



INTERNATIONAL DOCTORAL
SCHOOL OF THE USC

Almoutaz Mamdooh Ahmad
Mbaidin

PhD Thesis

Design of segmentation
algorithms to recognize
interested cells in microscopy
biological images

Santiago de Compostela, 2023

Doctoral Programme in Information Technology Research



ESCOLA DE DOUTORAMENTO
INTERNACIONAL DA USC

TESE DE DOUTORAMENTO

DESIGN OF SEGMENTATION ALGORITHMS TO RECOGNIZE INTERESTED CELLS IN MICROSCOPY BIOLOGICAL IMAGES

Almoutaz Mamdooh Ahmad Mbaidin

Directora: Eva Cernadas García

Tutora: Eva Cernadas García

**PROGRAMA DE DOUTORAMENTO EN
INVESTIGACIÓN EN TECNOLOXÍAS DA INFORMACIÓN**

SANTIAGO DE COMPOSTELA, 2023



**I dedicate this work to my father's soul, my compassionate mother, to my loving wife,
my wonderful children, my dear sister and brother , and brother in law for their
endless support and unwavering love.**

*I prefer to be a dreamer among the humblest, with
visions to be realized, than a lord among those
without dreams and desires.*

Gibran Khalil Gibran

Acknowledgments

I would like to express my special thanks and appreciation to my supervisor, Prof. Eva Cernadas García, for her patient guidance, encouragement, and invaluable advice. I have greatly benefited from her wealth of knowledge. I am extremely grateful to have had a supervisor who cared so much about my work and promptly responded to my questions.

I would also like to express my special thanks to Manuel Fernández Delgado for his support.

Thanks and great appreciation to my mother; your prayers for me have been instrumental in getting me this far. I would also like to thank my children for the unlimited happiness and pleasure they bring and my sister and her husband for their unwavering support.

Finally, my deepest appreciation goes to my wife, my inspiration and my world. Without her, I might never have completed this thesis. Having a partner with such infinite talent and dynamism who holds her own professional career to support the dreams of her husband is a powerful thing.

This work has received financial support from Xunta de Galicia (ED431G-2019/04) and the European Regional Development Fund (ERDF), which acknowledges the CiTIUS—Centro Singular de Investigación en Tecnoloxías Intelixentes da Universidade de Santiago de Compostela as a Research Center of the Galician University System.

Novembro de 2023

Contents

1	Introduction	7
1.1	Fish fecundity from histological images	8
1.2	Related work on image segmentation	11
1.2.1	K-means clustering	15
1.2.2	Mean Shift Algorithm	16
1.2.3	Chan-Vese Algorithm	16
1.2.4	Felzenszwalb method	17
1.2.5	Govocitos method	17
1.3	Objective of the thesis	17
2	Materials and methods	19
2.1	Materials	20
2.2	Methods	23
2.2.1	MSCF: Multiscale-Scale Canny Filter	23
2.3	Evaluation of cell segmentation methods	33
3	Results and discussion	35
3.1	MSCF algorithm	36
3.1.1	Performance at pixel level	37
3.1.2	Performance at region level	41
3.2	Comparison with other image segmentation algorithms	42
3.3	Computational time	56
3.4	Conclusions	56
4	The STERapp software	61

4.1	STERapp architecture and functionality	63
4.2	Description of modules in the application layer	67
4.2.1	Training of classifiers	68
4.3	Results and discussion for the use of STERapp	69
4.3.1	Experimental setup	70
4.3.2	Automatic cell recognition and classification	71
4.3.3	Analysis performance	73
4.3.4	Expert perception	74
4.3.5	Comparison with ImageJ	75
4.4	Conclusions	76
5	Conclusion	77
A	Listado de publicaciones	81
B	STERapp User Guide	83
B.1	Windows installation	84
B.2	Linux installation	84
B.3	Run STERapp	85
B.4	Set preferences	89
B.4.1	Calibration	91
B.4.2	Diameters	91
B.4.3	Working directories	91
B.4.4	Drawing configuration	92
B.4.5	Adding categories	93
B.4.6	Species selection	93
B.4.7	Setting the configuration	94
B.5	File menu	95
B.6	Edit menu	98
B.7	View menu or lateral panel	99
B.7.1	Supervised post-processing	102
B.7.2	Visualization of results	105
B.8	Analysis menu	105
B.9	Classification menu	108

<i>Contents</i>	iii
B.9.1 Training the classifier	111
B.10 Help menu	112
Bibliography	113
List of Figures	125
List of Tables	129

Resumo en galego

Este tese sitúase no ámbito da visión por computador, é concretamente na segmentación de imaxes. O obxectivo é o deseño de algoritmos de segmentación de imaxes para o recoñecemento de células en imaxes histolóxicas de gónadas de peixes coa finalidade de estimar a súa fecundidade. A tese estruturase nos seguintes cinco capítulos: introducción, materiais e métodos, resultados e discusión, o software STERapp e as conclusións.

No capítulo de **introducción** descríbese o problema e detállase brevemente as técnicas de segmentación de imaxes propostas na literatura. O posible agotamento dos recursos pesqueiros é un tema preocupante para a nosa sociedade, facendose necesaria a súa xestión en aras de acadar a sustentabilidade dos mares. A fecundidade é un dos parámetros críticos para estimar a capacidade de produción dunha poboación de peces. Ademais, tamén ten interese para a investigación en pesquerías como parámetro de avaliación das dinámicas das poboacións de peixes. A importancia en determinar este parámetro con exactitude propiciou moitos esforzos da ciencia de pequerías para deseñar métodos máis simples, rápidos e de menor coste. A efectos prácticos, a fecundidade é o número de ovos maduros que un peixe desova no mar, e que pode variar entre centos ou millóns. Hoxe en día, o método estereolóxico é aceptado como a técnica máis precisa para estimar a fecundidade a partir de imaxes histolóxicas das gónadas dos peixes, é dicir, do seu aparello reprodutor. A estereoloxía é unha interpretación tridimensional das seccións bidimensionais desa estrutura en 3D, permitindo a estimación do número de partículas, no noso caso células, e o volumen que ocupan dentro da estrutura. Este método estereolóxico vense utilizando nos laboratorios de pesquerías desde hai anos de forma manual contando e medindo as células en cada estado de desenvolvemento dentro da imaxe. Esta tarefa faina o personal especializado de pesquerías, consumindo moitos recursos humanos.

A automatización do método estereolóxico é un problema de visión por computador. Inicialmente, é necesario recoñecer as células maduras na imaxe, que é un problema de segmentación de imaxes. A continuación haberá que clasificar as células segundo o seu estado de desenvolvemento e segundo teñan ou non núcleo visible na imaxe, coa finalidade de medir só as células que son cortadas polo centro e que, por tanto, teñen o núcleo visible. Esta segunda parte xa foi abordada en traballos previos da equipa de investigación. Polo tanto, esta tese de doutoramento concéntrase na primeira etapa de deseñar novas técnicas para o recoñecemento de células maduras nas imaxes histolóxicas de gónadas de peixes. O problema é moi desafi-

ante xa que as células son estruturas moi complexas e existe unha gran variabilidade entre as imaxes, qué é causada, fundamentalmente, por: 1) diferencias na preparación de mostras nos laboratorios de pesquerías como concentración das tincións, igualdade nos cortes, presenza de artefactos, danos nos tecidos, etc; e 2) diferencias nos sistemas de adquisición como microscopio ou cámara utilizada, condicións de luz, etc. Ademáis, a propia heterogeneidade dentro de espécimes popiciada polo desenvolvemento dos seus ovocitos ou células reproductoras, tanto dentro da mesma especie como entre especies. A técnica de segmentación proposta debería manter un comportamento uniforme en todas estas situacións.

O proceso de segmentación de imaxes divide a imaxe en rexións que non se solapan. Clásicamente, as técnicas propostas na literatura soense clasificar en: umbralización, basadas na detección do contorno, basadas na detección de rexións, segmentación morfolóxica e estratexias híbridas. As técnicas basadas na detección do contorno intentan atopar as fronteiras entre as rexións localizando discontinuidades nas propiedades da imaxe como brillo, textura, cor, etc. As técnicas basadas en rexións constrúen rexións directamente basándose nas similitudes das propiedades da imaxe dentro da rexión. A umbralización é a técnica máis simple que só analiza a distribución das propiedades dos píxeles. Realmente existe unha dualidade entre as técnicas basadas no contorno e na rexión xa que é fácil construír rexións a partir da contorna dos obxectos e calcular a contorna de rexións. Tamén podemos clasificar as técnicas de segmentación segundo a familia a que pertencen en: 1) modelos de contornos activos; 2) grafos podados (*graph cuts*); 3) detectores de bordes; 4) agrupamentos ou *clustering*; 5) aprendizaxe profunda ou *deep learning*; e 6) métodos híbridos. Os modelos de contornos activos transforman o proceso de segmentación nun problema de minimización da enerxía, onde a función de enerxía especifica o criterio de segmentación e as variables descoñecidas describen os contornos das diferentes rexións. Os contornos activos é un modelo paramétrico que pode ter unha representación explícita (no caso das *snakes*) ou implícita (no caso dos algoritmos *level set*). Dependendo de cómo as propiedades da imaxe estean incrustadas na función de enerxía, os modelos de *level set* poden ser basados en contornos ou rexións. Chan-Vase model é a técnica máis representativa de modelo baseada en rexións. As técnicas de agrupamento agrupan os píxeles da imaxes cando teñen atributos similares. Estas técnicas poden estar basadas en grafos como a técnica proposta por Felzenszwalb, ou non como os algoritmos *k-means* ou *mean-shift*. A diferenza das técnicas de segmentación mencionadas, as técnicas de aprendizaxe profunda soen ser supervisadas, polo que non serán consideradas neste estudo.

Esta investigación foi realizada en colaboración co Instituto de Investigacións Mariñas (IIM) e Instituto Oceanográfico Español (IOE), ambos centros de investigación do Consejo Superior de Investigacións Científicas (CSIC) situados en Vigo. No capítulo de **Materiais e métodos** descríbese o proceso de adquisición de imaxes das cinco especies de peixes utilizadas neste tese e que están dispoñibles na páxina web <https://gitlab.citius.usc.es/eva.cernadas/sterappimagesdb>. Todas as bases de imaxes foron anotadas por persoal experto de ambos centros de investigación utilizando o software Govocitos ou STERapp.

Tamén neste capítulo propoñemos a técnica de recoñecemento de células chamada MSCF, Multi-scale Canny Filter, un algoritmo de segmentación baseado en contornos, e concretamente no popular operador de detección de bordes proposto por Canny. Existe un consenso na comunidade científica sobre a superioridade do operador de Canny sobre outros detectores de bordes. Este detector inclúe os seguintes pasos: 1) suavizar a imaxe cunha máscara gaussiana de anchura σ ; 2) aplicar un operador de gradiente; 3) atopar os máximos locais na dirección perpendicular ó borde; e 4) realizar a umbralización utilizando histérese con dous umbrais T_H and T_L . As implementacións existentes do operador de Canny teñen dúas limitacións: 1) non incorporan un proceso para establecer os valores óptimos para σ , T_H and T_L ; e 2) proporcionan unha imaxe binaria cos píxeles asociados a bordes en branco e os píxeles de fondo en negro, e non unha cadea de píxeles conectados que representen a contorna dos obxectos. O algoritmo MSCF proposto solventa a segunda limitación facendo unha implementación na que se devolven as cadeas de píxeles conectados que representan a contorna das células. Respecto a primeira limitación opta polas seguintes estratexias: 1) polo propio sistema de adquisición do microscopio que enfoca no centro da imaxe, a nosa hipótese é que nas fronteiras da imaxe as células están máis desenfocadas, polo que pensamos que cun único valor de σ (unha escala) non é posible recoñecer correctamente todas as células; e 2) a elección dos valores de umbralización para o proceso de histérese é crítico para un funcionamento óptimo da técnica, polo que propoñemos determinalos automaticamente a partir das características da imaxe de gradiente. Aplicando o operador de Canny a distintas escalas podemos producir máis dunha instancia da mesma célula, polo que será necesario incorporar unha etapa de post-procesamento que fusione a información das distintas escalas. O pseudo-código do algoritmo MSCF e a súa descrición detallada está na sección 2.2.1 deste capítulo.

O capítulo **Resultados e discusión** presenta os resultados de aplicar o algoritmo MSCF ó recoñecemento das células para as cinco especies de peixes estudadas. Realízase unha

avaliación estatística a nivel de píxel e rexión ou célula. Utilízanse as medidas de precisión, taxa de recuperación (*recall*) e F_1 para cuantificar a coincidencia entre os píxeles asociados a células polo algoritmo e os verdadeiros fixados polo persoal experto de pesquerías. Como era de esperar, os maiores valores de F_1 están comprendidos entre o 70% e o 80% dependendo da especie considerada. Nembargantes, imaxínade un caso no que para unha célula o algoritmo detecte case todos os píxeles da célula pero agrupados en varias rexións no canto dunha única rexión. Neste caso, aínda que o algoritmo sería moi bo desde o punto de vista da clasificación a nivel de píxeles, non proporcionaría un bo resultado para o noso obxectivo de medir e contar as células presentes na imaxe. Polo tanto, necesítase unha avaliación desde o punto de vista das rexións, considerando que unha célula é correctamente recoñecida cando a superposición entre a rexión (célula) anotada polo persoal experto e a proporcionada polo algoritmo se superpoñen máis dunha certa porcentaxe de píxeles, chamada tolerancia. Esixindo unha tolerancia do 70%, o algoritmo MSCF proporciona unha porcentaxe media de células correctamente recoñecidas entre o 47.44% para a especie *hake* e o 64.25% para a especie *Roughhead-grenadier*. A mellor configuración do algoritmo MSFC é distinta para cada especie, polo que se utilizou o ordeamento de Friedman para determinar cal é a mellor configuración globalmente. O primeiro posto foi para a combinación de tres escalas $\mathcal{S} = \{2, 4, 6\}$ e o par de umbrais $\mathcal{TH} = \{(0.55, 0.75)\}$. A segunda posición para dúas escalas, $\mathcal{S} = \{2, 4\}$ e o par de umbrais, $\mathcal{TH} = \{(0.4, 0.6)\}$. A mellor configuración cunha sóa escala está no posto 12, polo que se demostra que cunha única escala non se poden acadar os mellores resultados.

Ademáis o algoritmo proposto foi comparado con outras cinco técnicas de segmentación clásicas. Os resultados estatísticos conclúen que, globalmente, o algoritmo MSCF proporciona os mellores resultados, aínda que o mellor resultado para a especie *hake* proporcionouno o algoritmo de agrupamento baseado no algoritmo *k-means*. Todos os detalles están dentro do apartado 3.2 do capítulo.

O capítulo o **software STERapp** describe unha ferramenta software para a realización do traballo de estimación da fecundidade nos laboratorios de pesquerías utilizando o método estereolóxico. Dado que o recoñecemento automático de células non proporciona uns valores de calidade suficientes para a construción dun software que opere de xeito completamente automático, STERapp concebiuse como una ferramenta semi-automática de axuda á estimación da fecundidade en peixes utilizando o método estereolóxico. STERapp úsase como ferramenta de traballo cotiá no IIM e IOE de Vigo e está dispoñible a toda a comu-

nidade científica para propósitos non comerciais na páxina web <https://citius.usc.es/transferencia/software/sterapp>. STERapp é un software modular estruturado en capas e cunha interfaz gráfica amigable para que o persoal dos laboratorios de pesquerías poida interactuar co programa e correxir os erros no recoñecemento das células antes da estimación da fecundidade. As principais contribucións do software son:

1. Ter unha interfaz gráfica intuitiva para simplificar o fluxo de traballo diario do persoal experto na estimación da fecundidade dos peixes.
2. Incorporar un módulo de recoñecemento automático das células na imaxe histolóxica, utilizando o algoritmo MSCF, o suficientemente rápido para operar en tempo real nun ordenador de propósito xeral.
3. Que o persoal experto pode revisar e modificar o recoñecemento automático das células antes do proceso de cuantificación e estimación da fecundidade, superando as limitacións do popular software ImageJ.
4. O software permite exportar conxuntamente os resultados de análise de varias imaxes, facilitando o traballo de elaboración de informes do personal de pesquerías.
5. STERapp incorpora un módulo de aprendizaxe automático para a clasificación de tres estados de desenvolvemento das células (*cortical alveoli*, *vitellogenic* and *hydrated*) e se unha célula ten o núcleo visible ou non na imaxe. Hai que lembrar que para estimar a fecundidade necesítase contar o número de células en cada estado de desenvolvemento e medir as células maduras, aquelas que foron cortadas polo medio e teñen o seu núcleo visible.
6. O software é preciso, fiable e fácil de instalar para persoal sen coñecementos de informática.

O módulo de aprendizaxe automática son dous clasificadores (un para o estado de desenvolvemento das células e outro para a presenza/ausencia de núcleo visible). No módulo incorporouse o clasificador máquina de soporte vectorial, *Support Vector Machine* (SVM) en inglés, usando a librería `libsvm` coa interface en C++. Existen dous clasificadores entrenados para cada especie. O software permite re-entrenar os clasificadores para as especies en calquer momento ou engadir novos clasificadores entrenados para as novas especies que

demo de alta na ferramenta. Os detalles para o manexo de toda a funcionalidade de STERapp pode consultarse no manual de uso do apéndice B.

O funcionamento de STERapp foi avaliado no traballo rutineiro do personal experto dos laboratorios de pesquerías do IIM e IOE. Para iso STERapp rexista en arquivos a interacción que o persoal experto realiza co software. No caso do recoñecemento das células maduras, a sensibilidade, especificidade e precisión media entre especies foi $55.6 \pm 8.2\%$, $64.8 \pm 16.2\%$ and $43.1 \pm 11.3\%$ respectivamente. As desviacións típicas tan altas quere dicir que existen grandes diferenzas entre imaxes da mesma especie e entre especies, xa que cada especie ten o seu proceso reprodutivo característico e as imaxes histolóxicas son moi distintas. No caso da clasificación automática de células, a porcentaxe de acerto media foi de $84.8 \pm 14.8\%$ para determinar o estado de desenvolvemento da célula e de $78.5 \pm 13.7\%$ para determinar se a célula ter núcleo visible na imaxe. Aínda que os resultados, de xeito obxectivo, poden parecer moderados, supoñen un aforro de tempo na estimación rutineira da fecundidade que o persoal experto estimou entre un 40% e un 70% con relación ó método tradicional que estaban utilizando nos laboratorios de pesquerías. A percepción subxectiva do software foi avaliada polo cuestionario *system usability scale* (SUS) entre o persoal experto, proporcionado unha puntuación media de 81.9, que significa que o software foi percibido entre bo e excelente.

Finalmente, o capítulo **Conclusion** resume as aportacións máis relevantes da tese e propón as liñas futuras de traballo. No **apéndice A** lístanse as publicacións realizadas coa investigación presentada nesta tese.

Palabras chave: visión por computador, segmentación de imaxes, recoñecemento de células, ovocitos, fecundidade, aprendizaxe automática, clasificación, máquina de vectores de soporte.

CHAPTER 1

INTRODUCTION

White light microscopic imaging has a prominent role in life science and medical research. Due to the recent advances in technology for microscopic in the last decade (see special issue of journal [45]), computerized methods for microscopic image analysis are attracting increasing interest for the precise quantitative characterization of slices [37, 42]. Computerized methods for microscopic image analysis are attracting increasing interest for the precise quantitative characterization of slices [37, 42]. Microscopic study of a tissue provides detailed insight into cellular morphology. For visualization under a microscope, the tissue is normally prepared and stained following standard protocols. For white light microscopic, the most commonly used are histological and immunohistochemical staining protocols, in which different components of the tissue are dyed with different stains. Biologists often perform manual inspections for bioimage analysis. This task is very time consuming and requires a large effort and concentration of specialized personnel. The quantitative analysis of biological images is rarely performed, despite the useful information provided for diagnosis and therapeutic treatment in the case of biomedical images. Quantitative analysis of microscopic images to recognize, classify, measure and/or count structures of interest in the image. The quantitative analysis of biological images can be grouped into the following types:

1. Count and/or detect the position of the objects of interest in the image: J. Xu et al. [100] proposed a method for the nuclei detection of breast cancer histopathological images, and G.J.A. de Melo [21] proposed a method for counting bovine milk somatic cells in microscope slide images.

2. Divide the image into macroscopic regions of interest: A. B. Tosun [88] proposed an algorithm to divide the histopathology tissue images of colon into normal and cancerous regions of different grades.
3. Classify the whole image: H. Chang et al. [13, 47, 79] proposed systems to classify histological sections.
4. Recognize the objects of interest in the image: P. Wang et al. [92] segmented the cell nuclei in breast histological images to classify breast cancer, H. Kong et al. [46] proposed an integrated framework for supervised color-texture segmentation and cell splitting histopathology images. C. Gundez-Demir et al. [36] recognized glands in colon tissue images. Huang and Lai [40] proposed an automatic system for grading hepatocellular carcinoma biopsy images, which segment nuclei using a combination of watershed and active contour models to extract features for classification. Ta et al. [84] proposed a method based on graph-based regularization to segment nuclei in serous cytological and breast cancer histopathological images. Y. Al-kofahi et al. [1] proposed a graph cut- based method for segmentation of breast cancer nuclei.

The current thesis focuses on the last group and proposes a segmentation algorithm to recognize cells in microscopic images. It is used to recognize oocytes in histological images of fish ovaries to estimate fish fecundity. This chapter describes an overview of the problem, the related work in image segmentation, and the objective of the thesis.

1.1 Fish fecundity from histological images

Fecundity is one of the population parameters considered critical for estimating the productivity of a fish stock and thus of interest to fishery scientists as both a critical parameter of stock assessment [49], and a basic aspect of population dynamics [41]. There are fecundity regulatory mechanisms that operate during a gonad development cycle, and due to them, some oocytes do not complete the maturation process and are not ovulated; in this case, those follicles are reabsorbed by atresia processes. Fecundity in fishes can vary from thousands to millions of eggs spawned per individual. To assess stock reproductive potential (SRP) and its impact on recruitment (birth rate) variability, it is necessary to study both the development and resorption of oocytes to estimate the realized fecundity, which can be used not only as an indicator of stock productivity but also as a reference point for fisheries management [44], which

are fundamental to developing a precautionary approach to sustainable fisheries. A lack of awareness of variability in the SRP adds uncertainty to stock-recruitment relationships, which affects the reliability of associated reference points. The importance of determining accurate fecundity estimates has led to many research efforts to provide simpler, faster and lower cost methods since the beginnings of fisheries science. Fecundity studies from the early 1900s were based on the manual gravimetric method [60, 71], a basic, simple and inexpensive technique that is tedious and very time consuming. C. F. Hickling [38] (and references therein) described an indirect method to estimate fecundity based on the area occupied by the eggs when they were distributed in a layer whose height was equal to one egg. F. E. J. Fry established for the first time the relationship between the mean diameter of the oocytes from the leading cohort (more advanced maturity stage) [30] and the oocyte density in the ovaries that led to the semiautomatic autodiametric method developed by [87]. Some authors have used fish stripping to estimate the number of eggs released from a living fish; however, the efficiency of this method is not constant [51, 69] and can only be applied when females are close to ovulation. R. E. Burrows developed the volumetric method that relates the volume of fluid displaced by an ovary sample with the number of oocytes within this sample [7]; nevertheless, D. R. Wolfert demonstrated the inaccuracy of this method in comparison to the gravimetric method [99]. By summarizing the basics of the gravimetric method and with the evolution of computerized image analysis, new automatic tools to estimate fecundity have been developed [98]. T. B. Begenal and E. Braum summarized most of these methods based on counting the number of mature oocytes in whole mount preparations, assuming that all oocytes larger than a certain diameter threshold are mature and recruited to spawn [5]; nevertheless, it is not possible to determine the maturity stage of oocytes by the naked eye, especially at the beginning of the maturation process. In need of an accurate classification of the oocytes used in the fecundity estimates, microscopic and histology techniques were applied.

On the other hand, stereology [75] is a corpus of techniques and methodologies to estimate unbiased quantitative geometrical parameters (for instance, number, length, area, and volume) from solid material, based on the relationship between a 3D structure and 2D cross sections obtained from it, i.e., stereology is the tridimensional interpretation of bidimensional sections of a structure [96]. Stereology uses random, systematic samples to estimate these parameters robustly and is widely applied in different scientific fields from mineralogy to neurology. It allows us to estimate, applying the Delesse Principle, the number of particles and the volume they occupy within a structure. For that purpose, microphotographs of different tissues or

materials are commonly used [23], assuming that they are thin enough to be considered a plane (2D).

L. S. Emerson et al. [24], taking advantage of the precise classification of oocytes from histological images, developed a stereometric method based on the stereological method [97, 96] that allows estimating fecundity from histological images, guaranteeing the exclusivity in the selection of mature oocytes for fecundity estimations.

In need of an accurate classification of the oocytes used in the fecundity estimates, microscopic and histology techniques were applied. Weibel and Gómez [97] developed a method for particles having a constant shape that is frequently used in biology and relates the number of particles per unit of volume (NV) to the number of particle shapes per unit of area in each section (NA). For that purpose, a stereological analysis uses a grid of hexagonal cells (Weibel grid) whose cell size is adjusted to the size of the image and the objects (particles) to be measured. The grid is superimposed on the image to count the number of points (vertices of hexagonal cells) that are in contact with an individual particle and the number of particles included within the frame of the grid. Subsequently, each marked point is associated with the area of the corresponding hexagonal cell, where it is assumed that the number is proportional to the area of the particle in the tissue section. These areas are associated with the number of particles within the frame and used to estimate NA. Until now, this tedious work has been carried out manually, which is both time-consuming and inaccurate, because the area is based on extrapolation to hexagonal cells, instead of the real area of the particle. In that sense, Emerson et al. [24] developed a method to estimate fish fecundity (i.e., the number of mature oocytes in the ovaries) based on the work of Weibel and Gómez [97], which is currently largely used in fish biology research. Oocytes are spherical-shaped particles that increase in size and complexity as gonadal maturation advances. Thus, applying stereological methods to histological sections of ovarian tissue requires experienced technicians capable of identifying the different stages of oocyte maturation in histological slices.

The estimation of fish fecundity [49] is used to calculate fish stock productivity as well as spawning stock biomass by applying egg production methods (EPMs) to establish biological reference points that are used for sustainable fisheries management [41]. For example, EPM is currently used in the assessment of several commercial species targeted by European fleets, such as *Atlantic mackerel* (*Scomber scombrus*), *horse mackerel* (*Trachurus trachurus*), *European pilchard* (*Sardina pilchardus*) and *European anchovy* (*Engraulis encrasicolus*). Estimating fecundity by stereological methods implies counting and measuring the diameters

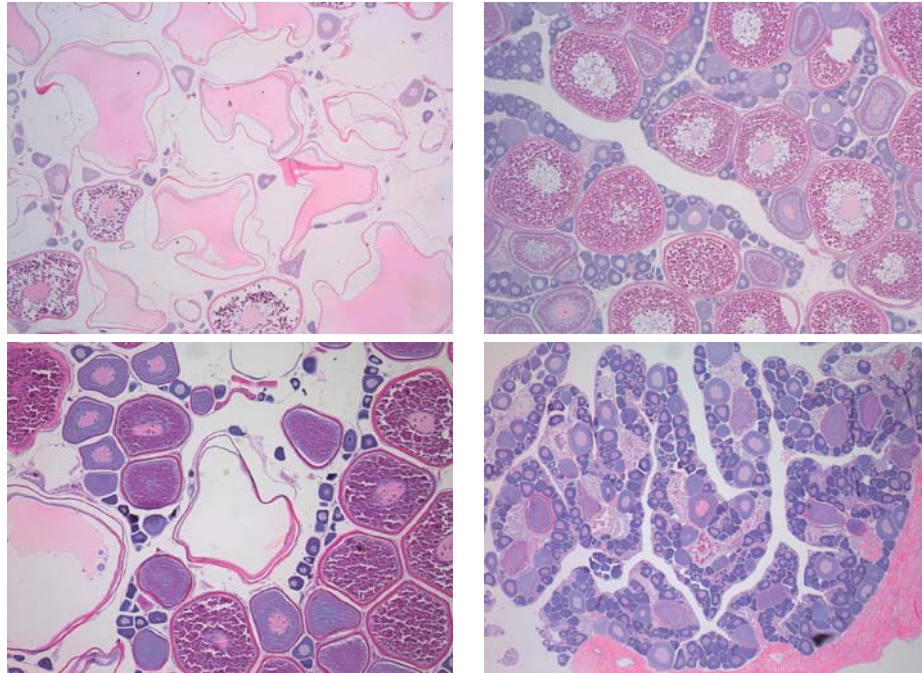


Figure 1.1: Examples of histological images of the fish species *European Hake* (upper panel) and *Trisopterus luscus* (lower panel).

of oocytes (spherical cells) at different developmental stages (*cortical alveoli*, *vitellogenic* and *hydrated*), as well as those cells that will be reabsorbed (*atretic*), in histological images. For the diameter measurement, only those oocytes with visible nuclei in the histological slide (i.e., cut closely through its maximum diameter) must be considered. Despite stereology being a robust and unbiased method for estimating fecundity, it still requires specialized technicians and is time consuming, even with software being available today. Figure 1.1 shows some examples of histological images of fish ovaries of different fish specie.

1.2 Related work on image segmentation

Image segmentation is a fundamental problem in image analysis, which partitions an image into non-overlapping regions. The main advantages of image segmentation are relaxing the

complexity of the image and make use of further analysis of each image segment instead of dealing with the entire image. Moreover, image segmentation enables object identification and the gathering and classification of other important elements in the image.

There are many surveys on image segmentation in the literature [70, 105, 64, 57], which give an idea of the importance of this field. Depending on the end application, the segmentation process can be classified into well-defined and ill-defined problems. In ill-defined problems, there are as many valid solutions as interpretations of the image. Nevertheless, the solution is unique in well-defined solutions. There are many techniques in the literature for automatic image segmentation applied to a wide range of applications. The problem of partitioning an image into its constituent components, frequently named the foreground and background, is based on the similarity and discontinuity properties of the image. Therefore, image segmentation methods can be categorized into thresholding, edge-based, region-based, morphological segmentation or watershed, and hybrid strategies [20, 21, 26, 91, 107]. Region-based segmentation constructs regions of images directly based on similarities in image properties. That is, region-based approaches try to separate homogeneous areas of images and group the image pixels into clusters. Two classical region-based methods: region growing algorithms and split-and-merge procedure. These algorithms use color, color gradient, and/or texture to define the region homogeneity [33, 83]. Edge based segmentation finds boundaries between regions based on local discontinuities in the image properties (brightness, texture, color or numerical measures over local image patches) [106, 54]. Thresholding segmentation was accomplished via thresholds based on the distribution of pixel properties [14]. This is the simplest and fastest approach but it is not suitable for complex images. Although, the edge-based and region-based results are not exactly the same, it is easy to construct regions from their borders and to detect the borders of existing regions.

The segmentation approach can also be categorized by families of algorithms into: active contours [20], graphs cuts [67], edge detectors [8], clustering and other hybrid methods. The active contour model family transforms image segmentation to an energy minimization problem where the energy functional specifies the segmentation criterion and the unknown variables describe the contours of different regions. The active contour is a parametric model, which is an explicit representation (called snakes) or an implicit representation (called level set algorithms). The level set methods can be categorized into edge-based and region-based methods according to the image property embedded in the energy functional. The Chan-Vese model is the most representative region-based model [11, 82]. The graphs cuts models rep-

represent the image by a weighted graph and the segmentation problem is translated to find a minimum cut in the constructed velocity graph via a maximum flow computation. which give an idea of the importance of this field. Depending on the end application, the segmentation process can be classified into well-defined and ill-defined problems. In ill-defined problems, there are as many valid solutions as interpretations of the image. Nevertheless, the solution is unique in well-defined solutions. There are many techniques in the literature for automatic image segmentation applied to a wide range of applications. The problem of partitioning an image into its constituent components, frequently named the foreground and background, is based on the similarity and discontinuity properties of the image. Therefore, image segmentation methods can be categorized into thresholding, edge-based, region-based, morphological segmentation or watershed, and hybrid strategies [103]. The most popular clustering algorithms can be broadly classified into graph-based algorithms, such as the normalized cuts [80] or Felzenszwalb-Huttenlocher segmentation algorithms [80] or Felzenszwalb-Huttenlocher segmentation algorithm [28], and non-graph-based algorithms, such as k-means [26, 103], mean-shift [18] or expectation maximization (EM).

The segmentation methods can also be classified into unsupervised, semi-supervised and supervised depending on the information provided by the user in the segmentation process [93, 35, 72]. In the semi-supervised segmentation, also called interactive segmentation, the user provides a few seeds to represent the object position or an initial contour. Thus, reduces the need for labeled data. The method of training can be achieved either by self-training [27], co-training [107], or by mean teacher-based methods [104, 95]. The main difference among them is that the self-training method contains only one model, while the co-training and mean teacher-based methods usually contain two or more models. Recently, an accurate semi-supervised segmentation network was obtained as in [35] by expanding the training dataset with high-quality pseudo labels. However, such a method requires a series of operations to optimize pseudo labels.

Other recent segmentation approaches combine different methods such as Peng et al. [66] proposed a Dynamic Region Merging (DRM) algorithm, which performs image segmentation by iteratively merging the regions according to a statistical test from an initially over-segmented image. Wang et al. [94] proposed a level set method where the local statistical analysis and global similarity measurement are incorporated in the construction of energy functional. Recently, Hu et al. [39] estimated the dynamic range and distribution of scale based on a stepwise evolution analysis (SEA) of region merging, and Xiao Tan et al. [85]

introduced High Dynamic Range (HDR) imaging for large-scale with dynamic scenes, and implemented a self-guided attention to alleviate the impact of saturated regions during the aligning and merging processes.

Currently, deep learning (DL) approaches have achieved remarkable breakthroughs in the field of image segmentation, outperforming the accuracy of conventional segmentation methods. DL approaches are based on a fully convolutional network (FCN) and they provided successful results for image semantic segmentation [56], [48]. In DL, convolutional neural networks (CNNs) are primarily composed of three types of layers: an input layer, hidden layers, and an output layer. The hidden layers are further divided into three types: extracting features layer named as convolutional layer; pooling layer, which reduces feature resolution; and aggregate layer or fully connected layer. The latter produces prediction results after compiling the extracted features from previous layers, filtering and selection feature maps. During the training process, the updated parameters across different layers can be efficiently achieved by utilizing the backpropagation algorithm [102], [50]. Moreover, unlike conventional segmentation methods, which employ multiple dense patches, CNNs utilize the entire image as input during both training and testing phases without the need for information about the size of the input image. DL has attracted growing interest in the biomedical image segmentation field. In addition to CNNs, U-net, U-net+, and generative adversarial networks (GANs) have recently been applied and share the same target in terms of improving the sensitivity of the details of the image, enhancing the accuracy, and extracting more complex image features [53].

