Tecnológico de Costa Rica

Escuela de Ingeniería en Computación

SegNema: Nematode segmentation strategy in digital microscopy images using deep learning and shape models

A thesis submitted in partial fulfillment of the requirements for the degree of Magíster Scientiae in Computer Science

José Jiménez-Chavarría

Cartago, February 10th, 2019

Declaro que el presente documento de tesis ha sido realizado enteramente por mi persona, utilizando y aplicando literatura referente al tema e introduciendo conocimientos y resultados experimentales propios.

En los casos en que he utilizado bibliografía he procedido a indicar las fuentes mediante las respectivas citas bibliográficas. En consecuencia, asumo la responsabilidad total por el trabajo de tesis realizado y por el contenido del presente documento.

José Jiménez-Chavarría

Cartago, March 5, 2019

Céd: 1-1439-0782

www.tec.ac.cr

TEC | Tecnológico de Costa Rica
Programa de Maestría en Computación
Teléfono: 2550-2402

APROBACIÓN DE TESIS

"SegNema: Nematode segmentation strategy in digital microscopy images using deep learning and shape models"

TRIBUNAL EXAMINADOR

Dr. Pablo Alvarado Moya
Profesor Asesor

Máster Saúl Calderón Ramírez
Profesor Lector

Dr. Juan Luis Crespo Marino
Profesor Externo

Dr. Roberto Cortés Morales
Coordinador a.i. del Programa
de Maestría en Computación

Febrero, 2019

www.tec.ac.cr

TEC | Tecnológico de Costa Rica
Programa de Maestría en Computación
Teléfono: 2550-2402

ACTA DE APROBACION DE TESIS

Con fundamento en lo que establecen los Artículos **22-24-25** del "Manual de Normas y Procedimientos para optar por el título de "MAGÍSTER SCIENTIAE EN COMPUTACION", el Tribunal Examinador de Tesis (TET), nombrado con el propósito de evaluar la tesis de grado.

SegNema: Nematode segmentation strategy in digital microscopy images using deep learning and shape models

Habiendo analizado el resultado general del trabajo presentado por el estudiante:

Primer Apellido	Segundo Apellido	Nombre	No. de carné
JIMENEZ	CHAVARRIA	JOSE MANUEL	200821732

Emite el siguiente dictamen:

☒ APROBADO

El TET, considerando que el trabajo realizado por el estudiante es SOBRESALIENTE, le otorga la siguiente MENCION HONORIFICA:

○ CUM LAUDE ☒ MAGNA CUM LAUDE ○ SUMMA CUM LAUDE

○ REPROBADO

○ SE RECOMIENDA ○ NO SE RECOMIENDA

Brindarle una nueva oportunidad para la DEFENSA PUBLICA de su Tesis

NUEVA FECHA: _____

Dando fe de lo aquí expuesto firmamos (*IDEM:* HOJAS DE APROBACION DE TESIS)

Dr. Pablo Alvarado Moya
Profesor Asesor

Máster Saúl Calderón Ramírez
Profesor Lector

Dr. Juan Luis Crespo Marino
Profesor Externo

Dr. Roberto Cortés Morales
Coordinador a.i. del Programa de Maestría en Computación

SINAES POST-GRADO ACREDITADO COSTA RICA

27 de Febrero, 2019

TEC | Tecnológico de Costa Rica
Sello Maestría en Computación

FT-01-MC

Resumen

Los nematodos son los animales pluricelulares más numerosos en la Tierra y su estudio tiene un impacto en el desarrollo de actividades agrícolas. En este documento se introduce *SegNema*, una estrategia para la segmentación de nematodos en imágenes de microscopia donde se utiliza aprendizaje profundo para clasificación de píxeles como nematodo o fondo, y modelos de forma para asociar hitos que describen la posición del nematodo en la imagen.

Para entrenar el modelo de segmentación se usan 2939 imágenes sin comprimir etiquetadas manualmente de tamaño 1024×768 píxeles obtenidas de 13 secuencias de imágenes de microscopia. Por otro lado, los hitos que describen la posición de los nematodos en estas imágenes de entrenamiento son utilizados para ajustar un modelo capaz de representar formas correspondientes a nematodo. La disparidad entre formas de las regiones clasificadas como nematodo en la etapa de segmentación y su posible representación truncada con el modelo de forma es usado para descartar posibles clasificaciones erróneas. Para la validación de este modelo se usan 321 imágenes de las secuencias de microscopia que no son utilizadas en la etapa de entrenamiento.

En cada imagen usada para entrenamiento y validación existe la información de la posición de hitos donde se delimita un único nematodo aunque otros nematodos pueden estar presentes.

Palabras clave: Aprendizaje profundo, Aprendizaje automático, Modelos de forma, Diccionarios activos de forma, Segmentación de nematodos.

Abstract

Nematodes are the most numerous multicellular animals on Earth and their study has a direct impact in the improvement and development of agricultural activities. This document introduces *SegNema*, a strategy for the segmentation of nematodes in microscopy images where deep learning is used for classification of pixels as nematode or background, and a shape model is used to associate landmarks that describe the position of the nematode in the image.

To train the segmentation model, a set of 2939 manually labeled uncompressed images of size 1024×768 pixels obtained from 13 different sequences of microscopy images is used. The landmarks that describe the position of the nematodes in these training images are used to adjust a model capable of representing shapes corresponding to a nematode. The disparity between the shapes of the regions classified as nematode in the segmentation stage and their possible truncated representation with the shape model is used to rule out possible erroneous classifications. The validation of this model was performed on 321 images of the microscopy sequences that were not used in the training stage.

In each image used for training and validation, there is information on the position of landmarks where a single nematode is delimited although more nematodes may be present.

321 / 2939 =

Keywords: Deep Learning, Machine Learning, Shape Models, Active Dictionary Models, Nematode Segmentation.

A mi padre

Agradecimientos

La culminación de esta investigación fue posible gracias al apoyo de mi familia. Mención especial a mi madre, Vilma Chavarría, mi hermana Gloria Jiménez y mi novia Samantha Urbina.

También quisiera agradecer a la empresa Ridgerun por su apoyo para continuar mis estudios de posgrado y formar parte integral en mi desarrollo como profesional.

Este trabajo no hubiera sido posible sin la excepcional asesoría brindada por el Dr. Pablo Alvarado Moya durante el proceso de investigación e implementación. Finalmente quisiera agradecer a todos los miembros del Programa de Maestría en Ciencias de la Computación del Tecnológico de Costa Rica por esta gran contribución a mi formación académica.

To those fighting their own giants, You're stronger than you think - I Kill Giants

José Jiménez-Chavarría

Cartago, March 5, 2019

Contents

List of Figures

List of Tables

List of symbols and abbreviations

Abbreviations

ADM	Active dictionary models
ASM	Active shape models
BMP	Barycentric Matching pursuit
CNN	Convolutional neural network
DL	Dictionary learning
FCN	Fully convolutional network
GP	Geodesic projection
MLP	Multilayer Perceptron
MP	Matching pursuit

General notation

$\mathbf{A}$	Matrix. $\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{bmatrix}$
$\mathbf{x}$	Vector. $\mathbf{x} = [x_1\ x_2\ \ldots\ x_n]^T = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$
$\mathbf{\Phi}$	Dictionary. $\mathbf{\Phi} = [\boldsymbol{\phi}_1\ \boldsymbol{\phi}_2\ \ldots\ \boldsymbol{\phi}_n]$
$\boldsymbol{\phi}_n$	Dictionary n-th atom. $\boldsymbol{\phi}_n = [\phi_1\ \phi_2\ \ldots\ \phi_n]^T = \begin{bmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_n \end{bmatrix}$
$\|\|\mathbf{x}\|\|_0$	l_0-"norm" of $\mathbf{x}$. Number of non-zero components of $\mathbf{x}$.
$\|\|\mathbf{x}\|\|_1$	l_1-"norm" of $\mathbf{x}$. $\|\|\mathbf{x}\|\|_1 = \sum_i \|x_i\|$
$\|\|\mathbf{x}\|\|_2$	l_2-"norm" of $\mathbf{x}$ or Euclidean norm. $\|\|\mathbf{x}\|\|_2 = \sqrt{\sum_i x_i^2}$
$\overline{\mathbf{S}}$	Simplex of k faces.

$\overline{\mathbf{S}} = [\mathbf{x}_1\ \mathbf{x}_2\ \ldots\ \mathbf{x}_k]$

$\mathbf{y}^T = \mathbf{x}^T\overline{\mathbf{S}}$ Product between vector $\mathbf{x}$ of k components and simplex $\overline{\mathbf{S}}$ of k faces.

$$\mathbf{x}^T\overline{\mathbf{S}} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_k \end{bmatrix}^T \overline{\mathbf{S}}$$

$\mathbb{R}$ The set of real numbers

Chapter 1

Introduction

Nowadays, the economy in the developing countries of Central America still depends on agriculture. In 2011 for example, coffee was an important export product in Central America; with a value of $3.70 trillion it was higher than bananas ($1.64 trillion), sugar ($1.03 trillion) and palm oil ($0.68 trillion) combined [49]. In Costa Rica, although its economy presents a growing diversification in services and goods, a part of the population still depends on agricultural activities where it is estimated that there are approximately 140,000 agricultural producers [49]. In this way; it is of interest for this sector to develop and apply technologies in order to face the demands and standards of today's local and international markets in areas such as pest control [30].

Nematodes are the most numerous multicellular animals on Earth [6]. Phytoparasitic nematodes can cause from insignificant damage to the total destruction of plantations, where the severity of the damage depends on factors ranging from the natural resistance of the plant species to external factors such as climate, soil types and reaction time to treat the plant biological agent. All fruit and vegetable crops are susceptible to nematodes [34]. Most of the nematodes are microscopic, and advances in the understanding of these organisms had to wait for the development of microtomes, and electron microscopes [6]. Figure 1.1 shows a nematode of the species *C. elegans*, used as a model for genetic studies.

Figure 1.1: Caenorhabditis elegans. **Source**: A.F. BIRD and J. BIRD, (1991) *The Structure of Nematodes (Second Edition)*, San Diego: Academic Press [6].

The nematodes are consistent in their anatomy; in general, they are symmetrical bilat-

eral pseudocelomata, not segmented, vermiform, with four major longitudinal epidermal cords, a triradiated pharynx, a circumternal nerve ring and do not possess circulatory or respiratory organs [6]. Usually, they have one or two tubular gonads, which open through a vulva in the female and in the rectum in the male [6]. In dry conditions, some nematodes can survive desiccation in a suspended state for long periods, and return to an awaken state when soil water conditions are restored [34]. In the dried state, nematodes are more resistant to high soil temperature and nematicides [34].

The search for biological control mechanisms of phytoparasitic nematodes is based on the analysis of hundreds of thousands of samples obtained from the soil. Typically performing the analysis of a sample image can take from 15 minutes to several hours. This time is directly proportional to factors such as the composition of the sample, the experience of the researcher, complexity of detection, etc. The effective use of the researcher's time, automating tedious tasks through the analysis of images presents a clear opportunity for improvement in the area of nematology [24].

The response of nematodes to a variety of stimuli in a given sample allows the analysis of movement capacity, mortality, fecundity and size [58]. For instance, death in C. Elegans is manually determined by an animal's inability to respond to a mechanical stimulus [47] and directional response to electrical fields can vary among nematode species [58]. Given these factors, is of interest to determine not only the number of nematodes in a given image, but also information for their position and edges.

The development of computational tools that provide support to biologists and nematologists is of interest since they potentially accelerate processes of identification, counting and behavior analysis; aimed at reducing the impact of these organisms on crops. The segmentation of nematodes in digital images is essential for a detection and identification system of these organisms, and their success depends on the accuracy of the algorithms used. However, the visual appearance of these biological organisms still represents an unresolved challenge in computer vision, and therefore it is an open research topic. In Costa Rica, particular applications of this technology include the analysis of pests associated with phytoparasitic nematodes, and the analysis of nematodes as indicators of environmental health.

Contributions to the development of such tools have been proposed in [24] and [45] and a recent proposal in [68] uses 1D CNN Networks. A fully developed strategy that integrates a cohesive image processing pipeline has not yet been presented. This document introduces *SegNema*; a proposal for the segmentation of vermiform structures in digital images. *SegNema* relies on a database of 3260 manually segmented images, each image includes only one manually segmented nematode.

Figure 1.2 shows the stages of segmentation approach in *SegNema*. The proposal uses a deep network to produce pixel-wise binary classification which discriminates nematode an background in the image. Based on that result, the method locates the corresponding nematode edges in order to verify if the detected shape corresponds to a nematode, based on possible deformations on vermiform structures.

