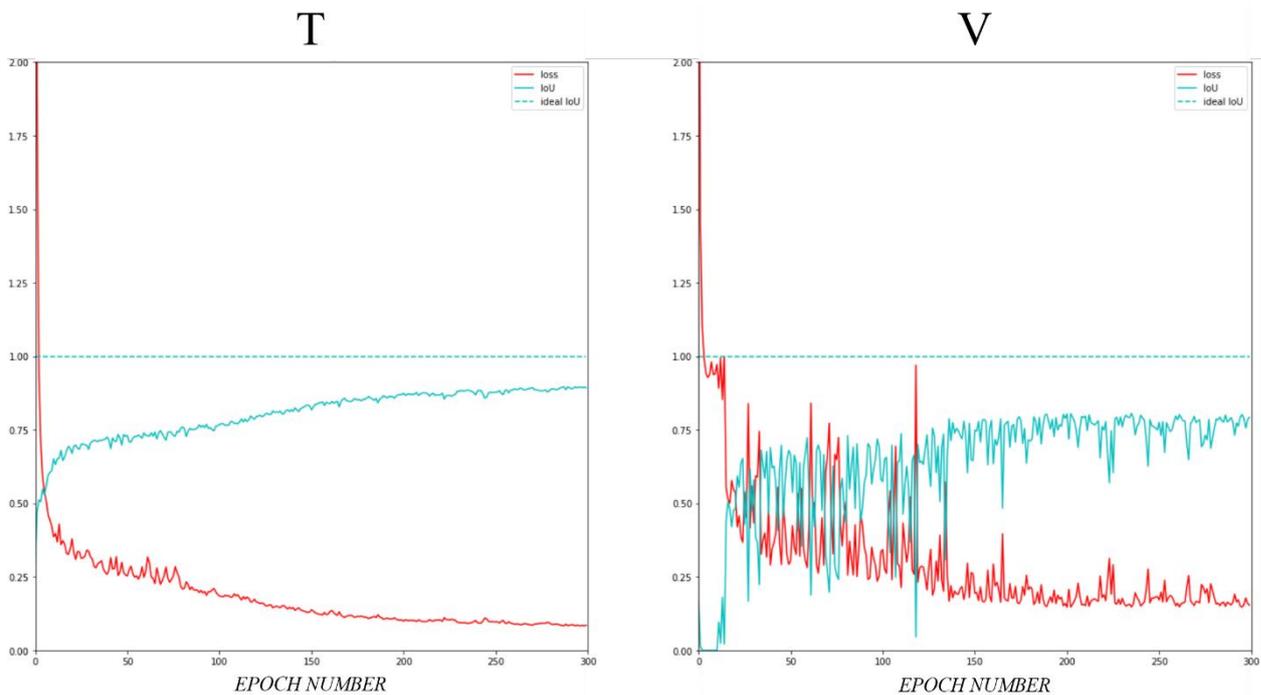


Figure 13: Loss (red) and mIoU metric (blue) trends over training samples (T) and validation samples (V) with training epochs. Ideal mIoU limit value dotted in blue.

Fig. 14 reports 4 examples (rows A, B, C, D) of validation samples (left column) with the associated predicted segmentation masks (middle column) compared to the expected manual labels (right column). The quantity of data did not represent a drawback given the meaningful validation results with the employed training size, as the instances reported in Fig. 14 exemplify. The quality of data ensured visually satisfying results, with the primary cause of the observed validation performance gap since some images represented incomplete, blurred, underexposed, coarse or fragmentary scaffold frameworks. Indeed, good quality validation samples were perfectly fit, as shown in Fig. 14, A, while others with such deficiencies were not as precisely predicted, as in Fig. 14, B, C and D.

The lack of consistency in assigning precise borders when manually labeling scaffold images further complicated the task accomplishment, as human hand is subject to intrinsic variability in reproducing perfectly coherent results among different samples, without ensuring a steadfast analysis. This affected the capability of the NN to learn a steady representation of the task. This was particularly determinant when scaffold structure gradually faded within the same image, either because underexposed or because its

framework prolonged slantwise through adjacent slices. As a result, outermost segments of scaffold structures were manually reconstructed with arbitrary borders and, after training, always predicted with a different length than expected, as exemplified in Fig. 14, B, C and D.

Additional examples are reported in Fig. 15.

Results were considered as satisfactory by biologists after inspection. Remarkably, segmentation outcomes sufficed both to correctly extract scaffold predominant orientation and to fully cover scaffold internal boundaries on collagen channel images when extracting fibril borders.

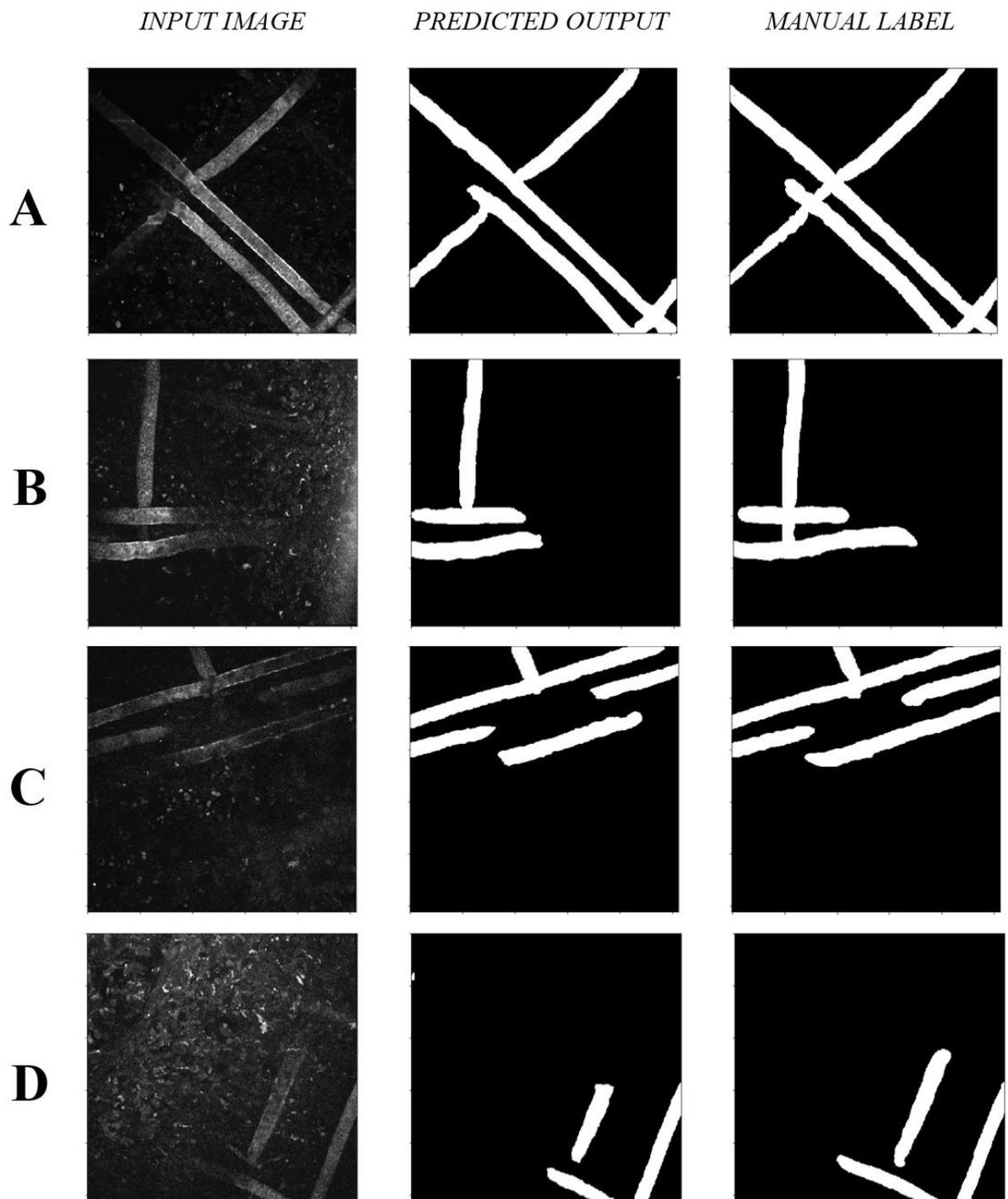


Figure 14: Scaffold segmentation: preprocessed input image (left), predicted output mask (middle) and target manual label (right) shown for 4 different validation instances (A, B, C and D) - white showing foreground in masks.