The DL methods are usually complex and require the support of powerful computing resources [2]. They have been used to segment different organs in radiological medical images [53], among others. In the field of medical microscopic images, Jiang et al. [43] reviewed the applications of DL in citological studies, principally to classify and detect cells or nuclei in citology images. Shujian Deng et al. reviewed the applications of DL for digital pathology image analysis [22], including classification, detection and semantic segmentation. Due to the computational requirements of DL methods, they are applied to image patches or downsampled images. The most popular DL techniques for image segmentation are U-Net versions [55], DeepLab [15] and SegNet [3]. Feixiao Long [55] proposed an enhanced U-net called U-net+ to minimize the less pre- and postprocessing of images. This approach boosts the accuracy of cell nuclei segmentation algorithms through DL under resource-constrained computing. Liang-Chieh Chen et al. [15] proposed a DeepLab technique that utilized upsampling

filters for dense prediction and feature extraction of images. Vijay et al. [3] based on FCN, introduced a technique to eliminate the need for learning to upsample for semantic pixelwise segmentation termed SegNet by using a decoder network to control the upsampling and then the resolution.

The aim of this work is to design and evaluate algorithms to segment cells in histological images of fish gonads, which require low computational time and resources to:

1. Be executed on general purpose computers to be used in all institutions around the world without any specific computing equipment to manage its marine resources, e.g., to be included in the STERapp software [59].
2. Be used in interactive systems, because the problem is too complex to provide a totally automatic recognition and it may be necessary the expert supervision before the image quantification.

In the following, we expand the description of some of the mostpopular publicly available image segmentation methods: 1) K-means clustering (subsection 1.2.1); 2) MeanShift (subsection 1.2.2); 3) Chan-Vase (subsection 1.2.3); 4) Felzenszwalb method (subsection 1.2.4); and Govocitos method in subsection 1.2.5.

1.2.1 K-means clustering

K-means clustering is the simplest and most commonly used clustering algorithm, which partition the image into K clusters (K is provided by the user) according to some property or numeric attributes of the image pixels on the basis of minimizing the error. For two clusters ($K = 2$), one represents objects or cells (foreground) and the other one is the background. The simplest property is the gray level of the pixel or brightness (pixel intensity), and other sophisticated properties include the pixel color in different color spaces, the texture features or a combination of both types of features [10]. To illustrate the working principle of K-means, let x_i and x_j be two feature vectors, and let the center of the cluster is E_k , $k = 1, \dots, K$ be the center of the cluster. The Euclidean distance between x_i and x_j is calculated to measure the similarity (i.e. the homogeneity of the clusters) and it is defined as [103]:

$$D(x_i, x_j) = \sqrt{(x_{i1} - x_{j1})^2 + \dots (x_{iL} - x_{jL})^2} \quad (1.1)$$

where L is the number of attributes of feature vector. The sum of squared errors (SSE) is an objective functions to measure the clustering quality for the k-Means algorithm define by [103, 81]:

$$SSE = \sum_{k=1}^K \sum_{x \in E_k} D(x, e_k) \quad e_k = \frac{1}{n_k} \sum_{x \in E_k} x \quad (1.2)$$

where n_k is the number of samples in the k^{th} cluster E_k .

1.2.2 Mean Shift Algorithm

The mean shift (MS) algorithm is a non-parametric method to estimate a probability density function (PDF). Mean shift avoids the need for prior knowledge about clustering number, centers, and shape. It simply can deal with any arbitrary shape of clusters. The MS algorithm is mostly applied to mode clustering. During the mean shift process, each point of the dataset is shifted iteratively by a small amount toward the mean. Thus, locating the mode for each pixel and grouping the pixels that has the same mode in a particular cluster [16, 101, 9].

Mean shift found its application in the fields of computer vision, image processing, and pattern recognition, and the MS algorithm is widely used, for example, in image filtering and segmentation, edge detection, and object tracking [17, 86, 31, 101, 9]. Its configuration parameters are the spatial bandwidth, $sbw = \{8, 16\}$ and the range bandwidth (rbw) with values from 4 to 20 with a step of 2.

1.2.3 Chan-Vese Algorithm

Based on the Mumford and Shah functional Model [63], Chan and Vese introduced a level set algorithm named the 'Chan-Vese Algorithm' [11],[32] for minimizing the Mumford-Shah function in image processing. Instead of using boundary edges, chan-vese model utilizes a level set function to represent the segmentation boundary where the initial curve can be set anywhere in the image. Additionally, the chan-vese model is mainly for region-based active contours to detect objects in a particular image. Furthermore, it eliminates the necessity of gradient-based boundaries and minimizes the energy of the minimal partition problem. However, chan-vese works under the assumption that the image intensities are homogeneous. In an attempt to overcome this issue, some modified versions of the chan-vese model were launched such as piecewise smooth (PS) models and the local region-based Chan-Vese (LRCV) model

[52, 73]. However, the computational efficiency of the chan-veese model remains a major challenge.

1.2.4 Felzenszwalb method

It is a method for a graph-based representation of the image segmentation based on pairwise region comparison. Its configuration parameters are the spread σ to smooth the input image before segmentation with values from 0.1 to 0.9 and a step of 0.1 and the k value to threshold the function with values from 250 to 1000 using a step of 250.

1.2.5 Govocitos method

Govocitos is a software tool for estimating fish fecundity based on digital analysis of histological images [68]. It includes a cell recognition algorithm where the Canny filter is applied at three scales, which are different values to the spread $\sigma = \{0.2, 0.4, 0.6\}$ of a Gaussian filter, following different pairs of threshold rates to perform the hysteresis process, $(T_{high}, T_{low}) = \{(0.4, 0.6); (0.55, 0.75); (0.6, 0.6)\}$. The output of the Govocitos algorithm is a set of contours C associated with the outline of the cells. The parameters provided to the algorithm are: the minimum diameter for the cell to be recognized and its maximum roundness.

1.3 Objective of the thesis

In previous research, we developed Govocitos software [68], which offers a software tool to estimate fecundity from histological images of fish gonads using the stereological method. However, Govocitos is quite difficult to install and maintain for a non-expert user due to its complexity, mainly caused by the use of a database for collaborative work and data sharing. It was statistically evaluated in a real environment with the fish species *European Hake* and *Trisopterus luscus*.

In this research, we extend the study to more fish species to develop more generic, robust and versatile tools. First we test the performance of state-of-the-art segmentation methods in the cell recognition problem. Afterwards, we design specific segmentation algorithms to recognize cells on microscopic images in biology and medicine. Our final objective is to develop a desktop application to estimate fish fecundity using the stereological principle, which overcomes the limitations of Govocitos and other available software. Therefore, it must fulfill the following requirements:

1. It will have a friendly graphical user interface (GUI), which simplifies the expert daily work flow to estimate fish fecundity. This requirement implies extending the following functionality of Govocitos:
 - a) Allows the user to define new stages of development of the oocyte or categories.
 - b) Define the category *void* (area in the image that must not be considered as the gonad area).
 - c) Allow the user to add new species to the system.
 - d) Export joint quantification analysis from multiple images.
2. The automatic image analysis will be fast enough to operate in real time.
3. The experts can review and modify the results of the automatic cell recognition and classification before image quantification.
4. It will allow us to train classifiers for different fish species.
5. The software will be accurate, trustworthy, and easy to install for non-expert users.

It is known that the processing of tissues and analysis of images from different species and taken with different image systems and magnifications increase the variability of oocyte shapes, sizes, aspects, abundance and distribution within the image. There is also variability among species due to differences in breeding strategies. For this reason, it is expected that the automatic algorithms to recognize the cells do not provide reliable cell recognition for all images. Therefore, it is important to emphasize that the algorithms developed to recognize cells must be fast enough to operate in real time on a general purpose computer. This restriction is very important to provide software tools, which can be adopted by experts in their daily work.

The proposed cell recognition method, called **Multi-Scale Canny Filter** (MSCF), is described in chapter 2, the statistical evaluation of MSCF and its comparison with other state-of-art segmentation techniques is presented and discussed in chapter 3, the description and evaluation of the developed software, called **STERapp**, is shown in chapter 4 and, finally, the last chapter (chapter 5) summarizes the conclusions of this research.

CHAPTER 2

MATERIALS AND METHODS

The content of the following paper is partially reproduced in this chapter. The publication details are:

A. Mbaidin, E. Cernadas, ZA. Al-Tarawneh, M. Fernández-Delgado, R. Domínguez-Petit, S. Rábade-Uberos, A. Hassanat. **MSCF: Multi-Scale Canny Filter to Recognize Cells in Microscopic Images**. *Sustainability*. Special issue *Latest Applications of Computer Vision and Machine Learning Techniques for Smart Sustainability* . 2023; 15(18):13693. <https://doi.org/10.3390/su151813693>. ISSN 2071-1050.

whose authors are:

1. A. Mbaidin from the Computer Science Department, Mutah University, Karak 61711, Jordan.
2. E. Cernadas from Centro Singular de Investigación en Tecnoloxías Intelixentes da USC (CiTIUS), Universidade de Santiago de Compostela, Spain.
3. ZA. Al-Tarawneh from the Computer Science Department, Mutah University, Karak 61711, Jordan.
4. M. Fernández-Delgado from Centro Singular de Investigación en Tecnoloxías Intelixentes da USC (CiTIUS), Universidade de Santiago de Compostela, Spain.
5. R. Domínguez-Petit from the Instituto Español de Oceanografía (IEO, CSIC), Centro Oceanográfico de Vigo, Vigo, Spain.

6. S. Rábade-Uberos from the Instituto de Investigaciones Marinas (IIM, CSIC), Vigo, Spain.
7. A. Hassanat from the Computer Science Department, Mutah University, Karak 61711, Jordan.

Segmentation of cells in routinely stained histological images is a challenging problem due to high variability in images caused by a number of factors including differences in slide preparation (dyes concentration, evenness of the cut, presence of foreign artifacts, damage of tissue sample, etc.) and image acquisition system (presence of digital noise, specific features of slice scanner, different lighting conditions or variations in microscope focus throughout the image). Furthermore, there is biological heterogeneity among specimens (cell type or development stage) and differences in the type of tissue under observation. A successful image segmentation approach will have overcome these issues in a robust way to maintain a high level of quality and accuracy in all situations.

This chapter describes the materials used to develop the experimental work, presents the cell recognition algorithm called MSCF, and describes the statistical measures to evaluate the performance of segmentation methods.

2.1 Materials

This research was performed in collaboration with the Instituto de Investigaciones Mariñas¹ (IIM) and the Instituto Español de Oceanografía² (IEO), both belonging to the State Agency Consejo Superior de Investigaciones Científicas (CSIC) in Spain. Ovaries at different maturity stages from different fish species were selected to develop the experiments. The sample processing and image acquisition were performed by IEO and IIM. One ovary lobe was fixed in 4% buffered formaldehyde and one slice per ovary of all females were embedded in paraffin. Then, 3 μm sections were cut and stained with hematoxylin and eosin for later microscopical analysis. Fecundity estimates were based on 4 microscope fields per ovary section using different microscopes and digital video cameras to acquire the images. Table 2.1 shows the acquisition system used for each fish species studied. In all cases, the exposure time and color balance were set automatically.

¹<http://www.iim.csic.es/>

²<http://www.ieo.es/en/home>

Dataset	Spice	#ima	System	Magnif.	Image size
hake	<i>European Hake</i>	31	1	2.5-10x	2088 × 1550
pouting	<i>Trisopterus luscus</i>	30	1	2.5-10x	2088 × 1550
pilchard	<i>European pilchard</i>	25	2	40x	3840 × 3072
megrin	<i>Four-spot megrim</i>	20	1	2.5-10x	2088 × 1550
grenadier	<i>Roughhead grenadier</i>	24	3	1.25x	3888 × 2916
			4	0.73x	2560 × 1920
ballan	<i>Ballan wrasse</i>	38	1	2.5-10x	3132 × 2325
mackerel	<i>Atlantic mackerel</i>	23	1	2.5-10x	2088 × 1550

Table 2.1: Datasets of images used: code of set (first column), fish species (second column), number of images (third column), acquisition system (column 4): 1) microscope Leica DMRE connected to a Leica DFC 320 camera; 2) microscope Nikon Eclipse 80i with a Nikon DXM 1200F camera; 3) Leica DM 4000B with Leica DFC 420 camera and 4) Leica M165C with Leica DMC 4500 camera. Magnification and image size in the last columns.

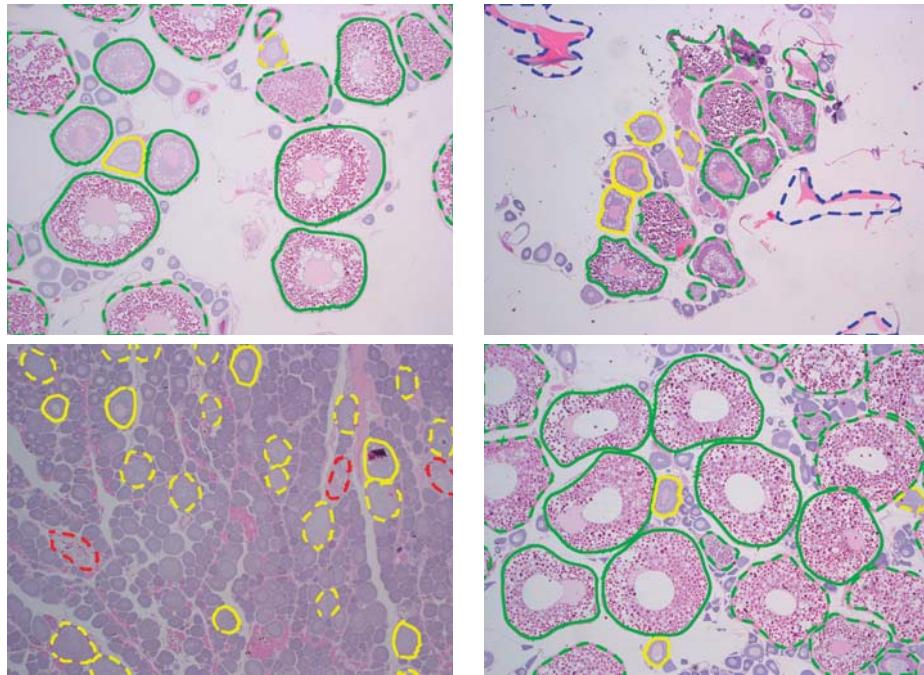


Figure 2.1: Examples of histological images of *European Hake* fish species with the outline of the ground truth cells overlapped. Different colors mean different development stage of matured cells.

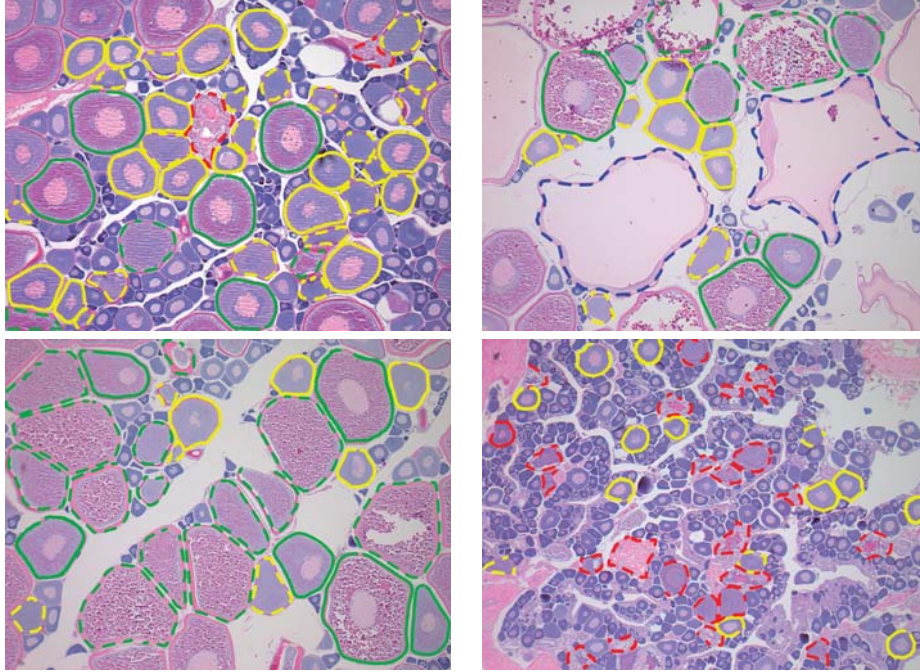


Figure 2.2: Examples of histological images of *Trisopterus Luscus* fish species with the outline of the ground truth cells overlapped. Different colors mean different development stage of matured cells.

The segmentation of the matured cells in images was supervised by experts of IEO and IIM using the software Govocitos [68] and STERapp [59]. The ground truth (optimal segmentation) for the cells in an image is reported in XML (Extensible Markup Language) files as the image points that define each cell (one per image). Table 2.1 presents the datasets used, which are available in <https://gitlab.citius.usc.es/eva.cernadas/sterappimagesdb>. Figures 2.1 to 2.7 show representative images of each dataset with the ground truth outline of the overlapping cells. The color indicates the development stage of the cell (*cortical alveoli* in yellow, *hydrated* in green, *vitelline* in blue, and without development stage in red) and the type of line the presence of visible nucleus (continuous for cells with visible nucleus and dashed for cells without visible nucleus).

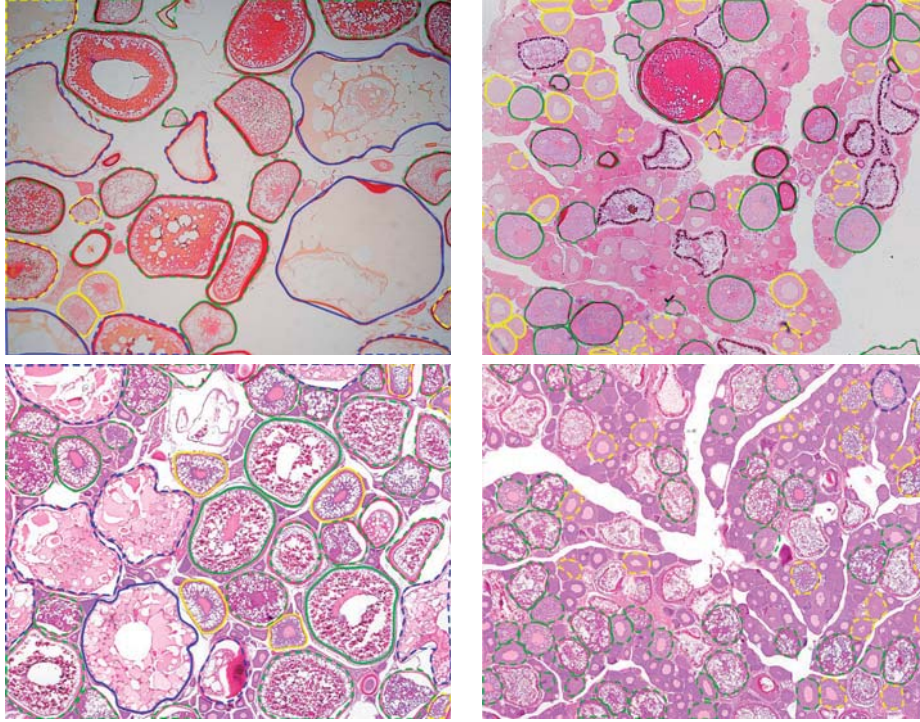


Figure 2.3: Examples of histological images of *Atlantic Mackerel* fish species with the outline of the ground truth cells overlapped. Different colors mean different development stage of matured cells.

2.2 Methods

There is a consensus in the scientific community about the superior performance of the Canny edge detector proposed by [8], in relation to other common differential operators, such as Sobel, Prewitt, Roberts, Laplacian, Laplacian of Gaussian (LoG) and difference of Gaussians (DoG).

2.2.1 MSCF: Multiscale-Scale Canny Filter

The classical Canny edge detector [8] includes the following steps: 1) smoothing of the input image using a σ -width Gaussian kernel; 2) differentiation using a first derivative operator; 3) perform the non-maxima suppression process, which finds local maxima in the direction per-

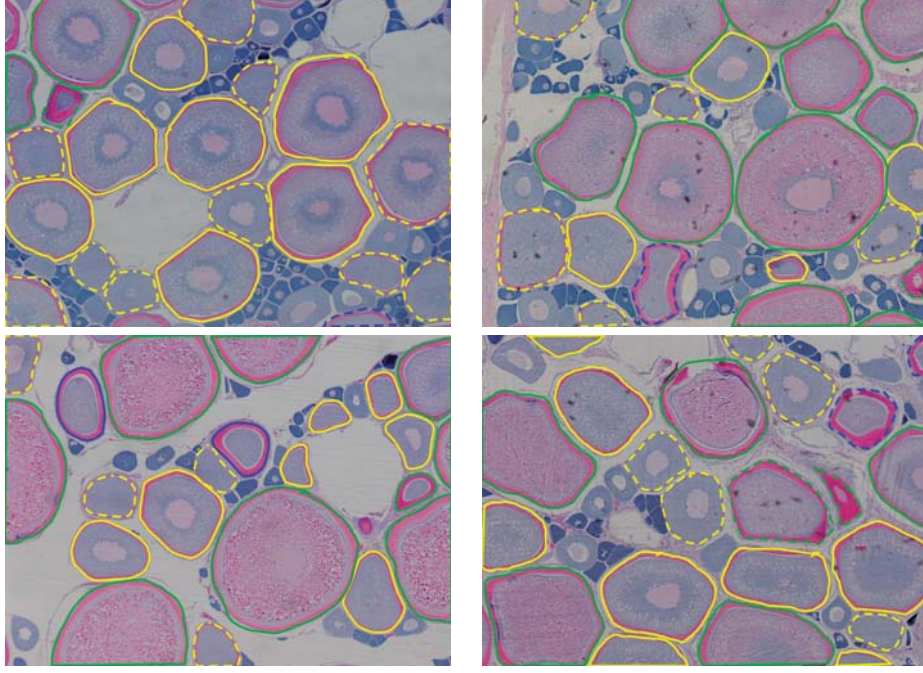


Figure 2.4: Examples of histological images of *Ballan Wrasse* fish species with the outline of the ground truth cells overlapped. Different colors indicates different developmental stages of matured cells.

pendicular to the edge; and 4) perform thresholding with hysteresis. The hysteresis process uses a high threshold (T_H) that allows a group of pixels to be classified as edge points without using information about connectivity. The process also uses a low threshold (T_L) to determine which pixels will not be edge points. Only those points that increase the connectivity of the previously determined edge points are aggregated as edge points. The output of the process is an edge map, i. e. a binary image with the detected edges in white. The filter performance depends critically on the tunable parameters: thresholds T_H and T_L , and smoothing parameter σ . Medina-Carnicer et al. [62] reviewed the approaches proposed to look for the best hysteresis thresholds for the Canny edge detector. Nevertheless, we experimentally proved in our problem of cell segmentation (see section 3.1) that a unique set of parameters does not allow us to recognize all the cells.

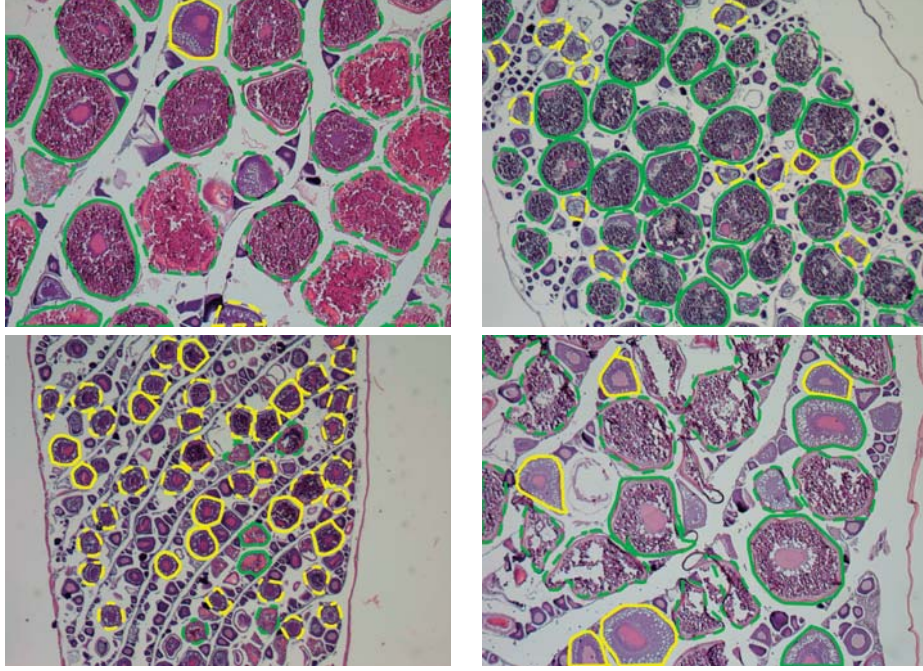


Figure 2.5: Examples of histological images of *European Pilchard* fish species with the outline of the ground truth cells overlapped. Different colors indicates different developmental stages of matured cells.

We propose applying the Canny filter using various sets of parameters followed by an information fusion step [58]. The idea behind our proposed model is to utilize various scales (σ values) and thresholds to obtain more information from the image. The objective is to achieve an efficient segmentation of microscopic images, where the strength of cell boundaries varies far away with the microscope focus and, probably, depends on sample preparation and specimen. The pixel neighborhood in the Canny filter is controlled by σ and its optimal value depends on the type and size of objects of interest in the image. High σ values smooth the inner regions of cells which can be textured or homogeneous. When cells are very close to each other, or when edges are weak, high σ values blur interesting edges as well. In the latter case, finer scales (i.e., lower σ values) must be used, although in this case the Canny filter will detect more noisy edges. For a given σ value, the variation of thresholds T_L and T_H controls the strength in the gradient image. High thresholds detect the strongest edges (normally true

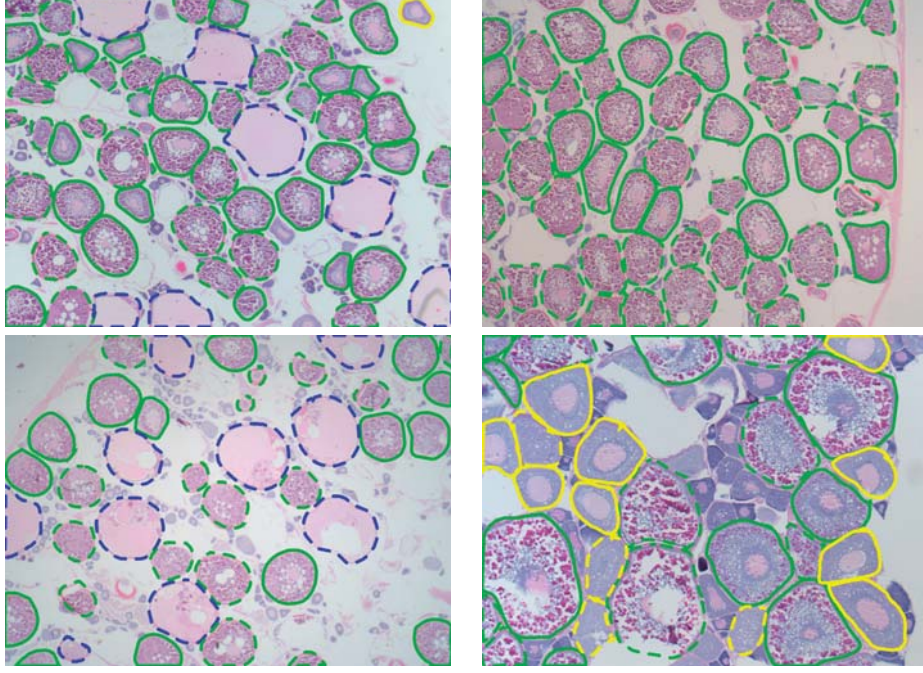


Figure 2.6: Examples of histological images of *Four-spot Megrim* fish species with the outline of the ground truth cells overlapped. Different colors indicates different developmental stages of matured cells.

edges), but they often miss other true edges. Therefore, low thresholds are also necessary to detect weak true contours.

In addition to the problem of tunable parameters, the output of the Canny edge detector (such as the implementation in MATLAB³ or OpenCV⁴) only provides an edge map of the image without exploiting the connectivity of the hysteresis process. Thus, we modified the filter implementation to provide the edges as sets or chains of connected points, representing pixels in the image, instead of an edge map. The MSCF function (summarized in algorithm 1) returns a list of detected edges, $\mathcal{E} = \{e_i\}_{i=1}^{N_e}$, in the input grey level image I . Let N_S and N_T be the number of smoothing values and thresholding rates, $\mathcal{S} = \{\sigma_i \mid i = 1, \dots, N_S \text{ and } \sigma_i > 0.1\}$ be the set of scales and let $\mathcal{R} = \{(R_L^j, R_H^j) \mid j = 1, \dots, N_T, \text{ and } 0 < R_L^j < R_H^j < 1\}$ be a set

³<http://es.mathworks.com>

⁴<http://opencv.org>

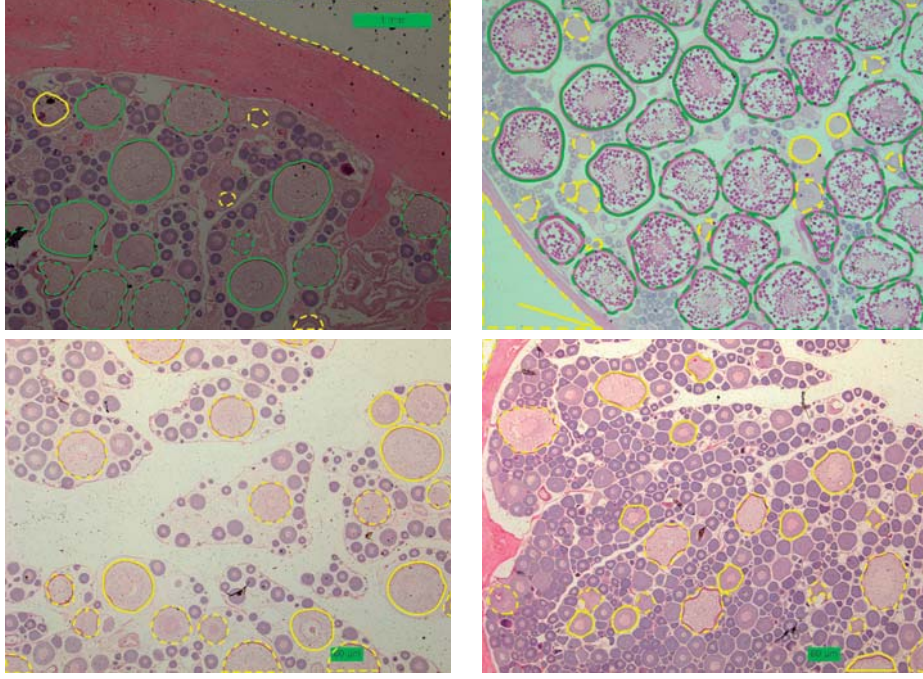


Figure 2.7: Examples of histological images of *Roughhead Grenadier* fish species with the outline of the ground truth cells overlapped. Different colors indicates different developmental stages of matured cells.

of threshold rates pairs. For each scale, a smoothed version I_S of I is obtained by applying a Gaussian filter (GaussianFilter function in algorithm 1). Afterwards, the gradient image I_G is calculated by applying a difference operator to I_S (GradientFilter function). The selection of the best values to apply hysteresis thresholding to the gradient image I_G is always a critical decision. In our approach, these values are determined from I_G and a pair of rates (R_L, R_H) using the CalculateThresholds function, which returns a different set of values (T_L, T_H) for each gradient image I_G . Finally, the hysteresis process (Hysteresis function) thresholds the gradient image and returns a set of edges. Each edge e_i is a set of points in the image representing contours or discontinuities in the image properties. Only the edges with more than a minimum number of points are considered to remove noisy edges. As the Canny filter is applied to different scales and threshold sets, some edges may correspond to the same cell. therefore, it is necessary to perform a fusion information step, which pro-

vides only one instance for each cell. The edges should also be with the points sorted to build a closed contour whose area can be measured.

Algorithm 1: Multi-scale Canny filter.

1 **Algorithm:** $\mathcal{E} = \text{MSCF}(I, \mathcal{S}, \mathcal{R}, m_p)$

Data: I : grey-level image; $\mathcal{S} = \{\sigma_i\}_{i=1}^{N_s}$; $\mathcal{R} = \{R_L^j, R_H^j\}_{j=1}^{N_T}$; m_p : minimum number of points

Result: $\mathcal{E}$: set of edges (sets of points)

2 $\mathcal{E} \leftarrow \emptyset$; $n_p \leftarrow \max(10, m_p)$

3 **foreach** $\sigma \in \mathcal{S}$ **do**

4 $I_S \leftarrow \text{GaussianFilter}(I, \sigma)$

5 $I_G \leftarrow \text{GradientFilter}(I_S)$

6 **foreach** $(R_L, R_H) \in \mathcal{R}$ **do**

7 $(T_L, T_H) \leftarrow \text{CalculateThresholds}(I_G, R_L, R_H)$

8 $\mathcal{E}' \leftarrow \text{Hysteresis}(I_G, T_L, T_H)$

9 $\mathcal{E} \leftarrow \mathcal{E} \cup \{e \in \mathcal{E}' \mid \text{size}(e) > n_p\}$

10 **end**

11 **end**

Algorithm 2: Algorithm for cell recognition.