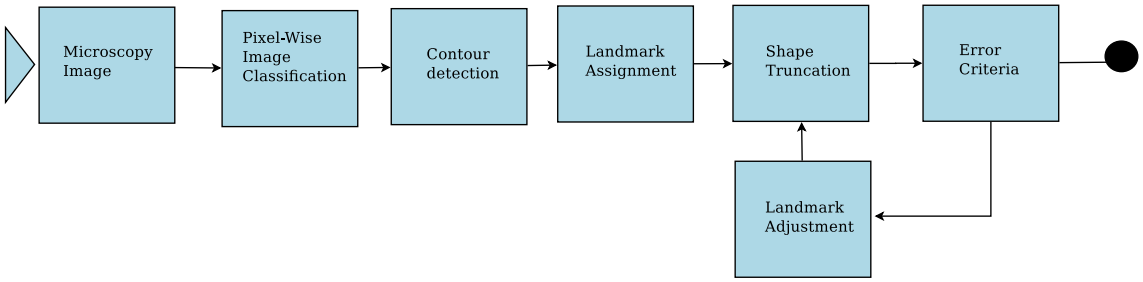


Figure 1.2: Image processing pipeline proposed in *SegNema*

A binary classification map for a given image is generated by means of a semantic segmentation network as discussed in Section 2.3. The map is further refined by applying closing and opening morphological operators. Contour detection for binary blobs on the binary map is performed using connected components (Section 2.6.1). Finally, landmark assignment and shape truncation is performed by means of ADM (Section 2.5) where shapes that do not meet the error criteria are discarded.

The main contributions of this work are:

- Implementation of a pixel-wise classifier for nematodes in microscopy images using deep neural networks.
- Shape analysis of the binary classifier result using manifold learning to discard shapes that do not correspond to nematodes.
- Shape adjustment using ADM as an iterative method for landmark positioning on nematode/background edges.

1.1 Objectives and document structure

The objective of this research is to design a computational tool capable of describing the position of nematodes in an image using shape models. This tool acts as a first stage in the development of a usable system in the analysis of microscopy images for the study of possible pests. In order to achieve this, a methodology it is shown to adapt a shape model to the image content under consideration of the information extracted via deep neural networks and active dictionary models. Additionally, it aims to optimize the edge detection in microscopy images between nematodes and background to identify a shape adjustment strategy for the placement of landmarks on a microscopy image, using an iterative process that fits a nematode shape model.

The document is structured as follows: in chapter 2 the theoretical foundations and related works are discussed. Next, in chapter 3 the proposed solution is introduced. In chapter 4 the accuracy of the developed system and its components are tested against manual segmentation. The results obtained are analyzed against the theory presented

in previous chapters. Finally, in chapter 5 the project conclusions are summarized along with suggestions for future work.

Chapter 2

Materials and Methods

2.1 Related Works

Shape analysis has been explored in the past for nematode segmentation using active shape models [45]. This analysis consists on a Procrustes shape alignment [64], followed by principal components analysis (PCA) and an image adjustment process that uses the edges in an image to position landmarks. In general terms PCA is applied in order to reduce the dimensionality of data. This result can be used as a way to parameterize shapes with the principal components. The deformations can be modeled with respect to a displacement of the landmarks that fit the edges detected on the image [18]. However, the results showed the limitation imposed by the use of PCA as a basis to create the possible deformation space of the nematodes. PCA assumes a linear variability model of the shapes and the vermiform structures change following a non-linear distribution [45]. The work on active dictionary models [26] was able to overcome that limitation and it is used on the proposed strategy.

In terms of image segmentation (i.e. a semantic partitioning of the image in regions), the use of artificial neural networks has been explored in [24][25]. In [24] the segmentation of nematodes in digital images is proposed in two levels of abstraction, so that it serves as a first stage of a complete automatic system for the detection and identification of nematodes. The main objective of the first level of segmentation corresponds to find samples of textures that are used to the train the second layer. To achieve this, a series of segmentation strategies were evaluated, in which CWAGM obtained the best performance. In the second level, corresponding to the object-based segmentation, the objective is to classify the regions of an over-segmentation according to their texture, and among the classification algorithms MLP presented the best performance. Considering the type of images and other difficulties associated with microscopic images, the system obtained a precision of up to 84.27 % of correct answers in the best of cases with the reference images used [24]. This strategy had some limitations:

- The quality of the images, particularly the blurring level, affects the classification.

In the first level it softens irregularities in the image and ends up giving less regions suitable for the classification step. On the second level it destroys the texture information by smoothing the image.

- The classifiers have problems in distinguishing which regions belong to the nematode and which are background regions because many sections of the nematode are translucent. This produces holes in the segmentation of the nematode. In order to separate these types of regions, a high level of adjustment by the classifier seems necessary, which disables the use of this architecture for samples of, for example, different microscopes.

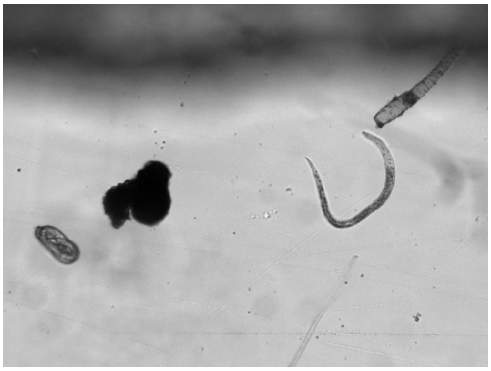
- The parameters chosen depend on the type of image content for the first segmentation level. Using the same parameters, over or under segmentation results can be expected. Small regions decrease the accuracy of the classifier as it will cause the regions to produce similar descriptors within the nematodes and the background. Large regions generate few training examples and regions that belong to nematodes and background can be merged in to a single region.

The work in [25] also relies on an initial texture based detection, after which a Markov random field model is used to refine the segmentation to infer which pixels belong to the nematode. The approach was tested on *C. elegans* under various dynamic environments, but it did not integrate a shape model to produce a static set of landmarks per nematode on the image.

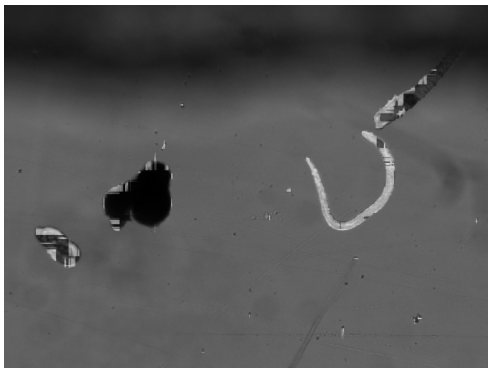
In [68] deep learning is used for the classification and detection of nematode edges as a two stage task; first a classification model is used to determine if a 1D vector contains nematode information and then a regressor model is used to determine which pixels correspond to a nematode or to the background. The obtained models are able to classify 1D vectors in nematode or non-nematode and estimate in which direction there is a nematode edge. During the CNN training, dropout layers were used to avoid overfitting. In addition, the use of ReLU activation features in all layers improves the convergence speed of the training without sacrificing precision and recall. Figure 2.1 shows the results obtained by the two tasks evaluating horizontal vectors of 37 pixels (in 1D). The network was trained with nematode data and background information close to the nematode. This caused that objects without a nematode shape could be wrongly classified and no method was proposed to discard those shapes. In cases where there is a nematode in the vector, it was well distinguished between nematode and background.

Deep learning frameworks have also been used to discern and count microscopic nematode eggs [1]. Similarly to what is proposed in this work, the architecture is trained with labeled data from microscopy images and used to build a machine learning model. The result of the segmentation performed was used to quantify via image analysis the number of eggs in the image.

Recently, other methods have been proposed with good results. A method for detecting rice root knot nematodes on microscopy images is presented in [67]. It is based on



(a) Original microscopy image



(b) Output of performing a classification and a regression task using deep learning

Figure 2.1: Nematode segmentation result, as obtained in the combination of a classification task and a regression task using 1D CNN. **Source**: M. Taylor, *Adjustment strategy of explicit shapes to images with vermiform structures*, ITCR (2017) [59].

mathematical morphology, where a binary image is obtained and the microscope lines are removed. The detected nematodes are counted after applying a threshold for the non-nematode particles. The method described in [44] relies on movement detection to discard objects in microscopy images that do not correspond to living organisms. A more recent area of deep learning, called *Geometric Deep Learning* [48][21][8][46], aims to transfer the high performance of CNN to structures that can be represented with no linear transformations by defining convolution operations for deep neural networks that can handle irregular input data. A way to limit the possible values learned by the weights of the convolutional networks so that filters are not able to learn representations outside a particular manifold is presented in [48]. On the other hand in [21] structured irregular data was received and mapped to a directed graph in order to being able to only represent data described by B-Splines.

At the time of this work, no proposal has been found that integrates a shape model stage with a segmentation stage using deep learning applied in microscopy images with one labeled nematode.

2.2 Metrics

This section describes the metrics used throughout this document for the validation of the different stages of *SegNema*. Categorical cross entropy loss and mean absolute error are used to measure the accuracy of the semantic segmentation model. The Jaccard index is used as a second stage accuracy measurement to compare binary masks generated by the segmentation model and ground truth reference values. The root mean squared error is used as a criterion to determine the convergence of the landmark adjustment strategy. Finally, the euclidean distance is used to measure distance between an arbitrary coordinate in a space of given dimensionality and a manifold in that space (more on section 2.5).

2.2.1 Cross entropy loss

In information theory [57], entropy refers to the average rate at which information is produced by a random source of data. Given a set of labels $\mathcal{I}$, each with a given probability of being produced p_i with $i \in \mathcal{I}$, provided by a classification distribution $\mathbf{p}$, the entropy H is given by:

$$H(\mathbf{p}) := -\sum_{i \in \mathcal{I}} p_i \log(p_i) \tag{2.1}$$

The more uncertain the setup is, the higher the entropy, since the average data will increase. The cross entropy for two probability distributions is used to quantify the rate of correct assignment of values from estimated predictions to a true set of predictions. The cross entropy between two probability distributions $\hat{\mathbf{y}}$ and $\mathbf{y}$ over the same underlying set of event measurements, being $\hat{\mathbf{y}}$ the predicted probabilities and $\mathbf{y}$ the ground-truth distribution, is given by:

$$H(\hat{\mathbf{y}}, \mathbf{y}) := -\sum_{i \in \mathcal{I}} \hat{y_i} \log(y_i) \tag{2.2}$$

In a model for probabilistic classification using supervised learning, the aim is to map the model inputs to probabilistic predictions, and the model is trained by incrementally adjusting the model parameters so that the predictions get closer and closer to ground-truth probabilities. This means that for a given classification model, the better the strategy is, the lower the cross entropy is. Hence, minimizing cross entropy will move the model configuration closer to ground-truth distribution. The cross-entropy can be used as a loss function when performing gradient descent.

2.2.2 Jaccard index

The Jaccard index (also known as intersection over union (IoU) and the Jaccard similarity coefficient) [54] measures similarity between finite sample sets. Is defined as the size of the intersection divided by the size of the union of the sample sets. Given two sample sets $\mathbf{A}$ and $\mathbf{B}$, the Jaccard index is calculated as:

$$J(\mathbf{A},\mathbf{B}) = \frac{|\mathbf{A} \cap \mathbf{B}|}{|\mathbf{A} \cup \mathbf{B}|} \tag{2.3}$$

Where $0 \leq J(\mathbf{A},\mathbf{B}) \leq 1$. In a binary pixelwise segmentation task, the Jaccard index is used as a measurement for the similarity between a ground truth classification mask and the predicted mask .

2.2.3 Mean absolute error

The mean absolute error (MAE) [29], is used to perform a measure of the average magnitude of the absolute errors in a set of predictions. In machine learning, it is the average over the test sample of the absolute differences between predictions and reference values. Given a set of n predictions $\hat{\mathbf{y}}$ for the set of ground truth values $\mathbf{y}$, the MAE is given by:

$$\text{MAE} = \frac{\sum_{i=1}^{n} |y_i - \hat{y}_i|}{n} \tag{2.4}$$

This value can be used as the prediction error for the difference between the reference value and the predicted value for a given set.

2.2.4 Root mean squared error

Root mean squared error (RMSE) [10] is the square root of the average of squared differences between predictions and reference values. Given a set of n predictions $\hat{\mathbf{y}}$ for set of ground truth values $\mathbf{y}$, the RMSE is given by:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{2.5}$$

The RMSE is a measure that allows to compare how the residuals spread around between the ground truth.