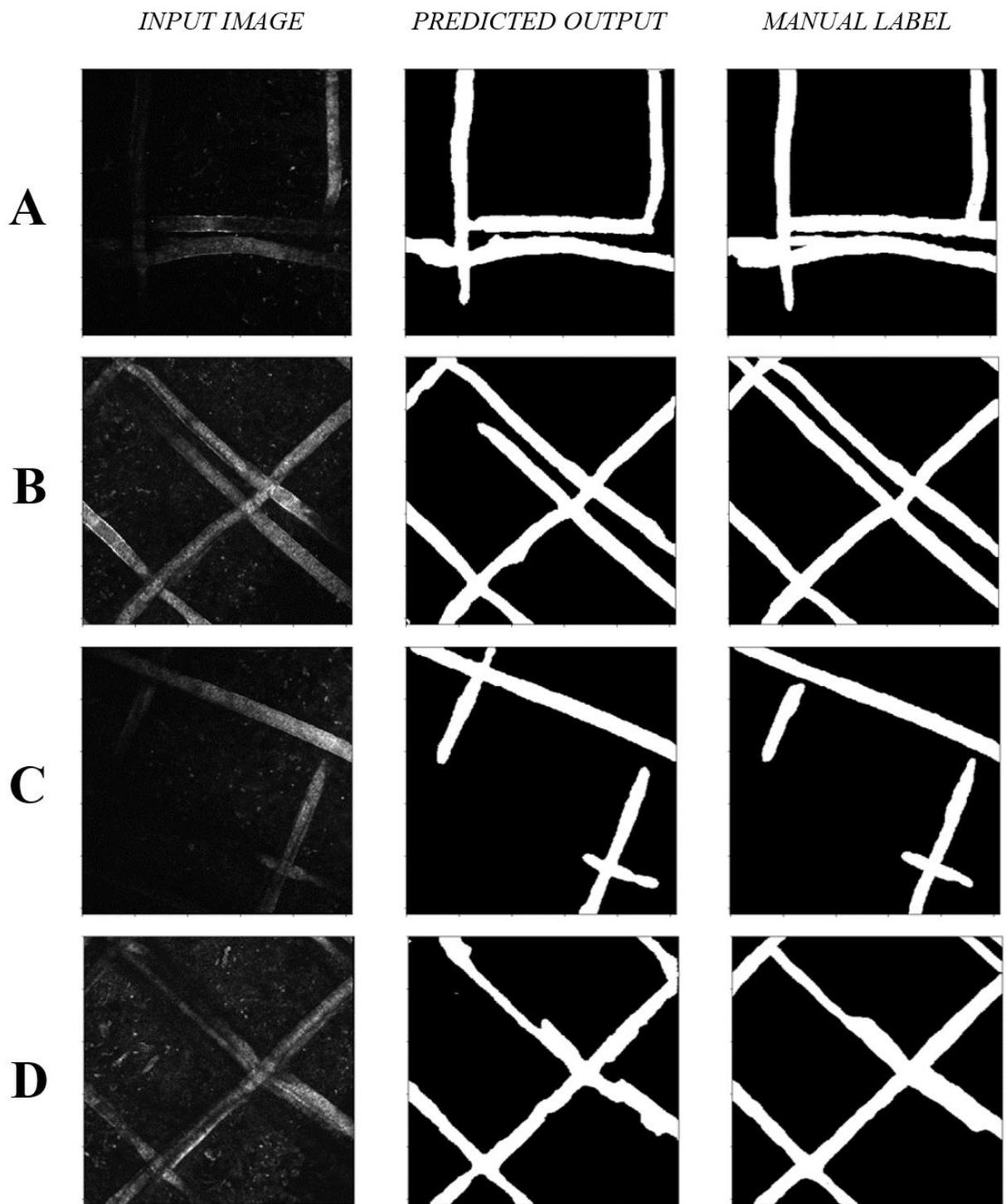


Figure 15: Scaffold segmentation: preprocessed input image (left), predicted output mask (middle) and target manual label (right) shown for 4 different validation instances (A, B, C and D) - white showing foreground in masks.

Once isolated the scaffold's framework, the directionality of scaffold and collagen fibrils was evaluated. Directional filtering results for both collagen and scaffold orientation quantification could not be compared to a gold-standard. The employed technique relied solely on the kernel definitions and produced according outcomes. As previously mentioned, the higher the pixel size of kernels, the finer the angular resolution was set. However, this also implies a coarser reconstruction of the boundaries (lower localization capability). Since the focus of the application was devoted to a rough quantification of collagen orientation to be related to the scaffold's one, a better angular resolution was favored to a better spatial resolution in order to obtain kernels with relevant distinctions among the edges they represented.

Kernel size was set to 15x15 pixels to reach an angular resolution of $\frac{\pi}{24}$. While the spatial map showing pixel-wise orientations could not localize borders precisely, the histogram of overall orientations yielded meaningful counts to quantify each directionality. Fig. 16, Fig. 17, Fig. 18 and Fig. 19 show the outcomes of the workflow stages, each for a different case. Each figure shows the input collagen image (A), the extracted spatial map of orientations of collagen boundaries in the whole image (B) with the associated colormap alongside, the input scaffold image (C), the extracted spatial map of orientations of the skeletonized scaffold mask (D) referring to the same colormap, the extracted spatial map of orientations of collagen boundaries in the subregion of interest where collagen directionalities are quantified (E), the resulting histogram of absolute collagen orientations in the subregion (F), the resulting histogram of absolute scaffold orientations (G) and the collagen histogram of (F) shifted relatively to scaffold predominant orientation (H). Thus, despite the defined kernels exhibited such a trade-off between localization precision and angular resolution, results were retained satisfactory for quantifying main scaffold and collagen orientations in terms of histograms. The overall computational time for a single inference, accounting both for NN prediction and orientation extraction, was of the order of seconds, allowing for a significant time saving compared to manual procedures.

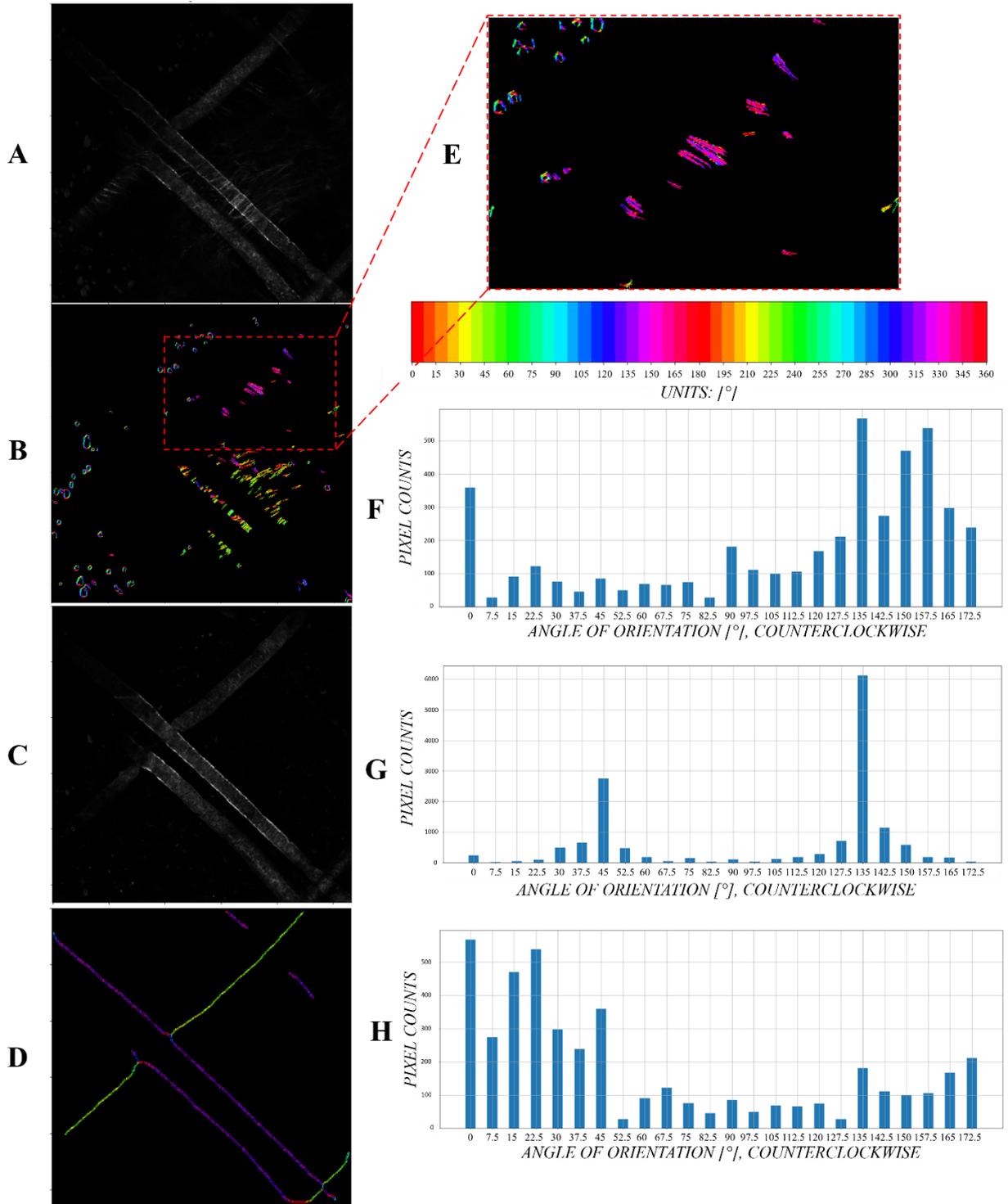


Figure 16: Exemplification of stages for extracting collagen orientations relatively to scaffold for a given case test: raw input collagen image (A), extracted spatial map of orientations of collagen boundaries in the whole image (B) and in the subregion of interest where collagen directionalities are quantified (E) with associated colormap, resulting histogram of absolute collagen orientations in the subregion (F), raw input scaffold image (C), extracted spatial map of orientations of skeletonized scaffold mask (D), resulting histogram of absolute scaffold orientations (G) and collagen histogram of F shifted relative to scaffold predominant orientation (H).