1 **Algorithm:** $\mathcal{C} = \text{CellDetector}(I, \mathcal{S}, \mathcal{R}, d_{\min}, d_{\max}, R_{\max}, SF, RF, OF)$

Data: I : input image; $\mathcal{S} = \{\sigma_i\}_{i=1}^{N_s}$; $\mathcal{R} = \{T_L^j, T_H^j\}_{j=1}^{N_T}$; $d_{\min}/d_{\max}$: min./max. cell diameter; $R_{\max}$: maximum roundness; SF : size flag; RF : roundness flag; OF : overlapping flag

Result: $\mathcal{C}$: set of cells detected

2 $\mathcal{C} \leftarrow \emptyset$; $\mathcal{E} \leftarrow \text{MSCF}(I, \mathcal{S}, \mathcal{R}, d_{\min})$

3 **foreach** $e' \in \mathcal{E}$ **do**

4 $e \leftarrow \text{edgeSortedFiltered}(e', d_{\min}, d_{\max}, R_{\max}, SF, RF)$

5 $\mathcal{C} \leftarrow \mathcal{C} \cup \{e\}$

6 **end**

7 **if** OF **then** // overlapping flag

8 $\mathcal{C} \leftarrow \text{computeOverlappingTest}(\mathcal{C})$

9 **end**

To recognize cells in the image, the set $\mathcal{E}$ of edges detected by MSCF is post-processed by the following steps: 1) sort the contour points of the edge; 2) apply a size filter to the edges;

3) cells are normally round, so a roundness filter is applied to them; and 4) apply an overlapping test to remove various instances of the same cell. This process is summarized by the `CellDetector` method in algorithm 2. The first and fourth steps are necessary to provide a unique segmentation with closed contours for a cell. Cell filtering by size and roundness is optional. The `edgeSortedFiltered` function, called in algorithm 2 and detailed in algorithm 3, sorts the points in a contour. This process is performed by calculating the convex hull of the contour and measuring the difference between the contour and its convex hull. When the difference is small, the convex hull is kept as the contour, and when the difference is high, interpolated points are created from the true and convex hull polygons. In many biological problems, experts are only interested in objects (cells) with diameters between d_{min} and d_{max} , whose values are specific for the problem and microscopic calibration. The size filter removes from $\mathcal{E}$ the edges whose convex hull has a diameter lower than d_{min} or higher than d_{max} . The object roundness r_i is defined as $r_i = \frac{P_i^2}{4\pi A_i}$, where P_i and A_i are the perimeter and area of the contour. The roundness filter removes the edges with a roundness higher than a certain value $R_{max} > 1$, because the roundness of a circle is 1. The `edgeSortedFiltered` function allows us to specify whether we want to apply a size filter (flag variable $SF = True$) or not ($SF = False$), and if we want to apply the roundness filter (flag variable $RF = True$) or not ($RF = False$).

Let $e = \{q_i\}$ be a set of unsorted image points $q_i = (x_i, y_i)$, which defines a contour, and let $h = \{p_i\}$ be the convex hull of e . Considering that the points in the convex hull are sorted, we start by considering the points of the convex hull to build an output contour (c in algorithm 3) with the points sorted. Therefore, if the distance between two consecutive points in the convex hull, p_i and p_{i+1} , is lower than a threshold (we assume $D_{max} = 10$ pixels), those points are included in the output contour c . Otherwise, we create new points following this procedure: 1) calculate a line segment l_i from p_i to p_{i+1} , and create n_p points on l_i separated by D_{max} using the `pointOnLine` function; 2) being z_l a point on l_i , calculate the points q_e in contour e and p_h in the convex hull h that are the closest ones to z_l using the `closestPoint` function; and 3) if $\text{distance}(q_e, z_l)$ is lower than $\text{distance}(p_h, z_l)$, then the point q_e is added to the output contour. This procedure is illustrated in figure 2.8.

After applying the MSCF algorithm, there may be various detections of each true object in the image. The overlapping test (`computeOverlappingTest` function called in algorithm 2 and detailed in algorithm 4) provides only one instance of each true object as a closed curve. Let $\mathcal{E} = \{e_i\}_{i=1}^{N_e}$ be a set of contours representing the cells. For each contour

Algorithm 3: EdgeSortedFiltered to return a sorted contour.

1 **Algorithm:** $c = \text{edgeSortedFiltered}(e, d_{\min}, d_{\max}, R_{\max}, SF, RF)$

Data: $e = \{q_i\}_{i=0}^{n_e}$: chain of points; $d_{\min}/d_{\max}$: min./max. cell diameter; $R_{\max}$: maximum roundness; SF : size flag; RF : roundness flag;

Result: c : cell contour

```

2  $h \leftarrow \text{convexHull}(e)$  //  $h = \{p_i\}_{i=0}^{n_h}$ 
3  $r \leftarrow \text{roundness}(h)$ 
4 if  $RF$  and  $r > R_{\max}$  then
5   |  $c \leftarrow \text{null}$ ; return
6 end
7  $d_h \leftarrow$  maximum distance among points in  $h$ 
8 if  $SF$  and  $(d_h < d_{\min} \text{ or } d_h > d_{\max})$  then
9   |  $c \leftarrow \text{null}$ ; return
10 end
11  $D_{\max} \leftarrow 10$  // max. distance between points
12  $c \leftarrow \{p_0\}$  // contour with sorted points
13 for  $i \leftarrow 0$  to  $n_h - 1$  do
14   |  $d \leftarrow \text{distance}(p_i, p_{i+1})$ 
15   | if  $d < D_{\max}$  then
16   |   |  $c \leftarrow c \cup \{p_{i+1}\}$ 
17   | else // create new points
18   |   |  $l_i \leftarrow$  line segment from  $p_i$  to  $p_{i+1}$ 
19   |   |  $n_p \leftarrow \lfloor d/D_{\max} \rfloor$ ;  $d_l \leftarrow d/(n_p + 1)$ 
20   |   | for  $j \leftarrow 0$  to  $n_p$  do // Points on line segment
21   |   |   |  $z_l \leftarrow \text{pointOnLine}(l_i, j, d_l)$ 
22   |   |   |  $q_e \leftarrow \text{closestPoint}(z_l, e)$ 
23   |   |   |  $p_h \leftarrow \text{closestPoint}(z_l, h)$ 
24   |   |   | if  $\text{distance}(q_e, z_l) < \text{distance}(p_h, z_l)$  then
25   |   |   |   |  $c \leftarrow c \cup \{q_e\}$ 
26   |   |   | end
27   |   | end
28   |   |  $c \leftarrow c \cup \{p_{i+1}\}$ 
29   | end
30 end

```

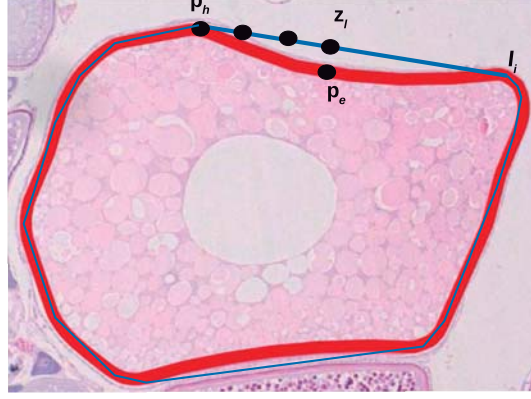


Figure 2.8: Example of the operation of algorithm 3 to sort the point into a cell contour.

e_i , we precalculate the following data: 1) the minimum rectangle b_i that encloses e_i , using the `boundingBox` function; 2) the radius r_i of e_i , calculated as half of the longest side of rectangle b_i ; 3) the mass center p_c^i , calculated as the center of rectangle b_i ; 4) the parameter d_i is the maximum distance between two consecutive points in e_i (taking in mind that when overlapping is tested, the points in the contour are sorted); and 5) the status s_i of e_i is set to 1, indicating that this contour is not still selected. Once these data are computed, for each contour e_i that is not still considered ($s_i = 1$), it is checked if other $e_j \in \mathcal{E}$, exist $e_j \neq e_i$, with $s_j = 1$ that overlaps e_i . To speed up this process, a simplified procedure is implemented. The first pre-selection of a cell candidate is calculated if the distance between the mass centers of both cells (p_c^i and p_c^j) is lower than their approximate radius, i.e. $e_i, e_j \in \mathcal{E}, i \neq j | \text{distance}(p_c^i, p_c^j) < r_i + r_j$. Second, for a finer test, the `pointInsideContour` function in algorithm 4 is used, which returns `true` if a point is inside a closed sorted contour and `false` otherwise. Then, if p_c^j is inside e_i and p_c^i is inside e_j , both contours are candidates to represent the same cell, i.e. both cells are overlapped and e_j is added as a candidate cell. If p_c^i is inside e_j but p_c^j is not inside e_i , it is probably that the e_i contour is a noisy object inside the e_j cell. Therefore, e_j is exchanged by e_i in the set of candidate cells $\mathcal{D}$. Finally, if p_c^j is inside e_i but p_c^i is not inside e_j , the contour e_j is desestimated because of it is probably a noisy object inside the e_i cell. Once the set of candidate cells $\mathcal{D}$ is built, the cell $e_k \in \mathcal{D}$ with the minimal d_k is selected, where d_k the maximum distance between two consecutive points in e_k .

Algorithm 4: Overlapping test to cells.

1 Algorithm: $\mathcal{C} = \text{computeOverlappingTest}(\mathcal{E})$

Data: $\mathcal{E}$: set of cells**Result:** $\mathcal{C}$: set of cells

```

2  $\mathcal{C} \leftarrow \emptyset$ ;  $\mathcal{E} = \{e_i\}_{i=1}^{N_e}$ 
3  $s \leftarrow \{s_i = 1, i = 1, \dots, N_e\}$  // cell status
4 foreach  $e_i \in \mathcal{E}$  do // precalculated data
5    $b_i \leftarrow \text{boundingBox}(e_i)$ 
6    $l_i \leftarrow$  longest side of the rectangle  $b_i$ 
7    $r_i \leftarrow l_i/2$ ;  $p_c^i \leftarrow$  mass center of  $b_i$ 
8    $d_i \leftarrow$  maximum distance between two consecutive points in  $e_i$ 
9 end
10 for  $i \leftarrow 1$  to  $N_e - 1$  do
11    $\mathcal{D} \leftarrow \emptyset$  // cells candidates to be overlapped
12   if  $s_i = 1$  then
13      $\mathcal{D} \leftarrow \mathcal{D} \cup \{e_i\}$ 
14     for  $j \leftarrow i + 1$  to  $N_e$  do
15       if  $s_j = 1$  then
16         if  $\text{distance}(p_c^i, p_c^j) < r_i + r_j$  then
17            $ji \leftarrow \text{pointInsideContour}(e_i, p_c^j)$ 
18            $ij \leftarrow \text{pointInsideContour}(e_j, p_c^i)$ 
19           if  $ji$  and  $ij$  then
20              $\mathcal{D} \leftarrow \mathcal{D} \cup \{e_j\}$ ;  $s_j \leftarrow 0$ 
21           else if  $ij$  then
22              $\mathcal{D} \leftarrow \mathcal{D} \cup \{e_j\} - \{e_i\}$ ;  $s_j \leftarrow 0$ 
23           else if  $ji$  then
24              $s_j \leftarrow 0$ 
25           end
26         end
27       end
28     end
29     if  $\text{size}(\mathcal{D}) > 0$  then
30        $\mathcal{C} \leftarrow \mathcal{C} \cup \{e_k \in \mathcal{D} | d_k \text{ is minimum}\}$ 
31     end
32   end
33 end

```

2.3 Evaluation of cell segmentation methods

Segmentation algorithms can be evaluated by quantifying the agreement of an automatic segmentation with some suitable defined ground truth [89, 25]. The evaluation measures can be enclosed in two types: those that compare the overlap between regions in the truth and computer segmentation; and those that evaluate the agreement between corresponding segmentation boundaries. The latter is oriented to evaluate the segmentation based on the boundary precision, and it is used in ill-posed problems to quantify the agreement of an automatic segmentation with the variation in a set of available manual segmentations. The first considers the segmentation by a partition image into its constituent regions. On the one hand, our problem is well-defined (only one ground truth) and the aim of segmentation is to recognize the cells to their later measure. Then, the first approach is more suitable. The segmentation performance can also be quantified at the pixel or region level.

The precision, recall and F_1 score allow to measure segmentation quality to pixel-based level. For each image segmentation, we record the number of true positives TP (the number of pixels that were classified as belonging to a cell by both the algorithm and the expert), true negative (the number of pixels that were classified as noncell pixels by both the algorithm and the expert), false positive FP (the number of instances where a noncell pixel was falsely classified as part of the cell by the algorithm) and false negative FN (the number of instances where cell pixels were falsely classified as noncell pixels by the algorithm). From this, we can calculate the precision (P), recall, sensitivity or true positive rate, specificity or false negative rate and F_1 given by the following equations:

$$P = \frac{TP}{TP + FP} \quad (2.1)$$

$$R = \frac{TP}{TP + FN} \quad (2.2)$$

$$SP = \frac{TN}{TN + FP} \quad (2.3)$$

$$F_1 = \frac{2PR}{P + R} \quad (2.4)$$

To quantify the rate of correctly segmented cells, let D^i be the number of recognized cells and A^i be the number of truth cells annotated by the expert in image I_i . Let the number of pixels of each recognized cell $R_d^i, d = 1, \dots, D^i$ be called P_d^i . Similarly, let the number of

pixels of each annotated cell R_a^i , $a = 1, \dots, A^i$ be called P_a^i . Let $O_{da}^i = R_d^i \cap R_a^i$ be the number of overlapped pixels between R_d^i and R_a^i . Thus, if there is no overlap between two regions, $O_{da}^i = \emptyset$, while if the overlap is complete, $O_{da}^i = R_a^i = R_d^i$. Let a threshold T be a measure for the strictness of the overlapping ($0.5 \leq T \leq 1$). A region can be classified into the following types:

1. A pair of regions R_d^i and R_a^i is classified as an instance of correct detection if $O_{da}^i \geq P_d^i T$ (at least percentage T of the pixels of region R_d^i in the detected image overlap with R_a^i) and $O_{da}^i \geq P_a^i T$ (at least percentage T of the pixels of region R_a^i in the annotated image overlap with R_d^i).
2. A region R_a^i that does not participate in any instance of correct detection is classified as missed.
3. A region R_d^i that does not participate in any instance of correct detection is classified as noise.

CHAPTER 3

RESULTS AND DISCUSSION

The content of the following paper is partially reproduced in this chapter. The publication details are:

A. Mbaidin, E. Cernadas, ZA. Al-Tarawneh, M. Fernández-Delgado, R. Domínguez-Petit, S. Rábade-Uberos, A. Hassanat. **MSCF: Multi-Scale Canny Filter to Recognize Cells in Microscopic Images**. *Sustainability*. Special issue *Latest Applications of Computer Vision and Machine Learning Techniques for Smart Sustainability* . 2023; 15(18):13693. <https://doi.org/10.3390/su151813693>. ISSN 2071-1050.

whose authors are:

1. A. Mbaidin from the Computer Science Department, Mutah University, Karak 61711, Jordan.
2. E. Cernadas from Centro Singular de Investigación en Tecnoloxías Intelixentes da USC (CiTIUS), Universidade de Santiago de Compostela, Spain.
3. ZA. Al-Tarawneh from the Computer Science Department, Mutah University, Karak 61711, Jordan.
4. M. Fernández-Delgado from Centro Singular de Investigación en Tecnoloxías Intelixentes da USC (CiTIUS), Universidade de Santiago de Compostela, Spain.
5. R. Domínguez-Petit from the Instituto Español de Oceanografía (IEO, CSIC), Centro Oceanográfico de Vigo, Vigo, Spain.

6. S. Rábade-Uberos from the Instituto de Investigaciones Marinas (IIM, CSIC), Vigo, Spain.
7. A. Hassanat from the Computer Science Department, Mutah University, Karak 61711, Jordan.

We evaluate the proposed algorithm to recognize cells, called Multiscale-Scale Canny Filter (**MSCF**), and compare with state-of-art segmentation algorithms on a collection of images belonging to five fish species. The creation of the datasets and the measures used to evaluate the cell recognition methods are described in sections 2.1 and 2.3 of chapter 2, respectively. The section 3.1 presents the results for different configurations of the MSCF algorithm and the section 3.2 compares the best configuration of the MSCF algorithm with other state-of-art segmentation methods.

3.1 MSCF algorithm

The MSCF algorithm described in section 2.2.1 of chapter 2 is applied to the grey level images of all datasets using the following parameters: 1) the Gaussian spread of the filter to smooth the images is some combination of the following set $\mathcal{S} = \{2, 4, 6\}$, $N_S = 3$; and 2) the rates for selecting the two thresholds to perform the hysteresis step on the gradient image is some combination of the following set $\mathcal{R} = \{(R_L, R_H)\} = \{(0.4, 0.6), (0.55, 0.75), (0.7, 0.9)\}$. In order to test the effect of varying both parameters, we develop experiments using one, two or three σ values. We also develop experiments varying the number of threshold pairs, one, two or three. We developed all the experiments using a minimum diameter $d_{min} = 100$ pixels because of it is the minimum size of matured cells for most of the fish species. The results analysing the performance of the MSCF algorithm from the point of view of pixel classification, if an image pixel belonging to a cell is assigned as a cell pixel by the segmentation algorithm, are presented in subsection 3.1.1. Since a segmentation algorithm can provide quite good results from the pixel classification point of view, but it could fail in the recognition of cells, for example, it can recognize a cell as many pixel regions. Therefore, its performance at the regional level is summarized in section 3.1.2.

hake dataset					
No. σ 's	σ value	Thresholds	Precision	Recall	F_1
1	{2}	{(0.55, 0.75)}	80.85	59.93	68.83
2	{2, 4}	{(0.55, 0.75)}	78.77	64.01	70.63
3	{2, 4, 6}	{(0.55, 0.75)}	77.67	65.72	71.20
pouting dataset					
1	{2}	{(0.55, 0.75)}	65.25	75.83	70.14
2	{4, 2}	{(0.4, 0.6)}	62.83	76.85	69.14
3	{2, 4, 6}	{(0.55, 0.75)}	62.1	75.69	68.23
pilchard dataset					
1	{4}	{(0.4, 0.6), (0.55, 0.75)}	71.7	67	69.27
2	{2, 4}	{(0.4, 0.6)}	70.5	71.61	71.05
3	{2, 4, 6}	{(0.55, 0.75)}	69.68	74.55	72.04
megrim dataset					
1	{2}	{(0.4, 0.6)}	85.84	70.25	77.27
2	{2, 4}	{(0.4, 0.6)}	85.93	75.41	80.33
3	{2, 4, 6}	{(0.4, 0.6)}	85.02	74.63	79.49
grenadier dataset					
1	{4}	{(0.55, 0.75)}	73.56	68.28	70.82
2	{2, 4}	{(0.55, 0.75)}	71.66	70.67	71.16
3	{2}	{(0.55, 0.75)}	70.15	71.58	70.86

Table 3.1: Precision, recall and F_1 in % for the classification of pixels for datasets hake, pouting, pilchard, megrim and grenadier using $D_{\min} = 100$ pixels. The columns No. σ 's, σ value and Thresholds show the configuration providing the highest performance.

3.1.1 Performance at pixel level

Table 3.1 shows the highest value F_1 using one, two or three values of σ for each fish species. The precision and recall and the best configuration for the rates to threshold are also presented. F_1 value gives a trade-off between the capability of the algorithm to detect pixels that belong to cells without detecting pixels outside the cells. The highest F_1 ranges from 70.14% for *Trisopterus luscus* (pouting dataset) to 80.33% for *four-spot megrim* species (see column F_1 in the table). The best configuration, number of σ 's used and values of the thresholds, depend on the fish species considered, as shown in Table 3.1. Nevertheless, the differences in performance (F_1 value) are small for different numbers of σ 's.

To know the best configuration globally, we calculate the Friedman rank of the F_1 value over all configurations and species. Table 3.2 shows the first 20 best MSCF configurations (σ

values and thresholds) using a minimum diameter $D_{\min} = 100$ pixels according to the Friedman rank of F_1 . The first position is for an MSCF that needs three scales $\mathcal{S} = \{2, 4, 6\}$ and one pair of thresholds $\mathcal{TH} = \{(0.55, 0.75)\}$, in the second and third positions the MSCF algorithm uses two scales $\mathcal{S} = \{2, 4\}$ and one pair of thresholds, and until the 12th position there is any configuration only with one scale $\mathcal{S} = \{2\}$. The best performance is achieved using more than one scale, so the classical Canny filter cannot provide a satisfactory performance to this problem. Hence, the best configuration uses three scales.

Position	Rank	σ values, $\mathcal{S}$	Thresholds, $\mathcal{TH}$
1	5.8	$\{2, 4, 6\}$	$\{(0.55, 0.75)\}$
2	6.4	$\{2, 4\}$	$\{(0.4, 0.6)\}$
3	6.6	$\{2, 4\}$	$\{(0.55, 0.75)\}$
4	8.2	$\{2, 4, 6\}$	$\{(0.55, 0.75), (0.7, 0.9)\}$
5	8.4	$\{2, 4\}$	$\{(0.55, 0.75), (0.7, 0.9)\}$
6	9.6	$\{2, 4, 6\}$	$\{(0.4, 0.6)\}$
7	12.0	$\{2, 4\}$	$\{(0.4, 0.6), (0.55, 0.75)\}$
8	12.8	$\{2, 6\}$	$\{(0.4, 0.6)\}$
9	13	$\{2, 4\}$	$\{(0.4, 0.6), (0.55, 0.75), (0.7, 0.9)\}$
10	14.4	$\{2, 6\}$	$\{(0.55, 0.75)\}$
11	14.6	$\{2, 6\}$	$\{(0.55, 0.75), (0.7, 0.9)\}$
12	15	$\{2\}$	$\{(0.4, 0.6)\}$
13	15.4	$\{2, 4, 6\}$	$\{(0.4, 0.6), (0.55, 0.75)\}$
14	16.6	$\{2, 4, 6\}$	$\{(0.4, 0.6), (0.55, 0.75), (0.7, 0.9)\}$
15	17.6	$\{2\}$	$\{(0.55, 0.75)\}$
16	17.8	$\{2\}$	$\{(0.55, 0.75), (0.7, 0.9)\}$
17	17.8	$\{2\}$	$\{(0.4, 0.6), (0.55, 0.75)\}$
18	18	$\{2, 6\}$	$\{(0.4, 0.6), (0.55, 0.75)\}$
19	18	$\{2\}$	$\{(0.4, 0.6), (0.55, 0.75), (0.7, 0.9)\}$
20	18.4	$\{4, 6\}$	$\{(0.4, 0.6)\}$

Table 3.2: List of the 20 best MSCF configurations (σ values and thresholds) using minimum diameter $D_{\min} = 100$ pixels according to the Friedman rank of F_1 .

The number of false positives is sometimes rather high using the parameter $D_{\min} = 100$ pixels, and some images only contain cells quite larger than that minimum diameter. To statistically analyse this issue, we develop experiments varying the value of $D_{\min}$ from 50 pixels to 200 pixels with a step of 50 pixels, i.e. $D_{\min} = \{50, 100, 150, 200\}$. The highest performance results are presented in Table 3.3 for all the fish species. The highest performance is always achieved using a minimum diameter higher than $D_{\min} = 100$ pixels, which is the

hake dataset						
$D_{\min}$	No. σ 's	σ value	Thresholds	Precision	Recall	F_1
150	1	{2,4,6}	{(0.4,0.6),(0.55,0.75)}	75.92	64.63	69.82
150	2	{2,4}	{(0.55,0.75)}	82.78	63.61	71.94
200	3	{2,4,6}	{(0.55,0.75),(0.7,0.9)}	82.68	65.24	72.93
pouting dataset						
150	1	{2}	All	68.47	74.48	71.35
150	2	{4,2}	{(0.4,0.6)}	66.56	77.91	71.79
150	3	{2,4,6}	{(0.4,0.6)}	65.14	77.49	70.78
pilchard dataset						
150	1	{4}	All	80.57	62.62	70.47
150	2	{2,4}	{(0.4,0.6),}	77.78	72.36	74.97
150	3	{2,4,6}	{(0.4,0.6)}	76.69	74.74	75.7
megrim dataset						
150	1	{2}	{(0.4,0.6)}	89.22	69.65	78.23
150	2	{2,4}	{(0.4,0.6)}	89.23	74.73	81.34
150	3	{2,4,6}	{(0.4,0.6)}	88.27	75.08	81.14
grenadier dataset						
200	1	{4}	{(0.55,0.75)}	85.57	68.38	76.02
200	2	{2,4}	{(0.55,0.75),(0.7,0.9)}	82.32	70.34	75.86
200	3	{2,4,6}	{(0.7,0.9)}	88.72	64.98	75.01
All= {(0.4,0.6),(0.55,0.75),(0.7,0.9)}						

Table 3.3: Precision, recall and F_1 in % for the classification of pixels for datasets *hake*, *pouting*, *pilchard*, *megrim* and *grenadier* varying the value of $D_{\min}$ pixels. The columns $D_{\min} = \{50, 100, 150, 200\}$, No. σ 's and σ value show the configuration providing the highest performance.

minimum diameter set by fishing experts to the matured cells. The improvement is similar for the different species: from 71.20% to 72.93% for *European hake* species, from 70.14% to 71.79% for *Trisopterus lucus* (*pouting* dataset), from 72.04% to 75.7% for *European pilchard*, from 80.33% to 81.34% for *Four-spot megrim* and from 71.16% to 75.86% for *Roughhead grenadier* species. In this case, the best configuration is also different for each species, but the highest performance is always achieved using two or three smoothed versions of the original grey-level images (two or three values of σ).

We again calculate the Friedman rank of the F_1 value over all configurations and species. Table 3.4 shows the first 30 best MSCF configurations (σ values and thresholds) and minimum diameter $D_{\min} = \{50, 100, 150, 200\}$ according to the Friedman rank of F_1 . The 15 first positions are provided by MSCF configurations using minimum diameter higher than 100 pix-

Position	Rank	$D_{\min}$	σ values, $\mathcal{S}$	Thresholds, $\mathcal{TH}$
1	14.8	150	$\{2, 4, 6\}$	$\{(0.4, 0.6)\}$
2	16.4	150	$\{2, 4\}$	$\{(0.4, 0.6), (0.55, 0.75), (0.7, 0.9)\}$
3	18.6	150	$\{2, 4, 6\}$	$\{(0.55, 0.75), (0.7, 0.9)\}$
4	19.2	150	$\{2, 4, 6\}$	$\{(0.55, 0.75)\}$
5	19.2	200	$\{2, 4, 6\}$	$\{(0.4, 0.6)\}$
6	19.4	150	$\{2, 4, 6\}$	$\{(0.4, 0.6), (0.55, 0.75), (0.7, 0.9)\}$
7	19.8	150	$\{2, 4, 6\}$	$\{(0.4, 0.6), (0.55, 0.75)\}$
8	22.2	200	$\{2, 4, 6\}$	$\{(0.4, 0.6), (0.55, 0.75)\}$
9	22.2	200	$\{2, 4, 6\}$	$\{(0.4, 0.6), (0.55, 0.75), (0.7, 0.9)\}$
10	27.4	200	$\{2, 4\}$	$\{(0.4, 0.6), (0.55, 0.75), (0.7, 0.9)\}$
11	27.4	150	$\{2\}$	$\{(0.4, 0.6)\}$
12	29.6	150	$\{2\}$	$\{(0.4, 0.6), (0.55, 0.75), (0.7, 0.9)\}$
13	31.2	150	$\{2\}$	$\{(0.4, 0.6), (0.55, 0.75)\}$
14	31.4	200	$\{2, 4, 6\}$	$\{(0.55, 0.75), (0.7, 0.9)\}$
15	31.8	200	$\{2, 4, 6\}$	$\{(0.55, 0.75)\}$
16	40.2	100	$\{2, 4, 6\}$	$\{(0.4, 0.6)\}$
17	41.4	150	$\{4, 6\}$	$\{(0.4, 0.6), (0.55, 0.75), (0.7, 0.9)\}$
18	41.6	100	$\{2, 4, 6\}$	$\{(0.55, 0.75)\}$
19	43.6	100	$\{2, 4\}$	$\{(0.4, 0.6), (0.55, 0.75), (0.7, 0.9)\}$
20	45.2	100	$\{2, 4, 6\}$	$\{(0.55, 0.75), (0.7, 0.9)\}$
21	46.6	100	$\{2\}$	$\{(0.4, 0.6)\}$
22	47.6	150	$\{2\}$	$\{(0.55, 0.75)\}$
23	47.8	150	$\{2\}$	$\{(0.55, 0.75), (0.7, 0.9)\}$
24	49.0	100	$\{2, 4, 6\}$	$\{(0.4, 0.6), (0.55, 0.75)\}$
25	50.2	150	$\{4\}$	$\{(0.4, 0.6)\}$
26	50.4	100	$\{2, 4, 6\}$	$\{(0.4, 0.6), (0.55, 0.75), (0.7, 0.9)\}$
27	50.8	150	$\{4\}$	$\{(0.4, 0.6), (0.55, 0.75)\}$
28	50.8	150	$\{4\}$	$\{(0.4, 0.6), (0.55, 0.75), (0.7, 0.9)\}$
29	51.0	100	$\{2\}$	$\{(0.4, 0.6), (0.55, 0.75)\}$
30	51.4	100	$\{2\}$	$\{(0.4, 0.6), (0.55, 0.75), (0.7, 0.9)\}$

Table 3.4: List of the 30 best MSCF configurations (σ values and thresholds) and minimum diameter $D_{\min} = \{50, 100, 150, 200\}$ according to the Friedman rank of F_1 .

els, which is the minimum diameter of the matured cells provided by fishing experts. This fact can be due to normally the majority of matured cells are being larger than $D_{\min} = 100$ pixels, and a smaller diameter detects more noise pixels.

hake dataset		
No. σ 's	σ value	Correct (%)
1	{2}	43.65
2	{2,4}	46.85
3	{2,4,6}	47.44
pouting dataset		
1	{2}	60.62
2	{4,2}	60.56
3	{2,4,6}	59.15
pilchard dataset		
1	{4}	53.6
2	{2,4}	51.61
3	{2,4,6}	57.67
megrim dataset		
1	{2}	50.69
2	{2,4}	55.51
3	{2,4,6}	53.57
grenadier dataset		
1	{4}	59.62
2	{2,4}	64.41
3	{2,4,6}	64.25

Table 3.5: Percentage of correct recognition of cells for hake, pouting, pilchard, megrim and grenadier datasets using $D_{\min} = 100$ pixels for different numbers of σ 's.

3.1.2 Performance at region level

Due to the properties of our problem, the best algorithm from the pixel classification point of view would not be the best option. For example, imagine that the segmentation algorithm correctly detects the majority of pixels of a cell, so from the pixel point of view, its performance is very nice. However, if these correctly detected pixels are distributed into various regions, its performance would be rather bad in counting and measuring the cells, which is our goal for estimating the fish fecundity. Table 3.5 shows the highest correct recognition rate of cells, column Correct(%), with a tolerance $T = 0.7$ using one, two or three values of σ for each fish species for a minimum diameter $D_{\min} = 100$ pixels.

3.2 Comparison with other image segmentation algorithms

This section reports a comparison of the best configuration for the MSCF algorithm with other state-of-art segmentation techniques described in section 1.2 of chapter 1. Specifically, we use the k-means clustering approach (implemented by the `kmeans` function in OpenCV library¹, Mean Shift² [18], Chan Vase³ [11, 32] and region merging⁴ [28] segmentation methods.

Deep learning approaches are not considered in this comparison because they are normally used as supervised segmentation techniques and require high computational resources for the size of the images in our datasets. In addition, these are high-resolution images, so DL should be applied to image patches or down-sampled images. Neither option is acceptable in our case because 1) image patches should be large enough to include a few number of whole cells, and therefore patches would be too large for a DL setting; and 2) image down-sampling until the size needed by DL networks may significantly reduce the image information to perform cell recognition.

The configuration used for each method is as follows:

1. **1. K-means clustering:** although the simplest property to apply clustering methods would be the grey level of the pixel, we also tested the color and texture features. We studied different types of input images and two pre-processing methods to compute from the images the feature vectors to be clusterized. Specifically, the available input images are: 1) grey-level intensity images; and 2) color images, either in RGB or Lab color spaces, representing each pixel by a vector with the intensities in the three color channels. The available pre-processing methods are as follows:
 - **Pixel intensity:** For grey-level images, we use the intensity of the pixel as the feature vector. For color images, using both RGB and Lab color spaces, the feature vector contains three values corresponding to the intensity value of the pixel for each color channel.
 - **Local Binary Patterns (LBP):** LBP is a well-known texture feature [65]. The LBP code describes each pixel by comparing its value with the neighboring pixels. For each neighboring pixel, the result will be set to one if its value is higher

¹<http://opencv.org>

²Authors code: <http://coewww.rutgers.edu/riul/research/code/EDISON/>

³<http://dx.doi.org/10.5201/ipol.2012.g-cv>

⁴Authors code: <http://cs.brown.edu/~pff/segment/>

than the value of central pixel, otherwise the result will be set to zero, developing a binary code for each pixel. We use the uniform LBP, which considers the binary patterns with only two transitions (from 0 to 1 and vice versa). In a circularly symmetric neighbor set of P pixels can occur $P + 1$ uniform binary patterns. The number of “1’s” in the binary pattern is the label of the pattern, while the nonuniform patterns are labelled by $P + 1$. The neighboring pixels can be placed at different distances or radius R . The most common values for these parameters are $\{(P, R) \in (8, 1), (12, 2), (16, 3)\}$, where P is the number of neighbors and R is the distance between the central pixel and the neighbors. The LBP code for each pixel is a vector of $P + 1$ values for each pixel in a grey-level image. This LBP code is applied to each color channel, resulting in a vector of $3(P + 1)$ values for each pixel.