2.2.5 Euclidean distance

The Euclidean distance between points $\mathbf{p}$ and $\mathbf{q}$ of n dimensions, is the length of the line segment connecting them ($\overline{\mathbf{pq}}$):

$$d(\mathbf{p},\mathbf{q}) = \sqrt{\sum_{i=1}^{n}(p_i - q_i)^2} \tag{2.6}$$

2.3 Semantic image segmentation

Semantic segmentation is the task of clustering image regions together which belong to the same object class [70]. Let $k \in \mathbb{N}$ be the number of classes and m the total number of pixels, $n_i \in \mathbb{N}$ with $i \in 1,...,\ k$ the number of pixels which belong to class i. The idea of semantic segmentation it to obtain n_i pixels correctly assigned to each of the k classes. In order to do so, semantic segmentation algorithms store information about the classes

they were trained to segment, while non-semantic segmentation algorithms try to detect consistent regions or region boundaries. This concept is associated as a step towards the high-level task of complete scene understanding [22].

The set of labels with which the algorithm is trained to classify is a central design decision. Most algorithms work with a fixed set of classes, which includes binary classes like foreground and background, or the object of interest and everything else. From a general perspective in the training of a deep learning based segmentation pipeline, data augmentation techniques such as image rotation or translation, among others, can be applied to raw data (see Section 2.4.1). There are two general strategies for training: using patches of the image, called windows, which are extracted from the labeled data, and downscaling input data to reduce dimensionality. The resulting semantic segmentation of an image typically can be refined by applying morphologic operations or by more complex approaches such as Markov Random Fields (MRFs) and Conditional Random Fields (CRF) [22].

2.3.1 Morphology

In order to soften edges in the binary predictions performed, *dilation* and *erosion* operations are perform on binary masks. As per morphological operation concepts, *dilation* is an operation that grows or thickens objects in a binary image. The dilation of an image $\mathbf{f}$ by a structuring element $\mathbf{s}$ (typically referred to as kernel) is denoted as $\mathbf{f} \oplus \mathbf{s}$. The result of this operation is a new binary image $\mathbf{g} = \mathbf{f} \oplus \mathbf{s}$ with ones in all locations (x,y) of a kernel origin on which that kernel hits the input image $\mathbf{f}$. This means that $g(x,y) = 1$ if $\mathbf{s}$ hits $\mathbf{f}$ and 0 otherwise, repeating for all pixel coordinates (x,y) [61]. *Erosion* on the other hand, shrinks or thins objects in a binary image. The extend of shrinking is controlled by the structuring element. The erosion of a binary image $\mathbf{f}$ by a structuring element $\mathbf{s}$, denoted by $\mathbf{f} \ominus \mathbf{s}$ produces a new binary image $\mathbf{g} = \mathbf{f} \ominus \mathbf{s}$ with 1 in all locations (x,y) of the kernel element origin on which that kernel fits the input image $\mathbf{f}$. This means that $g(x,y) = 1$ is $\mathbf{s}$ fits $\mathbf{f}$ and 0 otherwise, repeating for all pixel coordinates (x,y) [61].

Typical morphologic operations in refining the result of a semantic segmentation include *opening* and *closing*[52]. The opening operation is an erosion followed by an dilation; this removes small segments of the image. The closing operation is an dilation followed by an erosion. This removes gaps in otherwise filled regions [61].

2.3.2 Markov Random Fields

On the other hand, the overall idea of Markov Random Fields (MRF) is to assign a random variable to each feature and another random variable to each pixel which gets labeled. For instance, as proposed in [70], in computer vision segmentation problems where MRFs are applied, a random variable $\mathbf{y} \in \{1, ..., k\}$ represents the class of a single pixel, while a random variable $\mathbf{x}$ represents pixel values and edges represent pixel neighborhoods. The

join probability of $\mathbf{x}$, $\mathbf{y}$ is expressed as

$$P(\mathbf{x}, \mathbf{y}) = \frac{1}{Z} e^{-E(\mathbf{x},\mathbf{y})} \tag{2.7}$$

where $Z = \sum_{\mathbf{x},\mathbf{y}} e^{-E(\mathbf{x},\mathbf{y})}$ corresponds to the partition function which is a normalization term and E is the energy function. Similarly, CRFs are MRFs where instead of learning the distribution $P(\mathbf{x},\mathbf{y})$, the problem is reformulated to learn the conditional distribution $P(\mathbf{y} \mid \mathbf{x})$. The main advantage of this reformulation is that CRFs need less parameters since the distribution of $\mathbf{x}$ does not have to be estimated. On the other hand CRFs make no distribution assumption about $\mathbf{x}$ as opposed to what is proposed by MRFs [70][39].

2.4 Deep learning

Deep learning allows the generation of computational models that are trained to learn data representations with multiple levels of abstraction. Using these layers of representation, models are able to learn aspects for the interpretation of data making use of the underlying structure in the data through the parameters of each intermediate layer. Each layer is used to calculate a new representation from the representation of the previous layer [41]. While the classical methods require that the data must be preprocessed to obtain an intermediate representation, deep learning approaches learn during training the best intermediate representations for the data and generates a model. This technique has two main limitations that prevented popularization of its use in the past [42]:

1. It requires data for training that considers most of the variations of the input space. The ability of deep networks to learn intermediate representations comes at the price of having to increase the number of model parameters. A greater number of parameters implies that more data is required to train.

2. The training of a deep network requires computational resources that grow with respect to the number of training parameters. Each training data must be passed through a series of convolutions and linear calculations, to then use some optimization technique that modifies the parameters of the model.

These problems have been solved in recent years [22]. The Internet has allowed the exchange of huge amounts of images tagged in different areas; nevertheless images of microscopy of nematodes are not included among the available databases. On the other hand, advances in processing capacity, due to the rise of high-performance computing, have considerably reduced the time required to train this type of networks.

With the good performance obtained by a particular implementation with the ImageNet database, since 2012 progress in the area has grown exponentially [3]. In that year convolutional neural networks were applied in the recognition of objects, considerably reducing

the error obtained by solutions involving more complex synthesis operations. Since then, the computer vision community quickly adopted deep learning and is now widely used. The CNN model is made up of a series of parameter layers adjusted during training that synthesizes information on the data that should cause the highest activation. The trained model computes an output from an input. Table 2.1 summarizes some the most common layer architectures used in deep learning approaches.

Table 2.1: Deep learning common layers

Architecture	Description	References
Convolutional	Convolutional layers filter the input data, with filters learned during the training stage in order to highlight characteristics that allow the upper layers to classify the data. It generally learns an intermediate representation of the data. For each filter on the layer, it generates a mask that is the result of the convolution with the filter. This layer is parametrized by the amount of filters used and the size of the filter mask.	[3] [17]
Recurrent	As the name suggests; it is used when a model needs context to provide the output based on the present and past inputs. RNNs can use an internal state to process sequences of inputs.	[32][75] [23][11]
Batch Normalization	It reduces the amount by which the hidden unit values spread and allows each layer of a network to learn by itself more independently from other layers. It works by normalizing the output of a previous activation layer by subtracting the batch mean and dividing by the batch standard deviation.	[33][31]
Pooling	It progressively reduces the spatial size of the representation to reduce the amount of parameters. It also prevents overfitting. The Pooling Layer operates independently on every depth slice of the input and resizes it spatially, using the MAX operation.	[65][40] [56]
Dropout	It is a way to avoid overfitting through a method different than regularization. The idea is that during each step forward some neurons are deactivated. The percentage of these neurons is predefined in each layer and their selection is random. This prevents neurons from adapting too much.	[63]
Activation	It is a function applied to produce the output of a neuron. Possible activations include ReLu, Softmax, Softplus, Softsign, Sigmoid.	[53][19] [2]

Deep Learned Segmentation

Successful deep learning techniques for semantic segmentation are inspired from the approach taken by the Fully Convolutional Network (FCN) [59]. The Fully Convolutional Network (FCN) is a normal CNN, such as AlexNet [3], VGG (16-layer net) [60], GoogLeNet [66], and ResNet [27] where the last fully connected layer is substituted by another convolutional layer with a large receptive field, as shown in Figure 2.2. The idea is to capture the global context of the scene as output spatial maps instead of just obtaining classification scores. Those maps are upsampled using fractionally strided convolutions [76] to produce dense per-pixel labeled outputs. The success of this approach showed how CNNs can be trained to learn how to make dense predictions for semantic segmentation with inputs of arbitrary sizes.

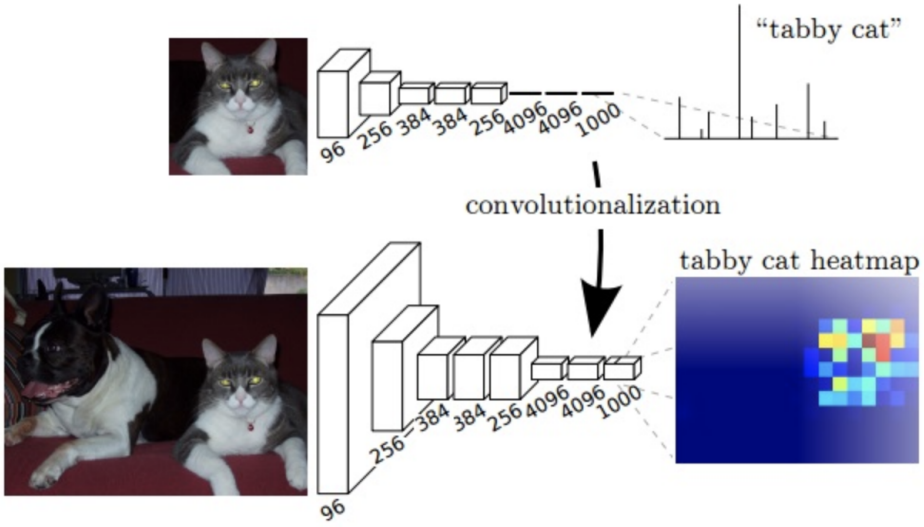


Figure 2.2: Fully Convolutional Network. **Source**: E. Shelhamer, J. Long, and T. Darrell, *Fully convolutional networks for semantic segmentation*, IEEE Trans on Pattern Analysis and Machine Intelligence. (2017) [59].

The FCN model has several limitations since its inherent spatial invariance does not use the global context information to refine the classification and there is no instance awareness, which means that the information obtained from the inference can not determine the amount of objects in the scene. Due to these limitations, a series of variations emerged that exploit the qualities of the FCN but at the same time take advantage of the characteristic information at the local level in each prediction, in order to make it suitable for segmentation. SegNet [4] is a clear example of this divergence and follows a different architecture to perform semantic pixel-wise labelling. It consists of an encoder network and a corresponding decoder followed by a pixel-wise classification layer used to map low resolution features obtained by the encoder stage to the input resolution, in order to obtain a pixel-wise classification result. The mapping produced by the network provides features which are useful for accurate boundary localization, since by obtaining a pixel by pixel classification in an inference result, it is possible to determine the relationship

where an edge can be located. A representation of this network, as proposed in [4], is showed in Figure 2.3.

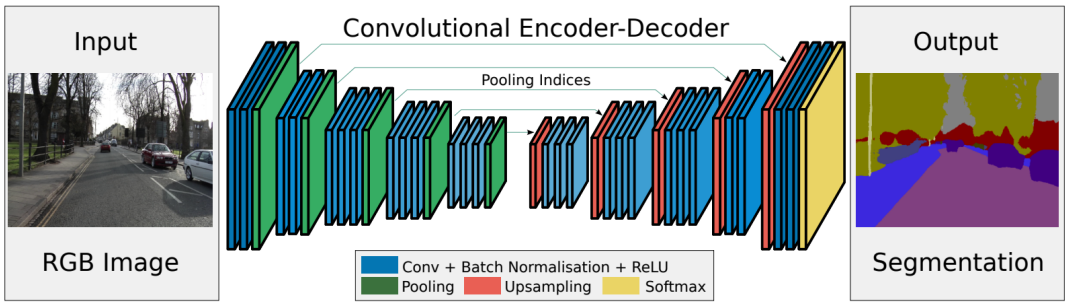


Figure 2.3: SegNet architecture. **Source**: V. Badrinarayanan, A. Kendall, and R. Cipolla, *Segnet: A deep convolutional encoder-decoder architecture for image segmentation*, CoRR (2015) [4].

In the SegNet architecture, encoder-decoder pairs are used to create feature maps for classifications of different resolutions. A collapsed view of this architecture is shown in Figure 2.4. Since the model decomposes the image at different resolutions it has the ability to delineate objects based on their shape, despite their small size. On the original proposal the pooling indices where shared between the subsampling and upsampling layers of the encoder and decoder, respectively, but the entire architecture can be trained end-to-end using stochastic gradient descent.