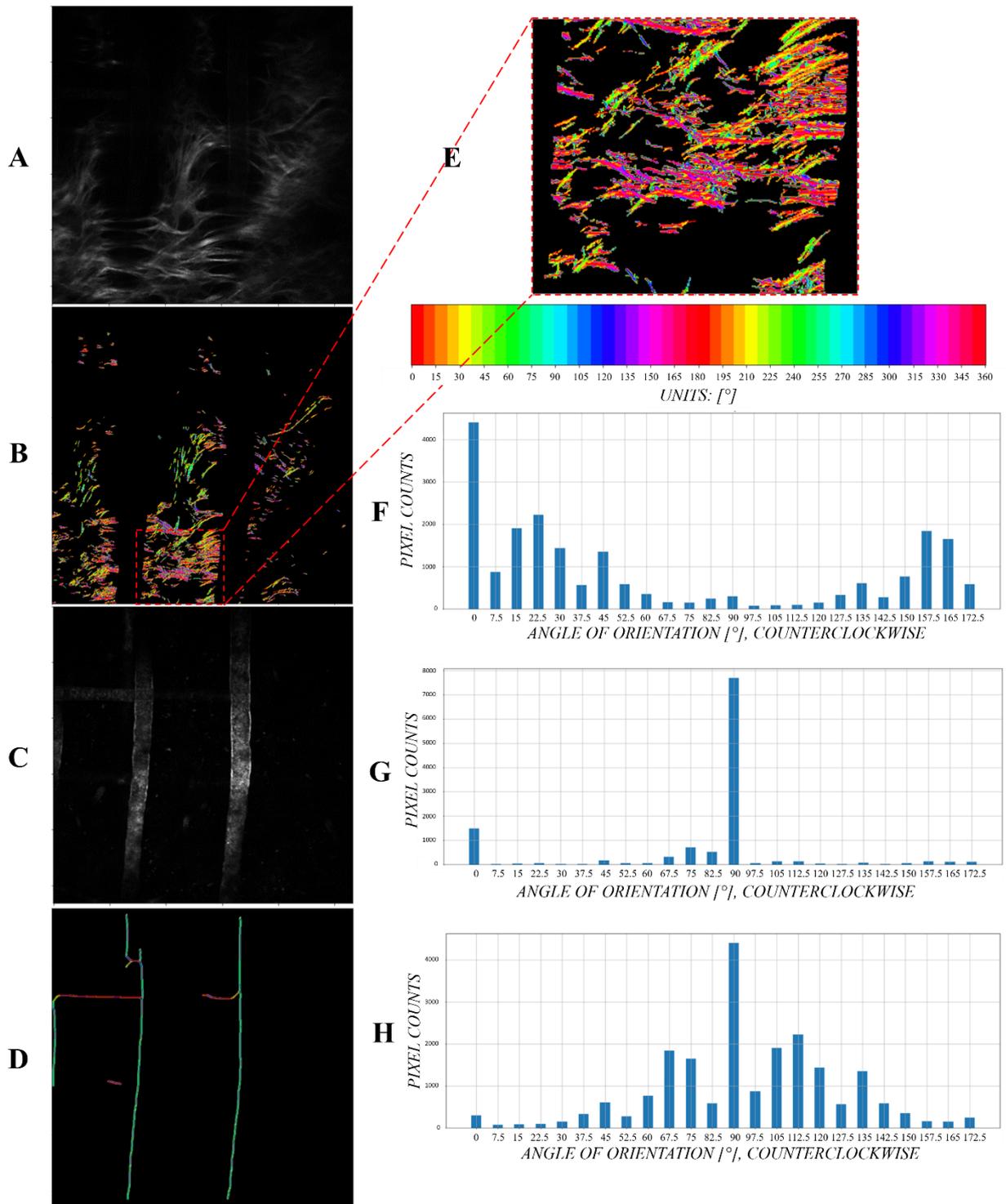


Figure 17: Exemplification of stages for extracting collagen orientations relatively to scaffold for a given case test: raw input collagen image (A), extracted spatial map of orientations of collagen boundaries in the whole image (B) and in the subregion of interest where collagen directionalities are quantified (E) with associated colormap, resulting histogram of absolute collagen orientations in the subregion (F), raw input scaffold image (C), extracted spatial map of orientations of skeletonized scaffold mask (D), resulting histogram of absolute scaffold orientations (G) and collagen histogram of F shifted relatively to scaffold predominant orientation (H).

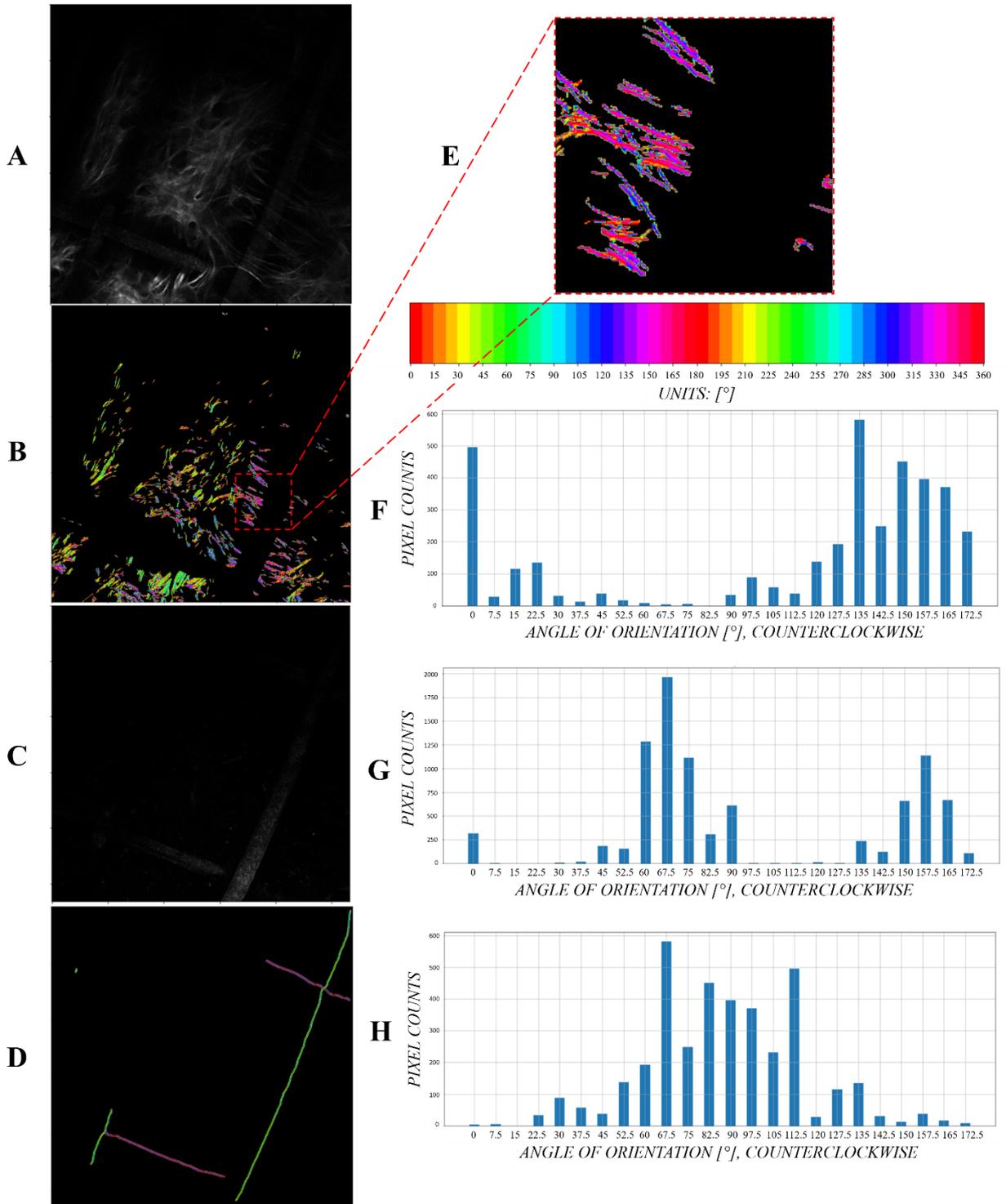


Figure 18: : Exemplification of stages for extracting collagen orientations relatively to scaffold for a given case test: raw input collagen image (A), extracted spatial map of orientations of collagen boundaries in the whole image (B) and in the subregion of interest where collagen directionalities are quantified (E) with associated colormap, resulting histogram of absolute collagen orientations in the subregion (F), raw input scaffold image (C), extracted spatial map of orientations of skeletonized scaffold mask (D), resulting histogram of absolute scaffold orientations (G) and collagen histogram of F shifted relative to scaffold predominant orientation (H).