- **Patch:** an image patch [90] in the neighborhood of the pixel is often used as a texture description with three patch sizes $s = 3, 5, 7$. We calculate the patch descriptors for grey level and color images. The patch code will be a vector of s^2 and $3s^2$ values for grey level and color images, respectively.
2. **MeanShift:** its configuration parameters are the spatial bandwidth, $sbw = \{8, 16\}$ and the range bandwidth (rbw) with values from 4 to 20 with a step of 2.
 3. **Chan-Vase:** the configuration parameter is the length $\mu = \{0.20, 0.25\}$.
 4. **Felzenszwalb method:** it is a method for a graph-based representation of the image segmentation based on pairwise region comparison. its configuration parameters are the spread σ to smooth the input image before segmentation with values from 0.1 to 0.9 using a step of 0.1 and the k value to threshold the function with values from 250 to 1000 using a step of 250.
 5. **Govocitos method:** this approach applies the Canny filter with 1) several scales, which are different values from the spread $\sigma = \{2, 4, 6\}$ of a Gaussian filter; and 2) different pairs of threshold rates to do the hysteresis process, $(T_{\text{high}}, T_{\text{low}}) = \{(0.4, 0.6), (0.55, 0.75), (0.6, 0.9)\}$.
 6. **MSCF:** the best configuration using the set of σ -values, $S = \{2, 4, 6\}$, and the set of pairs of thresholds, $\mathcal{R} = \{(R_L, R_H)\} = \{(0.4, 0.6), (0.55, 0.75), (0.7, 0.9)\}$.

hake dataset				
Algorithm	Configuration	Precision	Recall	F_1
Clustering	Color RGB WP	79.61	80.1	79.86
MeanShift	$sbw = 8, rbw = 4$	68	75.3	71.46
Chan-Vase	$\mu = 0.2$	72.24	63.64	67.67
Felzenszwalb	$\sigma = 0.5, k = 750$	68.63	74.62	71.5
Govocitos	–	70.87	65.6	68.13
MSCF	$\mathcal{S} = \{2, 4, 6\}, \mathcal{TH} = \{2\}$	77.67	65.72	71.2
pouting dataset				
Algorithm	Configuration	Precision	Recall	F_1
Clustering	Color LAB WP	56.93	63.73	60.14
MeanShift	$sbw = 8, rbw = 6$	48.93	77.85	60.09
Chan-Vase	$\mu = 0.2$	50.3	43.42	46.6
Felzenszwalb	$\sigma = 0.7, k = 250$	47.47	82.9	60.37
Govocitos	–	62.37	74.29	67.81
MSCF	$\mathcal{S} = \{2\}, \mathcal{TH} = \{1, 2, 3\}$	65.25	75.83	70.14
pilchard dataset				
Algorithm	Configuration	Precision	Recall	F_1
Clustering	Grey RGB WP	61.32	83.83	70.83
MeanShift	$sbw = 8, rbw = 1$	64.24	80.93	71.63
Chan-Vase	$\mu = 0.2$	59.35	74.24	65.97
Felzenszwalb	$\sigma = 0.7, k = 250$	54.22	79.22	64.38
Govocitos	–	68.63	72.12	70.33
MSCF	$\mathcal{S} = \{2, 4, 6\}, \mathcal{TH} = \{2\}$	69.68	74.55	72.04

WP = without pre-processing, $\mathcal{TH} = \{1 = (0.4, 0.6), 2 = (0.55, 0.75), 3 = (0.7, 0.9)\}$

Table 3.6: Precision, recall and F_1 in % for the classification of pixels for datasets hake (upper rows), pouting (middle rows) and pilchard (lower rows) using $D_{\min} = 100$ pixels. The last column *Correct* is the percentage of cells correctly recognized in the image. The algorithm configuration providing the highest performance is shown in the “Configuration” column.

The output of the MSCF algorithm is a set of contours $\mathcal{C}$ associated with the outline of the cells. The output of the remaining algorithms is a binary image I_B with cells and background. In the last case, the set of contours $\mathcal{C}$ is extracted from the I_B using the Suzuki algorithm [76]. The experts provide a minimum diameter for the matured cells, $D_{\min}$, which depends on the fish species and the spatial resolution at which the images are acquired. In our case, $D_{\min} = 100$ pixels for all fish species. At the same time, the cells are always rounded, so we assume that their roundness is lower than a value $R_{\max}$. We consider $R_{\max} = 1.2$, slightly

above 1, that is the circle roundness. Finally, in all algorithms the contours $c_i \in \mathcal{C}$ are filtered by size and roundness.

megrim dataset				
Algorithm	Configuration	Precision	Recall	F_1
Clustering	Color LAB WP	84.99	69.11	76.23
MeanShift	$sbw = 1, rbw = 4$	74.21	82.23	78.02
Chan-Vase	$\mu = 0.2$	76.48	56.66	65.09
Felzenszwalb	$\sigma = 0.5, k = 250$	74.29	73.53	73.91
Govocitos	–	84.42	72.42	77.96
MSCF	$\mathcal{S} = \{2, 4\}, \mathcal{TH} = \{1\}$	85.93	75.41	80.33

grenadier dataset				
Algorithm	Configuration	Precision	Recall	F_1
Clustering	Grey RGB WP	63.78	59.82	61.74
MeanShift	$sbw = 8, rbw = 8$	69.59	67.83	68.7
Chan-Vase	$\mu = 0.2$	54.48	36.27	43.55
Felzenszwalb	$\sigma = 0.5, k = 750$	76.32	66.73	71.2
Govocitos	–	67.06	67.79	67.43
MSCF	$\mathcal{S} = \{2, 4\}, \mathcal{TH} = \{2\}$	71.66	70.67	71.16

WP = without pre-processing, $\mathcal{TH} = \{1 = (0.4, 0.6), 2 = (0.55, 0.75), 3 = (0.7, 0.9)\}$

Table 3.7: Continuation of table 3.6 for megrim (upper rows) and grenadier (lower rows) datasets.

$D_{\min} = 100$			All $D_{\min}$		
Position	Rank	Algorithm	Position	Rank	Algorithm
1	1.8	MSCF	1	1.4	MSCF
2	3	MeanShift	2	2.4	Govocitos
3	3.4	Clustering	3	3	MeanShift
4	3.4	Felzenszwalb	4	3.8	Clustering
5	3.6	Govocitos	5	4.6	Felzenszwalb
6	5.8	Chan-Vase	6	5.8	Chan-Vase

Table 3.8: Friedman ranking of the diferent segmentation algorithms using the F_1 quality measure: using $D_{\min} = 100$ (left columns) and for the $D_{\min} = (50, 100, 150, 200)$ (right columns).

Tables 3.6 and 3.7 show the performance (classification of pixels) of all algorithms for the different fish species using the same minimum diameter, $D_{\min} = 100$ pixels, which is the parameter used by the fishing experts in their diary work to distinguish inmaturd from mature cells. Comparing the segmentation algorithms tested, the highest F_1 value is provided by the MSCF algorithm for all fish species ($F_1 = 70.14\%$ for *Trispterus luscus* (pouting dataset),

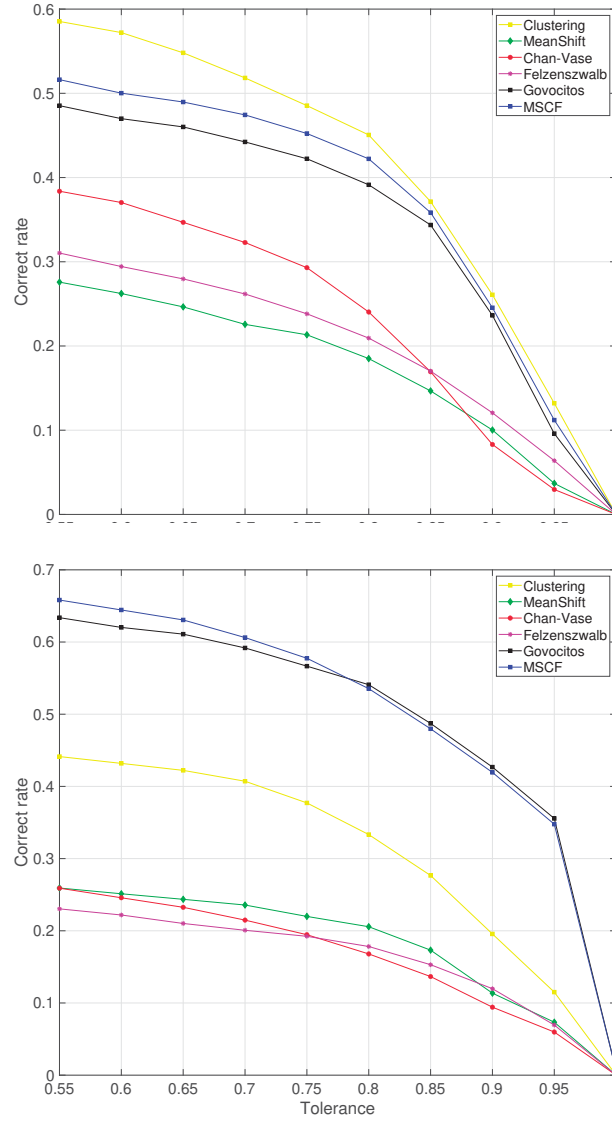


Figure 3.1: Correct rate recognition cells for different values of tolerance for the fish species *European hake* (upper panel) and *Trisopterus luscus* (lower panel) for the segmentation algorithms tested using the minimum diameter for cells of $D_{min} = 100$ pixels.

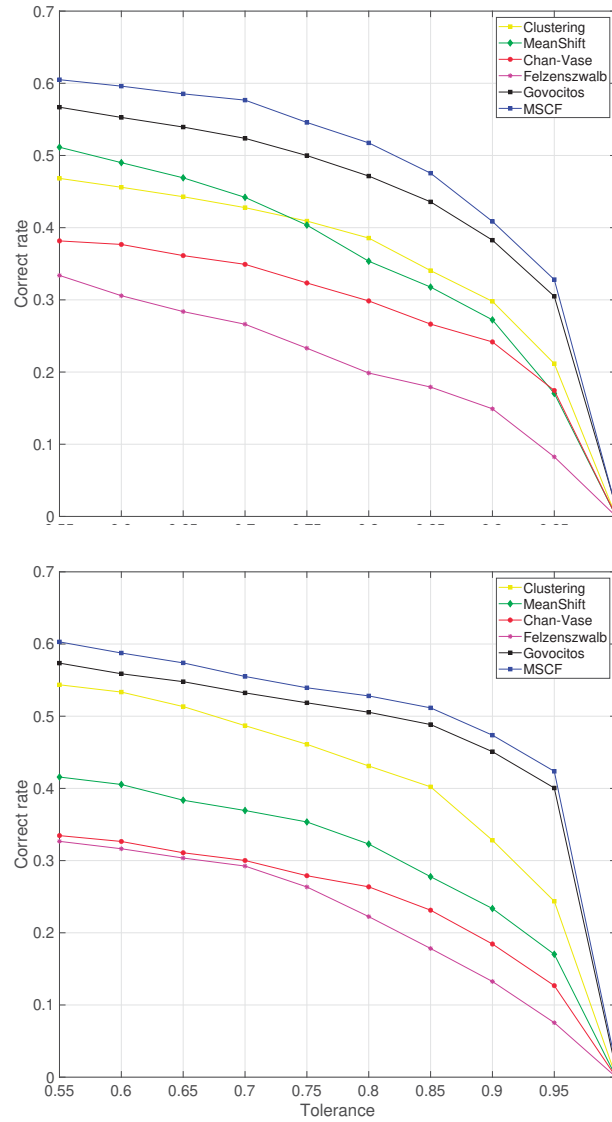


Figure 3.2: Correct rate recognition cells for different values of tolerance for the fish species *European pilchard* (upper panel) and *Four-spot megrim* (lower panel) for the segmentation algorithms tested using the minimum diameter for cells of $D_{min} = 100$ pixels.

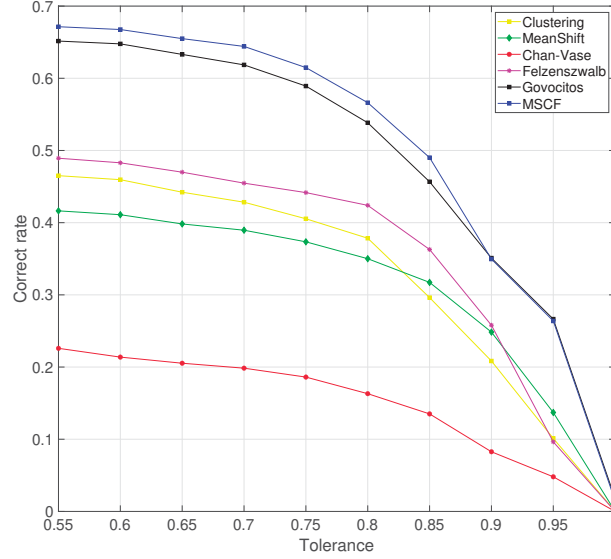


Figure 3.3: Correct rate recognition cells for different values of tolerance for the fish species *Roughhead grenadier* for the segmentation algorithms tested using the minimum diameter for cells of $D_{min} = 100$ pixels.

72.04 for *European pilchard*, 80.33 for *Four-spot megrim* and 71.16 for *Roughhead grenadier*) except for the *European hake* specie, where the best performance is provided by clustering one ($F_1 = 79.86\%$). The differences from the best algorithm to the poorest one are normally less than 10 points for all fish species: from 67.67% of Chan-Vase to 79.86% for clustering and *European hake* species, from 46.6% of Chan-Vase to 70.14% for MSCF and pouting, from 65.97% of Chan-Vase to 72.04 for MSCF and *European pilchard* specie, from the 65.09% of Chan-Vase to 80.33% for MSCF and *Four-spot megrim* species and from the 43.55% for the Chan-Vase to 71.16 for MSCF and *Roughhead grenadier* species. Therefore, the Chan-Vase is the worst segmentation algorithm for this problem. The rightest column of both tables (column *Correct* shows the average correct recognition rate of cells with a tolerance $T = 0.7$. This information is amplified in the figures from 3.1 to 3.3, which show the variation in the average percentage of correct recognition cells for different values of tolerance for all the fish species. In relation to the correct cell recognition rate, the proposed algorithm achieved the best results followed by the Govocitos algorithm, except to the *European hake* fish specie, in which the clustering provided the best results. The correct rate decreases as a stricter overlap

hake dataset					
Algorithm	$D_{\min}$	Configuration	Precision	Recall	F_1
Clustering	100	Color RGB WP	79.61	80.1	79.86
MeanShift	200	$sbw = 8, rbw = 4$	69.32	84.91	76.33
Chan-Vase	150	$\mu = 0.2$	73.32	65.14	68.99
Felzenszwalb	150	$\sigma = 0.5, k = 750$	70.94	72.18	71.56
Govocitos	–	–	75.84	68.34	71.9
MSCF	200	$\mathcal{S} = \{2, 4, 6\}, \mathcal{TH} = \{2, 3\}$	82.68	65.24	72.93
pouting dataset					
Algorithm	$D_{\min}$	Configuration	Precision	Recall	F_1
Clustering	150	Color RGB WP	58.73	64.33	61.4
MeanShift	150	$sbw = 1, rbw = 6$	49.94	86.09	63.21
Chan-Vase	150	$\mu = 0.2$	53.11	46.62	49.66
Felzenszwalb	150	$\sigma = 0.7, k = 250$	50.78	80.75	62.35
Govocitos	–	–	65.69	76.85	70.83
MSCF	150	$\mathcal{S} = \{4, 2\}, \mathcal{TH} = \{1\}$	66.56	77.91	71.79
pilchard dataset					
Algorithm	$D_{\min}$	Configuration	Precision	Recall	F_1
Clustering	100	Grey RGB WP	61.32	83.83	70.83
MeanShift	100	$sbw = 8, rbw = 1$	64.24	80.93	71.63
Chan-Vase	100	$\mu = 0.2$	59.35	74.24	65.97
Felzenszwalb	100	$\sigma = 0.7, k = 250$	54.22	79.22	64.38
Govocitos	–	–	76.3	73.76	75.01
MSCF	150	$\mathcal{S} = \{2, 4, 6\}, \mathcal{TH} = \{1\}$	76.69	74.74	75.7

WP = without pre-processing, $\mathcal{TH} = \{1 = (0.4, 0.6), 2 = (0.55, 0.75), 3 = (0.7, 0.9)\}$

Table 3.9: Precision, recall and F_1 in % for the classification of pixels for datasets *hake* (upper rows), *pouting* (middle rows) and *pilchard* (lower rows) for the best minimum diameter (column $D_{\min}$). The last column *Correct* is the percentage of cells correctly recognized in the image. The algorithm configuration providing the highest performance is shown in the “Configuration” column.

between the computer recognized and expert annotated cell is required, i.e., for higher values of tolerance T , being practically zero when we demand a perfect cell overlap. The best performance is provided by the MSCF algorithm for all datasets except the *European hake* dataset. Experts believe that an overlap of 70% could be acceptable for practical purposes. For that value, the percentage of cells correctly recognized is 51.83% for the *European hake* fish species and clustering segmentation algorithm and 60.62%, 57.67%, 55.55% and 64.41% for the MSCF algorithm and the *pouting*, *pilchard*, *megrin* and *grenadier* datasets respectively. The performance is different for all fish species, ranging from 51.83% for *hake* to 64.41% for the *grenadier* dataset.

To find the best technique globally, the segmentation algorithms are sorted using the Friedman rank. Table 3.8 shows the Friedman ranking for the different algorithms using the F_1 quality measure using the expert’s parameter $D_{\min} = 100$ pixels and varying the value of $D_{\min}$,

megrim dataset					
Algorithm	$D_{\min}$	Configuration	Precision	Recall	F_1
Clustering	100	Color LAB WP	84.99	69.11	76.23
MeanShift	150	$sbw = 8, rbw = 4$	75.32	83.55	79.22
Chan-Vase	150	$\mu = 0.2$	78.55	55.89	65.31
Felzenszwalb	100	$\sigma = 0.5, k = 250$	74.29	73.53	73.91
Govocitos	–	–	87.96	73.93	80.34
MSCF	150	$\mathcal{S} = \{2, 4\}, \mathcal{TH} = \{1\}$	89.23	74.73	81.34
grenadier dataset					
Algorithm	$D_{\min}$	Configuration	Precision	Recall	F_1
Clustering	200	Grey RGB WP	75.69	60.58	67.3
MeanShift	150	$sbw = 8, rbw = 8$	74.06	69.41	71.66
Chan-Vase	200	$\mu = 0.2$	60.37	41.55	49.22
Felzenszwalb	200	$\sigma = 0.9, k = 500$	78.11	70.03	73.85
Govocitos	–	–	78.14	70.02	73.86
MSCF	200	$\mathcal{S} = \{4\}, \mathcal{TH} = \{2\}$	85.57	68.38	76.02

WP = without pre-processing, $\mathcal{TH} = \{1 = (0.4, 0.6), 2 = (0.55, 0.75), 3 = (0.7, 0.9)\}$

Table 3.10: Continuation of table 3.9 for megrim (upper rows) and grenadier (lower rows) datasets.

$D_{\min} = 100$			All $D_{\min}$		
Position	Rank	Algorithm	Position	Rank	Algorithm
1	1.2	MSCF	1	1.6	MSCF
2	2.2	Govocitos	2	1.8	Govocitos
3	3	Clustering	3	3	Clustering
4	4.4	MeanShift	4	4.4	MeanShift
5	4.8	Felzenszwalb	5	4.8	Felzenszwalb
6	5.4	Chan-Vase	6	5.4	Chan-Vase

Table 3.11: Friedman ranking of the diferent segmentation algorithms using the correct cell recognition as a quality measure: using $D_{\min} = 100$ (left columns) and for the $D_{\min} = (50, 100, 150, 200)$ (right columns).

i.e., segmentation at pixel level. In both cases, the MSCF achieves the first position with ranks of 1.8 and 1.4, respectively, and the Chan-Vase algorithm achieves the last position with a rank of 5.8. The Govocitos algorithm is the fifth algorithm using $D_{\min} = 100$ and the second algorithm otherwise. For the remaining algorithms, the following best algorithm is MeanShift, followed by Clustering and Felzenszwalb.

Tables 3.9 and 3.10 show the best configuration, changing not only all the algorithm's configuration, but also the value of the minimum diameter. In the experimentation, we used $D_{\min} = \{50, 100, 150, 200\}$ pixels. The highest performance (F_1 value) increases slightly for all fish species except to the *European hake* one: from 70.14% to 71.79% for *Trisopterus luscus* using MSCF algorithm and $D_{\min} = 150$ pixels; from 72.04% to 75.7% for *European pilchard*

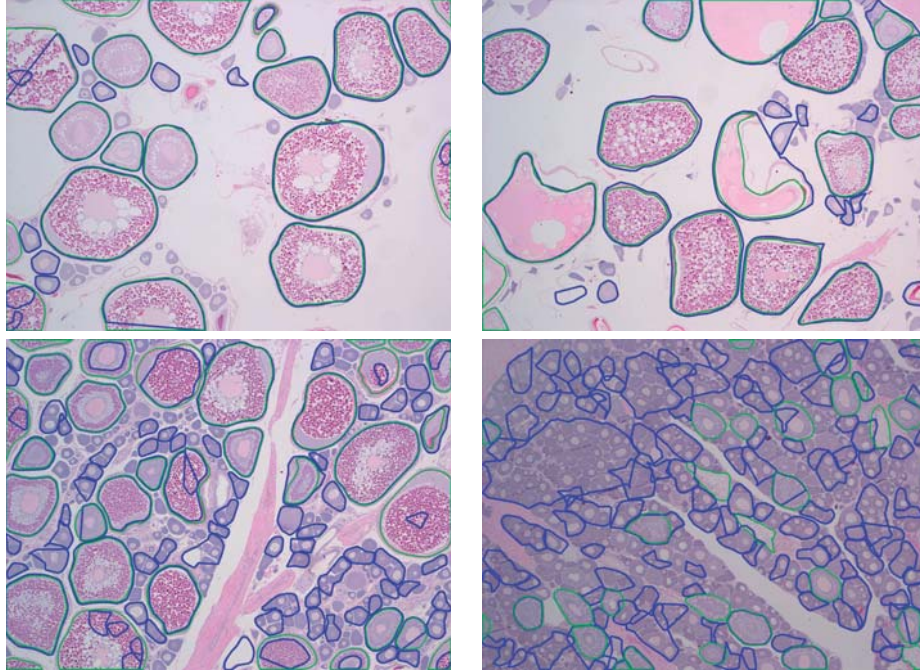


Figure 3.4: Examples of histological images of *European Hake* fish species with the outline of the ground truth cells (in green color) and computer recognized cells (in blue color) overlapped. Image g1-03-mz-4-13d (upper left), g1-03-mz-4-24c (upper right), g1-04-mz-40-32b (lower left) and g1-03-mz-26-7b (upper right).

using MSCF algorithm and $D_{\min} = 150$ pixels; from 80.33% to 81.34% for *Four-spot megrim* using MSCF algorithm and $D_{\min} = 150$ pixels; and from 71.16% to 76.02% for *Roughhead grenadier* using MSCF algorithm and $D_{\min} = 200$ pixels.

Repeating the comparison at the region level and considering the correct cell recognition at a tolerance of 0.70 as the quality measure, table 3.11 presents the Friedman ranking for all algorithms using $D_{\min} = 100$ (left columns) and for $D_{\min} = (50, 100, 150, 200)$ (right columns). In both cases, the MSCF algorithm achieves the first position with ranks of 1.2 and 1.6 respectively. The ranking of the techniques is equal for both cases. Therefore, the second algorithm is Govocitos, followed by Clustering, Meanshift, Felzenszwalb and Chan-Vase.

Figures 3.4 to 3.8 show visual examples of the performance of the MSCF algorithm for all the fish species considering a tolerance of 0.7. For example, for *European Hake* fish species

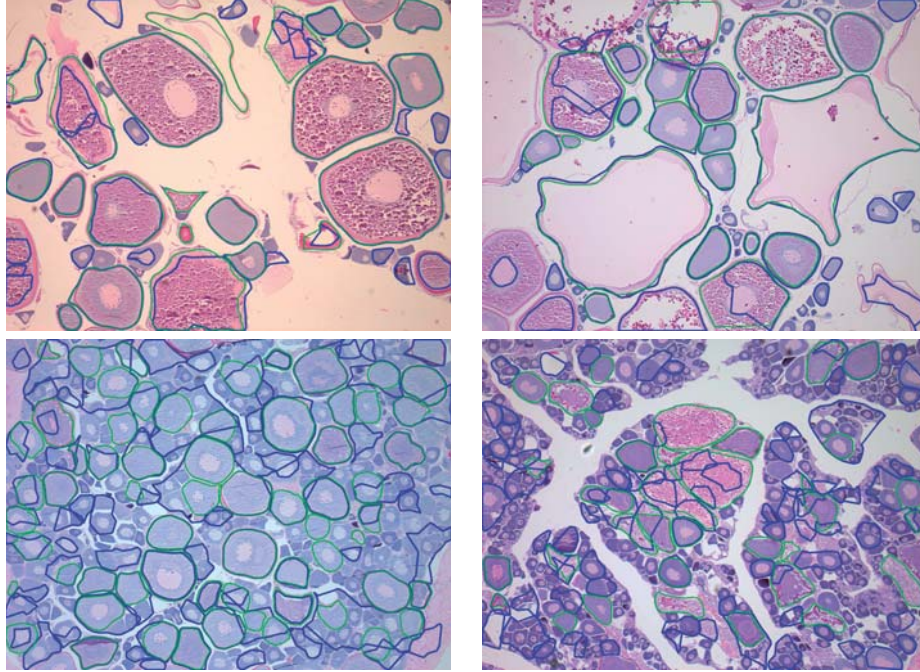


Figure 3.5: Examples of histological images of *Trisopterus Luscus* fish species with the outline of the ground truth cells (in green color) and computer recognized cells (in blue color) overlapped. Image g1-05-t1-18-6-d (upper left), g1-06-t1-32-3-c (upper right), g1-05-t1-30-31-a (lower left) and g1-06-t1-39-13c (upper right).

(see figure 3.4), the correct detection rate for images g1-03-mz-4-13d, g1-03-mz-4-24c, g1-04-mz-40-32b and g1-03-mz-26-7b are 90%, 77.78%, 72% and 19% respectively. For the first two images (the upper row in the figure 3.4), almost all matured cells are recognized and the errors are mainly due to the detection of inmaturred cells, which are counted as false positive. The third image, g1-04-mz-40-32b, follows similar behaviour, but detection more inmaturred cells and then more false positives. In the last image, g1-03-mz-26-7b, when almost all cells are inmaturred, the MSCF algorithm works very bad due to the cell contours are poorly defined. Then, MSCF detects many false positives cells and misrecognizes the true or matured cells.

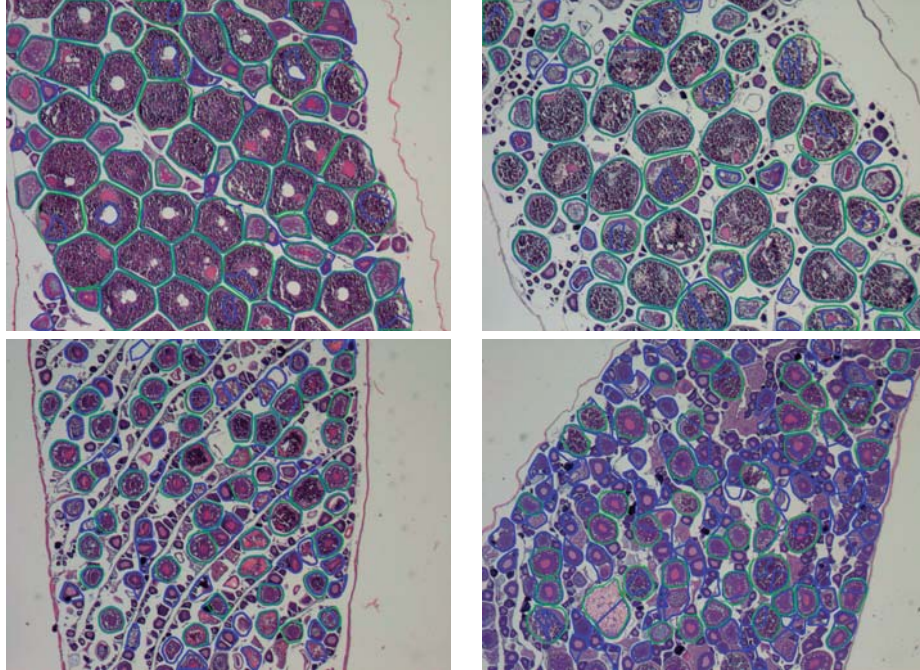


Figure 3.6: Examples of histological images of *European Pilchard* fish species with the outline of the ground truth cells (in green color) and computer recognized cells (in blue color) overlapped. Image nm2b (upper left), mn2 (upper right), pof-atr (lower left) and pof (upper right).

For *Trisopterus Luscus* fish species (see figure 3.5), the correct detection rate for images g1-05-t1-18-6-d, g1-06-t1-32-3-c, g1-05-t1-30-31-a and g1-06-t1-39-13c are 88%, 60%, 71% and 36% respectively. In these examples, the main causes of error are:

- The detection of immature cells, which are cells perfectly detected by the algorithm but that have not been annotated by the experts, probably due to their size. It should be noted that the development process of a cell is a continuum. So the difference between immature and mature is very narrow and may depend on the expert or even on the majority cells in the image.
- The MSCF algorithm missrecognize atretic cells, which already has broken their cell wall and will be reabsorbed by the gonad. See some missrecognized cells in the lower

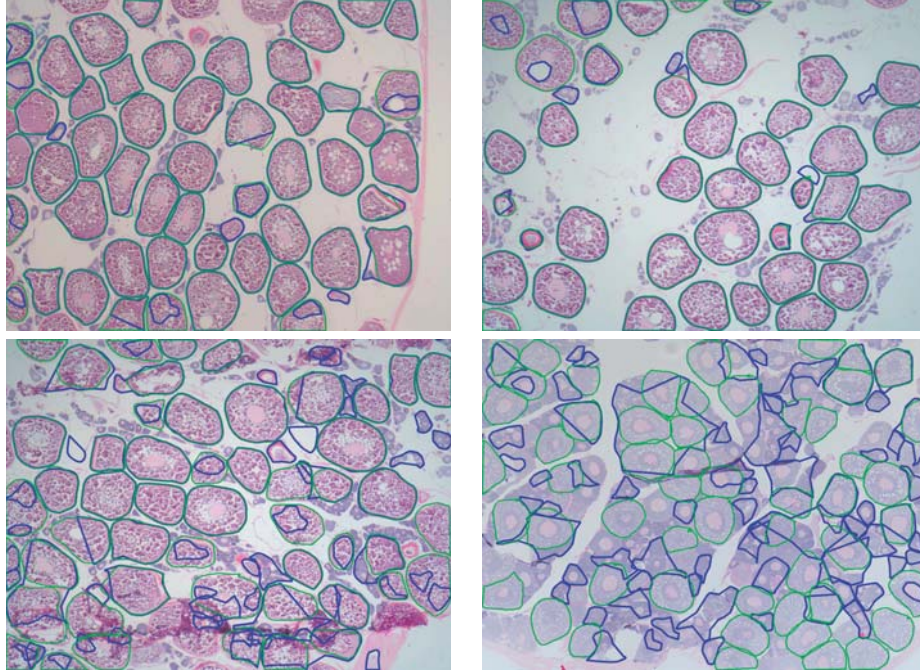


Figure 3.7: Examples of histological images of *Four-spot Megrin* fish species with the outline of the ground truth cells (in green color) and computer recognized cells (in blue color) overlapped. Image g1-3-8-1b-a (upper left), g1-07-1b-3-9 (upper right), g1-07-1b-3-15 (lower left) and g1-07-1w-2-6-ac (upper right).

right image of figure 3.5, drawn in green color in the image, but not in blue color. This fact is due to the MSCF algorithm is an edge-based method.

For *European Pilchard* fish species (see figure 3.6), the correct detection rate for images nm2b, mn2, pof-atr and pof are 87%, 71.7%, 73% and 43% respectively. This species exhibits the same behavior as previous species, detecting immature cells. The lower right image of the figure figure 3.6 shows a smaller number of automatically recognized cells because the cells are closer together and it is more difficult to correctly identify their contours.

For *Four-spot Megrin* fish species (see figure 3.7), the correct detection rate for images g1-3-8-1b-a, g1-07-1b-3-9, g1-07-1b-3-15 and g1-07-1w-2-6-ac are 95%, 79.49%, 80% and 37% respectively. In the upper images of the figure 3.7, the automatic recognition of matured cells is almost perfect with very low false positives. In the bottom right

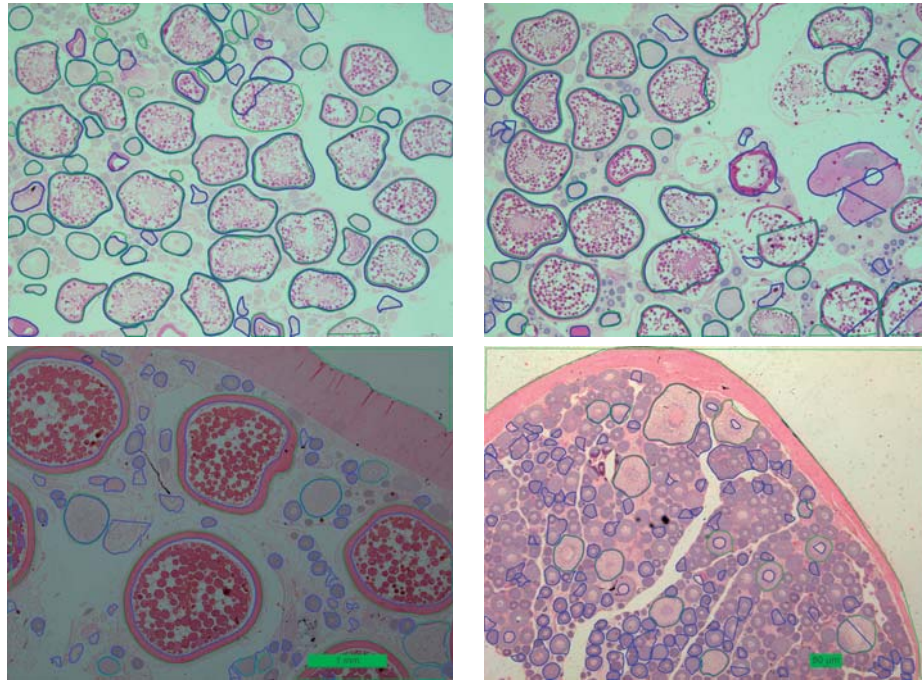


Figure 3.8: Examples of histological images of *Roughhead Grenadier* fish species with the outline of the ground truth cells (in green color) and computer recognized cells (in blue color) overlapped. Image Mb-F0813-051b (upper left), Mb-F0813-050a (upper right), Mb-FC05-039a (lower left) and Mb-F6112-069 (upper right).

image, the performance is much worse due to mainly the presence of immature cells in the image.