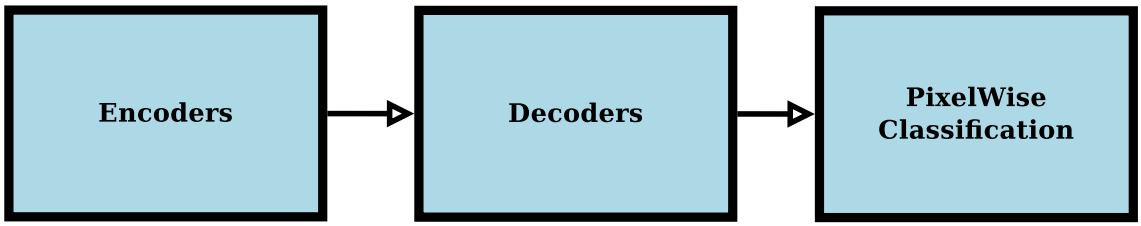


Figure 2.4: Simplified SegNet architecture

Each encoder on the model is constructed as the one depicted on Figure 2.5. Each one performs a convolution with a filter bank to produce a set of feature maps, which are then batch normalized. Batch normalization reduces the amount by what the hidden unit values spread (also known as covariance shift) in order to generate the feature maps created at each layer [31]. Then an element-wise rectified linear unit (ReLU) activation is added. Finally, max-pooling with a 2×2 window and stride 2 is applied. This causes that the resulting output after each encoder stage is a feature map subsampled by a factor of 2 from its input. The idea of using a max-pooling layer is to achieve translation invariance over small spatial shifts in the input image.

Combining translation invariance over small spatial shifts and subsampling leads to each pixel governing a larger input image context. This means that, at the end of the encoder stage, the network is able to classify the input pixels but in a reduced feature map size,

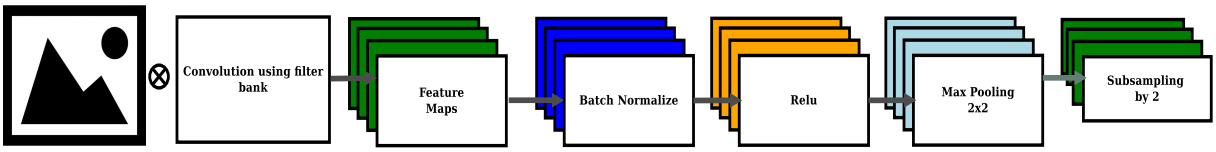


Figure 2.5: Segnet encoder architecture **Adapted from**: V. Badrinarayanan, A. Kendall, and R. Cipolla, *Segnet: A deep convolutional encoder-decoder architecture for image segmentation*, CoRR (2015) [4].

which can be translated as a lossy image representation with blurred boundaries, which is not ideal for segmentation purposes. The decoder section is added to overcome that limitation, since it is desired that the output image resolution is the same as in the input image. To achieve this SegNet applies upsampling in its decoder. For that, it stores the max-pooling indices from the encoding stage and provides them to the decoding upsampling layer. The sparse maps created by each upsampling layer are fed through a trainable filter bank to produce dense feature maps. The last decoder in the decoder network is connected to a *softmax* classifier which classifies each pixel to its corresponding class.

One of the advantages of the SegNet network is that it is smaller and easier to train than many other recent architectures to perform segmentation, while it is able to retain boundary information in the extracted image representation. Other approaches like Google DeepLab [14] which apply depth-wise separable convolution [13][12] and uses Atrous Spatial Pyramid as the pooling mechanism, needs to be trained using a multi-staged approach.

Recently, a SegNet based architecture was successfully applied in ear detection [20]. The detection problem was formulated as a two-class segmentation problem to distinguish between image-pixels belonging to either the ear or the non-ear class. The output of the network was refined using post-processing in order to not simply return a bounding box around the detected ear but also a detailed pixel-wise information about the location of the ears in the image. In this way the result of the pixelwise classification can be directly studied by shape models since it not only provides information on which pixels belong to a particular class, but also where the edge can be found.

The next step in pixelwise semantic segmentation is object detection where not only pixels of a particular class needs to be detected but also distinguish different instances of the same object. Pixelwise semantic segmentation is considered a first step in instance segmentation since that information is required while trying to get object instances, but further processing is required since neighboring pixels of a particular class might belong to different object instances and regions which are not connected might belong to the same object instance. For example, the intersection of two pencils will be classified all in a class but there are still two instances of the same object that without further shape analysis can be interpreted as being the same pencil [70][22]. The task of object detection in semantic segmentation is known as instance segmentation[22] and at the same time it is a challenging problem that is still under research with several applications in the

computer vision field. The main purpose is to separate objects that are represented with the same class label into different instances since the prediction obtained pixelwise does not contain information over the number of instances. Instance labeling provides extra information for reasoning about occlusion situations.

2.4.1 Data Augmentation

When training CNN based models data augmentation is a common technique that has been proven to benefit the performance in deep architectures in particular. A CNN deep learning network requires a large amount of data to compensate for the inherent high variance of the system. In other works, given that it has a high amount of parameters to train, the system has high capacity and to avoid overfitting the learning theory predicts higher requirements in the number of samples used as training data [22] [77] [51]. The reasoning behind this augmentation lies in applying transformations to a dataset to generate more samples and so decrease the chance of possible biases. If the capacity of the system fits the generalization requirements for the task, a large dataset allows the prevention of overfitting while training the model. The data augmentation techniques should be targeted at balancing the classes within that database, and synthetically produce new samples that are more representative than a small original dataset. Common data augmentation techniques include perform operations such as translations, rotations, cropping, noise addition (for instance, adding Gaussian noise) and optical distortions.

Examples can be found where data augmentation techniques allowed the model to reach better results in classification tasks. For instance, [79] showed that using a subset of the training data and applying a transformation has better effects later in the prediction that with the unaltered complete original set. On the other hand, in [35] 200×200 images were divided into smaller chunks in a segmentation task and applied data augmentation, which allowed to reach a 28% increase in the classification accuracy.

2.5 Active Dictionary Models

In order to generate landmarks that describe a shape in a manifold constituted by representations of nematodes, Active Dictionary Models (ADM) was used. ADM is based on a dictionary learning stage and a shape truncation step called geodesic projection both of which will be further explained in this chapter.

2.5.1 Geodesic Projection

A set of l two-dimensional landmarks that describe a particular shape can be mathematically interpreted as a point in a d-dimensional space with $d = 2l$ [43]. Since high-dimensional datasets can be difficult to represent, manifold learning techniques propose

that within a d-dimensional set $\mathbf{S} \in \mathbb{R}^d$ there is an embedded k dimensional manifold $\mathcal{M}$ which holds all shapes described in the d dimensional space, so that the condition $\mathcal{M} \in \mathbb{R}^k, k < d$ is fulfilled. The typical manifold learning problem is unsupervised and aims to learn the structure of the data from the data itself, without the use of predetermined classifications.

One of the techniques in manifold learning is Isomap [69] which stands for isometric mapping. Isomap is proposed as a non-linear dimensionality reduction method which tries to preserve the geodesic distances between points in spaces of lower dimension than the original set of points. In non-linear manifolds, the Euclidean metric for distance is a valid metric if the local neighborhood structure can be approximated as linear but it falls short for larger neighborhoods. This is depicted in the 2D projection onto a 1D space example shown in Figure 2.6; depending on the distribution of the points in the space, the Euclidean distances can be misleading, whereas if the distance between two points is calculated moving on the manifold, an alternative definition of the distance between two points in a lower dimensional space can be established.

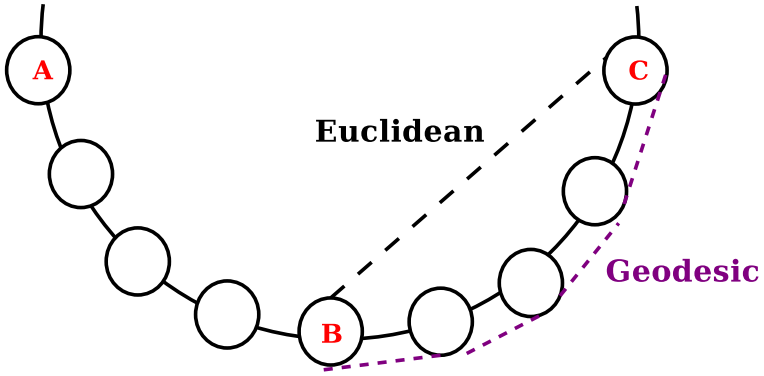


Figure 2.6: Example of the mapping for the geodesic distance that rely on the distance between the neighbors of points instead of the Euclidean distance between points

The first step in the Isomap method starts by creating a neighborhood network of the points in the higher dimensional space. After that, it uses the graph distance to approximate the geodesic distance between all pairs of points using the *k-nearest-neighbors* [69]. Then through the eigenvalue decomposition of the geodesic distance matrix, it finds the low dimensional embedding of the dataset [69].

The geodesic projection (GP), algorithm borrows the principle of projecting a shape from a higher dimension space into a subspace described by a discrete subset of points in a manifold [26] and interpolating using approximate geodesics. The GP procedure is described as algorithm 1 where $\mathbf{y}$ is the original point and $\mathbf{y}_{GP}$ the projected result.

Barycentric matching pursuit (BMP) as proposed in [26] is a modification of the classical matching pursuit algorithm [50] where the projection of the sample to be represented is confined into the interior of a simplex. In order to represent points in barycentric coordinates, let $\mathbf{t}$ be the finite place in a simplex of k vertices $\mathbf{x}_1, \cdots, \mathbf{x}_k$. We can represent

Algorithm 1 Geodesic Projection

Input: Signal to project $\mathbf{y}$, representative manifold $\mathbf{S}$

Output: Projected signal $\mathbf{y}_{GP}$

1: Let $\mathbf{S} \in \mathbb{R}^d$ be a representative set of samples from the manifold $\mathcal{M}$

2: Get the k-nearest-neighbors of $\mathbf{y}$ in $\mathbf{S}$

$$\mathbf{S}_k = \text{k-nearest-neighbors}(\mathbf{y}, \mathbf{S})$$

3: Find the closest neighbor of $\mathbf{y}$ in $\mathbf{S}_k$

$$\mathbf{y_{nn}} = \text{nearest-neighbor}(\mathbf{y}, \mathbf{S}_k)$$
$$\mathbf{y_0} = \mathbf{y} - \mathbf{y_{nn}}$$

4: Project $\mathbf{y_0}$ into a simplex spanned by $\overline{\mathbf{S}_k} = \{\mathbf{p} - \mathbf{y_{nn}} \mid \mathbf{p} \in \mathbf{S}_k\}$ using Barycentric matching pursuit. The vectors in the columns of $\overline{\mathbf{S}_k}$ conform a matrix that makes it suitable for BMP.

$$\mathbf{y}_{\mathcal{M}} = \text{BMP}(\mathbf{y_0}, \overline{\mathbf{S}_k})$$

5: Return to the original coordinate space

$$\mathbf{y}_{GP} = \mathbf{y}_{\mathcal{M}} + \mathbf{y_{nn}}$$

any point $\mathbf{p}$ with the barycentric coordinates $\mathbf{t}$ with

$$\mathbf{p} = \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_k \end{bmatrix} \mathbf{t} = t_1\,\mathbf{x}_1 + \cdots + t_k\,\mathbf{x}_k \tag{2.8}$$

with the restriction that all $t_i \geq 0$ and $\|\mathbf{t}\|_1 = 1$. The coefficients $(t_1, \ldots, t_k)$ are the barycentric coordinates of $\mathbf{p}$ with respect to $\mathbf{x}_1, \ldots, \mathbf{x}_k$. Algorithm 2 shows the confinement process of the point $\mathbf{y}$ into the simplex $\overline{\mathbf{S}}$ of k faces. The columns of $\overline{\mathbf{S}}$ span a subspace of $k-1$ dimensions.

2.5.2 Dictionary Learning

A shape model is a mathematical description of a shape that allows deformations according to a set of previously defined constraints. The shape models have diverse applications in industrial and scientific areas, and it is a basic element in tasks such as segmentation, elimination of noise, tracking and classification [16].