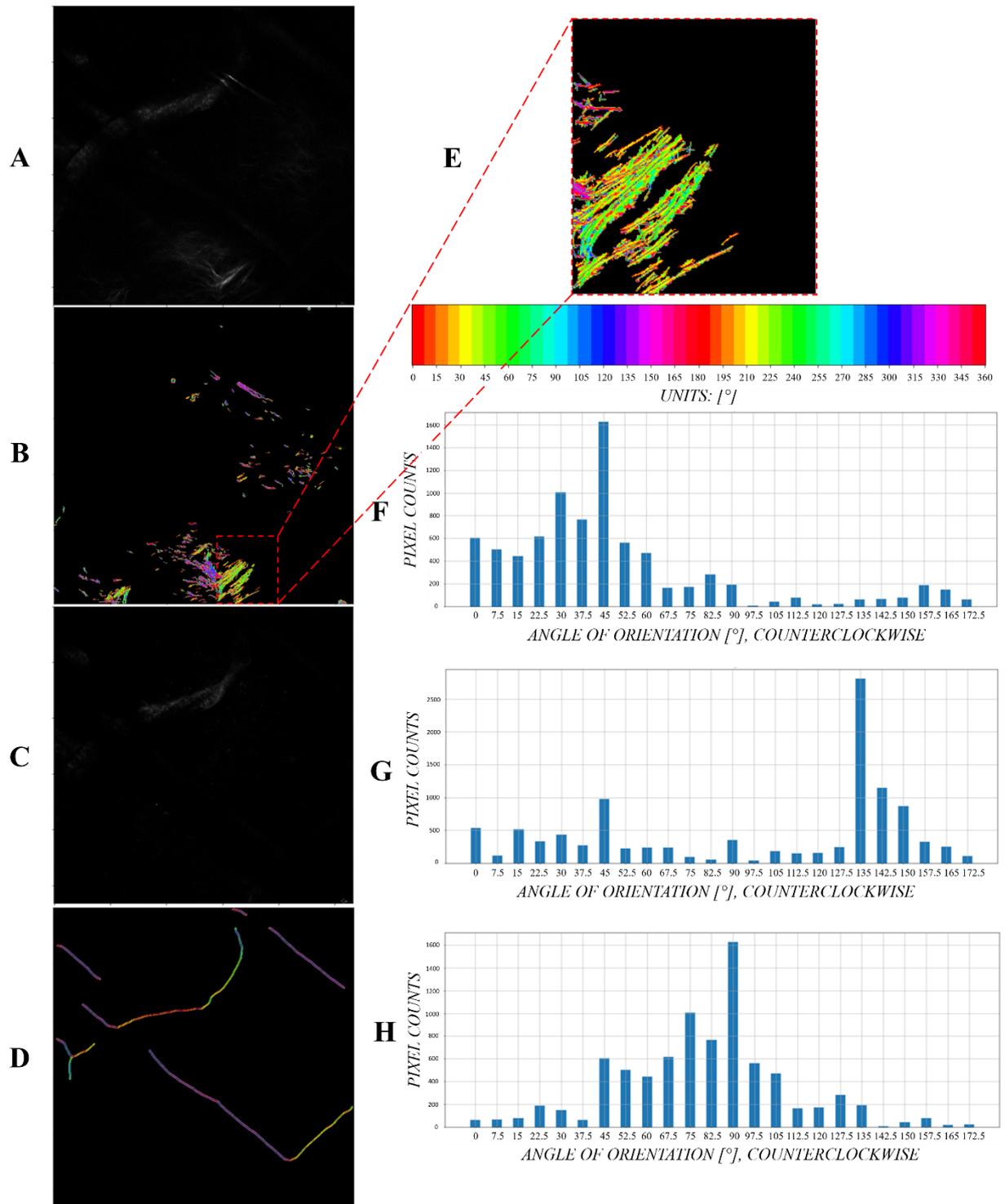


Figure 19: Exemplification of stages for extracting collagen orientations relatively to scaffold for a given case test: raw input collagen image (A), extracted spatial map of orientations of collagen boundaries in the whole image (B) and in the subregion of interest where collagen directionalities are quantified (E) with associated colormap, resulting histogram of absolute collagen orientations in the subregion (F), raw input scaffold image (C), extracted spatial map of orientations of skeletonized scaffold mask (D), resulting histogram of absolute scaffold orientations (G) and collagen histogram of F shifted relative to scaffold predominant orientation (H).

3.2 Application 2: Extraction of Fibroblast Parameters

Fibroblasts are one of the key cell types involved in the formation of the fibrotic capsule. In order to understand their biology, including shape, spatial distribution, reciprocal interplays and interactions with the surrounding environment, fibroblast images were processed and these parameters extracted. Thus, a system to segment fibroblasts from the respective fibroblast-rich channel of input images was developed and their individual properties measured.

Fig. 20 shows the results of the fibroblast segmentation procedure for 5 validation case studies (A, B, C, D, E), each reported in a separate row where, left to right, we can appreciate: the input image, the predicted semantic segmentation mask, the target manual label and the image of the final objects clustered based on 8-pixel connectivity from the predicted mask, where different colors denote separate fibroblasts. Circles on the predicted segmentation masks allow to note how contained the predicted structures were in order not to merge close, different fibroblasts together.

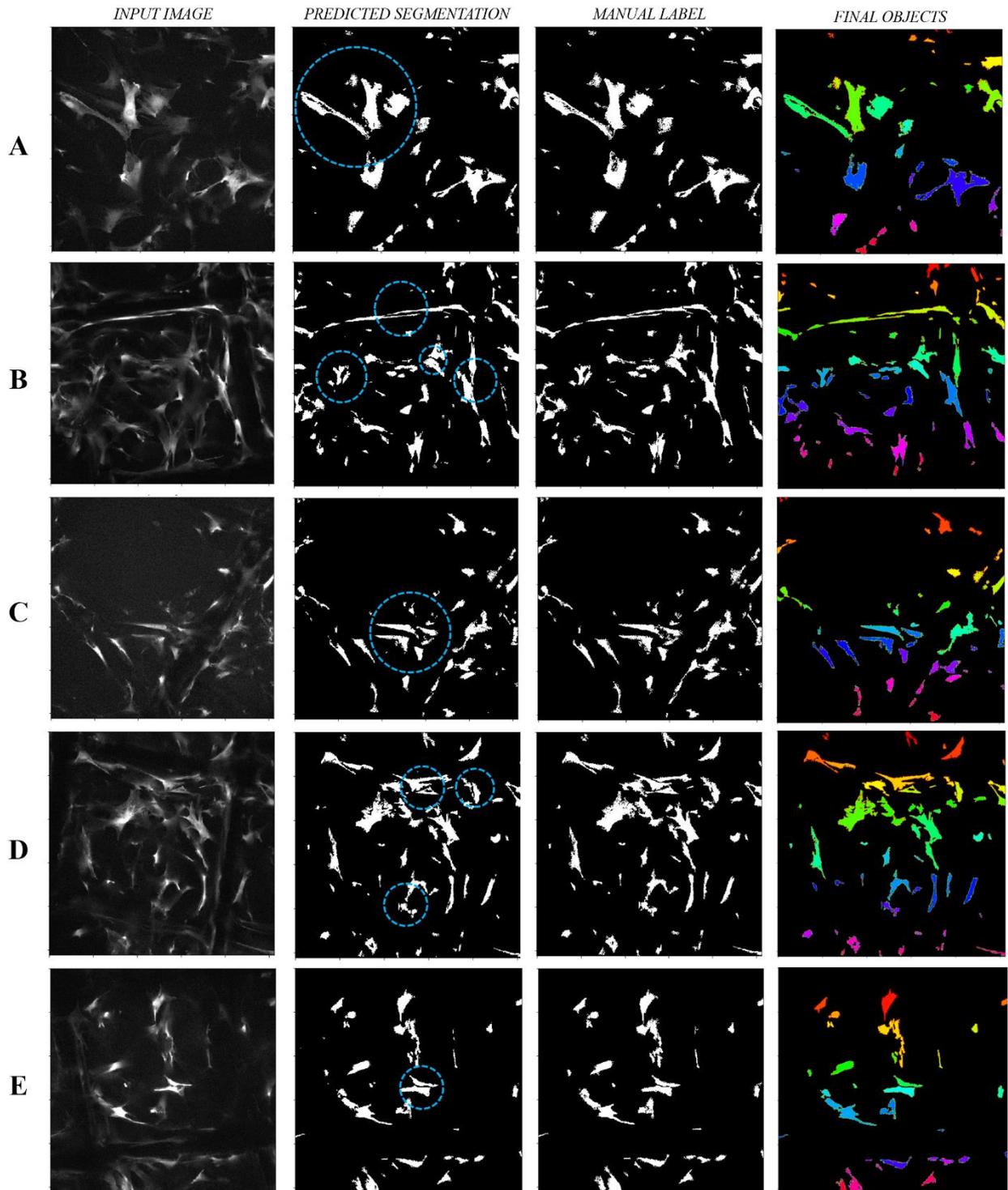


Figure 20: From left to right: input image, predicted segmentation mask (blue dotted circles highlighting touching cells correctly separated), target manual label and final predicted objects clustered based on 8-pixel connectivity (different colors for different objects) for different (A, B, C, D, E) validation instances.