For *Roughhead Grenadier* fish species (see figure 3.8), the correct detection rate for images Mb-F0813-051b, Mb-F0813-050a, Mb-FC05-039a and Mb-F6112-069 are 72.58%, 88.24%, 73.33% and 60% respectively. Despite the variability between images that exists in this species, the behavior of the MSCF algorithm is quite robust, automatically recognizing a significant number of mature cells. The main problem is, as in the previous species, the high number of recognized immature cells, especially in the lower right image of the figure 3.8.

The visual examples of the MSCF algorithm working in Figures 3.4 to 3.8 suggest that the main errors in automatic cell recognition are related to the presence of many *atretic* (matured

cell in the development stage *atretic*) or immature cells in the image. Cells in the *atretic* development stage are normally counted as in the *vitelline* stage of development by experts, so it is necessary to recognize and classify them. To improve the quality of automatic recognition of cells in the image, in the chapter 4, we propose to perform this process in two steps, first select a larger minimum diameter and, if the expert considers that there are still missrecognized cells, overlap the cell detection by MSCF algorithm using a lower cell diameter.

Although the performance of MSFC algorithm is moderate at both pixel segmentation level, using the F_1 -value measure, and region or cell level, using the correct rate of cell recognition, there are significant differences among the images within fish species and between different fish species. Figure 3.9 shows both metrics, F_1 and correct rate cell recognition, for each histological image within each fish species. In all fish species, there are images that the automatic cell recognition method is very poor and others in which the recognition is quite good, suggesting the great diversity among image within each species. Note that the experts selected the specimens to analyze in order to reflect the diversity between specimens and images. These results show the difficulty of the problem to design a fully automatic method to recognize cells in histological images.

3.3 Computational time

All the experiments were performed on a computer equipped by 32 GB of RAM and a 3.10 GHz processor under Linux Kubuntu 20.04. The average elapsed time of the segmentation algorithms per image, using the configuration that provides the best performance for each fish species is shown in Table 3.12. The fastest algorithm is MSCF with 2.69 s, but it followed very closely by Felzenszwalb, Govocitos and Clustering, which need only a few seconds to process an image. In fact, these methods are suitable for image processing in interactive applications. However, the remaining algorithms are too slow to be included in real-time applications because they need times above 1 minute to process an image.

3.4 Conclusions

In section 3.1 of chapter 2, we propose a new segmentation algorithm based on the Canny edge-detector. Our approach is called MSCF (Multi-scale Canny filter) and it was statistically evaluated to recognize cells in histological images of fish ovarian. The classical Canny edge detector combines the following steps: 1) smoothing of the grey-level image using a

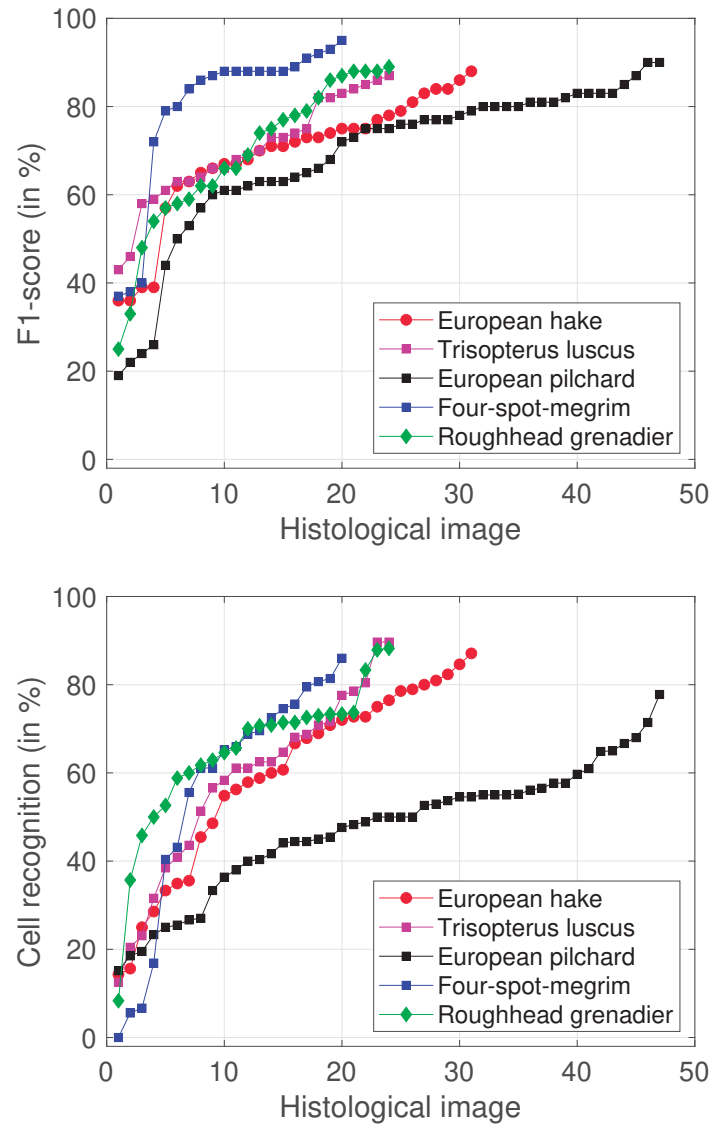


Figure 3.9: Performance of the best segmentation algorithm for each image and all fish species: F1 at pixel level (upper graph) and correct rate recognition cells using the minimum diameter for cells of $D_{min} = 100$ pixels.

Algorithm	Time in seconds	standard desviation
MSCF	2.69	0.63
Govocitos	3.83	0.20
Clustering	4.61	7.14
MeanShift	133.31	119.85
Felzenszwalb	2.97	0.16
Chan-Vase	165.04	108.38

Table 3.12: Average computational time spent to process an histological image of fish ovary for each segmentation algorithm. The last column shows the standard desviation

σ -width Gaussian kernel; 2) differentiate using a first derivative operator; 3) perform the non-maxima suppression process, which finds local maxima in the direction perpendicular to the edge; and 4) perform thresholding with hysteresis. MSCF considers different scales using a set of σ -values. Thus, the thresholds used to do the hysteresis process are selected automatically from the image characteristics. Finally, we design a fusion process with information from different scales, in order to provide a unique instance for each cell. Specifically, we use five datasets of histological images of fish ovaries annotated by experts in fish-ery, which are available at <https://gitlab.citius.usc.es/eva.cernadas/sterappimagesdb>. The statistical evaluation was performed from the pixel calssification and region recognition points of view using the F_1 and percentage of cells correctly recognized, respectively.

For our MSCF approach, the highest values of F_1 ranged from 70.14% to 80.33% for the different fish species. The average percentage of correctly recognized cells requiring a 70% overlap between the annotated and detected cells, i.e., a tolerance of 0.7, ranges from 47.44% for *European hake* dataset to 64.25% for *Roughhead-grenadier* one. Different configurations of MSCF algorithms provide the best performance for different fish species. Globally, the best configuration provided by the Friedman rank is the combination of three scales, $\mathcal{S} = \{2, 4, 6\}$, and the pair of thresholds $\mathcal{TH} = \{(0.55, 0.75)\}$ (rank of 5.8), followed by the use of two scales, $\mathcal{S} = \{2, 4\}$ and the pair of thresholds, $\mathcal{TH} = \{(0.4, 0.6)\}$ (rank of 6.4). The best configuration that uses only one scale is in the 12th position with a rank of 15. These results lead us to two conclusions:

1. Due to the complexity of this problem, the use of multiscalar information significantly improves the performance that can be provided using only one scale as the classical Canny edge-detector.

2. The differences in performance of using two or three scales is very small. However, the use of an additional scale significantly increases the computational time of the algorithm. So, we demonstrate that the use of two scales in the MSCF algorithm is an acceptable trade-off between performance and computational time to include this configuration in the STERapp software [59].

For these specific datasets, the increase in the minimum diameter provides the slightly highest performance results. But, we could miss true cells. Therefore, a possible solution in the STERapp software would be to perform the segmentation process in two stages: 1) recognize cells using a minimum diameter greater than the minimum diameter of the matured cells provided by experts; and 2) if that image contains small cells, recognize the cells with a smaller minimum diameter.

The MSCF algorithm was compared with five other state-of-art segmentation techniques described in section 1.2 of chapter 1 called clustering, Govocitos, MeanShift, Felzenszwalb and Chan-Vase in this chapter. The highest F_1 value was provided by the MSCF algorithm for all fish species ($F_1 = 70.14\%$ for *Trisopterus Luscus*, 72.04% for *European-pilchard*, 80.33% for *Four-spot-megrim* and 71.16% for *Roughhead-grenadier*) except for the *European hake* species, where the best performance is provided by clustering one ($F_1 = 79.86\%$). Nevertheless, MSCF algorithm achieved the first position using the Friedman rank, followed by Govocitos, Clustering, Meanshift, Felzenszwalb and Chan-Vase one.

In conclusion, our MSCF algorithm is very competitive, in computational time and performance, with other state-of-art segmentation algorithms. We verify that the problem of recognizing cells in histological images of fish gonads is very challenging and that an algorithm that can work completely automatically has not yet been proposed.

CHAPTER 4

THE STERAPP SOFTWARE

The content of the following paper is partially reproduced in this chapter. The publication details are:

A. Mbaidin, S. Rábade-Uberos, R. Dominguez-Petit, A. Villaverde, M.E. González-Rufino, A. Formella, M. Fernández-Delgado, E. Cernadas. **STERapp: semiautomatic software for stereological analysis. application in the estimation of fish fecundity.** *Electronics*, Special issue of *Applications of Computer Vision*, Vol. 10, No. 1432, pp. 1-15. ISSN 2079-9292.

whose authors are:

1. A. Mbaidin from Centro Singular de Investigación en Tecnoloxías Intelixentes da USC (CiTIUS), Universidade de Santiago de Compostela, Spain.
2. S. Rábade-Uberos from the Instituto de Investigaciones Marinas (IIM, CSIC), Vigo, Spain.
3. R. Domínguez-Petit from the Instituto Español de Oceanografía (IEO, CSIC), Centro Oceanográfico de Vigo, Vigo, Spain.
4. A. Villaverde from the Instituto Español de Oceanografía (IEO, CSIC), Centro Oceanográfico de Vigo, Vigo, Spain.
5. M. Fernández-Delgado from Centro Singular de Investigación en Tecnoloxías Intelixentes da USC (CiTIUS), Universidade de Santiago de Compostela, Spain.
6. M.E. González-Rufino from Laboratorio de Informática Aplicada (LIA2), Departamento de Informática, Campus Universitario, Universidade de Vigo, Ourense, Spain.

7. A. Formella from Laboratorio de Informática Aplicada (LIA2), Departamento de Informática, Campus Universitario, Universidade de Vigo, Ourense, Spain.
8. E. Cernadas from Centro Singular de Investigación en Tecnoloxías Intelixentes da USC (CiTIUS), Universidade de Santiago de Compostela, Spain.

The results shown in chapter 3 lead us to the conclusion that the problem of recognizing mature cells in histological images of fish ovaries cannot be automated. Therefore, our aim is to incorporate our algorithms into a software tool that simplifies the expert daily work flow to estimate fish fecundity in fishing labs.

ImageJ is the most popular tool to quantify particles from images in the biomedical field [78, 74]. It provides many common computer vision techniques to process the images before quantification and it allows plugins to be defined that may implement specific methods or customize processing routines. However, in its current configuration, ImageJ does not allow object outlines to be corrected manually in a versatile and easy way before starting the quantification. The software Govocitos [68] proposed an integrated solution to calculate fish fecundity using the Weibel grid [97, 24], overcoming the limitations of ImageJ through a friendly graphical user interface (GUI). However, Govocitos is quite difficult to install and to maintain for a non-expert user due to its complexity, mainly caused by the use of a database for collaborative work and data sharing. CystAnalyser [19] is a recent software developed by us to count and measure cysts in histological images of the liver and kidney, which also overcomes some of the limitations of ImageJ. In this chapter, we describe STERapp, an application to perform stereological analysis as an evolution of Govocitos using the technology of CystAnalyser to simplify its design and to increase both its versatility and its modularity. STERapp includes modules to automatically recognize and classify oocytes (reproductive cells) in histological images of fish ovaries. In addition, it includes a new design for the GUI to supervise the automatic recognition before stereological quantification. The recognized objects are stored in XML (extensive markup language) files are used to share and review the recognition results, and the stereological analysis itself is stored in CSV (comma separated values) files to simplify its use in other tools.

The main contributions of STERapp software are as follows:

1. Its friendly GUI simplifies the expert daily work flow to estimate the fish fecundity.
2. The automatic image processing is fast enough to operate in real time.

3. The experts can review and modify the results of the automatic oocyte recognition and classification before the image quantification step, overcoming other existing software packages such as ImageJ.
4. It allows the export of the combined results of various images.
5. It allows us to train the classifiers for different species.
6. The software is accurate, trustworthy, and easy to install for non-expert users.

The STERapp software is a desktop application that can run on a general purpose computer under the Linux and Windows operating systems. It has been written in the C++ programming language. The GUI was developed using the GTK+ (GIMP Tool Kit) library ¹. STERapp is available from <https://citius.usc.es/transferecia/software/sterapp> for research purposes. The rest of the chapter is organized as follows. First, the architecture, functionality and automatic recognition, and classification of objects of STERapp are described, along with the measurements used to evaluate the module of automatic processing to analyze the histological images of fish ovaries. Afterwards, the results are presented and discussed. Finally, the main conclusions are drawn.

4.1 STERapp architecture and functionality

STERapp is structured into two layers: the graphical user interface (GUI) and the application logic layer. Figure 4.1 shows the main window with an image loaded and displays the lateral panel of the module to analyse fish histological images. The GUI layer provides a friendly interface to draw the outline of objects and manage them, to interact with the software and to visualize data. It is composed of four modules (see below for a further description):

1. **Preferences**, which configures the preferences of the software, including colors and widths of lines to draw the objects, calibration settings, minimum and maximum size of the objects to be recognized, categories of the objects to manage, and the fish species defined in the software.
2. **Edit panel**, which manages the drawing, selecting, removing and, in general, managing the visualization of the objects and the results of the image analysis.

¹<https://www.gtk.org>

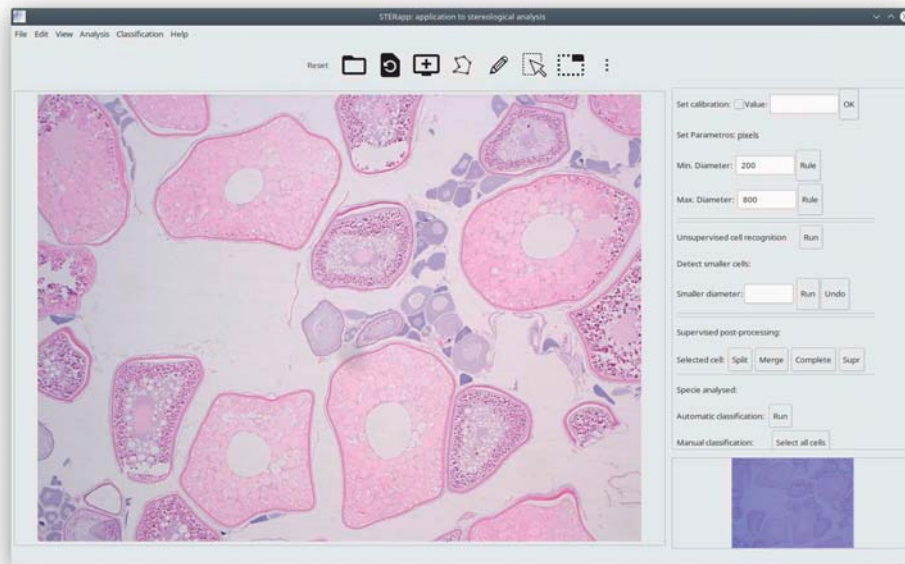


Figure 4.1: Main window of STERapp with an histological image of European hake ovary. On the right, we can see the full lateral panel showing the available functionality of the module to recognize and classify oocytes.

3. **The analysis panel** allows the options to be selected to calculate aggregated results of various images.
4. **Training panel**, which allows us to configure the classification training process. The user's guide supplied as complementary material provides a detailed description of the graphical appearance and settings of these modules.

The application layer refers to the data processing, including the following modules to automatically process the image, which will be described below in section 4.2:

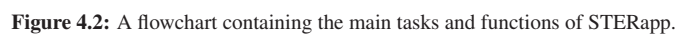
1. **Image segmentation** to automatically recognize objects in the image, i.e., cells in histological images of fish gonads.
2. **Feature extraction** to extract numerical characteristics from a recognized object in the image.

3. **Classification** automatically trains a classifier and classifies objects.
4. **Calculations** to compute the area and diameter of the objects.
5. **Data store** to save the calculated information into XML and CSV files.

The overall application is modular and can be adapted to other stereological issues, changing only the **image segmentation**, **feature extraction**, and **preferences** modules.

The main functionality provided by STERapp is shown in the flowchart of Figure 4.2. When we run STERapp the first time, we should configure the working preferences. One can work with default or preset configurations to recognize cells (oocytes) in the image, but we need to configure the species to train the classifier. By default, for the fish fecundity estimation, there are three maturation stages of oocytes configured in the preferences: *cortical alveoli*, *vitellogenic* and *hydrated*, which correspond to the mature oocytes to be counted in the fecundity calculations. STERapp allows setting the preferences such as the path of the images and other files in the system, the calibration, the working colors for each stage, the definition of new categories, e.g., such as *atresia* (oocytes that are reabsorbed before spawning), and the species to analyse. After configuring the preferences, a common use of STERapp starts loading an image in the view panel (see Figure 4.1) and doing unsupervised cell recognition by clicking on the button *Run* in the lateral panel. The detected cells are overlapped on the image in the drawing area. Due to the complexity of these types of images, the automatic cell recognition may not be perfect in the all cases. Therefore, the user can review and supervise the outlines of the cells using the tools named *supervised post-processing* in the lateral panel, which allow the user to complete, split and merge the objects having been automatically detected in the image. The user can also manually draw new cells or delete erroneously detected cells.

Once the outlines of the cells are satisfactory for the expert, the user runs the classifier to predict the categories of the objects. In our case, we used two classifications: 1) to discriminate oocytes with and without visible nucleus; and 2) to classify the three matured stages of the cells defined by default in the preferences. Again, if the automatic classification of cells is not suitable for the expert, he/she can change the categories of the cells using the functionality after the label *manual classification* in the lateral panel. The expert can also set other categories to the cells defined previously in the preferences using the drop down list named *other category* in the lateral panel. Finally, once all the objects are recognized and classified correctly after supervision by the expert, we can visualize the results or export them to XML



and CSV files. The *visualization of results* tool automatically shows information of each cell drawn on the drawing panel, allowing the user to explore the area, the diameter, or the category and whether the cell must be counted or not to the stereological analysis in real time by clicking the SHOW TABLE tool. The cells can be highlighted in the image by selecting them in the table and, vice versa, selecting a cell in the image highlights its corresponding row in the table. In addition, STERapp also allows the user to load recognition results from XML files. The XML file contains the overlays drawn on the image, which can be loaded any time later onto the image in order to check the image analysis, to share it with a colleague, or whatever. Taking in mind that in such fields as medicine or biology, there are subjectivity in the analysis among experts. The revision process is very very important in order to unify criteria among experts applying stereological analysis. The CSV file contains statistical information provided by quantitative analysis of the images:

- Spatial resolution, i.e., the relation between micrometers and pixels, at which the images were digitized (calibration), which must be pre-specified by the user.
- Information about the image analysis such as the image size in pixels, the cell index, the number of cells and the area, the diameter, the maturation stage, and if the cell has visible nucleus or not.

The user guide in the appendix B provides a more detailed description of the functionality of STERapp.

4.2 Description of modules in the application layer

As mentioned above, the application layer of STERapp is composed of following modules: image segmentation, feature extraction, classification, calculations and data store. Once the image is loaded into STERapp, the cells are automatically recognized calling the image segmentation module. After that, each cell is classified into a cell with/without visible nucleus and its development stage using the classification module. Finally, STERapp calculates some measures from the cells recognized and classified and stores the quantification analysis in XML and CSV files.

The **image segmentation** module includes the MSCF algorithm described in the section 2.2 of chapter 2. The configuration used to apply the MSCF algorithm is: the spread of a Gaussian filter $\sigma = \{0.2, 0.4\}$ and the following threshold rates to perform the hysteresis

process, $(T_{high}, T_{low}) = \{(0.4, 0.6); (0.55, 0.75); (0.6, 0.9)\}$. The maximum roundness is 1.4. The minimum diameter used in the post-processing is set by the user in the preferences of STERapp.

Once the cells are recognized, the software classifies the cells according to: 1) the presence or absence of visible nucleus in the cell (a two class problem); and 2) the development stage of the cell, considering only the three stages predefined in the preferences: *cortical alveoli*, *vitellogenic* and *hydrated*. The **classification** includes the support vector machine (SVM) classifier, one of the strongest classifiers in the literature [29]. Two SVM classifiers, one for each type of classification, are used by STERapp, both of them with radial basis function (RBF) kernel implemented by the LibSVM library [12] and accessed through its C++ binding. The input of the classifier is calculated in the **feature extraction** module, which computes a vector of features extracted from the image characteristics of the cell. Many features based on the image texture and color properties were tested in previous work [34]. For STERapp, we selected a combination of features that provided a good trade-off between the time needed for the calculation and the performance achieved by the SVM classifier. Eventually, the feature vector is composed of 1) the uniform rotation invariant local binary patterns (LBP) calculated on a gray-level version of the original image using irregular regions (10 features); and 2) the five statistics of each color channel of the RGB image into the cell ($5 \times 3 = 15$ features). The statistics used are the mean value, standard deviation, skewness, kurtosis and entropy.

The SVM classifier must be trained before predicting the class for a cell (presence/absence of visible nucleus or the development stage). STERapp allows the training of the SVM classifier for the fish species defined in the preferences. The functionality *Training panel*, through the submenu *Classification* $\rightarrow$ *Train classifier*, allows the user to set the fish species and the XML files (generated by STERapp) that are used to train the classifiers. These files must contain the objects recognized for a collection of images, alongside their classes. The collection of images should be representative enough for the classification problem, and must contain objects of all classes. The next subsection describes the training.

4.2.1 Training of classifiers

For the training, we consider a maximum number of 1,000 cells randomly selected from the XML files provided by the user, with the same number of patterns for each class whenever possible (a minimum number of objects is needed for a class to be included in the training). STERapp implements the K-fold cross-validation methodology, with $K = 4$, using training

and validation sets to tune the two hyper-parameters of the SVM, i.e., the regularization (λ) and the RBF kernel spread (σ). The performance is evaluated by the Cohen kappa statistic [61], which measures the agreement between the true and predicted category labels excluding the agreement by chance. Kappa (in %) is defined as:

$$\text{kappa} = 100 \frac{p_a - p_e}{s - p_e} \quad (4.1)$$

where

$$p_a = \sum_{i=1}^C N_{ii}$$

$$s = \sum_{i=1}^C \sum_{j=1}^C N_{ij}$$

$$p_e = \frac{1}{N^2} \sum_{i=1}^C \left(\sum_{j=1}^C N_{ij} \right) \left(\sum_{j=1}^C N_{ji} \right)$$

where C_{ij} is the number of validation patterns that belong to class i and that are assigned by the SVM to class j ; N is the total number of validation patterns. In each cross-validation trial, $K - 1$ folds (3 folds for $K = 4$) are used to train the classifier using each combination of hyper-parameter values, and the remaining fold is left for validation by evaluating kappa using the current combination of hyper-parameter values. This process is repeated K times, and the kappa on the validation fold is averaged over the K folds. The values of λ and σ used are: $\lambda = 2^i$ for $i = -3, \dots, 12$, and $\sigma = 2^i$ for $i = -15, \dots, 0$. Finally, each SVM is trained with the combination of hyper-parameter values showing the highest average kappa over the K validation sets using the entire dataset, to classify new cells.

4.3 Results and discussion for the use of STERapp

STERapp was tested by expert technicians from the Instituto de Investigaci3n Marinas (IIM) and the Instituto Espa3ol de Oceanograf3a (IEO) from the State Agency Consejo Superior de Investigaciones Cient3ficas (CSIC) since 2020 with the aim of evaluating the software operating in a real environment. The biologists use STERapp in their daily work perform stereological analysis, for which they need to recognize and classify the cells in the images. Their operations using STERapp were logged into XML files for later statistical evaluation

of the automatic computer vision and machine learning algorithms incorporated in the software. After the description of the experimental setup and the statistical measures to evaluate the software in section 4.3.1, the following subsections summarize the results achieved from different points of view: the automatic detection and classification of cells (subsection 4.3.2), the performance analysis (subsection 4.3.3), the perception of the system (subsection 4.3.4 and the comparison with ImageJ software (subsection 4.3.5).

4.3.1 Experimental setup

In the evaluation of STERapp software, the analyzed images correspond to histological sections of ovaries of several species with different reproductive strategies and oocytes characteristics: *Atlantic mackerel* (*Scomber scombrus*), *Four-spot megrim* (*Lepidorhombus bosci*), *Roughhead grenadier* (*Macrourus berglax*), *Ballan wrasse* (*Labrus bergylta*) and *European pilchard* (*Sardina pilchardus*). As mentioned, two fish fishing research centers participate in this research. Therefore, the processing of samples and the acquisition equipment are different for the fish species (see the section 2.1 of chapter 2 for the details). So, all the microscopic images used exhibit a great variability of oocyte shapes, sizes, aspects, abundance and distribution within the image, which was used to test the robustness of the software. Figures from 2.3 to 2.7 show visual examples of the histological images for each species and the table 2.1 summarizes the number of images per species and acquisition system.

As mentioned previously, to perform the statistical evaluation of the automatic recognition algorithms, STERapp registers the use of the software by the experts, saving additional information in the XML files. Let us define a true positive (TP) hit when a cell is correctly recognized and a false positive (FP) hit whenever the user manually deletes the cell using the GUI. A cell is considered a false negative (FN) if the user manually adds it. Once we counted the values for TP, FP, and FN for an image, we calculate the sensitivity (Se), specificity (Sp) and average precision (AP), in %, as:

$$Se = 100 \frac{TP}{FN + TP} \quad (4.2)$$

$$Sp = 100 \frac{TP}{FP + TP} \quad (4.3)$$

$$AP = 100 \frac{TP}{TP + FP + FN} \quad (4.4)$$

The automatic cell recognition algorithm of STERapp often provides detections close to those expected by the experts, who only use the *tools for supervision* (complete, split and

merge) to fit the true cell contour. For example, the *complete* tool is used when the contour provided is close to the true cell contour, the *split* tool is used when two cells are detected together, and the *merge* tool is used when a cell is detected as several small pieces. In all these cases, the cells were detected, but their outline was not totally perfect. STERapp only registers the cells which are modified by these tools, and not the type of tool used in each case. So, we define the ratio MC as the number of cells modified by the tools for supervision divided by the number of cells automatically recognized by the algorithm. Additionally, the scores Se, Sp and AP are updated considering that the cells have been modified using *tools for supervision* are counted as true positive, because they are correctly detected and their outline is quasi-suitable (note that if the recognition were not almost perfect, the expert would delete the cell instead of modifying it).

4.3.2 Automatic cell recognition and classification

As our purpose is to evaluate the robustness and generalization capability of the software, STERapp was used in the fisheries labs with images of the five fish species: *Atlantic mackerel*, *Four-spot megrim*, *Roughhead grenadier*, *Ballan wrasse* and *European pilchard*, which have been acquired under different conditions of sample preparation and digitization. We analyzed 132 images: 27 of *European pilchard*, 20 from the *Four-spot megrim*, 24 from the *Roughhead grenadier*, 38 from the *Ballan wrasse* and 23 from the *Atlantic mackerel*. The average number of cells by image is 35.8, varying from 22.9 in the *Ballan wrasse* up to 44.8 in the *Four-spot megrim*. Table 4.1 shows the results achieved for every species including sensitivity (Se), specificity (Sp), average precision (AP) and the ratio of modified cells (MC), in %.

The MC varies among species, ranging from 21.4% for *Roughhead Grenadier* to 45.3% for the *European pilchard*. As expected, Se, Sp and AP increase significantly when the modified cells are considered hits, normally more than 10 points. In this case, the average sensitivity, specificity and average precision among species were $55.6 \pm 8.2\%$, $64.8 \pm 16.2\%$ and $43.1 \pm 11.3\%$, respectively. These somewhat high standard deviations mean that there is a large difference among species. For example, the average precision ranges from 27.5% for the *Ballan wrasse* up to 60.3% for the *Four-spot megrim*. This variability is also observed among the images of a given species, as we reported in the previous software Govocitos with the species *European hake* and *Trisopterus lucus* [68]. This was one of the facts that led us to propose a semi-automatic software that allows the expert to supervise the cell recognition before starting the quantification and eventually estimate the fish fecundity.

Dataset	Spices	N	Nc	MC	Se	Sp	AP
pilchard	<i>European pilchard</i>	25	36.5	45.3	39.8 57.8	70.5 80.3	34.8 50.5
megrim	<i>Four-spot megrim</i>	20	44.8	28.4	58.5 87.5	65.0 53.6	83.4 60.3
grenadier	<i>Roughhead grenadier</i>	24	33.0	21.2	47.3 52.6	58.6 28.0	42.1 37.6
ballan	<i>Ballan wrasse</i>	38	22.9	23.1	26.3 45.6	40.3 18.8	31.5 27.4
mackerel	<i>Atlantic mackerel</i>	23	42.0	35.0	46.1 56.1	43.8 58.2	33.4 39.5
Average		26.4	35.8	30.6	43.6 55.6	54.3 64.8	33.7 43.1
SD		6.2	7.7	8.7	10.5 8.2	19.4 16.2	11.4 11.3

Table 4.1: Performance of the automatic algorithm recognizing the cells in the image for all fish species studied. N and Nc are the number of images and the average number of cells per image, respectively. Se, Sp and AP are the sensitivity, specificity and average precision, respectively. MC is the ratio (in %) of modified cells by using the supervision tools in the lateral panel. Se, Sp and AP are calculated considering only the cells perfectly recognized and not modified by the expert (first row), and considering that the cells modified by the tools for supervision are correctly detected but their outlines were modified by the expert (second row). Average values in the last two rows are presented with the standard deviation (SD) to provide an insight into the variation among species.

Dataset	Spices	N	Stage	With or without nucleus
pilchard	<i>European pilchard</i>	25	93.0	83.6
megrim	<i>Four-spot megrim</i>	20	98.7	86.5
grenadier	<i>Roughhead grenadier</i>	24	95.1	89.3
ballan	<i>Ballan wrasse</i>	38	58.5	51.6
mackerel	<i>Atlantic mackerel</i>	23	78.5	81.6
Average		26.4	84.8	78.5
SD		6.2	14.8	13.7

Table 4.2: Average accuracy of the automatic classification for all fish species studied. The column “Stage” shows the accuracy for classifying the development stages: *cortical alveoli*, *vitellogenic* and *hydrated*. The column “With or without nucleus” shows the accuracy of classifying whether the cell has a visible nucleus. Average values and standard deviations (SD) are presented to provide insight into the variation among species.

Table 4.2 shows the average accuracy to classify the cells into the three development stages, *cortical alveoli*, *vitellogenic* and *hydrated*, and into the presence/absence of visible nucleus for each species. The average accuracy among species for the development stage

is 84.8%, ranging from 58.5% for *Ballan wrasse* up to 98.7% for the *Four-spot megrim*. The average accuracy for the presence/absence of visible nucleus ranges from 51.6% for the *Ballan wrasse*, which comes close to random classification, up to 89.3% for the *Roughhead grenadier*, which is a very good score. These results agree with the study in [10], which used *European hake* and *Trisopterus luscus* and where the development stage classification achieved better results than the nucleus classification. *Ballan wrasse* eggs have a characteristic thick chorion, a cellular coat surrounding the eggs that protects them physically from the environment. This coat contains the automatic recognition algorithm, which tends to recognize the outline of the cell as the interior margin of the chorion instead of the exterior. In addition, the extremely high affinity of the vitelline granules for eosin blur the lines between the nuclei and the vitelline granules. These two characteristics could explain the rather bad results obtained for this species.

4.3.3 Analysis performance

In this subsection we compare the stereological analysis of an image using STERapp and the traditional method, based on the Weibel grid system [96]. This method uses a grid of hexagonal cells of which only the vertices are displayed, and further involves: 1) superimposing on the image the previous grid of dots separated by a distance equivalent to the size of the smallest particle to be measured, approximately 100 microns. for fish fecundity; 2) marking each of the dots on each object of interest (oocytes) to estimate the associated area; and 3) counting each object within the grid [24]. This work, in addition to being extremely tedious and visually uncomfortable, decreases the accuracy of the analysis since the area occupied by each object is estimated based on the theoretical area of the hexagonal cell associated with the points over the object, i.e., it is not based on the real area of the object in the image, an advantage the present software exhibits. Moreover, STERapp uses the entire image (all pixels) for its calculations, whereas the Weibel grid, as is commonly employed, usually does not occupy the entire image. The traditional Weibel grid system does not automatically classify the objects (oocyte maturity stages and with or without nucleus), so it is the user who must associate the marked points and objects to each class manually, increasing the probability of making mistakes. In addition, the image correction tools provided by STERapp (merge, split, select, etc.) make the histological image analysis and detection of errors much easier, saving the contours and class labels of oocytes in XML files that allow a review or interruption of the work at any time or replicate, compare, or share the results of processed images. The

stereological analysis of the images can also be exported to CSV files, a file format easily suited to be managed in other tools. A further added value of STERapp is the opportunity to customise preferences and save them in xml file for different users; in this way, users from different locations can easily share preferences or work on the same computer in different work sessions. The simple installation and use of STERapp (see Section B.1 and B.2 in appendix B), its versatility for the creation of new object classes, as well as the ability to train its classifiers (in our case, for different species), also make it an application with great potential in other disciplines such as Medicine or Geology. The time needed for processing histological images with STERapp depends on the size of the image and the computing power, as well as on the characteristics of the ovarian structures of each species. We did not develop a formal statistical comparison of the time required by STERapp and the traditional Weibel grid system. However, the time required by experts is much shorter using STERapp, because its intuitive and user-friendly GUI greatly facilitates the technical work. Specifically, the time needed by the automatic processing of one image is approximately one minute on a general purpose computer, plus the time needed by the expert's supervision, which may range from 2 to 5 minutes, depending on the review needs. Comparatively, the traditional methods require approximately 10 minutes per image. Therefore, STERapp saves between $100(10-6)/10=40\%$ and $100(10-3)/10=70\%$ of the time compared to the traditional method.