Classical algorithms such as PCA are based on linear methods and unimodal normal distributions that are not appropriate for modeling deformations present in natural signals, such as the vermiform structures associated with the nematodes. As opossed to these methods, dictionary learning is capable of representing deformations of non-linear signals, such as those present in natural signals [26]. Matching a shape and its possible deformations from a previously learned representation to image contents is a problem studied in the area of computer vision and the tendency in recent years has been to solve

Algorithm 2 Barycentric matching pursuit, adapted from [26]

Input: Signal to truncate $\mathbf{y_0}$, Simplex to project the image $\overline{\mathbf{S}}$ of k faces

Output: Signal projected inside the simplex after the j iteration $\mathbf{y}^j$

1: Set

$$\begin{aligned} j &= 1 \\ \mathbf{r}^j &= \mathbf{y_0} \\ \mathbf{y}^j &= \begin{bmatrix} 0 & 0 & \cdots & 0 \end{bmatrix} \overline{\mathbf{S}} \end{aligned}$$

where j is the iteration step, $\mathbf{r}^j$ is the residual of the projection of $\mathbf{y_0}$ inside $\overline{\mathbf{S}}$ at iteration j, and $\mathbf{y}^j$ the approximation of $\mathbf{y_0}$ inside $\overline{\mathbf{S}}$ at iteration j.

2: Compute the product

$$\mathbf{v}^T = \mathbf{r}^{jT}\overline{\mathbf{S}}$$

3: Find the n-th vertex of $\overline{\mathbf{S}}$, such that

$$|v_n| \geq \alpha \sup_i |v_i|$$

where $0 < \alpha \leq 1$.

4: Truncate so that $0 \leq v_n \leq 1$

5: Update

$$\mathbf{y}^{j+1} = \mathbf{y}^j + v_n \overline{\mathbf{S}}_n$$

where $\overline{\mathbf{S}}_n$ is the n-th column of $\overline{\mathbf{S}}$.

6: Confine the hypertriangle

$$\mathbf{y}^{j+1} = \mathbf{y}^{j+1} / \|\mathbf{y}^{j+1}\|_1$$

7: Update the residual

$$\mathbf{r}^{j+1} = \mathbf{r}^j - \mathbf{y}^{j+1}$$

8: Increment j and repeat 2-7 until some convergence criterion has been satisfied

it for three-dimensional images [5]. However, a generalized method to represent all possible deformations in space has not yet been reached, particularly when they can not be represented in a linear manner.

Signals can often be represented in a simpler way than that provided by the acquisition process. For example, when a pure tone is described, it is not necessary to list all the samples with their respective values to understand the nature underlying the signal, but it is sufficient to indicate the frequency of the tone and its sampling frequency. This provides a compact description of the measurement, that highlights the useful information of the signal and facilitates its storage, transport and consumption. The same principle applies when the example of pure tone extends to a more complex signal, such as that produced by a musical instrument [38].

It is said that natural signals are sparse. Sparsity is a characteristic that states that by using the appropriate base vectors, the signals can be described with a reduced set of components. For instance, in the pure tone example, all frequency coefficients are zero, except those corresponding to the tone [26]. The signal over time is dense, but it is sparse when it is described in the frequency domain. This means that in the process of generating a representation of a sparse signal, less significant components are zeroed and an accurate approximation of the original signal is still recovered. In other words, a natural signal in the frequency can be represented as as sparse combination of several components, and dictionary learning aims to create the frame vectors from which these components can be gathered to represent the natural signals [38]. Mathematically a signal $\mathbf{y}$ of n discrete components is sinthesized with a dictionary $\mathbf{\Phi}$ of m atoms with:

$$\mathbf{y} = \mathbf{\Phi a} + \boldsymbol{\lambda} = \sum_{k=1}^{m} a_k \boldsymbol{\phi}_k + \boldsymbol{\lambda} \tag{2.9}$$

where $\mathbf{\Phi} \in \mathbb{R}^{n \times m}$ is known as the dictionary (the frame vectors, $\boldsymbol{\phi}_k$ as its columns), $\mathbf{a} \in \mathbb{R}^n$ is the sparse vector that ponders the dictionary, $\mathbf{y} \in \mathbb{R}^n$ is the reconstructed signal and $\boldsymbol{\lambda}$ represents the error from the dictionary representation. For $\mathbf{\Phi}$ to induce dispersion, it has to hold a number of m of words or codes larger than the dimension of the original signal; then, when $n < m$ it is said that $\mathbf{\Phi}$ represents an overcomplete dictionary [71].

K-SVD [55] is a dictionary learning algorithm for creating a dictionary for sparse representation of signals. K-SVD uses singular value decomposition approach for creating a dictionary, this method consists of a generalization of the k-means clustering method. It works by alternating between sparse coding the input data based on the current dictionary using orthogonal matching pursuit (OMP) [50], and updating the atoms in the dictionary to fit the data.

2.5.3 ADM

ADM, or active dictionary model, is an iterative algorithm for image segmentation using shape models [26]. It is especially useful when delineating objects with non-linear shape

models. It is also useful for scenarios where noise components are present. These may include measurement noise, overlaps with other objects and auto-occlusions. The working principle of the ADM is summarized in Algorithm 3.

Algorithm 3 Active Dictionary Models

Input: Initial image-level approximation for signal $\mathbf{y}^1$, representative manifold $\mathbf{S}$, acceptance criterion ϵ

Output: Truncated signal after i-th iteration $\mathbf{y}^i$

1: Assign $i = 1$
2: Learn the $\mathbf{\Phi}$ dictionary using KSVD and OMP.
3: Adjust $\mathbf{y}^i$ using image-level information.
4: Minimize $\min_{\mathbf{a}}\{\|\mathbf{y}^i - \mathbf{\Phi a}\|_2 + \|\mathbf{a}\|_0\}$ using OMP. The vector $\mathbf{a}$ induce both the minimal error and the highest sparsity.
5: Assign $\mathbf{y}^i = \mathbf{\Phi a}$
6: Project $\mathbf{y}^i$ on the manifold using GP
7: **if** $\|\mathbf{y}^i - \mathbf{y}^{i-1}\|_2 < \epsilon$ then
8: exit
9: **else**
10: Assign $i = i + 1$
11: repeat from 3
12: **end if**

This process is repeated until the established convergence criterion is reached [26]. An example result of the adjustment of a shape model fitted to a vermiform structure is presented in Figure 2.7. After 19 iterations, a model adjusted to a vermiform structure is obtained, even though it is superimposed with another shape with the same characteristics.

2.6 Image processing

In order to position landmarks under the consideration of information extracted from deep neural networks, the operations of skeletonization and oriented Gaussian derivatives (OGD) are used. The skeletonization process is useful to determine endpoints on binary blobs and OGD can be used in landmark adjustment based on the edge strength in the original image.

2.6.1 Skeletonization

Skeletonization is the process of extracting skeletons from an object in a digital image. This process is usually applied to binary images with black and white pixels, which represent foreground and background pixels [62]. This morphological operation deletes

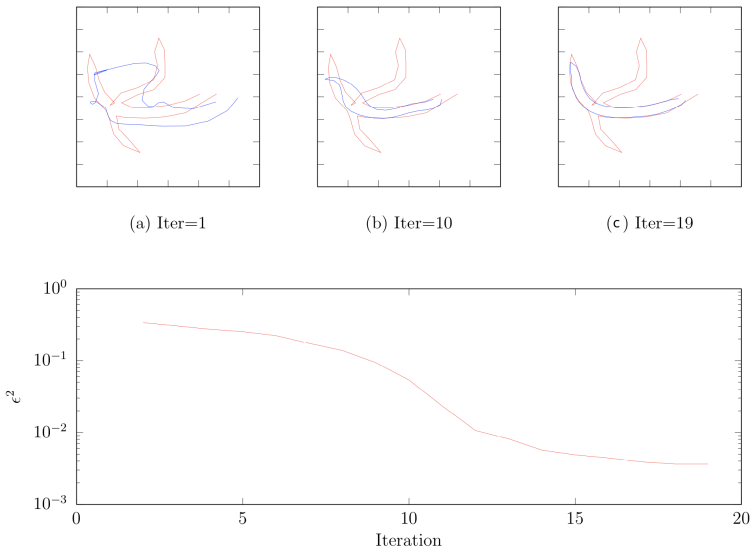


Figure 2.7: Modeling example of ADM to vermiform structures. The blue lines in a), b) and c) correspond to the shape adjustment result by means of ADM to the vermiform structure in red. Below is the error in terms of the iteration number. **Source**: M. Grüner, *Active dictionary models: A framework for non-linear shape modeling*, ITCR (2015) [26].

black foreground pixels iteratively until a one-pixel width structure, known as skeleton, is obtained. Skeletonization is essentially a pre-processing step used in image analysis [62].

The Zhang-Suen thinning algorithm [78] is frequently used to aproximate the skeleton of an image. The algorithm is a two-pass algorithm, meaning that for each iteration it performs two sets of checks to remove pixels from the image. The process is iterative and it continues until no more pixels can be removed. One example of a pixel in the evaluation process of the algorithm is showed in Figure 2.8.

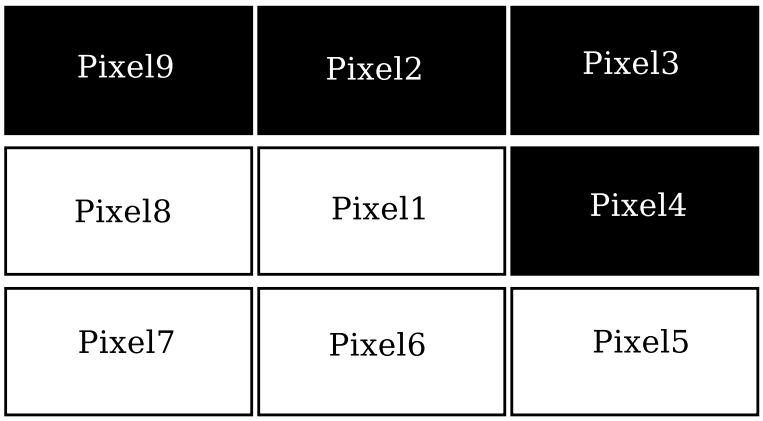


Figure 2.8: Example of grid of pixels on a binary image. In this particular case *Pixel1* corresponds to one pixel on the evaluation process

The Zhang-Suen algorithm operates on pixels within an 8-neighborhood. This means that pixels found at the border and corners of the image are not analyzed. We first define $A(Pixel1)$ as the number of transitions from black to white (0 to 1). Following the sequence $Pixel2 \rightarrow Pixel3 \rightarrow Pixel4 \rightarrow Pixel5 \rightarrow Pixel6 \rightarrow Pixel7 \rightarrow Pixel8 \rightarrow Pixel9 \rightarrow Pixel2$ (in the example of Figure 2.8, $A(Pixel1) = 1$). We then define $B(Pixel1)$ as the number of white pixel neighbours of *Pixel1* (in the example of Figure 2.8, $B(Pixel1) = 4$).

Pixels are tested using the two-pass tests. At the first stage pixels satisfying five conditions simultaneously are set to black ($Pixel1 = 0$). Pixels are set to black after all the qualified pixel are determined.

1. The pixel analyzed is white ($Pixel1 == 1$) and has eight neighbours (is not a pixel on the border or corner).
2. $2 <= B(Pixel1) <= 6$
3. $A(Pixel1) == 1$
4. $Pixel2 \vee Pixel4 \vee Pixel6 == 0$ (at least one is black)
5. $Pixel4 \vee Pixel6 \vee Pixel8 == 0$ (at least one is black)

Once the first stage is completed, at the second stage all pixels satisfying five conditions simultaneously are set to black ($Pixel1 = 0$). Pixels are set to black after all the qualified pixel are determined.

1. The pixel analyzed is white ($Pixel1 == 1$) and has eight neighbours (is not a pixel on the border or corner).
2. $2 <= B(Pixel1) <= 6$
3. $A(Pixel1) == 1$
4. $Pixel2 \vee Pixel4 \vee Pixel8 == 0$ (at least one is black)
5. $Pixel2 \vee Pixel6 \vee Pixel8 == 0$ (at least one is black)

If a pixel is chosen for removal by either pass, then it is removed. These stages are both repeated until there is no pixel chosen for removal by either step. It is to notice that this algorithm is suitable for parallel processing.

2.6.2 Oriented Gaussian Derivatives

The main objective of applying Oriented Gaussian Derivatives or OGD filtering on a image relies on the need of obtaining a scalar value that determines the strength of an edge with a controllable radius of action. It arises from mixing the properties of two filters. The Gaussian filter acts as a low-pass filter to eliminate noise and to build a scale space [37] [74] and the derivative filters allow to find edges by means of the gradient [9]. Applying this type of filters allows not only to get a rough estimate of the edgeness but also properly discriminate the elements of the image depending on the scale. The multidimensional Gaussian filter can be separated into multiple one-dimensional filters oriented to the main axes [72]: one for the horizontal direction, and one for the vertical direction. The one-dimensional Gaussian filter is defined by the Gaussian function:

$$G(x;\sigma,\mu) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}} \tag{2.10}$$

In order to create a Gaussian filter, the mean μ is taken as 0 and the normalization constants can be ignored to simplify the expression. Normalization can be done once the filter is calculated by dividing the obtained matrix between its algebraic norm:

$$\hat{G}(x;\sigma) = e^{\frac{-x^2}{2\sigma^2}} \tag{2.11}$$

The first derivative of (2.11) corresponds to :

$$\frac{\partial \hat{G}(x;\sigma)}{\partial x} = -\frac{x}{\sigma^2} \exp\left(-\frac{x^2}{2\sigma^2}\right) = -\frac{x}{\sigma^2}\hat{G}(x;\sigma) \tag{2.12}$$

where the scale in which it is filtered is defined by σ^2. Normally this scale is selected with the approximation:

$$\sigma^2 = \frac{s+1}{6} \tag{2.13}$$

where s is the scale (in pixels) that acts as the cutoff limit of the filter mask. To obtain the gradient on a particular direction, OGD can be applied using a perpendicular Gaussian filter with the same variance σ^2.