For the same reasons as section 3.1, validation performances were initially quantified in terms of IoU. The segmentation network achieved a mIoU of 0.8869 when evaluated over validation samples.

The importance of BN layers for a successful training outcome results clear from Fig. 21. By adding BN to the network, the loss (Dice, at the time) value exhibited an almost monotonically decreasing trend during training, while the same network without BN, although initially faster, was not able to further minimize the loss after initial epochs.

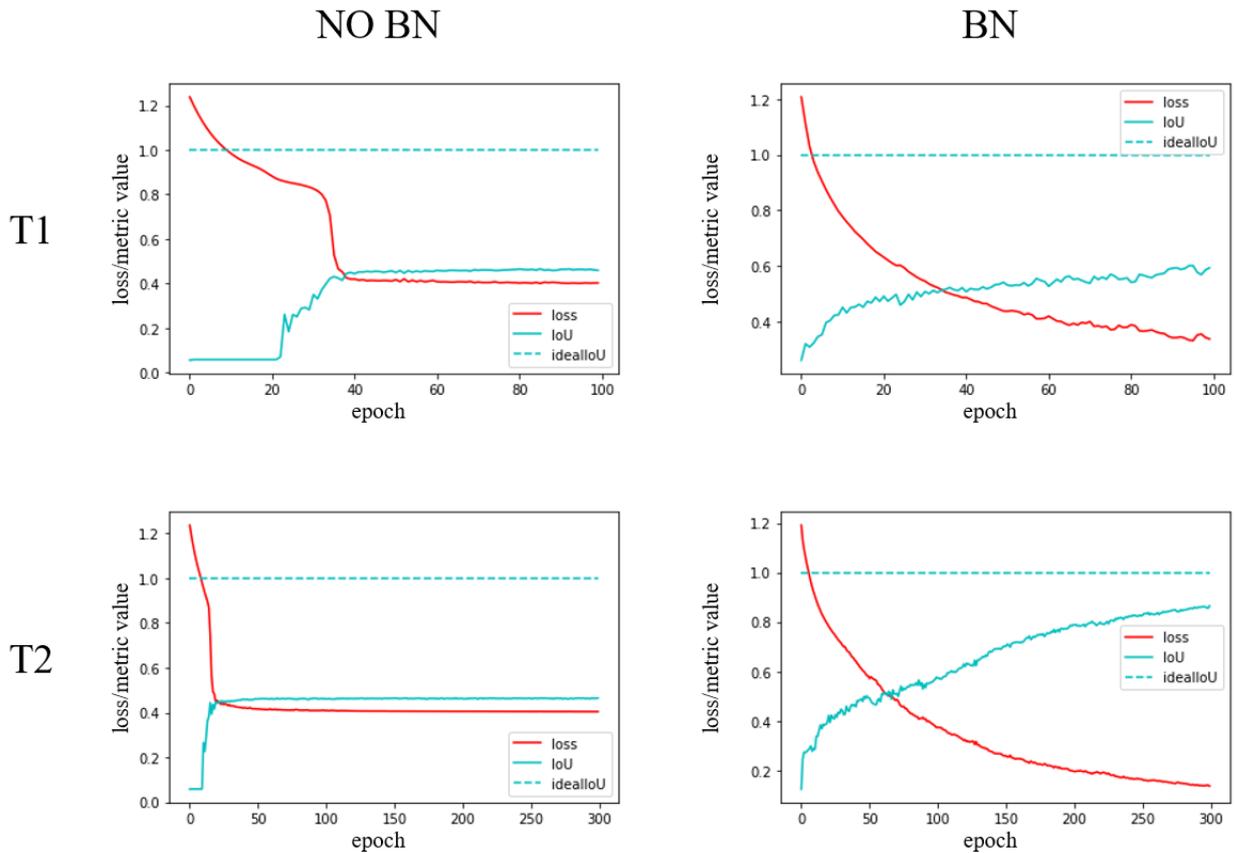


Figure 21: Training loss (red) and mIoU (blue) trends with training epochs: same network with (BN) and without (NO BN) batch-normalization layers during different trials of 100 (T1) and 300 (T2) epochs.

The maximum amplitude of weights for separation of touching cells was arbitrary. As a consequence, discretionary importance was attributed to optimize segmentation of border pixels rather than other structures. The employed weight amplitude empirically represented

a good compromise between separation of touching borders and IoU metric, the latter meaning only good adhesion to segmentation results without considering touching boundaries as relevant. Higher amplitudes allowed for sharp separation of close cells at the expense of IoU, while lower amplitudes yielded opposite effects. Hence, IoU was not an exhaustively representative metric for the task achievement.

In response, a new metric (M) was introduced by computing a weighted average of pixel-wise probability likelihoods based on the same weights defining relevance for touching objects:

$$M = \frac{1}{N_S} \sum_S \frac{\sum_{x \in S} w_{C-B}(x)(y\hat{y} + (1-y)(1-\hat{y}))}{\sum_{x \in S} w_{C-B}(x)} \quad (16)$$

Symbolism is maintained as in Eq. (14). M emulated a bounded version of the employed loss to the $[0;1]$ range. Indeed, its value over validation samples equaled 0.9822 and, considering $M = 1$ as the ideal case, reflected the fact that the loss was minimized on the validation set.

As in section 3.1, the applied preprocessing pipeline was fundamental to achieve the final results, with previous preprocessing trials (e.g. only normalization, only normalization and gamma correction) yielding unsatisfactory outcomes. Although depending on the number of fibroblasts, each inference required an overall computational time of the order of seconds, negligible compared to the one for manual procedures.

Geometrical parameters were extracted, as exemplified in Fig. 22. For each of three representative images (A, B, C - marks showing fibroblast centroids), fibroblasts were counted (D) and their singular area was quantified (E, bar: mean, whiskers: standard deviation). In the subregion marked in (A), instanced fibroblasts (F, a different color each) were characterized in terms of main orientation (quantified as in Application 1), centroid coordinates, area, and distances between centroids were reported (G).

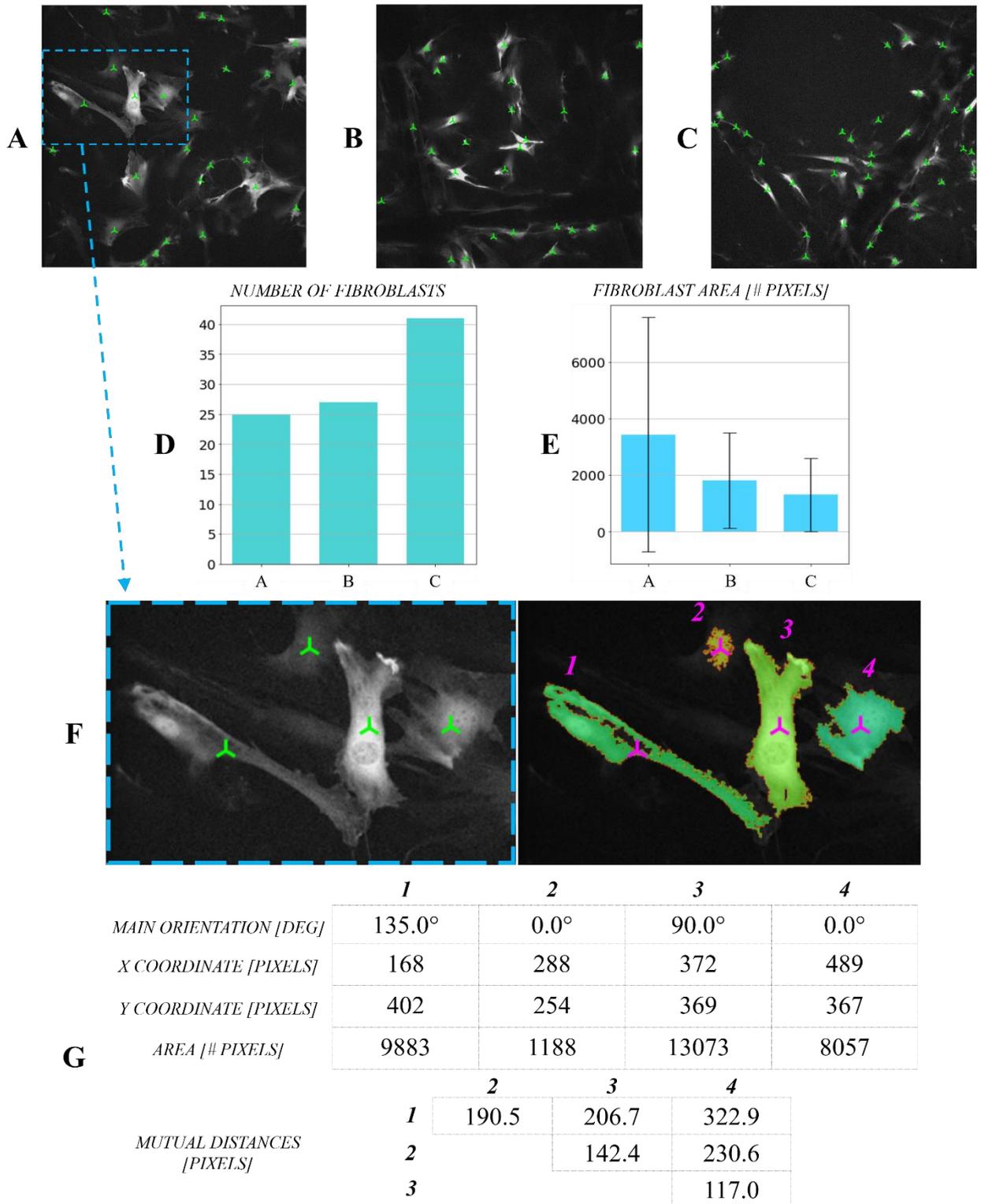


Figure 22: Exemplification of extracted parameters: fibroblasts in each input images (A, B, C) counted (D) with their area distribution quantified (E, bar: mean, whisker: standard deviation); instanced fibroblasts (F) in the highlighted subregion of (A) shown with different colors and characterized (G) each in terms of main orientation, centroid coordinates and area, with mutual distances between centroids reported.