4.3.4 Expert perception

We evaluated the expert's perception about STERapp using the system usability scale (SUS), a free questionnaire to measure the learning ability and subjectively perceived usability of computer systems [4, 6]. This questionnaire has the following 10 items:

1. I think that I would like to use this product frequently.
2. I found the product unnecessarily complex.
3. I thought the product was easy to use.
4. I think that I would need the support of a technical person to be able to use this product.
5. I found that the various functions in the product were well integrated.
6. I thought there was too much inconsistency in the product.
7. I imagine that most people would learn to use this product very quickly.

8. I found this product very awkward to use.
9. I felt very confident using the product.
10. I needed to learn a lot of things before I could get going with this product.

The items are scored on a 5-point scale ranging from 1 (strongly disagree) to 5 (strongly agree), providing a final system score between 0 and 100. The score is calculated by adding up the positively worded items (1, 3, 5, 7 and 9), subtracting one from the user responses, and the negatively worded items (2, 4, 6, 8 and 10), subtracting the user responses from five. The SUS score for each user ranges from 0 to 40, which is multiplied by 2.5 to rescale the value to the range from 0 to 100. This score is translated to people's rating of systems and products in terms of adjectives to provide a meaning to them [77] as follows: SUS < 25 is the worst imaginable system; from 25 to 39 is from the worst imaginable to poor; from 40 to 52 is from poor to OK; from 53 to 73 is OK to good; from 74 to 85 is good to excellent; and above 85 is excellent to the best imaginable system. Usually, a small sample (8–12 users) is enough to give a good assessment of how people see the software. STERapp was evaluated using the SUS questionnaire by 8 expert technicians from the Instituto de Investigaciones Mariñas (IIM) and the Instituto Español de Oceanografía (IEO). The mean score achieved was 81.9 (ranging from 67.5 to 95), which means “from good to excellent”. These positive ratings are mainly associated with the ease of use, performance and versatility of the software as highlighted in Section 4.3.3.

4.3.5 Comparison with ImageJ

As mentioned, ImageJ is the most widely used software for image analysis in the field of biology and biomedicine [78, 74]. ImageJ allows the development and installation of plugins to analyse a specific type of image. Nevertheless, the main drawback of ImageJ is that its GUI does not provide an easy way to correct or modify the automatic recognition of objects in the images before their quantification. Considering the complexity and variability of the histological images of fish gonads, we do not know any approach that automatically recognizes oocytes in a completely satisfactory way for the majority of images, so the expert's revision is an essential part of any software. STERapp overcomes many limitations of ImageJ such as: 1) its friendly GUI provides an easy way to expert's supervision of the fecundity calculation process and 2) it allows the monitoring of each result through the information registered in

the xml files and share analysis criteria among experts. In fact, ImageJ was never used for fish fecundity estimation purposes due to its low usability, while STERapp was qualified by the users as “from good to excellent” with a mean score of 81.9 on the system usability scale.

4.4 Conclusions

This chapter proposes the software tool STERapp perform do stereological analysis on images. It was evaluated for the analysis of histological images of fish gonads to estimate fecundity, a parameter commonly used in stock assessment for the sustainable management of fisheries. STERapp combines the automatic recognition and classification of cells in histological images with a friendly graphical user interface, that allows the expert to modify the automatic cell recognition and/or classification before starting the quantification or stereological analysis of the image. STERapp calculates the cell area and diameter and the number of cells in each development stage, which are the key parameters to estimate fish fecundity. Additionally, it checks if the cells are on the image border to determine if they are counted or measured in the stereological analysis. STERapp has been tested by two Spanish fisheries research groups since 2020, analyzing images of five fish species coming from different image acquisition systems. The automatic processing of the images is quite robust providing the following average performance over the five species studied: 1) cell recognition with a sensitivity of 55.6%, specificity of 64.8% and average precision of 43.1%; and 2) cell classification of 84.8% to discriminate three development stages and 78.5% to discriminate whether the cell has visible nucleus or not. These performance values lead us to conclude that although STERapp cannot operate fully automatically, it can considerably reduce the analysis time, facilitate image processing and correction and improve the precision of stereological estimations by using real areas of the objects among other characteristics that make this software a major advance over what is currently available. In fact, the user perception of the system was evaluated using the SUS questionnaire, achieving a mean score of 81.9, which means that the system is from good to excellent from the experts’ point of view. Therefore, we can conclude that STERapp is a useful software tool that can simplify and improve stereological analysis in fisheries research, specifically in those groups or institutions that use fecundity data for biological studies or fisheries assessment.

CHAPTER 5

CONCLUSION

The current thesis includes the proposal of an algorithm to segment cells in microscopic images and a software tool to quantify images using stereological analysis. The segmentation algorithm proposed to recognize cells in microscopic images, called MSCF (Multi-scale Canny filter), is based on the Canny edge-detector. The classical Canny edge detector combines the following steps: 1) smooth the gray-level image using a σ -width Gaussian kernel; 2) differentiating using a first derivative operator; 3) performing the non-maxima suppression process, which finds local maxima in the direction perpendicular to the edge; and 4) performing thresholding with hysteresis. MSCF considers different scales using a set of σ -values, and the thresholds to the hysteresis process are selected automatically from the image characteristics. Finally, there is a fusion process of the information coming from different scales.

Our approach has been statistically evaluated over five datasets of histological images of fish ovaries providing F_1 values from 70% to 80% for the five fish species used. Compared with other state-of-art segmentation techniques, namely, k-means clustering, Mean Shift, Chan Vase, Felzenszwalb and Govocitos, MSCF globally achieved the highest performance both in pixel classification (the highest F_1 values) and cell recognition, measured as the percentage of cells correctly recognized for a level of overlapping tolerance. This percentage for the different fish species ranges from 51.83% for *European hake* to 64.41% for the *Roughhead grenadier* fish species. Globally, the MSCF algorithm achieved the first position using the Friedman rank, followed by Govocitos, Clustering, Meanshift, Felzenszwalb and the Chan-Vase method. The time needed by the first four algorithms to process each image is approximately a very few seconds, which makes them suitable for image processing in inter-

active applications, while the last two applications spend more than there minute and they are too slow for real time applications.

In conclusion, our MSCF algorithm is very competitive, both in computational time and performance, with other state-of-art segmentation algorithms. We verified that the problem of recognizing cells in histological images of fish gonads is very challenging, and that a completely automatic approach is not still available.

The software tool proposed to perform stereological analysis on images is called STERapp. It was evaluated in a real environment for the analysis of histological images of fish gonads to estimate fecundity, a parameter commonly used in stock assessment for the sustainable management of fisheries. STERapp combines the automatic recognition and classification of cells in histological images with a friendly graphical user interface that allows the expert to modify the automatic cell recognition and/or classification before starting the quantification or the stereological analysis of the image. STERapp calculates the cell area and diameter and the number of cells in each development stage, which are the key parameters to estimate fish fecundity. Furthermore, it checks if the cells are located on the image border to determine if they have been counted or measured in the stereological analysis. Since 2020, STERapp has been tested by two Spanish fisheries research groups that analyzed images of five fish species coming from different image acquisition systems. The automatic processing of the images is quite robust providing the following average performance over the five species studied:

- Cell recognition was performed with a sensitivity of 55.6%, specificity of 64.8%, and average precision of 43.1%
- A cell classification of 84.8% to discriminate three development stages, and 78.5% was used to discriminate whether the cell had a visible nucleus.

These performance values led us to conclude that although STERapp cannot operate fully automatically, it can considerably reduce the analysis time. Thus, STERapp facilitates image processing and correction and improves the precision of stereological estimations by using real areas of the objects, among other characteristics that make this software a major advance over what is currently available. Hence, experts estimate that the sterapp software saves between 40 and 70% of the time needed to analyse an image compared to the traditional method of estimating fish fecundity. In fact, the user perception of the system was evaluated using the SUS (system usability scale) questionnaire, achieving a mean score of 81.9, which means that the system is rated from good to excellent by experts.

Therefore, we can conclude that STERapp is a useful software tool which that can simplify and improve stereological analysis in fisheries research, specifically in those groups or institutions that use fecundity data for biological studies or fisheries assessment.

Future work will focus on the development of new algorithms to recognize the cells in the image, which improves the performance of the automatic analysis for some fish species to reduce the time and effort of expert supervision, perhaps taking into account these annotations in the segmentation and classification process. We would like to extend the software to the automatic recognition of objects in other stereological analysis problems coming from mineralogy or medicine. We also try to include in STEPapp some algorithms of semi-supervised or interactive segmentation in which the expert provides a cue of the matured cells to recognize its contour.

APPENDIX A

LISTADO DE PUBLICACIONES

As publicacións asociadas á tese son as seguintes:

1. A. Mbaidin, S. Rábade-Uberos, R. Dominguez-Petit, A. Villaverde, M.E. González-Rufino, A. Formella, M. Fernández-Delgado, E. Cernadas. **STERapp: semiautomatic software for stereological analysis. application in the estimation of fish fecundity.** *Electronics*, Special issue of *Applications of Computer Vision*, Vol. 10, No. 1432, pp. 1-15. Índice de impacto 2.9. Posición 131 de 275 en Engineering, Electrical and Electronic (Q2). Posición 72 de 160 en applied physics (Q2). Posición 99 de 158 en Computer Science, Information Systems (Q3).
2. A. Mbaidin, E. Cernadas, Z. Al-Tarawneh and M. Fernández-Delgado, **Recognizing cells in histological images of fish gonads**, 2022 *International Conference on Emerging Trends in Computing and Engineering Applications (ETCEA)*, Karak, Jordan, 2022, pp. 1-6, doi: 10.1109/ETCEA57049.2022.10009859.
3. A. Mbaidin, E. Cernadas, Z.A. Al-Tarawneh, M. Fernández-Delgado, R. Domínguez-Petit, S. Rábade-Uberos, A. Hassanat. **MSCF: Multi-Scale Canny Filter to Recognize Cells in Microscopic Images.** *Sustainability*. Special issue *Latest Applications of Computer Vision and Machine Learning Techniques for Smart Sustainability*. 2023; 15(18):13693. <https://doi.org/10.3390/su151813693>. Índice de impacto 3.9. Posición 114 de 275 en Environmental Sciences (Q2). Posición 48 de 128 en Environmental Studies (Q2).

As tres publicacións anteriores foron realizadas exclusivamente polo doutorando, polo tanto a súa contribución é do 100% das mesmas, é orixinal e propia do doutorando.

APPENDIX B

STERAPP USER GUIDE

STERapp is a free software tool to do stereological analysis. The present distribution includes a plugin to estimate fish reproductive parameters like fecundity, based on digital analysis of histological samples of fish ovaries. STERapp was developed by the following four Spanish research groups: Instituto de Investigaciones Mariñas (IIM)¹ part of the CSIC (Centro Superior de Investigaciones Científicas), Oceanographic Center of Vigo from the Spanish Institute of Oceanography (IEO)², Laboratorio de Informática Aplicada³ of the University of Vigo and Centro de Investigación en Tecnoloxías Intelixentes (CiTIUS)⁴ of University of Santiago de Compostela. STERapp is a multi-platform software written in the programming language C/C++. The current implemented plugin analyses automatically ovary histological images, using advanced computer vision and machine learning techniques. Due to the different histological processes and fish species analyzed in each laboratory, precision of the results obtained during the automatic processing may differ, because of this, STERapp includes a friendly GUI (Graphical User Interface) to visualise, review and interact with the classification of the fish gonad histological manually.

STERapp works with three types of files: image files, XML files (eXtensible Markup Language) and CSV files (Comma-Separated Values). The image formats supported are the most frequently used like GIF, TIF, PNG, BMP, PPM, JPG, etc. Every processed image generates a XML file which contains the contour and classification of each oocyte detected.

¹<http://www.iim.csic.es/>

²<http://www.ieo.es/es/web/vigo/>

³<http://lia.esei.uvigo.es/projects/gonadas/index.php>

⁴<http://citi.usc.es/>

This XML file is stored and its graphical content can be superimposed to the image in any moment. The statistical information of the quantitative analysis of every image is exported in a CSV file, such as number of objects, cell area and diameter, type of oocyte, etc. Calculated data can also be saved by STERapp in order to be reviewed by experts any time later.

The present user guide is organized as follows: Sections B.1 and B.2 describe the installation steps for the operating systems Windows and Linux; Section B.3 describes how to run the Graphical User Interface (GUI) of STERapp. Section B.4 describes the configuration of the preferences. Sections B.5, B.6 and B.7 describe the File, Edit and View menus, respectively. In the Sections B.8 and B.9 users can find information about the analysis of the results with STERapp and the classification process. Finally, Section B.10 refers to Help menu.

B.1 Windows installation

Once the user has downloaded the installation file `setupSTERapp.exe`, a double click on the file will open a dialog window which asks for permission to install a foreign program on your computer, click on “Yes” and a window will show you the Setup Wizard. Clicking on “Next” will start the installation.

The next window allows the user to choose the installation folder, click on “Next” and continue with the installation. A dialog will ask if you want to create a desktop icon. Click on “Next” to continue and confirm if you want to install STERapp, clicking on “Install”, or cancelling the installation.

Once the installation process has started, a pop up window will show the progress; this overall process may take a few seconds. If you click on “Cancel”, the installation process will be aborted. When the installation has finished the system will inform you and ask for launching the program. Click on “Finish” and check the box *Launch STERapp* if you want to run STERapp immediately.

B.2 Linux installation

In Linux, there are several types of packages, and every distribution has its own preferred package format. Ubuntu distributions used the Debian packages (format DEB). The .deb/Debian files containing STERapp is the file `sterapp_1.0_all.deb`. It is provided the linux package for Ubuntu 20.04 version.

To install STERapp, go to terminal, change to the folder where the file `sterapp_1.0_all.deb` is and type the following command: `sudo dpkg -i sterapp_1.0_all.deb` and the system asks you by the administrator password.

If the installation gives errors due to the lack of some packages required by STERapp, the user must uninstall STERapp with the command `sudo apt --fix-broken install`, install the required packages with the command `sudo apt install package-name`, and install STERapp again with the command `sudo dpkg -i sterapp_1.0_all.deb`. Then, you can run STERapp using the following command: `sterapp`.

STERapp can be removed from the computer using the command: `sudo apt remove sterapp`.

B.3 Run STERapp

After installing, click on the desktop icon in Windows. If it is the first time you run STERapp, a pop-up window (Figure B.1) will inform you that there are no species defined in the program, click **ok** and the main program window (Figure B.2) will be opened. This window encloses a menu bar with several sections, a toolbar containing the main functionality commands of STERapp, and a window panel where the histological images are visualized.

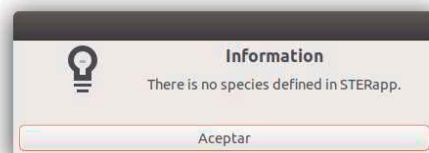


Figure B.1: Pop-up message to inform that there are no species defined in STERapp.

The menu bar lists all **STERapp** commands and it is organized in six menus:

1. **File:** basic file operations, opening images and XML files, saving the overlays (as xml) and the statistical results (as csv); set and load the working preferences; and exit STERapp.
2. **Edit:** fitting image size and set the image to its original size.
3. **View:** show or hide the processing (lateral) panel.



Figure B.2: Main window of STERapp.

4. **Analysis:** provides functionalities to calculate the results from several images of the same sample.
5. **Classification:** provides functionalities to train the classifier (machine learning model) with different species, and to classify the detected cells by developing stage and presence or absence of visible nucleous in the oocyte.
6. **Help:** provides information about the development team.

The toolbar provides a faster access to the main functionalities of STERapp, and contains the following icons (a pop up label shows a short description when the mouse is placed over the icon):

1. **Reset:** clears all the objects drawn on the image.
2. **Open:** opens a dialog to select the histological image (see Section B.5).
3. **Zoom Fit:** fits the zoom of the image window.
4. **Original Zoom:** resets to the original image size.

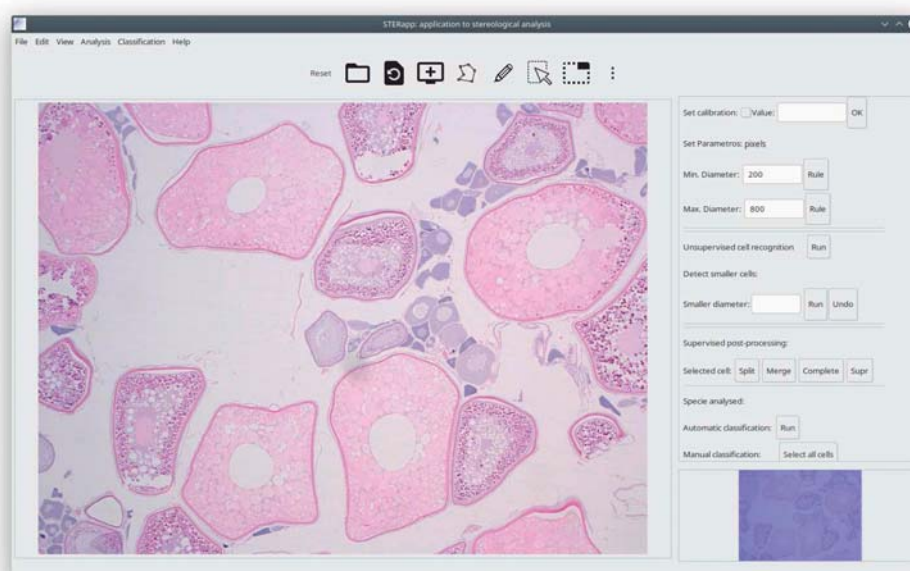


Figure B.3: Main window of STERapp with the lateral panel being open, which shows the plugin GONAD to process histological images of fish gonads.

5. **Draw by points:** marks the oocyte contour by points; use the left button of the mouse to draw them. The contour will be closed by clicking the middle button of the mouse.
6. **Draw freehand:** marks the oocyte contour freehand. To draw a contour press the left button of mouse and keep it pressed while drawing the region. The contour will be closed when the button is released.
7. **Select:** selects a drawn contour on the image by clicking into the object with the left button of the mouse. To select more than one object, keep the key **Ctrl** or key “Control” pressed while selecting objects. You can also select various objects with the following option **Select with rectangle** or select all the objects with the button *Select all cells* in the lateral panel.

8. **Select with rectangle:** to select several drawn objects on the image, click the left button of the mouse and keep it pressed, drag the mouse to draw a rectangle and release the left button. The objects under the rectangle area will be selected.
9. **Lateral panel:** open the lateral panel that shows the commands to process and analyse the image (see the section B.7 and Figure B.3).

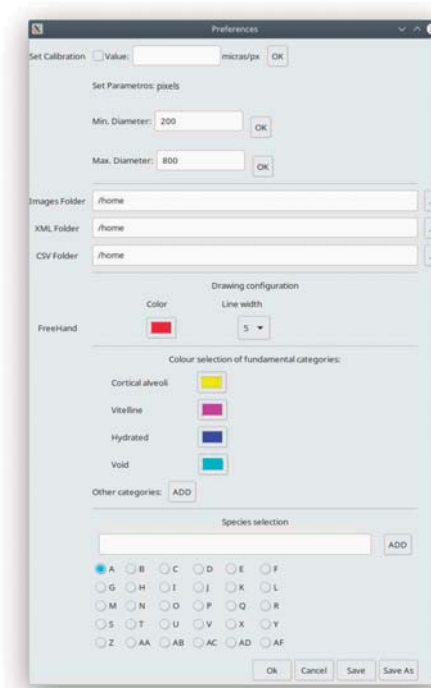


Figure B.4: Preferences panel: default configuration.

In the following sections the functionalities of the Graphical User Interface (GUI) of STERapp will be described. In the File menu there are several commands to work with the required files:

1. **Open image:** opens a dialog window to load a new image.

2. **Open image and XML:** opens a dialog window to select the image to load the XML file with the contours of the cells superimposed, when the XML file for that image already exists in the XML directory and the option is set in the preferences.
3. **Open XML:** opens a dialog window to select the XML file and loads it on a preloaded image. The user should check that the XML file matches with the image.
4. **Save XML:** opens a dialog window to enter the name of the XML file and chooses the path to save it. By default, STERapp uses the name of the image with extension xml, and saves the contours in the XML directory as set in the preferences.
5. **Export CSV:** opens a dialog window to enter the name of a CSV file in which you want to save the statistical analysis of the image. By default, STERapp uses the name of the image with extension CSV, and saves the statistical data in the CSV directory as set in the preferences.
6. **Preferences:** opens a panel to set the working preferences (see Figure B.4), such as working directories, colour and width of lines, calibration of the images, fish species, maximum and minimum diameters of the oocytes for the selected species, and developing stage of the oocytes. The working preferences can be saved to be available in future sessions, or exported to a XML file.
7. **Reset preferences:** resets the preferences as previously set in the program.
8. **Load preferences:** allows to load the working preferences from a XML file given by the user.
9. **Exit:** exits the program.

In the on-going, Section B.4 describes the configuration of the working preferences in STERapp.

B.4 Set preferences

Selecting “Preferences” in the **File menu**, the window shown in Figure B.4 (left panel) will be opened. There, you can configure the following items:

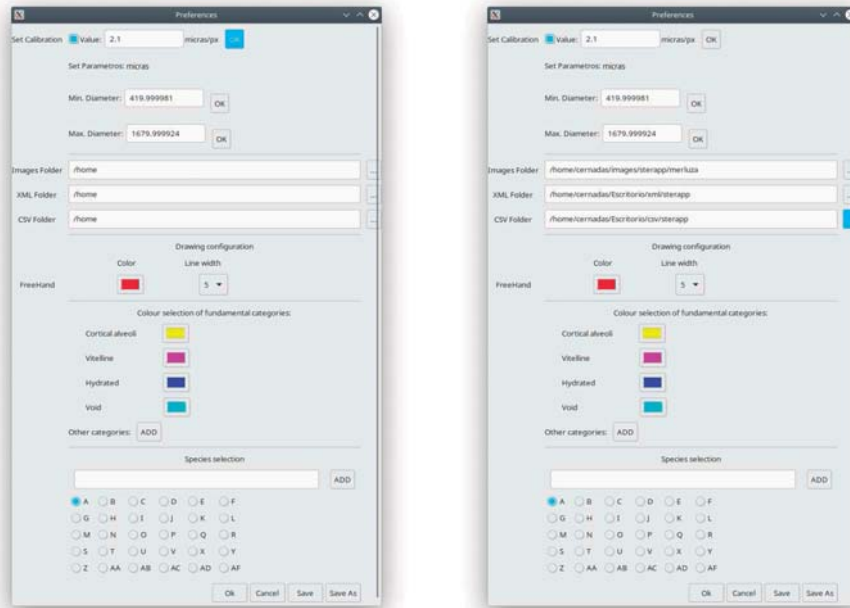


Figure B.5: Setting preferences: (left panel) after setting the calibration, i.e., the relation between micrometers and pixels; (right panel) after setting the working directories.

1. **Calibration:** It is the size (in microns) of a pixel in the image. The user must set the value of every image to get the measuring of the objects in their real units. Otherwise, values will be given in pixels.
2. **Diameters:** sets the minimum and maximum size of the oocytes in the images. These parameters can be set in pixels or microns depending of the Calibration setting.
3. **Working directories:** sets the default directories for images, xml and CSV files.
4. **Drawing configuration:** sets the colour and width of the line to draw the contours over the oocytes.
5. **Colour selection of fundamental categories:** As default, STERapp only can automatically classify the following three oocytes development stages: cortical alveoli, vitelline and hydrated. The user can select a colour to represent every category and also the cat-

egory “void” that corresponds to these parts of the image that corresponds to space out of the ovary tissue, i.e. it will be excluded from the tissue area estimates. New categories can be added manually using the GUI of STERapp.

6. **Species Selection:** defines new fish species.
7. **Setting or changing configuration:** use buttons **Ok**, **Cancel**, **Save** and **Save As** at the bottom (Figure B.4) to set the preferences to this working session, cancel, save the preferences in the system or save the preferences in a xml user file, respectively.

B.4.1 Calibration

The calibration of an image is fixed in the digitalization process, and depends on the magnification used in the microscope and the spatial resolution of the digital camera connected to it. If the user knows the actual calibration of the image, or set of images, he/she should introduce the *Value* in microns per pixel. Therefore press “Enter” or press the button **OK** and check the box after the label *Set Calibration* to confirm that the real calibration value has been added. If it is non-checked, the calibration will be considered in pixels. When this value is fixed, the minimum and maximum diameters will be converted from pixels to microns according to the calibration, as you can see in the left panel of Figure B.5.

B.4.2 Diameters

The range of variation of oocyte diameters can be a species specific parameter. Keeping this in mind, STERapp provides more versatility allowing the user to choose the minimum and maximum diameters of the oocytes according to the species. These parameters can be set in the boxes *Min. Diameter* and *Max. Diameter*, respectively, by inserting the values in microns or in pixels depending if the calibration is active or not. After setting the values, press the “Intro” or “Enter” key or the button **OK** after the label box to update. This parameter can also be set on the lateral panel.

B.4.3 Working directories

In this panel the user can introduce the path to set the default directories to store the working files, *Images Folder*, *XML Folder* and *CSV Folder*, as in the right panel of Figure B.5. Clicking

the button next to the box, users can choose the working directory in an open dialog window (see Figure B.6).

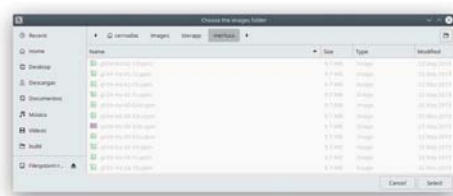


Figure B.6: Open dialog box to choose the working directory.

B.4.4 Drawing configuration

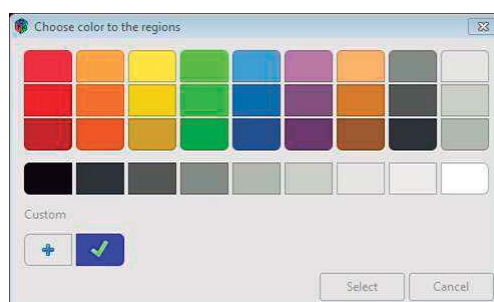


Figure B.7: Chart colour for contour lines.

The colour and width of the line to draw the oocytes over the image can be changed in this panel. Clicking the coloured button after the label *FreeHand*, opens the colour chart dialog shown in Figure B.7, choose the colour and press “Select”. Clicking on the button “Line width” (Figure B.5) the width of the line can be chosen in a drop-down menu. The colour of every oocyte category can be changed in their own coloured button, below the section *Colour selection of fundamental categories*, with the same process as the *FreeHand* colour.

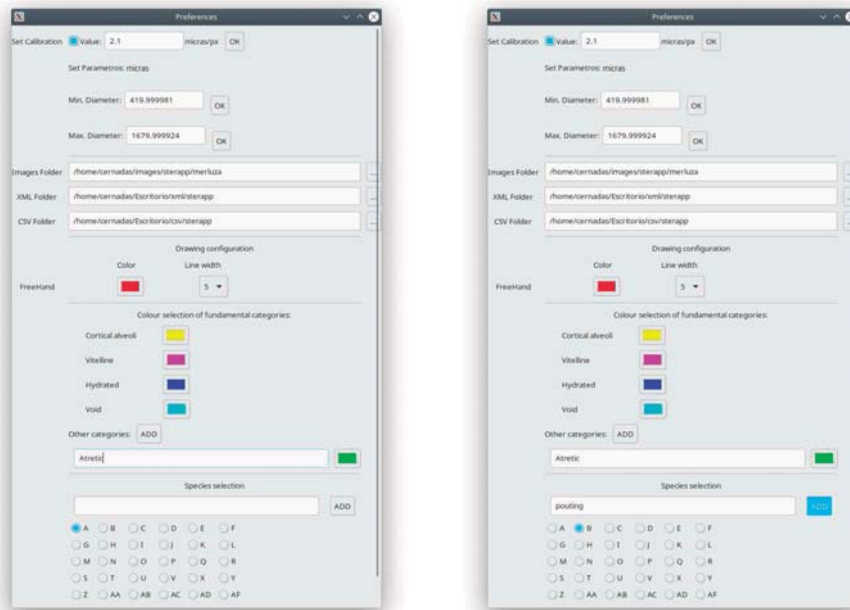


Figure B.8: Setting preferences: (left panel) after adding new cell categories; (right panel) after setting some fish species.

B.4.5 Adding categories

With the button **ADD** after the label “Other categories”, in section *Colour selection of fundamental categories* of Figure B.8, you can add new categories of oocytes to STERapp. As mentioned before, these new categories will not be considered in the automatic classification, but they can be manually managed in the program. You can manually assign these categories to oocytes contours on the images and STERapp will use them for the statistics. The left panel of Figure B.8 shows, as an example, where a new category “Atritic” is added with an associated green colour.

B.4.6 Species selection

STERapp manages a different classifier (or machine learning model) to each fish specie, so is necessary to train the classifier with each new species. In the current version, STERapp

can manage 30 different fish species. By default there is no species defined in the system. To add a new fish species, select a radio button with one of the characters (A, B, C, D, E, ...), write the name of the species in the entry widget below the label “Species selection” and press “Intro” or “Enter” to update the value or click the button **ADD** after the text box. This process creates an internal link between the label used for the software (A, B, C, ...) and the name of the species, which allows to preserve data consistency. We defined in the radio button A the specie European hake (*Merluccius merluccius*) and in the radio button B the specie Pouting (*Trispterus luscus*) (see the right panel of Figure B.8). After that, the radio button A was selected and the lateral panel only shows the fish selected species, but not the character associated (see Figure B.11 after the label “Species analysed”).

B.4.7 Setting the configuration

1. Button **Ok**, sets these preferences to the present working session.
2. Button **Cancel**, cancels the operation of setting the preferences. Changes will not be saved.
3. Button **Save**, saves the preferences for present and future working sessions. The first time the user runs STERapp, the pop-up message of Figure B.9 informs you that “There is no trained classifiers”, the user must complete this process in order to use the automatic classification (see Section B.9). Once the classifier is trained, the program will classify the oocytes in an image and will show the development stage of each oocyte using different colours (set in the preferences). The presence or absence of visible nucleus in each oocyte will be indicated using a continuous line to show oocytes with visible nucleus and dashed line to oocytes with non-visible nucleus. This information is relevant as STERapp only uses those cells with visible nucleus to perform the diameter calculations, since they are the only one from which the measurement of the real diameter can be assured. It is important to emphasize that the user must check that the loaded image belongs to the working specie, shown in the lateral panel after the label “Species analysed” (otherwise, the user must go to the preferences and change the species in the radio buttons at the bottom of the preferences window in Figure B.8).
4. Button **Save As**, saves the preferences of the present working session and saves them in an XML file selected by the user, which can be loaded using the submenu “Load

preferences” of menu “File”. This option allows to share the preferences files between users.

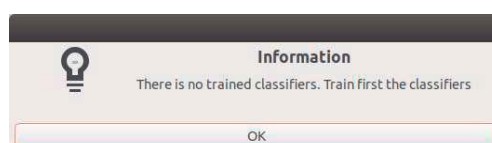


Figure B.9: Pop-up message to inform that the classifier must be trained in the future.

B.5 File menu

The commands available in the **File** menu are: **Open Image**, **Open Image and XML**, **Open XML**, **Save XML**, **Export CSV**, **Preferences**, **Reset preferences**, **Load Preferences** and **Exit**. STERapp works with three types of files: images, xml (eXtensible Markup Language) and csv (Comma-Separated Values) files. The supported image formats are the most frequently used ones, such as .gif, .tif, .png, .bmp, .ppm, .jpg, etc. Every image file will have its associated XML file, which will include the contours and categories of the oocytes, and can be loaded into the software at any time. The CSV file will include the statistical information of the quantitative analysis of every image.

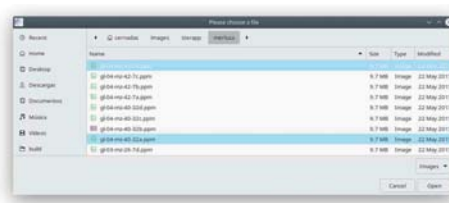


Figure B.10: File chooser dialog to choose the image to be loaded in STERapp.

Open Image opens a dialog (Figure B.10) where you can choose an image file from the directory set in Preferences (see Section B.4) or where you can select another path. Choose the file and click **Open** to load the image in STERapp. Be aware that the path must not be larger than 256 characters and must not contain rare symbols (“/”, “%”, “ ”, “*”, etc).



Figure B.11: The image selected in Figure B.10 is loaded in STERapp.

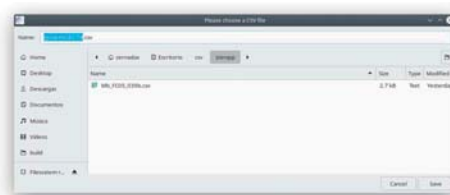


Figure B.12: Dialog to choose the CSV file to store the statistical analysis of the image.