Chapter 3

SegNema: A novel strategy for nematode segmentation

Figure 3.1 shows the strategy used for the proposed stages in *SegNema*. A binary map generated by a deep learning network is used to classify each pixel as nematode or background in an image, and opening and closing operations are performed to smooth irregularities out on the result. A structural analysis of binary blobs (regions) is then performed in order to separate and locate the connected regions with possible nematode information. Next, landmarks are assigned using the information of the endpoints of the regions (obtained by a thinning process). The contour information and the endpoints are used to approximate the closest nematode shape performing a geodesic projection. Finally ADM is used to adjust landmarks into a valid vermiform shape in an iterative process. Landmarks are adjusted to the image using information extracted from the image and truncated until a rejection or convergence criterion is met.

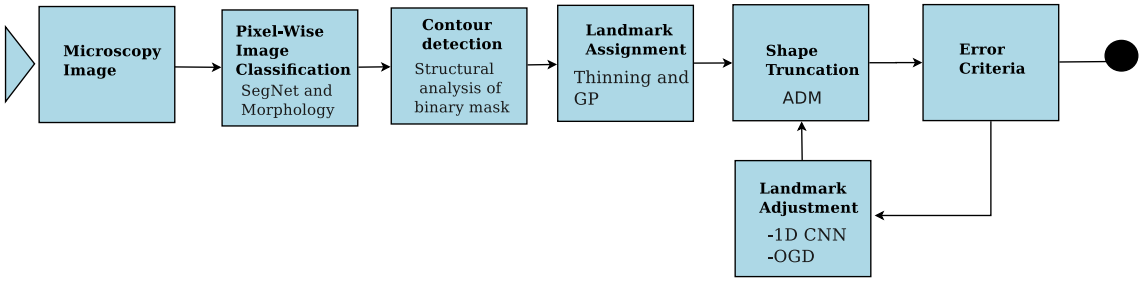


Figure 3.1: *SegNema* processing pipeline

The following sections specify the operations performed in each of the *SegNema* stages. In the case of the pixel-wise classification stage, the training strategy and the network structure are specified. The contour detection stage relies on the detection of connected components [28]. For the landmark adjustment stage two strategies were proposed: a one-dimensional classification using CNN, and an edge detection strategy using OGD. *SegNema* associates a shape model with the CNN prediction instead of just generating a bounding box. Hence, the shape of the nematode and the enclosed image region can be

further processed to extract information about species, size or orientation. Furthermore the use of shape models allows to discard predictions based on CNNs.

3.1 Pixel-Wise Image Classification

The object detection stage has to determine if a nematode exists in a certain region or not. The result is a binary mask with the same size as the input image with blobs that correspond to regions with a high probability of being part of a nematode. 1D CNNs are used in [68] to classify nematode and the results suggest that training a deep learning model with convolutional layers using 2D patches should improve the results of the classification since more contextual data can improve the results of the regressor model. To achieve this task, a pixel-wise segmentation architecture similar to SegNet (Section 2.3) is selected, where the coefficients of the upsampling layer in the decoder section are learned in the training process. The decision is made according to what is described in Table 3.1: the SegNet architecture is implementable and trainable on most 8 deep learning frameworks and can perform pixel-wise classifications.

Table 3.1: Comparison of requirements for the deep network structure for object segmentation

Architecture	Training requirements	Classification Result
SegNet [4]	Fully trained using gradient descent	Pixel-wise classification
FCN [59]	Fully trained using gradient descent	Downsampled classification
DeepLab V2 [12]	Layers need to be trained separately. Training not currently supported by all frameworks.	Pixel-wise classification

On [68] the length of the 1D patches used was 37, and the results suggests that 37×37 2D patches should improve the pixel classification results. Since this is a odd number and given that the selected architecture has downsampling stages performed by max pooling layers, the input size is increased to 40×40 pixels to keep the same size at the output layer for the pixelwise prediction, to avoid adding stride and to allow the use of up to three encoders in the arquitecture. In each convolutional layer the size for the filters is 3×3. The number of parameters for the encoder and decoder sections for 4 decoders is shown in Tables 3.2 and 3.3, respectively. The number of encoder/decoder units can be modified to tune the model.

In order to generate training data for the proposed model, a binary mask is generated from the inner area of a polygon delimited by the landmarks that indicate the nematode position on an image, as shown in Figure 3.2. The model proposed is trained using patches

Table 3.2: Parameters of 4 encoders for pixelwise segmentation

Layer	Output	#Param
Input	(40,40, 1)	0
Conv2D	(40,40,64)	640
Batch Normalization	(40,40,64)	256
Activation (ReLU)	(40,40,64)	0
Max Pooling2D	(20,20,64)	0
Conv2D	(20,20,128)	73856
Batch Normalization	(20,20,128)	512
Activation (ReLU)	(20,20,128)	0
Max Pooling2D	(10,10,128)	0
Conv2D	(10,10,256)	295168
Batch Normalization	(10,10,256)	1024
Activation (ReLU)	(10,10,256)	0
Max Pooling2D	(5,5,256)	0
Conv2D	(5,5,512)	1180160
Batch Normalization	(5,5,512)	2048
Activation (ReLU)	(5,5,512)	0
Total	-	1552761

of 40×40 pixels from the original image and the mask identifying the pixels belonging to nematode and the background.

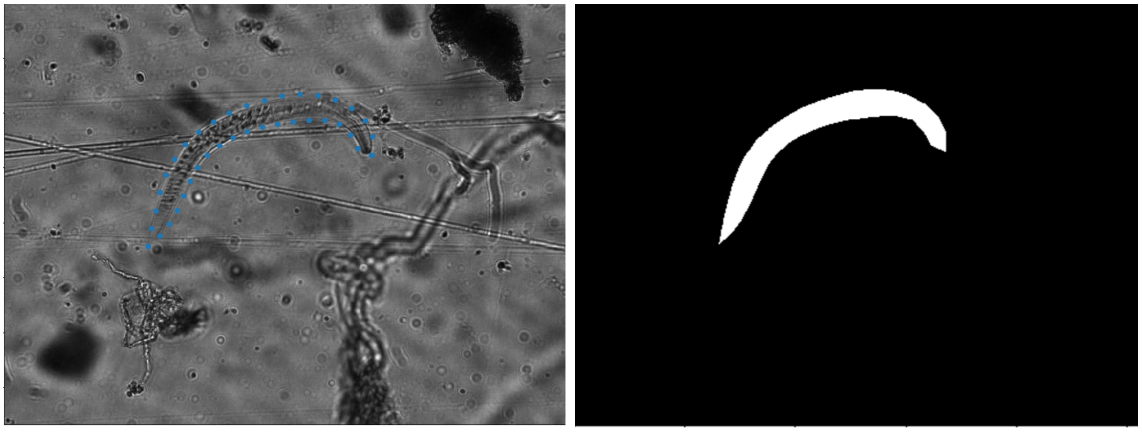


Figure 3.2: Example of a nematode and the mask generated using the landmark information on the training data. (a) An example of a nematode in Sequence *1*, with blue landmarks provided for that nematode (b) Mask generated using the information provided by the landmarks

For the class that corresponds to nematode information, the 40×40 pixel section is selected perpendicular to the spline for each landmark at the right or left side of the nematode on the original image and its corresponding place on the binary map using different offsets.

Table 3.3: Parameters of 4 decoders for pixelwise segmentation

Layer	Output	#Param
Input	(5,5,512)	0
Conv2D	(5,5,512)	2359808
Batch Normalization	(5,5,512)	2048
Activation (ReLU)	(5,5,512)	0
Upscaling2D	(10,10,512)	0
Conv2D	(10,10,256)	1179904
Batch Normalization	(10,10,256)	1024
Activation (ReLU)	(10,10,256)	0
Upscaling2D	(20,20,256)	0
Conv2D	(20,20,128)	295040
Batch Normalization	(20,20,128)	512
Activation (ReLU)	(20,20,128)	0
Upscaling2D	(40,40,128)	0
Conv2D	(40,40,64)	73792
Batch Normalization	(40,40,64)	256
Activation (ReLU)	(40,40,64)	0
Upscaling2D	(40,40,64)	0
Conv2D	(40,40,2)	130
Activation (SoftMax)	(40,40,2)	0
Total	-	3912514

In order to create the perpendicular section for a given landmark l, a line between the landmarks $l+1$ and $l-1$ is drawn and a perpendicular segment to this line and landmark l is calculated. This second segment corresponds to a possible offset for a position of the center of the 40×40 pixel patch. The length of the offset is given by the average width of the nematodes. An example of such training data example is shown in Figure 3.3.

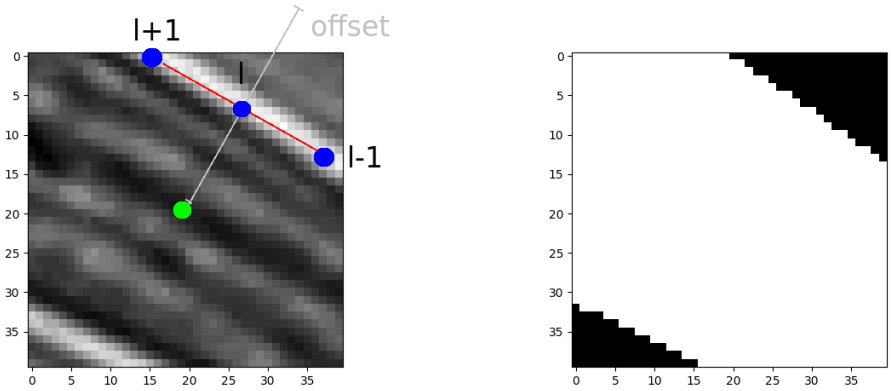


Figure 3.3: Example of a nematode and the mask using a window of 40×40 pixels. (a) Information from the original image on Sequence *1*, on grey offset posible values for a given point in l (b) Mask generated using the information provided by the the landmarks at the given offset

For the class that corresponds to the background the image is first preprocessed with a OGD filter such as the one introduced in Section 2.6.2. The width and standard deviation of the Gaussian function used to create the kernel linearly depend on a scale term, which is calculated using the landmarks as shown on Figure 3.4. The landmarks for the head and the tail are discarded, and the width and standard deviation are calculated using the euclidean distance between the corresponding landmarks on each side of the nematode.

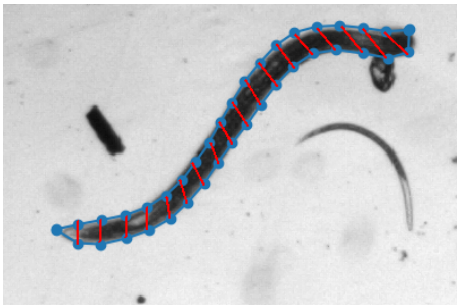


Figure 3.4: Example of the width calculated using landmarks for a given nematode

The rationale of this step is to take into consideration information of non-nematode areas near a tagged nematode. After applying a mask that removes the nematode information enclosed by the landmark polygon, the OGD allows gradients near the nematode to be accentuated and gradients distant from the nematode to be attenuated. Data augmentation techniques: rotations and translations are applied to both, nematode and background samples, in order to avoid overfitting and improve the generalization of the network.

In order to evaluate an image and generate a binary mask of the same size Algorithm 4 is proposed. An image must be divided into regions of 40×40 pixels to perform the evaluation; so the height and width must be multiple of 40. For example, in image of size of 1024×768 pixels the nearest multiple is 1000×760 pixels, leaving a region of 24 pixels of height and 8 pixels width without evaluation, if the evaluation is performed only once. The proposed algorithm allows to define a height and a width stride, which acts as an offset for the starting point to divide the image into sections of 1000×760. The maximum stride is given by the difference between the size of the image and the nearest multiple, so it is 24 for width and 8 for height in the example mentioned previously. This allows to evaluate the image for a given number of times and include predictions at offsets at the interval 0 to $stride - 1$ to increase accuracy.