3.3 Application 3: Segmentation of Scaffold-Associated Cells

FBGCs are the key immune cells at the basis of the FBR, that recruit fibroblasts by means of production of soluble factors, and adhere to the foreign materials to digest it. In order to monitor their evolution over time, a system to automatically discern Scaffold-Associated cells (SA), INterstitial cells (IN) and the remaining acellular BackGround (BG) from the same input image was implemented.

Fig. 23 shows the results of the semantic segmentation network differentiating SA, IN, and BG classes. 4 validation case studies (A, B, C, D) are reported, one in each row, showing the input image on the left, the target manual label on the right and the predicted semantic segmentation mask in the middle, where red, blue and black respectively stand for SA, IN and BG, allowing to note meaningful accomplishments (all instances and A in particular). The results obtained questioned human performances for gold-standard label creation. Several validation examples confirmed that equivocal structures, either misinterpreted or simply missed during manual labeling, were individuated and rectified by our algorithm. For instance, in Fig. 23, B the network seemed to correctly infer that the scaffold prolongation wrapped by scaffold-associated cells re-appeared at the bottom-right of the image, even if not explicitly assessable by a human observer (manual label) without knowing the adjacent content outside the image. In the outcome of Fig. 23, C, on the left, the network generalized predicting a structure, whose shape reminded of a scaffold-associated cell lying on a scaffold branch, licitly classified as SA, while a human observer could not recognize it because of the low image exposure. In Fig. 23, D prediction, several unclear, doubtful cells were classified as IN despite either on or adjacent to the scaffold framework, reasonable given their shape and size. Finally, the example in Fig. 23, A also reported that SA borders around scaffold framework were better reconstructed by the algorithm than by manual labeling. Additional results are reported in Fig. 24.

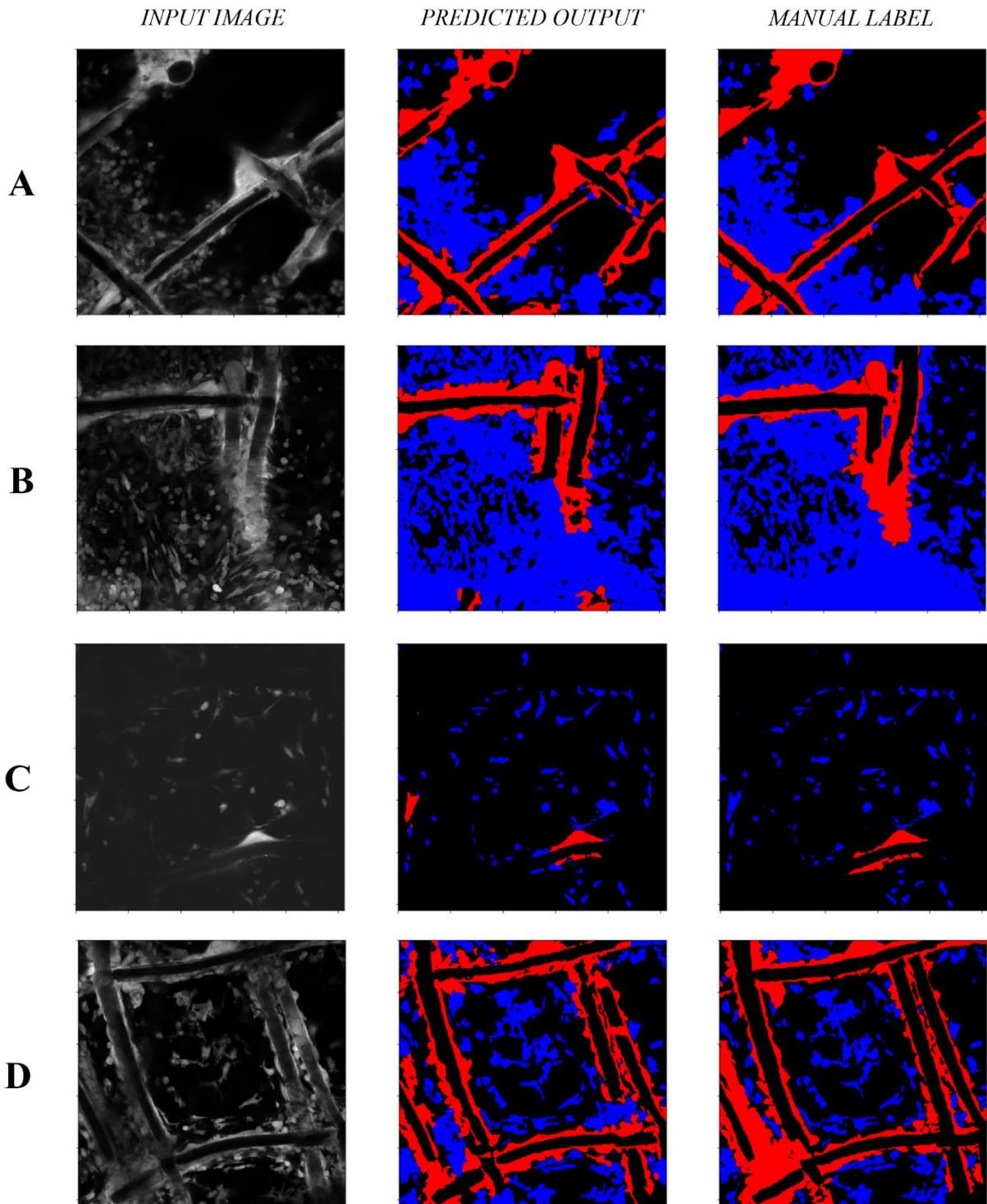


Figure 23: Segmentation of scaffold-associated cells (red), interstitial cells (blue) and background (black): preprocessed input image (left), predicted output mask (middle) and target manual label (right) shown for 4 different validation instances (A, B, C and D).