Figure B.11 shows the main window of the program with an image loaded and with the oocytes contours overlapped. When an image is loaded, the lateral panel opens and the state bar at the left-bottom corner of the main window shows the name of image file and its resolution. If the image was already analysed and the XML file was stored, you can load it selecting the submenu **Open XML** and the contours will be shown overlapped to the image, as it can be seen in Figure B.11. The user must reassure that the selected XML file is the corresponding one for each image, STERapp does not check this automatically. Using the submenu **Open**

	Image area	Calibration	Species	O	O diameter	Area	Stage	Visible nucleus	Count	Measure
1	14272522_704	2,100	European hake	1	272,514	58326,655	Vitellogene	No	No	No
2	14272522_704	2,100	European hake	2	87,899	6068,159	Vitellogene	No	No	No
3	14272522_704	2,100	European hake	3	148,814	17593,038	Vitellogene	No	No	No
4	14272522_704	2,100	European hake	4	370,840	198009,710	Vitellogene	No	No	No
5	14272522_704	2,100	European hake	5	561,664	247767,007	Vitellogene	No	No	No
6	14272522_704	2,100	European hake	6	359,613	101568,800	Vitellogene	Yes	Yes	Yes
7	14272522_704	2,100	European hake	7	118,906	11104,379	Vitellogene	No	No	No
8	14272522_704	2,100	European hake	8	769,907	465594,428	Vitellogene	No	No	No
9	14272522_704	2,100	European hake	9	275,047	58415,925	Vitellogene	No	No	No
10	14272522_704	2,100	European hake	10	63,627	3179,610	Vitellogene	No	Yes	No
11	14272522_704	2,100	European hake	11	437,262	150235,458	Vitellogene	No	Yes	No
12	14272522_704	2,100	European hake	12	77,221	4683,420	Vitellogene	No	Yes	No
13	14272522_704	2,100	European hake	13	1034,844	841093,944	Vitellogene	Yes	Yes	Yes
14	14272522_704	2,100	European hake	14	725,538	413426,427	Vitellogene	No	Yes	No
15	14272522_704	2,100	European hake	15	250,920	49449,326	Vitellogene	No	Yes	No
16	14272522_704	2,100	European hake	16	428,598	144266,522	Vitellogene	No	Yes	No
17	14272522_704	2,100	European hake	17	286,368	64408,044	Vitellogene	No	No	No
18	14272522_704	2,100	European hake	18	385,768	116880,424	Vitellogene	No	No	No
19	14272522_704	2,100	European hake	19	98,402	7605,044	Vitellogene	No	No	No
20	14272522_704	2,100	European hake	20	248,189	48371,081	Vitellogene	No	Yes	No
21	14272522_704	2,100	European hake	21	509,438	203832,386	Vitellogene	Yes	Yes	Yes
22	14272522_704	2,100	European hake	22	611,027	601897,476	Vitellogene	No	Yes	No
23	14272522_704	2,100	European hake	23	536,532	226089,654	Vitellogene	No	Yes	No
24	14272522_704	2,100	European hake	24	235,557	43979,616	Alveolo	No	Yes	No
25	14272522_704	2,100	European hake	25	393,196	121671,889	Vitellogene	No	Yes	No
26	14272522_704	2,100	European hake	26	1057,818	878511,610	Vitellogene	Yes	Yes	Yes

Figure B.13: An example of CSV file of the image of Figure B.11 imported in LibreOffice Calc.

Image and XML (open the window of Figure B.10 to select the image), STERapp will check automatically in the xml directory set in Preferences, if there is a file with the same name as the image. If this is the case, STERapp will open the image file and overlap its xml in a single process (see Figure B.11).

Once the contours of the oocytes have been drawn on the image, the user should open the submenu **Export CSV** to open the window of Figure B.12 and set the file name (by default the name of the image with extension csv is used). The CSV file will store the statistical results of the image in the directory selected in “Preferences”. This CSV file contains the directory path, the selected area, the calibration, the fish species, the development stage of every oocyte, the presence/absence of a visible nucleus, and if the oocyte will be used in the statistical analysis or not. Following the rules of the stereological method, when you use more than one image of a sample and they are adjacent, the probability of counting the same object twice is high. Thus, in order to minimize this probability, the method suggests to count only the cells in touch with the bottom and right border of the image and discard the oocytes touching upper and left side of the image. The information of the CSV file can be loaded in a spreadsheet such as LibreOffice Calc⁵, as it can be seen in Figure B.13, or in Excel.

The command **Reset preferences** removes all preferences previously set and shows the message of Figure B.14. The program closes if you click on the **OK** button and does nothing

⁵<https://www.libreoffice.org/discover/calc/>

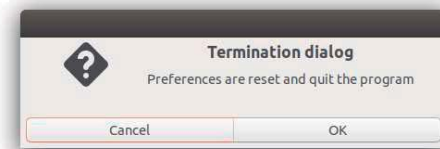


Figure B.14: Pop-up message shown when the user resets the preferences.

if you click on the **Cancel** button. The user can also remove the set preferences manually. The preferences file (a XML file named `preferences.xml`) and the configuration of classifiers to each fish specie are located in the directory `.govocitos` of the home path. In Windows, the home path ususally is the directory `C:\users\myUserName` or `C:\usuarios\myUserName` where `myUserName` is the session's name where the program was installed in the computer. Deleting the directory `C:\users\myUserName\govocitos`, all the preferences are removed. In Linux, the home path is the directory `/home/myUserName`.

The **Load preferences** command opens a file chooser dialog to select the XML file preferences (`preferences.xml` file to be loaded. Preferences do not include the configuration files of the classifier which cannot be loaded. All the preferences (file `preferences.xml`) and the configuration files of the classifiers for each fish specie can be transferred manually to another computer copying the `.govocitos` directory from one computer to another in the same path, e.g., `C:\users\myUserName` or `C:\usuarios\myUserName` for a Windows system.

B.6 Edit menu

All the items available within this menu are also available in the toolbar (see Section B.3). They are: **Fit Image** and **Original Size**.

Fit Image fits the zoom of the image panel and **Original Size** sets the original size of the image. Image zoom can be changed rolling up and down the mouse wheel to increase or decrease the zoom. The portion of the area which is visible in the image window can be seen in the iconized image located in the bottom of the lateral panel (see Figure B.15). Keeping the zoom, you can move to another part of the image by two methods: 1) pressing simultaneously the left and right buttons of the mouse over the image panel and displacing the visible area; or 2) pressing the left button of the mouse on the blue square in the iconized

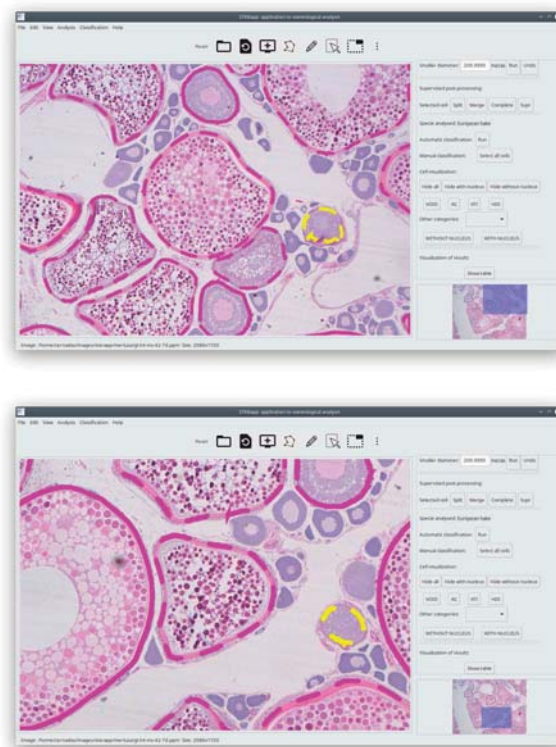


Figure B.15: An example of the zoom functionality in STERapp.

image of the lateral panel and displacing the square. Both movements, the visible area in the window image and the iconized image of the lateral panel are synchronized. The overlays of the image are zoomed with the image as well.

B.7 View menu or lateral panel

The lateral panel is a simple way of visualizing and managing the commands needed in the processing of an image; it contains the same functionalities already seen in other sections as **Calibration** and **Diameters**. **Diameters** in the lateral panel has also a tool to introduce the values both graphically and manually: the button **Rule** allows to measure one object of

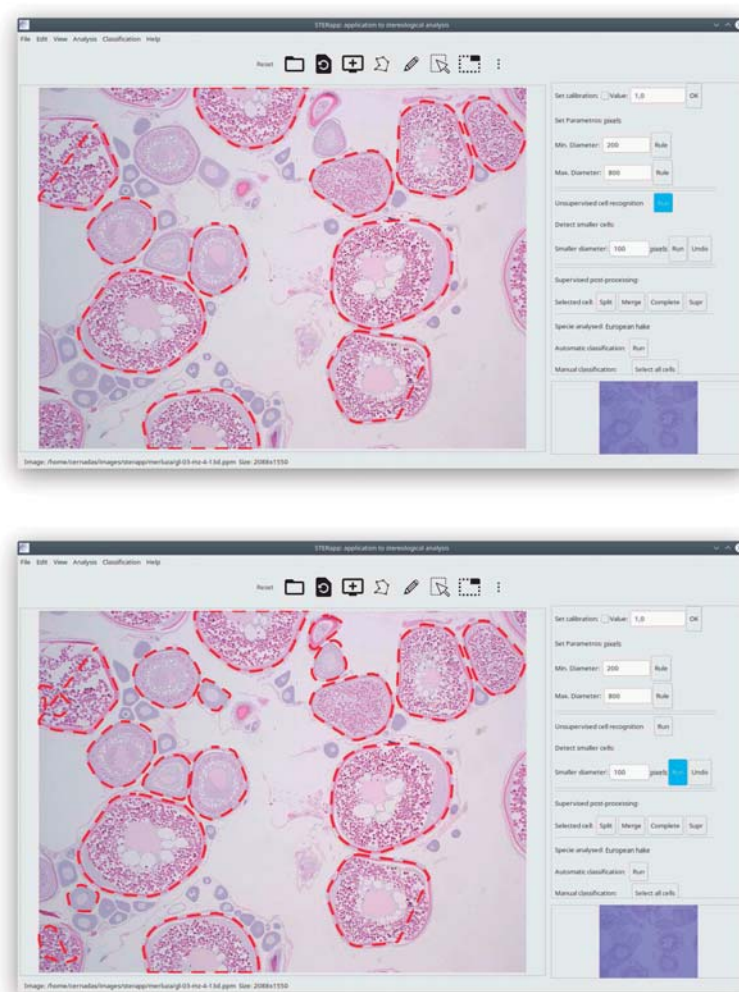


Figure B.16: Output after automatic processing of the histological image (upper panel) and output after automatic processing using the option “detect smaller cells” (bottom panel).

the loaded image and will transfer the result to the diameters box when the left button of the mouse is released. The value will be referred in pixels or microns depending on the current calibration. When the calibration or diameters are modified, all the overlays of the image are

deleted. The value will not be updated if there is an inconsistency between maximum and minimum values. Other utilities of the lateral panel are:

1. **Unsupervised cell Recognition** automatically processes the image and shows, overlapped on the image window, the recognized oocytes. Clicking on the **Run** button, the image panel will show, in a few seconds, the results of the operation (see the upper panel of Figure B.16).
2. **Detect smaller cells** can be used to detect oocytes smaller than the minimum diameter set in the program after the run of the **Unsupervised cell Recognition**. In the entry widget after the label “smaller diameter”, you can set a value lower than the minimum diameter $d_{smaller}$, and press the key “Enter” to update the value. Pressing **Run** after this entry widget, STERapp automatically recognizes the oocytes, whose diameter is between $d_{smaller}$ and the minimum diameter set in the preferences, and ensures that the newly recognized oocytes do not overlap the previously recognized ones. If the result provides many false positive or it is not suitable, you can undo this step clicking in the button **Undo**. As well, this process can provide less false positives, because of its two phases of recognition and revision. The lower panel of Figure B.16 shows an example of oocyte detections added when the automatic algorithm to recognize oocytes is used with $d_{smaller} = 100$ and the minimum diameter is 200 pixels.
3. **Supervised post-processing**, in this block there are some tools to help the user to supervise the recognition results of the automatic algorithms. These tools are detailed in Section B.7.1.
4. **Classification utilities**, there are a set of functionalities to manage the issues related to the assignment of categories to the oocytes being automatically or manually recognized (see Section B.9 for a further description).
5. **Visualization of results**, pressing the **Show Table** button, a table containing analysis information will be opened at the bottom of the main window. The table will show the diameter, the area, the development stage, the visible/not visible nucleus, if the cell is counted or not, and if the object is measured or not. A detailed description of these functionalities is provided in Section B.7.2.
6. **Iconized image for visualize position**, at the bottom of the lateral panel, there is a miniature image of the original image loaded in STERapp. Over this miniature there is

a blue shaded square representing the area of the original image currently shown in the image window. The position and size of this area depends on the zoom used, as it can be seen in Figure B.15.

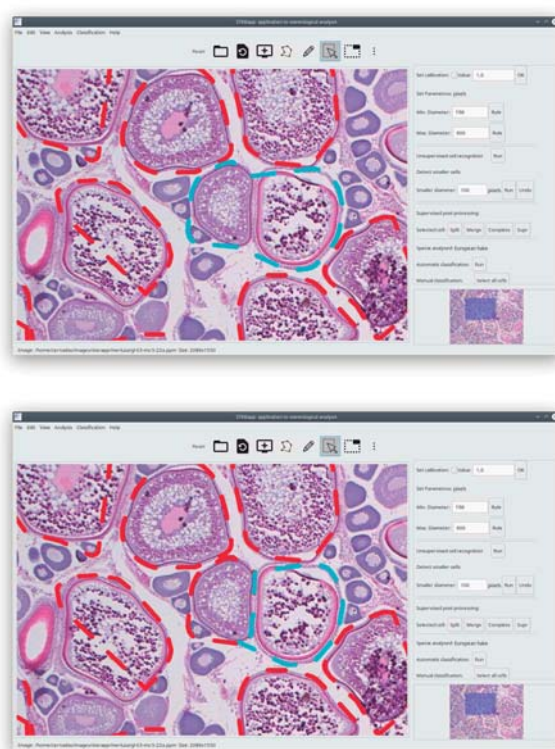


Figure B.17: Split operation (button **Split** in the lateral panel) to divide one selected object (magenta in the upper panel) into two regions.

B.7.1 Supervised post-processing

After the automatic processing to recognize the oocytes (see Figure B.16), the user can review and edit the results using the tools to modify, add and/or remove misclassified oocytes. To add an oocyte you must select the drawing tool on the toolbar and draw the contour using the

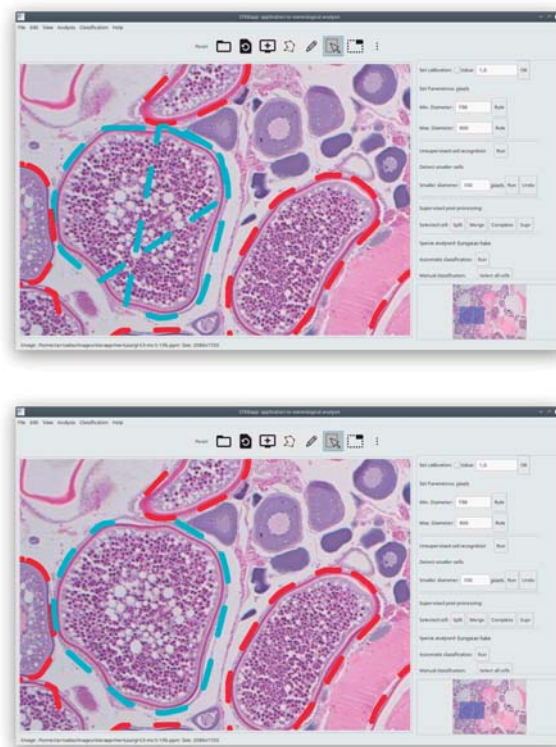


Figure B.18: Merge operation (button **Merge** in the lateral panel) to merge various selected cells into one cell. The selected oocytes to merge in the upper panel are in magenta colour.

mouse. To remove one or more oocytes you must first select them and then press the **Supr** button of the keyboard or the **Supr** button on the lateral panel. To select oocytes, press the **Select cells** icon on the toolbar and click inside the oocyte with the left button of the mouse. To select more than one object repeat the process keeping pressed “Ctrl” in the keyboard. Buttons **Split**, **Merge**, **Complete** and **Supr** must be applied for a selection of oocytes.

1. **Split**: divides one selected object into two oocytes when the automatic processing fails in the recognition of more than one particle. To do the split operation, the user must first select one object, press then the button **Split** (this button will be activated), and draw a line or an arc on the selection which follows the contour of the desired oocyte

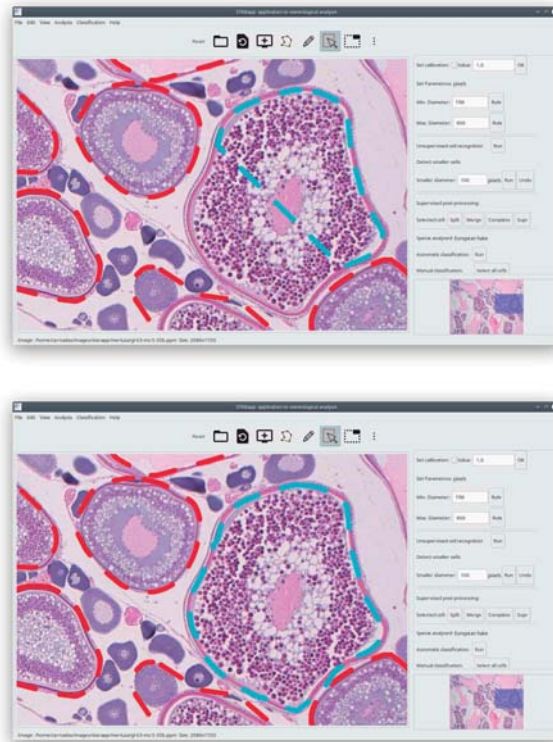


Figure B.19: Result of the operation **Complete** of the lateral panel. Complete the contour of a selected oocyte that was inaccurately recognized (magenta colour).

contour. The **Split** button will be deactivated and the selected object will be divided into two. Figure B.17 shows two images representing the state before and after the split operation. If one of the oocytes as a result of the splitting process is smaller than allowed by the minimum diameter, this oocyte or object is automatically removed.

2. **Merge**: fuses the selected contours, which do not correspond to single oocytes, into one. To do the merge operation, you must select the set of objects you want to merge (i.e., using “Ctrl”) and click **Merge** in the lateral panel. After that, the separated contours will not be visible any more and they will appear as a single oocyte. Figure B.18 shows two images that represent the state before and after the merge operation.

3. **Complete:** adds an outline to a selected oocyte when the automatic processing has not recognized the entire contour. To do such a completing operation, the user must select the incomplete oocyte and click **Complete** in the lateral panel. This button will be activated as long as the user draws on the image to complete the actual outline of the selected oocyte. When the user releases the left button of the mouse, the selected oocyte will be completed and the **Complete** button will be deactivated, as it is shown in the Figure B.19.
4. **Supr:** selects an oocyte or a set of oocytes. You can click the button **Supr** to remove them. The selected objects can also be removed pressing “Supr” on the keyboard.

B.7.2 Visualization of results

Press the button **Show Table** of the lateral panel to open a table at the bottom of the main window (see Figure B.20) that shows the information of the detected oocytes in the image. The results table will show diameter, area, development stage, visible/not visible nucleous and whether the oocyte is counted (measured) or not. The user can select an oocyte outline on the image (cyan contour) and the row of the table containing the information of that cell will be activated, as it can be seen in the upper panel of Figure B.20. Inversely, if you select a row in the table, its corresponding oocyte in the image window will be appear as selected (cyan colour), as it can be seen in the lower panel of Figure B.20. The information of this table can be exported to a CSV file, as it has been described in Section B.5.

B.8 Analysis menu

The only item available in this menu is **XML files**, which opens a pop-up window (Figure B.21). There, the user can choose the XML files of every processed image in order to gather the results in one single CSV file, as they can correspond to a single individual, samples, etc. For this purpose, the user must first process individually each image and save the recognition results (outline of the oocytes of each image) in an XML file; once finished, run **Analysis**, select the **XML files**, and get one single results sheet.

In Figure B.21, by default the selected xml directory is the folder of the preferences for XML files, but it can be changed by clicking on the three points button. After that, click in the button **Click to select files** to select the XML files to be included in the anlysis, opening the

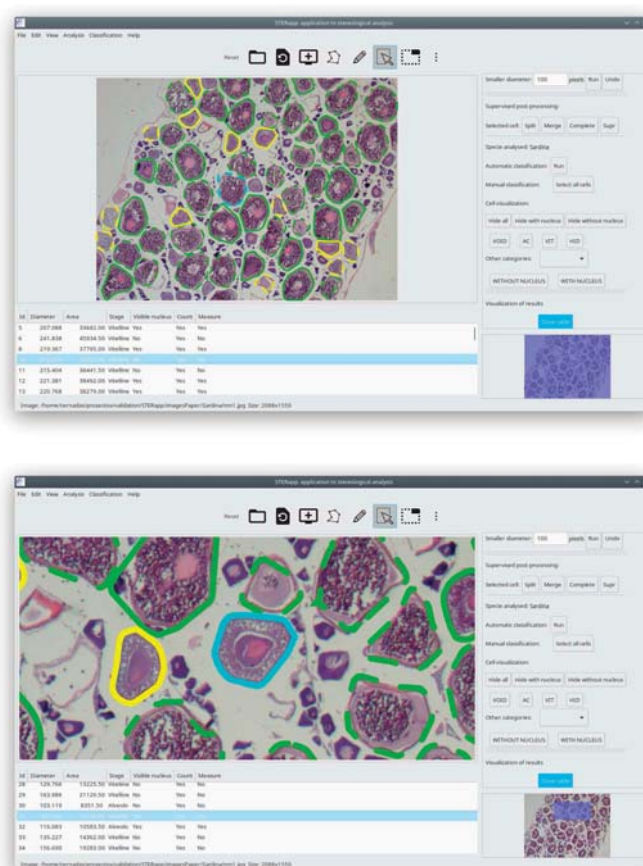


Figure B.20: Visualization of information about the oocytes overlapped to the image in the bottom table (button **Show Table** in the lateral panel). The cyan contour is the selected cell.

window of Figure B.22. To select contiguous files, click on the file name with the left button mouse and keeping the key **Alt** pressed, choose the last file. To select non-contiguous files, click with the left button mouse on the file and, keeping the **Ctrl** key pressed, click on other files. Once the selection is made, click **Select** and the files selected appear in the selection XML files box (Figure B.23). Clicking on button **Export CSV** (see Figure B.23) to export the statistical results to a CSV file (Figure B.24 shows the CSV files generated); or press the

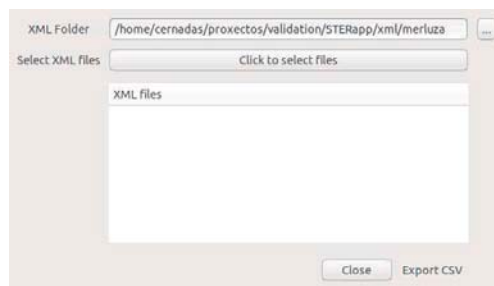


Figure B.21: Window open when the submenu **XML File** of the menu **Analysis** is chosen.

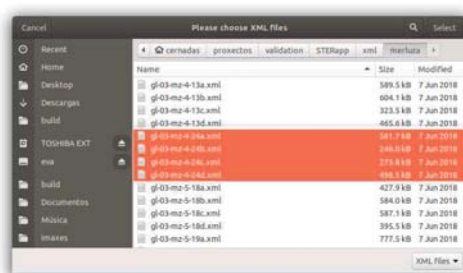


Figure B.22: File chooser dialog to choose the XML files included in the joint analysis.

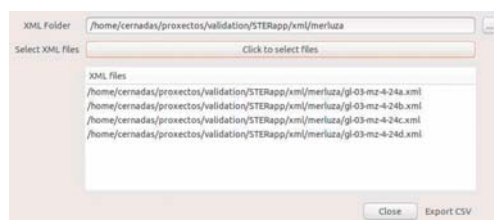


Figure B.23: Window open when the submenu **XML File** of the menu **Analysis** is chosen and after selecting the XML files used in the analysis.

Close button to cancel the operation. Following, the user will be asked for the name of the results CSV file. The operation could require some seconds depending of the amount of files to analyse. After finishing the process, the user must close the window in the **Close** button (in Figure B.23) or perform another procedure.

Image	Image area	Calibration	Species	ID	Diameter	Area	Stage	Visible nucleous	Count	Measure
G-1Mb Govocitos/Mb_F0813_043a.tif	87911556,324	4,320	Granadero	14	1316,462	1361151,583	Vitelline	Yes	Yes	Yes
G-1Mb Govocitos/Mb_F0813_043a.tif	87911556,324	4,320	Granadero	52	611,981	294147,441	Alveolo	No	Yes	No
G-1Mb Govocitos/Mb_F0813_043a.tif	87911556,324	4,320	Granadero	55	1327,274	1383602,452	Vitelline	Yes	Yes	Yes
G-1Mb Govocitos/Mb_F0813_043a.tif	87911556,324	4,320	Granadero	59	678,261	361313,424	Alveolo	No	Yes	No
G-1Mb Govocitos/Mb_F0813_043a.tif	87911556,324	4,320	Granadero	236	584,505	268328,009	Alveolo	No	Yes	No
G-1Mb Govocitos/Mb_F0813_043a.tif	87911556,324	4,320	Granadero	275	451,815	160328,691	Alveolo	No	Yes	No
G-1Mb Govocitos/Mb_F0813_043a.tif	87911556,324	4,320	Granadero	404	511,008	205090,461	Vitelline	No	No	No
G-1Mb Govocitos/Mb_F0813_043a.tif	87911556,324	4,320	Granadero	406	317,561	79203,232	Alveolo	No	Yes	No
G-1Mb Govocitos/Mb_F0813_043a.tif	87911556,324	4,320	Granadero	407	2079,680	3396902,324	Hydrated	No	Yes	No
/home/cernadas/proyectos/validation/STERapp/imagesPa...	24924149,371	1,483	Granadero	70	703,907	389152,943	Vitelline	Yes	Yes	Yes
/home/cernadas/proyectos/validation/STERapp/imagesPa...	24924149,371	1,483	Granadero	73	458,539	165136,058	Alveolo	Yes	Yes	Yes
/home/cernadas/proyectos/validation/STERapp/imagesPa...	24924149,371	1,483	Granadero	86	505,865	200983,156	Vitelline	No	Yes	No
/home/cernadas/proyectos/validation/STERapp/imagesPa...	24924149,371	1,483	Granadero	156	122,898	11862,562	Vitelline	No	No	No
/home/cernadas/proyectos/validation/STERapp/imagesPa...	24924149,371	1,483	Granadero	157	163,890	21095,839	Vitelline	No	No	No
/home/cernadas/proyectos/validation/STERapp/imagesPa...	24924149,371	1,483	Granadero	158	304,513	72828,571	Alveolo	Yes	Yes	Yes
/home/cernadas/proyectos/validation/STERapp/imagesPa...	24924149,371	1,483	Granadero	160	276,547	60065,764	Alveolo	No	Yes	No
/home/cernadas/proyectos/validation/STERapp/imagesPa...	24924149,371	1,483	Granadero	162	363,960	104039,245	Vitelline	Yes	No	No
/home/cernadas/proyectos/validation/STERapp/imagesPa...	19219220,743	1,302	Granadero	1	752,471	444702,897	Vitelline	Yes	No	No
/home/cernadas/proyectos/validation/STERapp/imagesPa...	19219220,743	1,302	Granadero	6	410,599	132411,546	Alveolo	Yes	Yes	Yes
/home/cernadas/proyectos/validation/STERapp/imagesPa...	19219220,743	1,302	Granadero	8	820,241	528412,076	Vitelline	No	Yes	No
/home/cernadas/proyectos/validation/STERapp/imagesPa...	19219220,743	1,302	Granadero	9	455,472	162934,544	Vitelline	No	Yes	No
/home/cernadas/proyectos/validation/STERapp/imagesPa...	19219220,743	1,302	Granadero	12	383,946	115779,051	Alveolo	No	Yes	No
/home/cernadas/proyectos/validation/STERapp/imagesPa...	19219220,743	1,302	Granadero	39	719,371	406439,597	Vitelline	No	Yes	No
/home/cernadas/proyectos/validation/STERapp/imagesPa...	19219220,743	1,302	Granadero	40	342,487	92125,020	Alveolo	No	Yes	No
/home/cernadas/proyectos/validation/STERapp/imagesPa...	19219220,743	1,302	Granadero	42	226,502	40293,307	Vitelline	No	Yes	No
/home/cernadas/proyectos/validation/STERapp/imagesPa...	19219220,743	1,302	Granadero	44	272,784	58442,162	Alveolo	No	Yes	No

Figure B.24: An example of a CSV file of the analysis of various XML files imported in LibreOffice Calc.

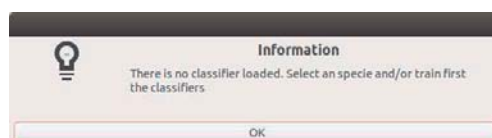


Figure B.25: Pop-up message informing that the classifier is not trained for the fish specie considered.

B.9 Classification menu

The classification menu has two submenus: **Classify** and **Train classifier**, whose use will be described below.

The oocytes as recognized in the histological image can be classified following two criteria, “Classes” and “Stage”.

1. **Classes:** describes the way in which the oocyte has been sectioned, i.e., showing or not showing nucleous. It deeply affects the diameter estimates. An oocyte without visible nucleous in the image indicates that the oocyte has not been cut by its middle section so, its diameter does not correspond to the real diameter of the spherical cell. Once an oocyte or set of oocytes are selected on the image, the class of the selected

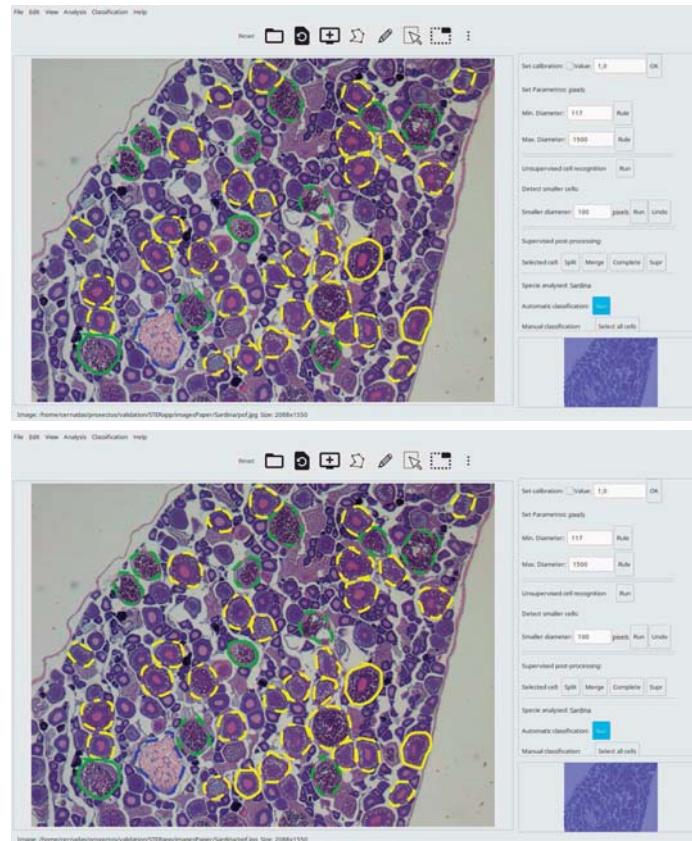


Figure B.26: An example of oocytes classified manually by expert (upper panel) and after applying the classifier (lower panel).

oocytes can be manually set clicking the buttons **WITHOUT NUCLEOUS** or **WITH NUCLEOUS** in the lateral panel. The oocytes with visible nucleous are shown with a continuous line and the oocytes with no visible nucleous are shown with a dashed line.

2. **Stages:** the oocytes can be at different developmental stage within the same ovary section depending on the specie. These maturity stages can be cortical alveoli, vitelogenic, atretic, hydrated, primary growth, etc. For automatic classification, STERapp allows, by default, only three of this categories: cortical alveoli, vitelogenic and hydrated, plus the category “Void”, that represents the area in the image that does not correspond to

gonadal tissue and therefore shouldn't be evaluated. As it was described in Section B.4, STERapp allows to define or register new categories or stages and the user can associate a colour to show them on the image window. If the user wants to assign a non predefined category, she must select an item in the drop-down list after the label *Other categories* (see Section B.4 to define new categories in the working preferences). Once an oocyte or set of them are selected, the category or stage can be manually set by clicking the corresponding buttons **VOID**, **AC**, **VIT** and **HID** of the lateral panel to assign the stage "void", "cortical alveoli", "vitelogenic" and "hydrated", respectively.

The user can only apply the classification process over recognized oocytes in an image loaded in the image window. To classify automatically the oocytes, go to the submenu **Classify** or click the button **Run** in the lateral panel after the label "Automatic classification". Before running the classification, the user must make sure that the selected fish species in the preferences is correct (shown in the lateral panel after the label "Specie Analysed"). If the user tries to do the classification before the training, the program will show a pop-up message (Figure B.25), warning that the classifier for the mentioned species was not trained. If there already exists a trained classifier for that species, STERapp runs it and shows the results (the upper panel of Figure B.26 shows the cells manually classified by the expert, and the lower panel shows the output provided by the automatic classifier).

The stage or class of the oocytes can be manually assigned using the GUI. STERapp also implements the automatic classification of oocytes for the predefined stages. This procedure operates over the recognized oocytes, but again, the classifier needs to be trained. The classifier is a supervised machine learning technique, which needs to learn the function to predict, in this case, the stage and class of the oocytes which have not been seen before. To learn this function it is necessary to provide the classifier a set of images which contains oocytes of every predefined stage and class. So first, the user must manually analyse a limited number of images for a specific fish species, save the annotated oocytes overlapped to the image in their corresponding XML files, and train the classifier for that species. After the training, the classification operation can be applied with new images or individuals, using the submenu **Classify** or clicking the button **Run** in the lateral panel after "Automatic classification".

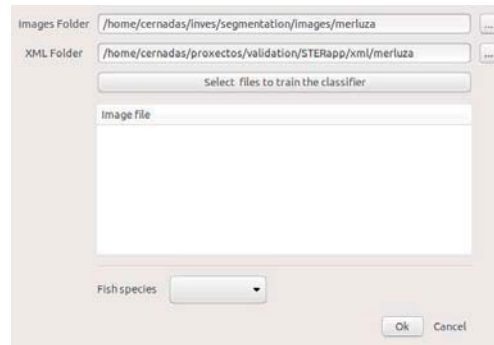


Figure B.27: Dialog window to train the classifier.

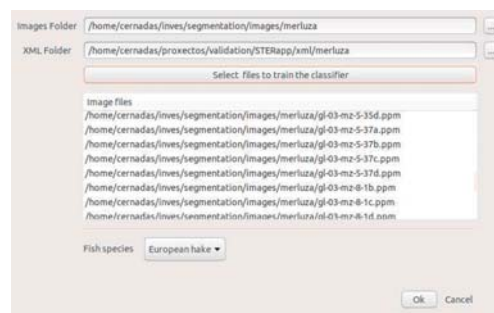


Figure B.28: Figure B.27 after the configuration of the parameters to train the classifier.

B.9.1 Training the classifier

To train the classifier, please select the submenu **Train classifier** in the “Classification” menu, which will open the dialog window of Figure B.27. Select the image and XML files used in the training process, and click on the button **Select files to train the classifier** which will open a window similar to Figure B.22 to select the files used. STERapp will train the classifier. At the bottom of the panel of Figure B.27, there is a drop-up list “Fish species” where the user has to select the fish species for which he wants to train (among the fish species defined in STERapp when setting the preferences in Section B.4). Clicking **Ok** starts the training, and clicking on **Cancel** closes the training process (Figure B.28). This process can take several minutes, depending on the number of images used. During the training, the program cannot be used.