Algorithm 4 Image evaluation using 40×40 model

Input: Image to evaluate *image*
Output: Binary mask *result*

1: set $base = 40$
2: set $iteration = 0$
3: set $width = image.width$, $height = image.height$
4: $width\_max = round(int(\frac{float(width)}{base}) * base$
5: $height\_max = round(int(\frac{float(width)}{base}) * base$
6: $width\_stride\_max = width - width\_max$
7: $height\_stride\_max = height - height\_max$
8: Assign $width\_stride \in [0, width\_stride\_max]$
9: Assign $height\_stride \in [0, height\_stride\_max]$
10: $Image\_window = image[width\_stride : width\_max + width\_stride, height\_stride : height\_max + height\_stride]$
11: partial_result= $predict(Image\_window)$
12: $result[width\_stride : width\_max + width\_stride, height\_stride : height\_max + height\_stride] += partial\_result$
13: iteration+ = 1
14: **if** $iteration < iteration\_max$
15: repeat from 8
16: **else**
17: $result = result/iteration\_max$
18: **end if**

Finally, morphological operations are carried out to refine the result of the pixel-wise classification; using a disk-shaped structuring element, as shown in the Figure 3.5. The idea of this step is to smooth out possible irregularities in the detected edges for sections that are detected as nematode.

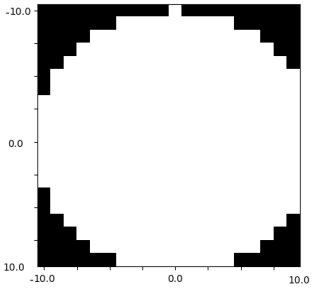


Figure 3.5: Example of the type of kernel used during morphological opening and closing

3.2 Contour detection

The object detection stage produces regions (blobs) that may contain nematodes in the image and also regions erroneously classified as nematodes. Adding an analysis stage of the blobs detected in the prediction mask should improve the accuracy of the results. Each blob is separated using pixel 8-connectivity, and the number of pixels for each blob is counted. If the number of pixels is beneath a threshold then the blob is discarded. This is done to avoid analyzing the shape of a blob for which it does not exist enough information to generate a reliable shape model. The remaining blobs are analyzed to get their contour. This corresponds to the pixels at the edges of the region classified as nematode, and these are used to generate a shape model associated with the blob on the landmark assignment stage. Figure 3.6 summarizes the contour detection in a flow chart process diagram.

3.3 Landmark assignment

The contour provides an approximation of the nematode edge but alone it is not restricted to follow any particular shape. Therefore, the contour can include components of the prediction mask which correspond to other microorganisms or biological material present in the image, that are erroneously classified as nematode. In order to associate a shape model to these blobs an initial assignment of landmarks is performed, based on the skeleton approximation constructed using the Zhang-Suen thinning algorithm (Section 2.6.1). The skeleton is a one-pixel-wide structure associated to a binary blob an it allows to determine the possible tail and head landmarks of a nematode by checking for endpoints. An example of such a skeleton is depicted on Figure 3.7. It is shown that irregularities on a binary blob can cause more than two endpoints in the same skeleton, so in order to avoid possible conflicts, the skeleton is interpreted as a non-directed graph where the nodes correspond to the endpoints and intersections between branches of the skeleton. To select the position of the head and tail, the shortest path with largest weight is selected, where the weight is determined from the geodesic distance between nodes of the graph.

the distance between two vertices in a graph is the number of edges in a shortest path (also called a graph geodesic) connect

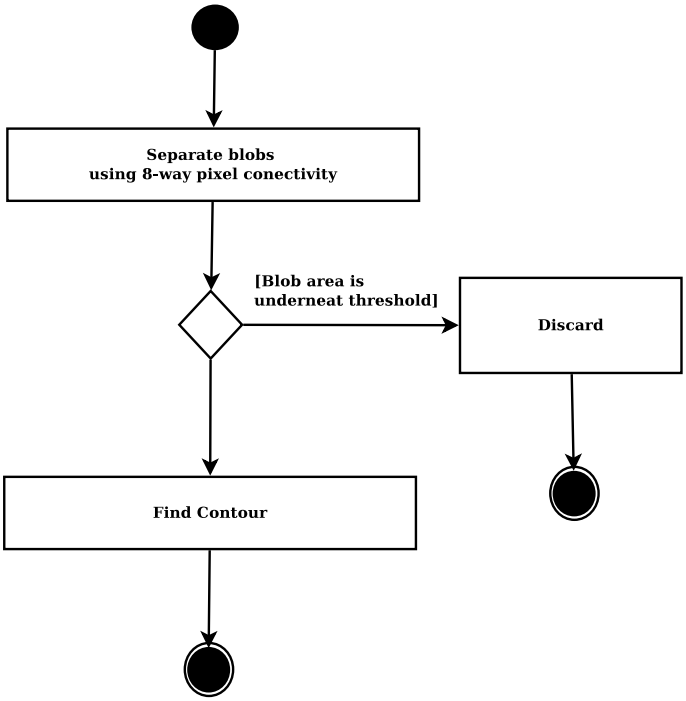


Figure 3.6: Flowchart for the process proposed to obtain the contour of binary blobs in a binary image

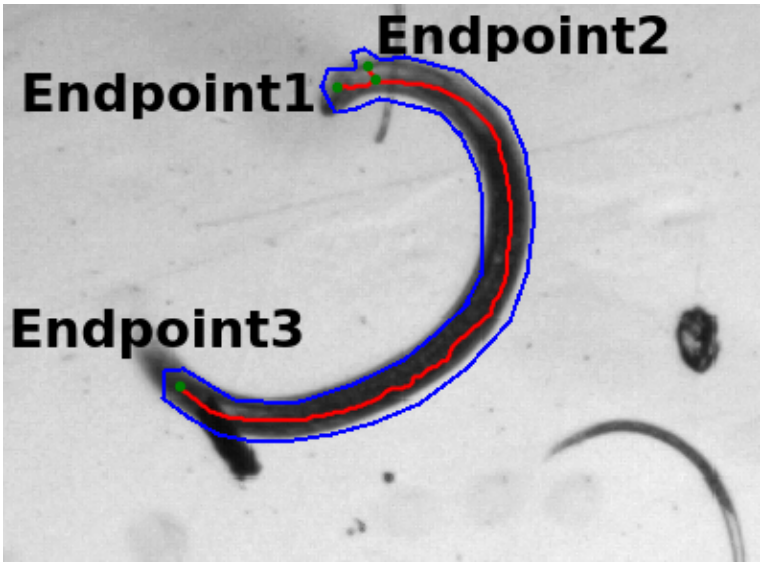


Figure 3.7: Skeleton constructed from a region predicted as nematode that has irregularities at the edges. The skeleton is marked in red, its endpoints in green, and the contour of the binary region that is classified as nematode is showed in blue.

Once the possible head and tail points of the skeleton have been obtained, the closest point on the previously obtained contour is searched for. The points of the contour are divided into two sections between the head and the tail in order to obtain a spline for each side of the nematode as shown in Figure 3.8. These two cubic degree splines are sampled to have 40 landmarks. Hence, each shape is represented by a point in an 80-dimensional space (since each landmark has two coordinates). Finally, these shapes are projected onto a 79-dimensional manifold embedded in the shape space and approximated with training shape samples using the geodesic projection algorithm (Section 2.5). The euclidean distance between a shape in the 80 dimensional space and the nearest neighbor on the manifold is proposed as a first rejection mechanism.

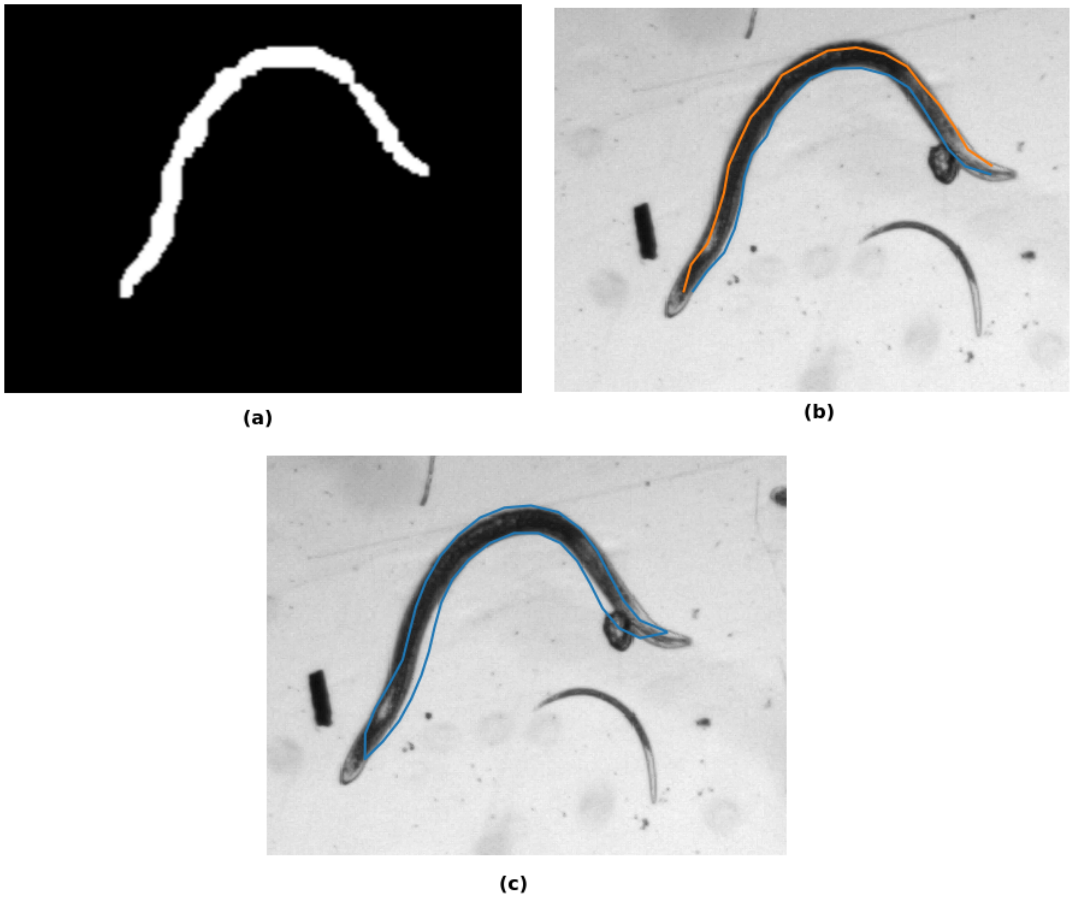


Figure 3.8: Example of the first shape approximation obtained at the landmark assignment stage.
(a) Example of the a binary mask where in white are sections classified as nematode and in black background , (b) Spline nematode division for one of the blobs using the head and tail approximated using the skeleton (c) Geodesic projection using 40 landmarks obtained by spline interpolation.

3.4 Shape truncation

The shape adjustment stage is done by means of ADM as described in Section 2.5. In order to train the dictionary all available nematode shapes on a training set are aligned using Procrustes analysis [64] where a horizontal shape is used as a reference. No distinction is made between the head and the tail in the available landmarks, so the number of shapes available is increased by building synthetic shapes considering the cases were the first landmark describes the tail or the head. The shapes corresponding to the mirror of the shapes available are also considered. The shapes are normalized and the average nematode is computed, which is then used as the reference for aligning nematode shapes using Procrustes analysis. An example of the average nematode computed from the normalized shapes is shown in Figure 3.9.

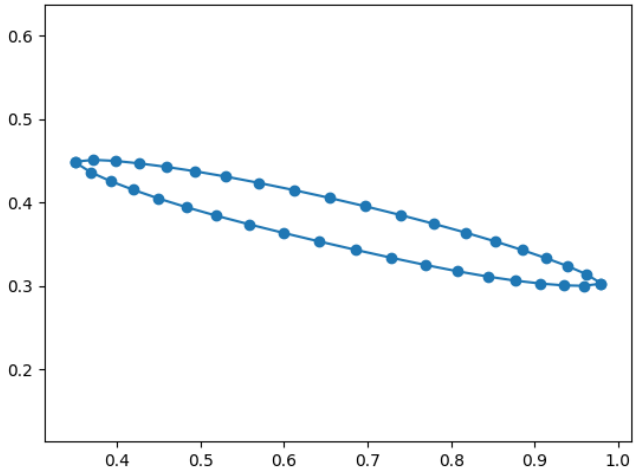


Figure 3.9: Example of an average nematode shape computed from normalized shape data.