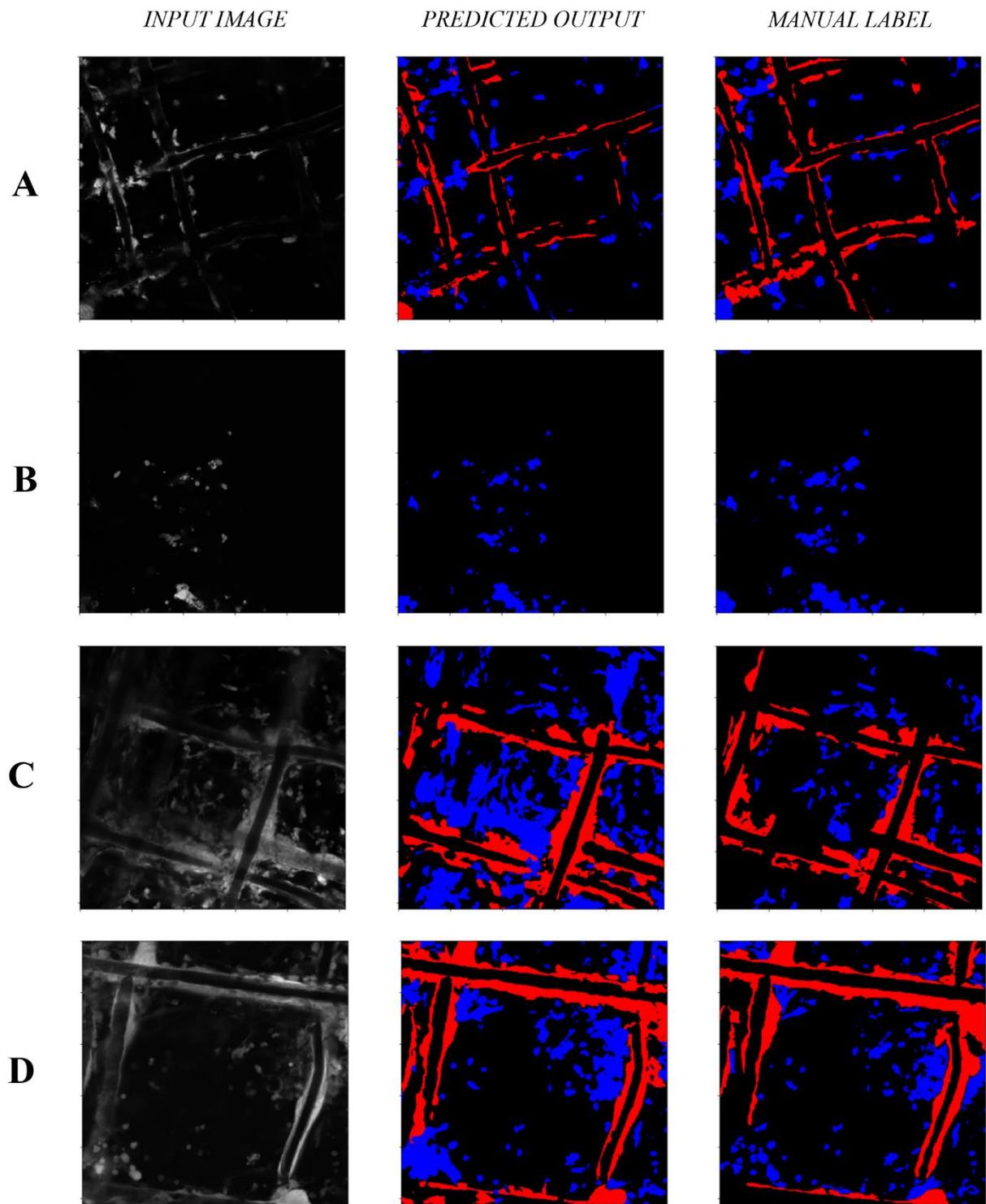


Figure 24: Segmentation of scaffold-associated cells (red), interstitial cells (blue) and background (black): preprocessed input image (left), predicted output mask (middle) and target manual label (right) shown for 4 different validation instances (A, B, C and D).

For the same rationales as in section 3.1, segmentation results were assessed by employing IoU metric separately for each class identified, i.e. SA, IN and BG. Our UNet variant yielded mIoU values over the validation set of 0.5330 for SA, 0.5795 for IN and 0.9019 for BG. Thus, BG mIoU confirmed the ability to associate scaffold structure to the a-cellular background, without an evident gap from the ideal value of 1.0. Given that metric class values over training samples were respectively 0.6416, 0.7375 and 0.8884 for SA, IN and BG, similar considerations as for Application 1 held both for SA and IN classes.

The network showed limitations in properly fitting the training set, so issues concerned the model capability of being correctly trained on the available data rather than the generalization. The observed gaps in validation performances for SA and IN classes were plausible, as validation metric values are never likely to exceed or match training ones.

The model complexity was bounded to the training size, the loss trend did not exhibit further improvements by increasing training epochs, and the quantity of data was retained sufficient to achieve the task as a meaningful accomplishment was observed on validation data, as shown in Fig. 23, all and A in particular. Hence, issues concerned the quality of training data, specifically, given the good resolution of microscopy images, the quality of labels. Some structures from the input images appeared as jagged, underexposed, blurred and faded. As a consequence, the manual image labeling remained uncertain in terms of structure boundaries. This resulted in slightly incoherent training examples, leading to conflicting weight updates both within and among mini-batches. Thus, a coherent representation of the task was not reached.

Nevertheless, segmentation state-of-the-art performances in terms of mIoU metric proved to be highly variable depending on the task to achieve and on the data domain of application [37]. Values of mIoU between 0.5 and 0.6 are not uncommon in commensurate problems. Hence, results must be considered relatively to the application and, in addition, weighted on the nature of the input data. Results were considered as satisfactory by biologists for the developed methodology to be applied to their analyses. Inference was again of the order of seconds, guarantying high time savings.

Once segmentation was achieved, area was computed separately for each class of interest, i.e. SA and IN, to quantify their total space occupancy, as shown in Fig. 25.

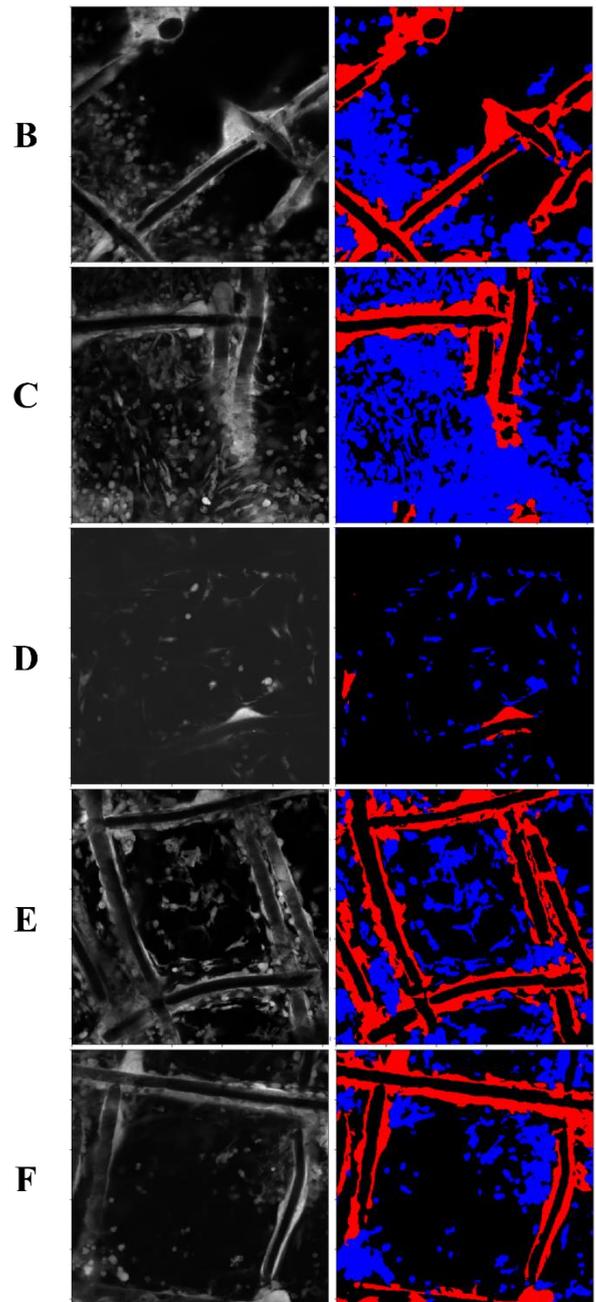
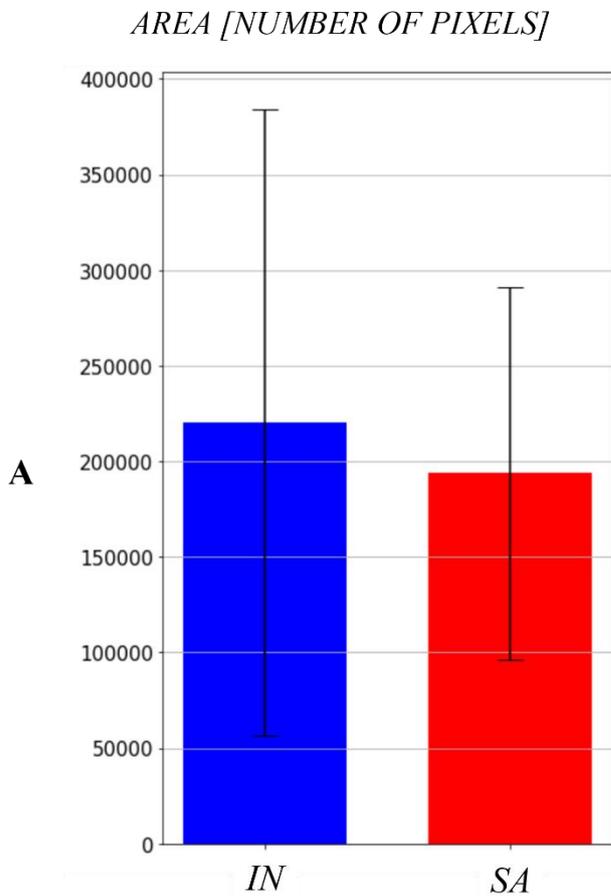


Figure 25: Example of area quantification in pixels both for SA class and IN class in terms of mean (bar) and standard deviation (whiskers) over five segmentation results (B, C, D, E, F), each illustrated as input image (left) and segmentation outcome (right) - red and blue always associated to SA and IN, respectively.

4. Conclusions

Three different applications were developed to endow biologists with automated software tools for discerning FBR dynamics of interest. All applications relied on deep learning to accomplish semantic segmentation of structures of interest in respective input images with data-driven approaches, and parameters were extracted from predicted masks.

The first application allowed to quantify collagen orientations relatively to the segmented scaffold framework, thanks to the implemented directional filtering based on customized kernels.

The second application yielded geometrical parameters describing fibroblasts distinguished from the segmented image via connectivity, and for this reason touching cells were segmented with particular attention.