B.10 Help menu

The **Help** menu is loaded within the menu bar with the submenus **User Manual** and **About Us**. The submenu **About Us** pop-ups a window with a short description of STERapp and information about its developers and the license (see Figure B.29).



Figure B.29: Pop-up dialog to inform about help.

Bibliography

- [1] Y. Al-Kofahi, W. Lassoued, W. Lee, and B. Roysam. Improved automatic detection and segmentation of cell nuclei in histopathology images. *IEEE Transactions on Biomedical Engineering*, 57(4):841–852, 2010.
- [2] Rowel Atienza. *Advanced Deep Learning with TensorFlow 2 and Keras: Apply DL, GANs, VAEs, deep RL, unsupervised learning, object detection and segmentation, and more*. Packt Publishing Ltd, 2020.
- [3] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(12):2481–2495, 2017.
- [4] A. Bangor, P. Kortum, and J. Miller. Determining what individual SUS scores mean: Adding and adjective rating scale. *Journal of Usability Studies*, 4(3):114 – 123, 2009.
- [5] T. B. Begenal and E. Braum. Eggs and early life history. In *Method for assessment of fish production in fresh waters. Third edition. IBP Handbook No3. Blackwell scientific publication*, pages 165–201, 1978.
- [6] John Brooke. SUS: A retrospective. *Journal of Usability Studies*, 8(2):29 – 40, 2013.
- [7] R. E. Burrows. A method for the enumeration of salmon and trout eggs by displacement. *The Progressive Fish-Culturist*, 13:25–30, 1951.
- [8] J. F. Canny. A computational approach to edge detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986.
- [9] Claude Cariou, Steven Le Moan, and Kacem Chehdi. A novel mean-shift algorithm for data clustering. *IEEE Access*, 10:14575–14585, 2022.

- [10] E. Cernadas, M. Fernández-Delgado, E. González-Rufino, and P. Carrión. Influence of normalization and color space to color texture classification. *Pattern Recogn.*, 61:120 – 138, 2017.
- [11] T.F. Chan and L.A. Vese. Active contours without edges. *IEEE Trans. Image Process.*, 10(1):266–277, 2001.
- [12] C.C. Chang and C.J. Lin. LIBSVM: a library for support vector machines. *ACM T Intel. Syst. Tec.*, 2:27:1–27:27, 2011.
- [13] Hang Chang, Yin Zhou, Alexander Borowsky, Kenneth Barner, Paul Spellman, and Bahram Parvin. Stacked predictive sparse decomposition for classification of histology sections. *International Journal of Computer Vision*, 113(1):3–18, 2015.
- [14] Jiaquan Chen, Binglei Guan, Hailun Wang, Xuguang Zhang, Yinggan Tang, and Wenzhao Hu. Image thresholding segmentation based on two dimensional histogram using gray level and local entropy information. *IEEE Access*, 6:5269–5275, 2018.
- [15] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):834–848, 2018.
- [16] Hanjoo Cho, Suk-Ju Kang, and Young Hwan Kim. Image segmentation using linked mean-shift vectors and global/local attributes. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(10):2132–2140, 2017.
- [17] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5):564–577, 2003.
- [18] Dorin Comaniciu and Peter Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(5):603–619, 2002.
- [19] Adrián Cordido, Eva Cernadas, Manuel Fernández-Delgado, and Miguel A. García-González. Cystanalyser: A new software tool for the automatic detection and quantification of cysts in polycystic kidney and liver disease, and other cystic disorders. *PLOS Comput Biol*, 16(10):1–18, 10 2020.

- [20] Daniel Cremers, Mikael Rousson, and Rachid Deriche. A review of statistical approaches to level set segmentation: Integrating color, texture, motion and shape. *International Journal of Computer Vision*, 72(2):195–215, 2007.
- [21] G.J.A. de Melo, V. Gomes, C.C. Baccili, L.A.L. de Almeida, and A.C. de C. Lima. A robust segmentation method for counting bovine milk somatic cells in microscope slide images. *Computers and Electronics in Agriculture*, 115, 2015.
- [22] Shujian Deng, Xin Zhang, Wen Yan, Eric I-Chao Chang, Yubo Fan, Maode Lai, and Yan Xu. Deep learning in digital pathology image analysis: a survey. *Frontiers of medicine*, 14:470–487, 2020.
- [23] R. Domínguez-Petit. *Study of reproductive potential of European hake European hake in the Galician Shelf*. PhD thesis, University of Vigo, Vigo (Spain), 2007.
- [24] L. S. Emerson, M. Greer-Walker, and P. R. Witthames. A stereological method for estimating fish fecundity. *Journal of Fish Biology*, 36:721–730, 1990.
- [25] Francisco J. Estrada and Allan D. Jepson. Benchmarking image segmentation algorithms. *International Journal of Computer Vision*, 85:167–181, 2009.
- [26] Absalom El-Shamir Ezugwu, Moyinoluwa B. Agbaje, Nahla Aljojo, Rosanne Els, Haruna Chiroma, and Mohamed Abd Elaziz. A comparative performance study of hybrid firefly algorithms for automatic data clustering. *IEEE Access*, 8:121089–121118, 2020.
- [27] Deng-Ping Fan, Tao Zhou, Ge-Peng Ji, Yi Zhou, Geng Chen, Huazhu Fu, Jianbing Shen, and Ling Shao. Inf-net: Automatic covid-19 lung infection segmentation from ct images. *IEEE Transactions on Medical Imaging*, 39(8):2626–2637, 2020.
- [28] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181, 2004.
- [29] Manuel Fernández-Delgado, Eva Cernadas, Senén Barro, and Dinani Amorim. Do we need hundreds of classifiers to solve real world classification problems? *J Mach Learn Res*, 15:3133–3181, 2014.
- [30] F. E. J. Fry. Statistics of a lake trout fishery. *Biometrics*, 5:27–67, 1949.

- [31] Nupur J. Gandhi, Vandana J. Shah, and Ravindra Kshirsagar. Mean shift technique for image segmentation and modified canny edge detection algorithm for circle detection. In *2014 International Conference on Communication and Signal Processing*, pages 246–250, 2014.
- [32] Pascal Getreuer. Chan-Vese Segmentation. *Image Processing On Line*, 2:214–224, 2012.
- [33] Mohsen Ghanbari, David A. Clausi, and Linlin Xu. Cp-irgs: A region-based segmentation of multilook complex compact polarimetric sar data. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 14:6559–6571, 2021.
- [34] E. González-Rufino, P. Carrión, E. Cernadas, M. Fernández-Delgado, and R. Domínguez-Petit. Exhaustive comparison of colour texture features and classification methods to discriminate cells categories in histological images of fish ovary. *Pattern Recognition*, 46:2391 – 2407, 2013.
- [35] Leo Grady. Random walks for image segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, pages 1768–1783, 2016.
- [36] Cigdem Gunduz-Demir, Melih Kandemir, Akif Burak Tosun, and Cenk Sokmensuer. Automatic segmentation of colon glands using object-graphs. *Medical Image Analysis*, 14(1):1 – 12, 2010.
- [37] Metin N. Gurcan, Laura E. Boucheron, Ali Can, Anant Madabhushi, Nasir M. Rajpoot, and Bulent Yener. Histopathological image analysis: A review. *IEEE Reviews in Biomedical Engineering*, 7:97–114, 2014.
- [38] C. F. Hickling. The fecundity of herring of the southern North Sea. *J. Mar. Biol. Ass. UK.*, 24:619–632, 1940.
- [39] Zhongwen Hu, Qian Zhang, Qin Zou, Qingquan Li, and Guofeng Wu. Stepwise evolution analysis of the region-merging segmentation for scale parameterization. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 11(7):2461–2472, 2018.

- [40] Po-Whei Huang and Yan-Hao Lai. Effective segmentation and classification for {HCC} biopsy images. *Pattern Recognition*, 43(4):1550 – 1563, 2010.
- [41] J. R. Hunter, J. Macewicz, N. C. H. LO, and C. A. Kimbrell. Fecundity, spawning, and maturity of female Dover Sole, *Microstomus pacificus*, with an evaluation of assumptions and precision. *Fishery Bulletin*, 90:101–128, 1992.
- [42] H. Irshad, A. Veillard, L. Roux, and D. Racoceanu. Methods for nuclei detection, segmentation, and classification in digital histopathology: A review – current status and future potential. *IEEE Reviews in Biomedical Engineering*, 7:97–114, 2014.
- [43] Hao Jiang, Yanning Zhou, Yi Lin, Ronald C.K. Chan, Jiang Liu, and Hao Chen. Deep learning for computational cytology: A survey. *Medical Image Analysis*, 84:102691, 2023.
- [44] T. Katsukawa. Points of view: Introduction of spawning potential: improvement in the threshold management theory. *Review in Fish Biology and Fisheries*, 7:2085–289, 1997.
- [45] C. Kervrann, S. T. Acton, J.-C. Olivo-Marin, C.O.S. Sorzano, and M. Unser. Introduction to the issue on advanced signal processing in microscopy and cell imaging. *IEEE Journal of Selected Topics in Signal Processing*, 10(1):3–5, 2016.
- [46] Hui Kong, M. Gurcan, and K. Belkacem-Boussaid. Partitioning histopathological images: An integrated framework for supervised color-texture segmentation and cell splitting. *IEEE Transactions on Medical Imaging*, 30(9):1661–1677, 2011.
- [47] J. Kong, O. Sertel, H. Shimada, K.L. Boyer, J.H. Saltz, and M.N. Gurcan. Computer-aided evaluation of neuroblastoma on whole-slide histology images: Classifying grade of neuroblastic differentiation. *Pattern Recognition*, 42(6):1080 – 1092, 2009.
- [48] Florian Kromp, Lukas Fischer, Eva Bozsaky, Inge M Ambros, Wolfgang Dörr, Klaus Beiske, Peter F Ambros, Allan Hanbury, and Sabine Taschner-Mandl. Evaluation of deep learning architectures for complex immunofluorescence nuclear image segmentation. *IEEE Transactions on Medical Imaging*, 40(7):1934–1949, 2021.

- [49] R. Lasker. An egg production method for estimating spawning biomass of pelagic fish: Application to the Northern anchovy, *Engraulis mordax*. Technical Report 36, U.S. Dep. Comm., 1985.
- [50] Haichun Li, Ao Li, and Minghui Wang. A novel end-to-end brain tumor segmentation method using improved fully convolutional networks. *Computers in biology and medicine*, 108:150–160, 2019.
- [51] A. Lindroth. Salmon Stripper, Egg Counter, and Incubator. *The Progressive Fish-Culturist*, 18:165–170, 1956.
- [52] Shigang Liu and Yali Peng. A local region-based chan–vese model for image segmentation. *Pattern Recognition*, 45(7):2769–2779, 2012.
- [53] Xiangbin Liu, Liping Song, Shuai Liu, and Yudong Zhang. A review of deep-learning-based medical image segmentation methods. *Sustainability*, 13(3), 2021.
- [54] Yang Liu, Zongwu Xie, and Hong Liu. An adaptive and robust edge detection method based on edge proportion statistics. *IEEE Transactions on Image Processing*, 29:5206–5215, 2020.
- [55] Feixiao Long. Microscopy cell nuclei segmentation with enhanced u-net. *BMC bioinformatics*, 21:1–12, 2020.
- [56] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440, 2015.
- [57] Almoutaz Mbaidin, Eva Cernadas, Zakaria Al-Tarawneh, and Manuel Fernández-Delgado. Recognizing cells in histological images of fish gonads. In *2022 International Conference on Emerging Trends in Computing and Engineering Applications (ETCEA)*, pages 1–6, 2022.
- [58] Almoutaz Mbaidin, Eva Cernadas, Zakaria A. Al-Tarawneh, Manuel Fernández-Delgado, Rosario Domínguez-Petit, Sonia Rábade-Uberos, and Ahmad Hassanat. MSCF: Multi-Scale Canny Filter to Recognize Cells in Microscopic Images. *Sustainability*, 15(18), 2023.

- [59] Almoutaz Mbaidin, Sonia Rábade-Uberos, Rosario Dominguez-Petit, Andrés Villaverde, María Encarnación González-Rufino, Arno Formella, Manuel Fernández-Delgado, and Eva Cernadas. STERapp: Semiautomatic software for stereological analysis. application in the estimation of fish fecundity. *Electronics*, 10(12), 2021.
- [60] E. A. McGregor. Observations on the egg yield of Klamath river king salmon. *Californi Fish and Game*, 8:160–164, 1922.
- [61] Mary L McHugh. Interrater reliability: the kappa statistic. *Biochemia medica: Biochemia medica*, 22(3):276–282, 2012.
- [62] R. Medina-Carnicer, R. Muñoz-Salinas, E. Yeguas-Bolivar, and L. Diaz-Mas. A novel method to look for the hysteresis thresholds for the canny edge detector. *Pattern Recognition*, 44(6):1201 – 1211, 2011.
- [63] David Mumford and Jayant Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on Pure and Applied Mathematics*, 42(5):577–685, 1989.
- [64] Wan Azani Mustafa, Wan Khairunizam, Zunaidi Ibrahim, Shahrman A. B., and Zuradzman M. Razlan. A review of different segmentation approach on non uniform images. In *2018 International Conference on Computational Approach in Smart Systems Design and Applications (ICASSDA)*, pages 1–6, 2018.
- [65] T. Ojala, M. Piatikäinen, and T. Mäenpää. Multiresolution grey-scale and rotation invariant texture classification with Local Binary Pattern. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24 (7):971–987, 2002.
- [66] Bo Peng, D. Zhang, and D. Zhang. Automatic image segmentation by dynamic region merging. *IEEE Transactions on Image Processing*, 20(12):3592–3605, 2011.
- [67] Bo Peng, Lei Zhang, and David Zhang. A survey of graph theoretical approaches to image segmenetation. *Pattern Recognition*, 42:1020–1038, 2013.
- [68] J.M. Pintor, P. Carrión, E. Cernadas, E. González-Rufino, A. Formella, M. Fernández-Delgado, R. Domínguez-Petit, and S. Rábade-Uberos. Govocitos: A software tool for estimating fish fecundity based on digital analysis of histological images. *Comput Electr Agric*, 125:89 – 98, 2016.

- [69] J. A. Pope, D. H. Mills, and W. M. Shearer. The fecundity of the Atlantic salmon (*Salmo salar* L.). *Freswat. Salm. Fish. Res. Scotland*, 26:1–12, 1961.
- [70] R. C. González and R. E. Woods. *Digital Image Processing (Third Edition)*. Pearson Prentice Hall, 2008.
- [71] D. S. Raitt. The fecundity of the haddock. *Scientific Investigations. Scotland Fishery Board*, 1:1–40, 1933.
- [72] Hiba Ramadan, Chaymae Lachqar, and Hamid Tairi. A survey of recent interactive image segmentation methods. *Computational Visual Media*, 6(4):355–384, 2020.
- [73] Michael Roberts and Jack Spencer. Chan-veye reformulation for selective image segmentation, 2019.
- [74] Curtis T. Rueden, Johannes Schindelin, Mark C. Hiner, Barry E. DeZonia, Alison E. Walter, Ellen T. Arena, and Kevin W. Eliceiri. ImageJ2: ImageJ for the next generation of scientific image data. *BMC Bioinf*, 18:529, 2017.
- [75] J. C. Russ and R. T. Dehoff. *Practical Stereology*. Springer Science and Business Media, 2012.
- [76] S. Suzuki and K. Be. Topological structural analysis of digitized binary images by border following. *Comput Vis Graph Image Proc*, 30(1):32 – 46, 1985.
- [77] Jeff Sauro. *A practical guide to the System Usability Scale: Background, Benchmarks & Best Practices*. CreateSpace Independent Publishing Platform, 2011.
- [78] Johannes Schindelin, Ignacio Arganda-Carreras, Erwin Frise, Verena Kaynig, Mark Longair, Tobias Pietzsch, Stephan Preibisch, Curtis Rueden, Stephan Saalfeld, Benjamin Schmid, Jean-Yves Tinevez, Daniel James White, Volker Hartenstein, Kevin Eliceiri, Pavel Tomancak, and Albert Cardona. Fiji: an open-source platform for biological-image analysis. *Nat Methods*, 9:676–682, 2012.
- [79] O. Sertel, J. Kong, H. Shimada, U.V. Catalyurek, J.H. Saltz, and M.N. Gurcan. Computer-aided prognosis of neuroblastoma on whole-slide images: Classification of stromal development. *Pattern Recognition*, 42(6):1093 – 1103, 2009.

- [80] Jianbo Shi and Jiterdra Malik. Normalized cuts and image segmentation. *IEEE Trans. on Pattern Analysis and Machine Ingelligence*, 22(8):888–905, 2000.
- [81] Kristina P. Sinaga and Miin-Shen Yang. Unsupervised k-means clustering algorithm. *IEEE Access*, 8:80716–80727, 2020.
- [82] Jintao Song, Huizhu Pan, Wanquan Liu, Zisen Xu, and Zhenkuan Pan. The chan-vese model with elastica and landmark constraints for image segmentation. *IEEE Access*, 9:3508–3516, 2021.
- [83] Liyilei Su, Xianjun Fu, Xiaodong Zhang, Xiaoguang Cheng, Yimin Ma, Yungen Gan, and Qingmao Hu. Delineation of carpal bones from hand x-ray images through prior model, and integration of region-based and boundary-based segmentations. *IEEE Access*, 6:19993–20008, 2018.
- [84] Vinh-Thong Ta, Olivier Lezoray, Abderrahim Elmoataz, and Sophie Schüpp. Graph-based tools for microscopic cellular image segmentation. *Pattern Recognition*, 42:1113–1125, 2009.
- [85] Xiao Tan, Huaian Chen, Kai Xu, Chunmei Xu, Yi Jin, Changan Zhu, and Jinjin Zheng. High dynamic range imaging for dynamic scenes with large-scale motions and severe saturation. *IEEE Transactions on Instrumentation and Measurement*, 71:1–15, 2022.
- [86] Wenbing Tao, Hai Jin, and Yimin Zhang. Color image segmentation based on mean shift and normalized cuts. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 37(5):1382–1389, 2007.
- [87] A. Thorsen and O. S. Kjesbu. A rapid method for estimation of oocyte size and potential fecundity in atlantic cod using a computer-aided particle analysis system. *J. Sea Res.*, 46(3-4):295–308, 2001.
- [88] A. B. Tosun and C. Gunduz-Demir. Graph run-length matrices for histopathological image segmentation. *IEEE Transactions on Medical Imaging*, 30(3):721–732, 2011.
- [89] Ranjith Unnikrishnan, Caroline Pantofaru, and Martial Hebert. Toward objective evaluation of image segmentation algorithms. *IEEE Trans. on Pattern Analysis and Machine Ingelligence*, 29(6):929–944, 2007.

- [90] M. Varma and A. Zisserman. A statistical approach to material classification using image patch exemplars. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 31 (11):2032–2047, 2009.
- [91] Dan Wang, Guoqing Hu, Qianbo Liu, Chengzhi Lyu, and Md Mojahidul Islam. Region-based nonparametric model for interactive image segmentation. *IEEE Access*, 7:111124–111134, 2019.
- [92] Pin Wang, Xianling Hu, Yongming Li, Qianqian Liu, and Xinjian Zhu. Automatic cell nuclei segmentation and classification of breast cancer histopathology images. *Signal Processing*, 122:1 – 13, 2016.
- [93] Tao Wang, Quansen Sun, Zexuan Ji, Qiang Chen, and Peng Fu. Multi-layer graph constraints for interactive image segmentation via game teory. *Pattern Recognition*, 44:2478–2501, 2011.
- [94] Xiao-Feng Wang, Hai Min, Le Zou, and Yi-Gang Zhang. A novel set method for image segmentation by incorporating local statistical analysis and global similarity measurement. *Pattern Recognition*, 48:189–204, 2015.
- [95] Yixin Wang, Yao Zhang, Jiang Tian, Cheng Zhong, Zhongchao Shi, Yang Zhang, and Zhiqiang He. Double-uncertainty weighted method for semi-supervised learning. *CoRR*, abs/2010.09298, 2020.
- [96] E. R. Weibel. *Steorological method: Practical methods for Biological Morphometry*. Academic Press, 1979.
- [97] E. R. Weibel and D. M. Gómez. A principle for counting tissue structures on random sections. *Journal of Applied Physiology*, 17:343, 1962.
- [98] P. R. Witthames and M. G. Walker. An automated method for counting and sizing fish eggs. *Journal of Fish Biology*, 30:225–235, 1987.
- [99] D. R. Wolfert. Maturity and fecundity of walleyes from the eastern and western basins of Lake Erie. *J. Fish. Res. Bd. Can.*, 26:1877–1888, 1969.
- [100] Jun Xu, Lei Xiang, Qingshan Liu, H. Gilmore, Jianzhong Wu, Jinghai Tang, and A. Madabhushi. Stacked sparse autoencoder (ssae) for nuclei detection on breast

- cancer histopathology images. *IEEE Transactions on Medical Imaging*, 35(1):119–130, 2016.
- [101] Ryoya Yamasaki and Toshiyuki Tanaka. Properties of mean shift. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(9):2273–2286, 2020.
- [102] Rikiya Yamashita, Mizuho Nishio, Richard Kinh Gian Do, and Kaori Togashi. Convolutional neural networks: an overview and application in radiology. *Insights into imaging*, 9:611–629, 2018.
- [103] Gang Yao, Yang Wu, Xiaoxu Huang, Qinfeng Ma, and Jiang Du. Clustering of typical wind power scenarios based on k-means clustering algorithm and improved artificial bee colony algorithm. *IEEE Access*, 10:98752–98760, 2022.
- [104] Lequan Yu, Shujun Wang, Xiaomeng Li, Chi-Wing Fu, and Pheng-Ann Heng. Uncertainty-aware self-ensembling model for semi-supervised 3d left atrium segmentation. *CoRR*, abs/1907.07034, 2019.
- [105] Nida M. Zaitoun and Musbah J. Aqel. Survey on image segmentation techniques. *Procedia Computer Science*, 65:797 – 806, 2015.
- [106] Min Zhang, Kai Xie, Yu-Hang Zhang, Chang Wen, and Jian-Biao He. Fine segmentation on faces with masks based on a multistep iterative segmentation algorithm. *IEEE Access*, 10:75742–75753, 2022.
- [107] Hongming Zhou, Kang Song, Xianglei Zhang, Wenyong Gui, and Qiusuo Qian. Wails: Watershed algorithm with image-level supervision for weakly supervised semantic segmentation. *IEEE Access*, 7:42745–42756, 2019.

List of Figures

Fig. 1.1	Examples of histological images of the fish species <i>European Hake</i> (upper panel) and <i>Trisopterus luscus</i> (lower panel).	11
Fig. 2.1	Examples of histological images of <i>European Hake</i> fish species with the outline of the ground truth cells overlapped. Different colors mean different development stage of matured cells.	21
Fig. 2.2	Examples of histological images of <i>Trisopterus Luscus</i> fish species with the outline of the ground truth cells overlapped. Different colors mean different development stage of matured cells.	22
Fig. 2.3	Examples of histological images of <i>Atlantic Mackerel</i> fish species with the outline of the ground truth cells overlapped. Different colors mean different development stage of matured cells.	23
Fig. 2.4	Examples of histological images of <i>Ballan Wrasse</i> fish species with the outline of the ground truth cells overlapped. Different colors indicates different developmental stages of matured cells.	24
Fig. 2.5	Examples of histological images of <i>European Pilchard</i> fish species with the outline of the ground truth cells overlapped. Different colors indicates different developmental stages of matured cells.	25
Fig. 2.6	Examples of histological images of <i>Four-spot Megrin</i> fish species with the outline of the ground truth cells overlapped. Different colors indicates different developmental stages of matured cells.	26
Fig. 2.7	Examples of histological images of <i>Roughhead Grenadier</i> fish species with the outline of the ground truth cells overlapped. Different colors indicates different developmental stages of matured cells.	27

Fig. 2.8	Example of the operation of algorithm 3 to sort the point into a cell contour.	31
Fig. 3.1	Correct rate recognition cells for different values of tolerance for the fish species <i>European hake</i> (upper panel) and <i>Trisopterus luscus</i> (lower panel) for the segmentation algorithms tested using the minimum diameter for cells of $D_{min} = 100$ pixels.	46
Fig. 3.2	Correct rate recognition cells for different values of tolerance for the fish species <i>European pilchard</i> (upper panel) and <i>Four-spot megrim</i> (lower panel) for the segmentation algorithms tested using the minimum diameter for cells of $D_{min} = 100$ pixels.	47
Fig. 3.3	Correct rate recognition cells for different values of tolerance for the fish species <i>Roughhead grenadier</i> for the segmentation algorithms tested using the minimum diameter for cells of $D_{min} = 100$ pixels.	48
Fig. 3.4	Examples of histological images of <i>European Hake</i> fish species with the outline of the ground truth cells (in green color) and computer recognized cells (in blue color) overlapped. Image g1-03-mz-4-13d (upper left), g1-03-mz-4-24c (upper right), g1-04-mz-40-32b (lower left) and g1-03-mz-26-7b (upper right).	51
Fig. 3.5	Examples of histological images of <i>Trisopterus Luscus</i> fish species with the outline of the ground truth cells (in green color) and computer recognized cells (in blue color) overlapped. Image g1-05-t1-18-6-d (upper left), g1-06-t1-32-3-c (upper right), g1-05-t1-30-31-a (lower left) and g1-06-t1-39-13c (upper right).	52
Fig. 3.6	Examples of histological images of <i>European Pilchard</i> fish species with the outline of the ground truth cells (in green color) and computer recognized cells (in blue color) overlapped. Image nm2b (upper left), mn2 (upper right), pof-atr (lower left) and pof (upper right).	53
Fig. 3.7	Examples of histological images of <i>Four-spot Megrim</i> fish species with the outline of the ground truth cells (in green color) and computer recognized cells (in blue color) overlapped. Image g1-3-8-1b-a (upper left), g1-07-1b-3-9 (upper right), g1-07-1b-3-15 (lower left) and g1-07-1w-2-6-ac (upper right).	54

Fig. 3.8	Examples of histological images of <i>Roughhead Grenadier</i> fish species with the outline of the ground truth cells (in green color) and computer recognized cells (in blue color) overlapped. Image Mb-F0813-051b (upper left), Mb-F0813-050a (upper right), Mb-FC05-039a (lower left) and Mb-F6112-069 (upper right).	55
Fig. 3.9	Performance of the best segmentation algorithm for each image and all fish species: F1 at pixel level (upper graph) and correct rate recognition cells using the minimum diameter for cells of $D_{min} = 100$ pixels.	57
Fig. 4.1	Main window of STERapp with an histological image of European hake ovary. On the right, we can see the full lateral panel showing the available functionality of the module to recognize and classify oocytes.	64
Fig. 4.2	A flowchart containing the main tasks and functions of STERapp.	66
Fig. B.1	Pop-up message to inform that there are no species defined in STERapp. . .	85
Fig. B.2	Main window of STERapp.	86
Fig. B.3	Main window of STERapp with the lateral panel being open, which shows the plugin GONAD to process histological images of fish gonads.	87
Fig. B.4	Preferences panel: default configuration.	88
Fig. B.5	Setting preferences: (left panel) after setting the calibration, i.e., the relation between micrometers and pixels; (right panel) after setting the working directories.	90
Fig. B.6	Open dialog box to choose the working directory.	92
Fig. B.7	Chart colour for contour lines.	92
Fig. B.8	Setting preferences: (left panel) after adding new cell categories; (right panel) after setting some fish species.	93
Fig. B.9	Pop-up message to inform that the classifier must be trained in the future. . .	95
Fig. B.10	File chooser dialog to choose the image to be loaded in STERapp.	95
Fig. B.11	The image selected in Figure B.10 is loaded in STERapp.	96
Fig. B.12	Dialog to choose the CSV file to store the statistical analysis of the image. . .	96
Fig. B.13	An example of CSV file of the image of Figure B.11 imported in LibreOffice Calc.	97
Fig. B.14	Pop-up message shown when the user resets the preferences.	98
Fig. B.15	An example of the zoom functionality in STERapp.	99

Fig. B.16 Output after automatic processing of the histological image (upper panel) and output after automatic processing using the option “detect smaller cells” (bottom panel).	100
Fig. B.17 Split operation (button Split in the lateral panel) to divide one selected object (magenta in the upper panel) into two regions.	102
Fig. B.18 Merge operation (button Merge in the lateral panel) to merge various selected cells into one cell. The selected oocytes to merge in the upper panel are in magenta colour.	103
Fig. B.19 Result of the operation Complete of the lateral panel. Complete the contour of a selected oocyte that was inaccurately recognized (magenta colour). . . .	104
Fig. B.20 Visualization of information about the oocytes overlapped to the image in the bottom table (button Show Table in the lateral panel). The cyan contour is the selected cell.	106
Fig. B.21 Window open when the submenu XML File of the menu Analysis is chosen.	107
Fig. B.22 File chooser dialog to choose the XML files included in the joint analysis. .	107
Fig. B.23 Window open when the submenu XML File of the menu Analysis is chosen and after selecting the XML files used in the analysis.	107
Fig. B.24 An example of a CSV file of the analysis of various XML files imported in LibreOffice Calc.	108
Fig. B.25 Pop-up message informing that the classifier is not trained for the fish specie considered.	108
Fig. B.26 An example of oocytes classified manually by expert (upper panel) and after applying the classifier (lower panel).	109
Fig. B.27 Dialog window to train the classifier.	111
Fig. B.28 Figure B.27 after the configuration of the parameters to train the classifier. .	111
Fig. B.29 Pop-up dialog to inform about help.	112

List of Tables

Tabla 2.1	Datasets of images used: code of set (first column), fish species (second column), number of images (third column), acquisition system (column 4): 1) microscope Leica DMRE connected to a Leica DFC 320 camera; 2) microscope Nikon Eclipse 80i with a Nikon DXM 1200F camera; 3) Leica DM 4000B with Leica DFC 420 camera and 4) Leica M165C with Leica DMC 4500 camera. Magnification and image size in the last columns.	21
Tabla 3.1	Precision, recall and F_1 in % for the classification of pixels for datasets hake, pouting, pilchard, megrim and grenadier using $D_{\min} = 100$ pixels. The columns No. σ 's, σ value and Thresholds show the configuration providing the highest performance.	37
Tabla 3.2	List of the 20 best MSCF configurations (σ values and thresholds) using minimum diameter $D_{\min} = 100$ pixels according to the Friedman rank of F_1	38
Tabla 3.3	Precision, recall and F_1 in % for the classification of pixels for datasets hake, pouting, pilchard, megrim and grenadier varying the value of $D_{\min}$ pixels. The columns $D_{\min} = \{50, 100, 150, 200\}$, No. σ 's and σ value show the configuration providing the highest performance.	39
Tabla 3.4	List of the 30 best MSCF configurations (σ values and thresholds) and minimum diameter $D_{\min} = \{50, 100, 150, 200\}$ according to the Friedman rank of F_1	40
Tabla 3.5	Percentage of correct recognition of cells for hake, pouting, pilchard, megrim and grenadier datasets using $D_{\min} = 100$ pixels for different numbers of σ 's.	41

Tabla 3.6	Precision, recall and F_1 in % for the classification of pixels for datasets hake (upper rows), pouting (middle rows) and pilchard (lower rows) using $D_{\min} = 100$ pixels. The last column <i>Correct</i> is the percentage of cells correctly recognized in the image. The algorithm configuration providing the highest performance is shown in the “Configuration” column.	44
Tabla 3.7	Continuation of table 3.6 for megrim (upper rows) and grenadier (lower rows) datasets.	45
Tabla 3.8	Friedman ranking of the diferent segmentation algorithms using the F_1 quality measure: using $D_{\min} = 100$ (left columns) and for the $D_{\min} = (50, 100, 150, 200)$ (right columns).	45
Tabla 3.9	Precision, recall and F_1 in % for the classification of pixels for datasets hake (upper rows), pouting (middle rows) and pilchard (lower rows) for the best minimum diameter (column $D_{\min}$). The last column <i>Correct</i> is the percentage of cells correctly recognized in the image. The algorithm configuration providing the highest performance is shown in the “Configuration” column.	49
Tabla 3.10	Continuation of table 3.9 for megrim (upper rows) and grenadier (lower rows) datasets.	50
Tabla 3.11	Friedman ranking of the diferent segmentation algorithms using the correct cell recognition as a quality measure: using $D_{\min} = 100$ (left columns) and for the $D_{\min} = (50, 100, 150, 200)$ (right columns).	50
Tabla 3.12	Average computational time spent to process an histological image of fish ovary for each segmentation algorithm. The last column shows the standard desviation	58

Tabla 4.1 Performance of the automatic algorithm recognizing the cells in the image for all fish species studied. N and Nc are the number of images and the average number of cells per image, respectively. Se, Sp and AP are the sensitivity, specificity and average precision, respectively. MC is the ratio (in %) of modified cells by using the supervision tools in the lateral panel. Se, Sp and AP are calculated considering only the cells perfectly recognized and not modified by the expert (first row), and considering that the cells modified by the tools for supervision are correctly detected but their outlines were modified by the expert (second row). Average values in the last two rows are presented with the standard deviation (SD) to provide an insight into the variation among species. 72

Tabla 4.2 Average accuracy of the automatic classification for all fish species studied. The column “Stage” shows the accuracy for classifying the development stages: *cortical alveoli*, *vitellogenic* and *hydrated*. The column “With or without nucleus” shows the accuracy of classifying whether the cell has a visible nucleus. Average values and standard deviations (SD) are presented to provide insight into the variation among species. 72



Fish fecundity is one of the most relevant parameters for estimating reproductive potential of fish stocks used for assessing stock status to guarantee a sustainable fisheries management. Fecundity is the number of matured eggs that each female fish can spawn each year. The stereological method is the most accurate technique to estimate fecundity using histological images of fish ovaries, in which matured oocytes must be measured and counted. This thesis propose a multi-scale Canny filter (MSCF) algorithm to recognize the outlines of cells. It also develop the graphical software STERapp, which includes the MSCF algorithm and other machine learning technique to help the quantitative analysis of images in the fishing labs. STERapp saves between 40% to 70% of time in the fecundity estimation.