The third application differentiated two cellular macro-populations, i.e. cells associated to scaffold structure and interstitial cells, from background within the same input image to quantify their area and growth.

For each application, images used to train and to validate the segmentation NN were accordingly preprocessed and manually labeled with expected results under biologists' supervision. The limited datasets employed, coupled to the extensive use of data augmentation, led to quick implementations, fast trainings and inferences but also to considerable results, thanks to our architectural choices (UNet variants). The developed applications automated the workflow of biologists, yielding at the same time substantial time saving, accuracy and reproducibility of results.

Deep learning intrinsic generalizability casted doubts on manual gold-standard labels of some validation instances. The same inference capability offers the application also to different data domains, with the possibility to extend tools to similar images acquired with different modalities or microscopy parameters.

Future developments will solve some issues encountered, mostly regarding precision and coherence of manual labels adapted. Samples quality will be improved and their quantity will be increased to allow the segmentation network to distinguish several cellular species

with complex characteristics within the same input image. Also, the network generalization capability on different domains will be systematically tested and, in case not robust, improved by adding similar images acquired with different modalities to the training set.

Finally, segmentation will be extended to 3D microscopy volumes to reconstruct cellular entities through slices and to extract exhaustive parameters for complete descriptions. The proposed UNet architecture can be easily extended to the 3D case by exploiting volumetric convolutions, but with a significant increase in the number of weights and a subsequent worsening in computational efficiency, training time, inference time and number of required labeled samples. Moreover, each sample will require manual labeling of several slices, dilating preparation times. Furthermore, a non-isotropic physical resolution along all spatial dimensions could represent a limit for a satisfactory reconstruction of structures.

References

- [1] Dondossola, E, et al., Examination of the Foreign Body Response to Biomaterials by Nonlinear Intravital Microscopy, *Nat Biomed Eng*, 2016
- [2] Ronneberger, O., et al., U-Net: Convolutional Networks for Biomedical Image Segmentation, 2015, arXiv:1505.04597
- [3] Anderson, J. M., et al., Foreign Body Reaction to Biomaterials, *Seminars in Immunology* 20, 86–100, 2008, <https://doi.org/10.1016/j.smim.2007.11.004>
- [4] Vogler, E. A., Siedlecki, C. A., Contact activation of blood-plasma coagulation. *Biomaterials*, 30, 1857–1869, 2009, <https://doi.org/10.1016/j.biomaterials.2008.12.041>
- [5] Horbett, T. A., Principles underlying the role of adsorbed plasma proteins in blood interactions with foreign materials. *Cardiovascular Pathology*, Chapter 13, 2, 137–148, 1993, [https://doi.org/10.1016/1054-8807\(93\)90054-6](https://doi.org/10.1016/1054-8807(93)90054-6)
- [6] Jung, F., et al., Haemocompatibility Testing of Biomaterials Using Human Platelets, *Clin Hemorheol Microcirc* 53, 97–115, 2013, <https://doi.org/10.3233/CH-2012-1579>
- [7] Scapini, P., et al., The Neutrophil as a Cellular Source of Chemokines. *Immunological Reviews* 177, 195–203, 2000, <https://doi.org/10.1034/j.1600-065x.2000.17706.x>

[8] Kobayashi, S. D., et al., Neutrophils in the Innate Immune Response. *Arch Immunol Ther Exp* 53, 505–517, 2005

[9] Al-Saffar, N., et al., 1997. Modulation of the Phenotypic and Functional Properties of Phagocytic Macrophages by Wear Particles from Orthopaedic Implants, *Journal of Materials Science: Materials in Medicine* 8, 641–648, 1997, <https://doi.org/10.1023/A:1018575504518>

[10] Berton, G., Lowell, C. A., Integrin Signalling in Neutrophils and Macrophages. *Cellular Signalling* 11, 621–635, 1999, [https://doi.org/10.1016/s0898-6568\(99\)00003-0](https://doi.org/10.1016/s0898-6568(99)00003-0)

[11] Kwee, B. J., Mooney, D. J., Manipulating the Intersection of Angiogenesis and Inflammation. *Annals of Biomedical Engineering* 43, 628–640, 2014

[12] Jones, J. A., et al., Proteomic Analysis and Quantification of Cytokines and Chemokines from Biomaterial Surface-Adherent Macrophages and Foreign Body Giant Cells. *Journal of Biomedical Materials Research Part A* 83A, 585–596, 2007, <https://doi.org/10.1002/jbm.a.31221>

[13] Brodbeck, W. G., et al., Influence of Biomaterial Surface Chemistry on the Apoptosis of Adherent Cells. *Journal of Biomedical Materials Research* 55, 661–668, 2001, [https://doi.org/10.1002/1097-4636\(20010615\)55:4<661::aid-jbm1061>3.0.co;2-f](https://doi.org/10.1002/1097-4636(20010615)55:4<661::aid-jbm1061>3.0.co;2-f)

[14] Sinha, V. R., et al., Poly- ϵ -caprolactone Microspheres and Nanospheres: an Overview, *International Journal of Pharmaceutics* 278, 1–23, 2004, <https://doi.org/10.1016/j.ijpharm.2004.01.044>

[15] Bhardwaj, U., et al., Controlling Acute Inflammation with Fast Releasing Dexamethasone-PLGA Microsphere/PVA Hydrogel Composites for Implantable Devices, 2007, <https://doi.org/10.1177/193229680700100103>

[16] Galeska, I., et al., Controlled Release of Dexamethasone from PLGA Microspheres Embedded within Polyacid-containing PVA Hydrogels. *The AAPS Journal* 7, E231–E240, 2005, <https://doi.org/10.1208/aapsj070122>

[17] Morais, J. M., et al., Biomaterials/Tissue Interactions: Possible Solutions to Overcome Foreign Body Response. *The AAPS Journal* 12, 188–196, 2010, <https://doi.org/10.1208/s12248-010-9175-3>

[18] Ward, W. K., et al., The Effect of Microgeometry, Implant Thickness and Polyurethane Chemistry on the Foreign Body Response to Subcutaneous Implants. *Biomaterials* 23, 4185–4192, 2002, [https://doi.org/10.1016/s0142-9612\(02\)00160-6](https://doi.org/10.1016/s0142-9612(02)00160-6)

[19] Zeplin, P. H., et al., Surface Modification of Silicone Breast Implants by Binding the Antifibrotic Drug Halofuginone Reduces Capsular Fibrosis. *Plastic and Reconstructive Surgery* 126, 266–274, 2010

[20] Klueh, U., et al., Enhancement of Implantable Glucose Sensor Function in Vivo Using Gene Transfer-induced Neovascularization. *Biomaterials* 26, 1155–1163, 2005, <https://doi.org/10.1016/j.biomaterials.2004.04.017>

[21] Kastellorizios, M., et al., Multiple Tissue Response Modifiers to Promote Angiogenesis and Prevent the Foreign Body Reaction around Subcutaneous Implants, *Journal of Controlled Release* 214, 103–111, 2015, <https://doi.org/10.1016/j.jconrel.2015.07.021>

[22] Lehninger, A. L., et al., *Lehninger Principles of Biochemistry, Biosignaling*, New York, W.H. Freeman, 2008

[23] Ioffe, S., Szegedy, C., Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift, 2015, arXiv:1502.03167v3

[24] Pedamonti, D., Comparison of Non-Linear Activation Functions for Deep Neural Networks on MNIST Classification Task, 2018, arXiv:1804.02763v1

[25] Scherer, D., et al., Evaluation of Pooling Operations in Convolutional Architectures for Object Recognition, 20th International Conference on Artificial Neural Networks (ICANN), Thessaloniki, Greece, 2010,

[26] Dumoulin, V., Visin, F., A Guide to Convolution Arithmetic for Deep Learning, 2018, arXiv:1603.07285v2

[27] Li, H., et al., Visualizing the Loss Landscape of Neural Nets, 2018, arXiv:1712.09913

[28] Drozdal, M., et al., The Importance of Skip Connections in Biomedical Image Segmentation, 2016, arXiv:1608.04117v2

[29] Masters, D., Luschi, C., Revisiting Small Batch Training for Deep Neural Networks, 2018, arXiv:1804.07612v1

[30] Kingma, P. D., Ba, J. L., ADAM: A Method for Stochastic Optimization, 2017, arXiv:1412.6980v9

[31] <https://keras.io/>

[32] <https://keras.io/>

[33] <https://colab.research.google.com/notebooks/intro.ipynb>

[34] Shen, C. et al, On the Influence of Dice Loss Function in Multi-Class Organ Segmentation of Abdominal CT Using 3D Fully Convolutional Networks 2018, arXiv:1801.05912v1

[35] Novikov A. A., et al., Fully Convolutional Architectures for Multi-Class Segmentation in Chest Radiographs 2018, arXiv:1701.08816v4

[36] Abraham, N., Khan, N. M., A Novel Focal Tversky Loss Function with Improved Attention U-Net for Lesion Segmentation, 2018, arXiv:1810.07842v1

[37] <https://paperswithcode.com/task/semantic-segmentation/latest>