Using this information, the dictionary is trained using KSVD with 160 atoms which is enough to represent nematode shapes [26]. Next, the manifold is built to carry out the geodesic projection. In principle, because the first iteration is obtained directly from a geodesic projection, it is not expected that the landmarks suffer any alteration, but once the landmarks are adjusted this stage serves to eliminate noise and ensures that the shape does not deviate from a valid nematode.

3.5 Landmark adjustment

The strategy of adjusting points to the image, seeks a match of the landmarks detected with the deep neural network to the actual image, using information such as edges, which might correspond to nematodes. Two strategies are evaluated to drive the adjustment. The first one relies on the information extracted by applying OGD filters to the image.

Using the previously located landmarks, the width of the possible nematode in that section is determined and a line segment half that size is obtained in a direction perpendicular to the spline, as shown in Figure 3.10. Then the landmark is moved towards to the position where the gradient is higher. Any noise around the nematode that takes the point too far will be corrected by the shape truncation step.

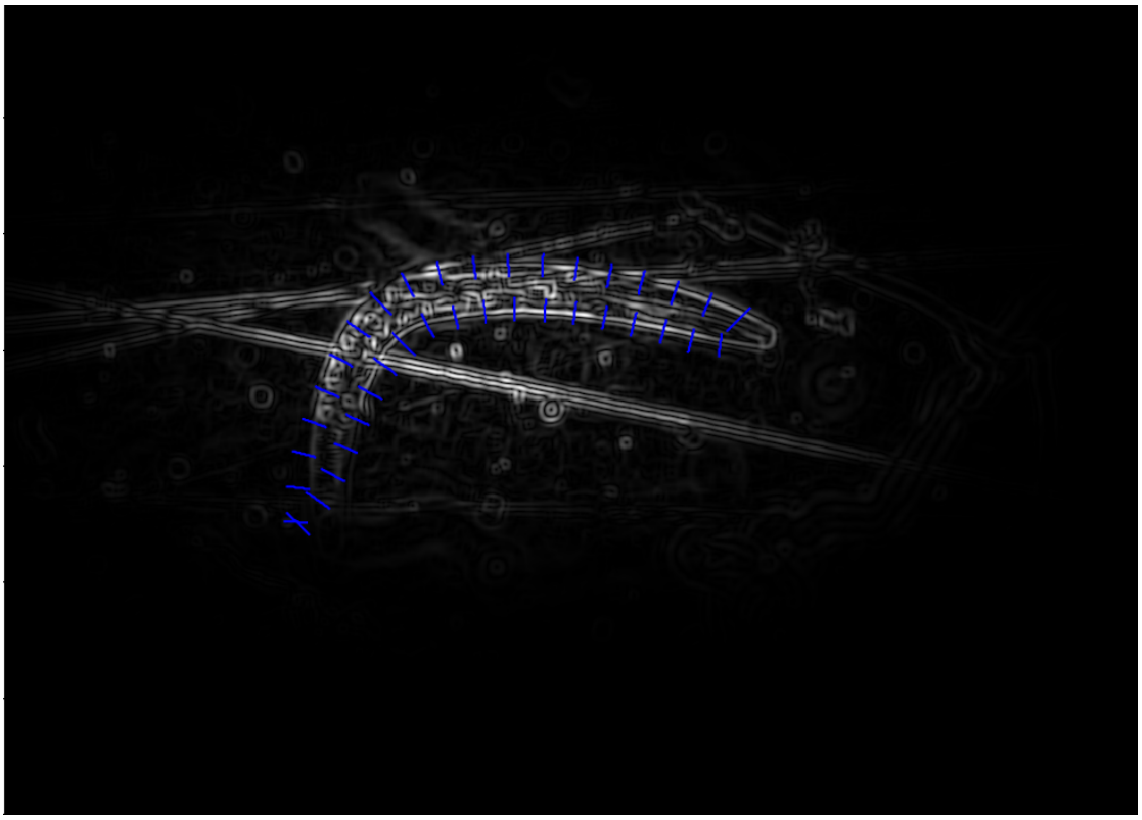


Figure 3.10: Oriented gradient derivative based landmark adjustment. In blue the line segment perpendicular to the spline from which the landmark is adjusted.

The second strategy is based on a process that reuses 1D CNN regressor developed in [68]. A line segment of 37 pixels perpendicular to the spline in every landmark is computed. All pixels in the line segment are classified and an derivative of Gaussian filter is applied to determine the point of highest variation between regions classified as nematode or as background and the landmark is moved to that point. As in the pure OGD method, noise around the nematode that misplaces a landmark will be corrected by the shape truncation step. An example of this is shown on Figure 3.11 where the landmarks are moved to the strongest edge in the clasification result made by the 1D CNN network.

This stage is also used for discarding structures that do not correspond to nematodes. Two main discarding criteria where used on the error computation step:

1. In each iteration i the resulting shape after the landmark adjustment and confinement to a valid shape with ADM ($\mathbf{y}_i$), is compared with the landmark positions at the previous iteration ($\mathbf{y}_{i-1}$). Taking $\mathbf{y}_{i-1}$ as a reference and $\mathbf{y}_i$ as the predicted

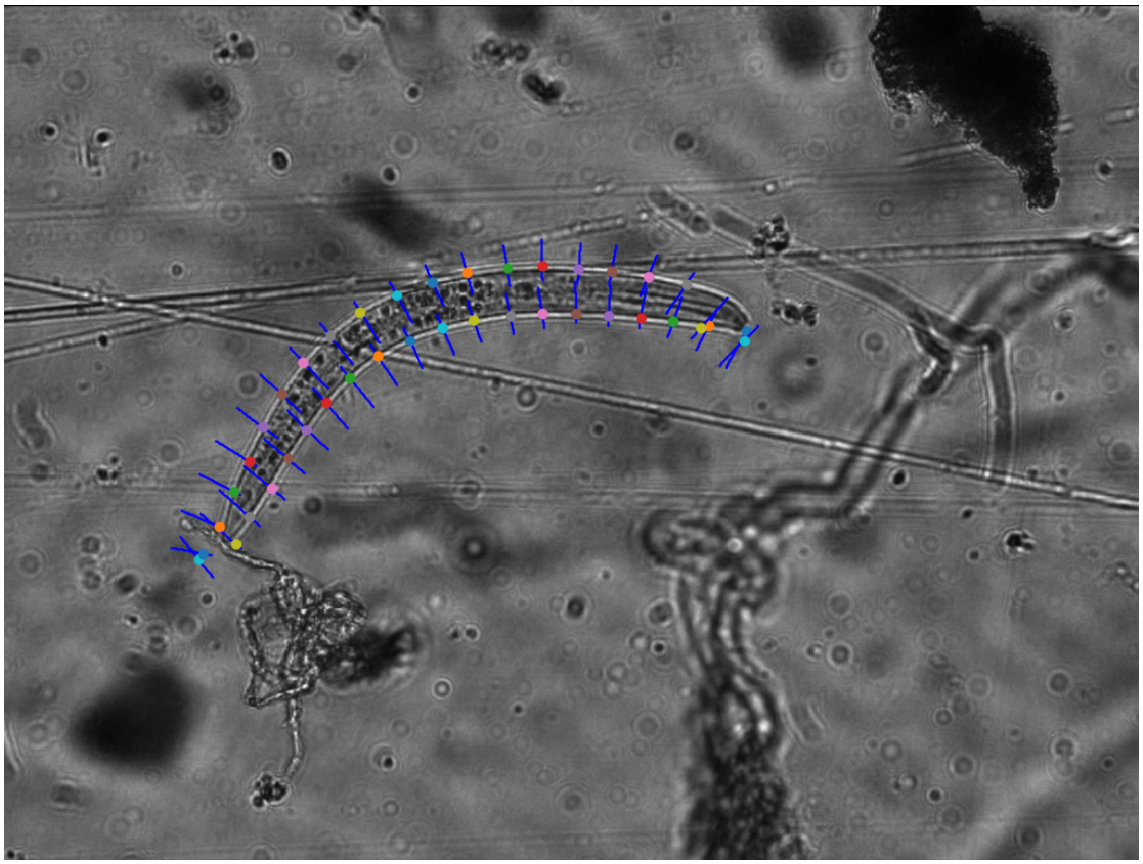


Figure 3.11: 1D nematode edge regressor strategy for point adjustment. The result of the 1D CNN network is passed to a 1D derivative of Gaussian and then the point is moved to the position of maximum response

value, the first discard criteria is fullfiled if the root mean square error (RMS) between those shapes does not decrease after 20 iterations.

2. Let $\mathbf{y}_0$ be the localization of landmarks using the geodesic projection and $\mathbf{y}_i$ the shape obtained at the i-th iteration for which the convergence criterion was met. The shape is discarted if the root mean square error between $\mathbf{y}_0$ and $\mathbf{y}_i$ is higher than 30%.

Chapter 4

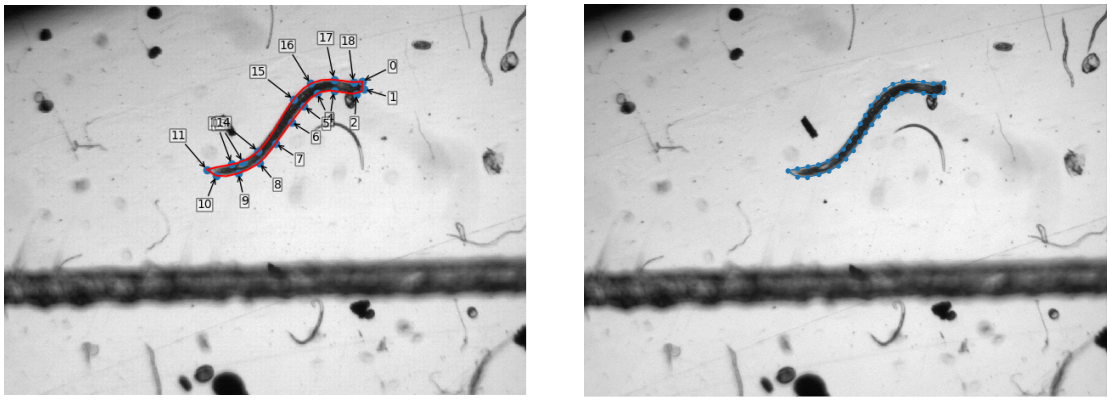
Experimental Results and Analysis

The validation experiments for the *SegNema* stages have two main objectives: first, to determine the accuracy of the segmentation of nematodes using the proposed deep neural network over a validation set of microscopy images; second, to determine the accuracy of the landmark adjustment as well a comparison of the two landmark adjustment strategies proposed.

4.1 Test and training data

The data set used for training and testing this system encompasses 3260 manually labeled uncompressed images of size 1024×768 pixels obtained from 13 different sequences of microscopy images. Figure 4.1 shows an example of the training data used. Each image has a single nematode marked with landmarks set by a human operator; however there might be more than one nematode depicted in one image. This implies that no assumptions can be made regarding the complementary regions of the labeled nematode as being non-nematode, such as the case showed in the example. The number of manually placed landmarks varies between samples within each sequence, so spline interpolation [36] is used to produce 40 landmarks per nematode on each sample image. This allows the landmark data to be suitable for training and testing in the shape truncation stage.

A fraction of 10% of the data is randomly selected and reserved for validation and testing. The rest of the data is used for training the stages of *SegNema*. One example for each image sequence is shown on Figure 4.2. Each sequence varies on factors such as the number of nematodes present, the size of the nematodes, the configuration of the microscope, the organic material on the sample, among others. The partition of the data is performed per sequence. Table 4.1 shows the total number of images available in each of the sequences as well as the partition used for testing and training. For the case of the deep neural network, the sequences are partitioned in 40×40 patches as described in Section 3.1, 80% of which is used to train the network. The remaining 20% is used as the validation set in order to verify that any increase in accuracy over the training data set yields an increase



(a) **(b)**

Figure 4.1: Example of one manually labeled image on the training and test set. (a) Nematode with landmarks set by a human operator (b) Nematode with 40 landmarks interpolated from the original data using splines

in accuracy over a data set.

Table 4.1: Partition of available data for use in *SegNema*. 10% of the data is reserved for testing. The data is selected proportional to the total number of images per sequence

Sequence	Total of images	# Used for training	# Used for testing
1	581	523	58
2	158	143	15
3	415	374	41
4	160	144	16
5	59	54	5
6	435	392	43
7	113	102	11
8	579	522	57
9	135	122	13
10	15	13	2
11	135	122	13
12	373	336	37
13	102	92	10

4.2 Pixel-Wise Classification

The *Keras* framework [15] is used for implementation, training and evaluation of the selected deep learning model architecture. Figure 4.3 shows the metrics for the cross entropy loss computed for the train and validation sets during the training process using

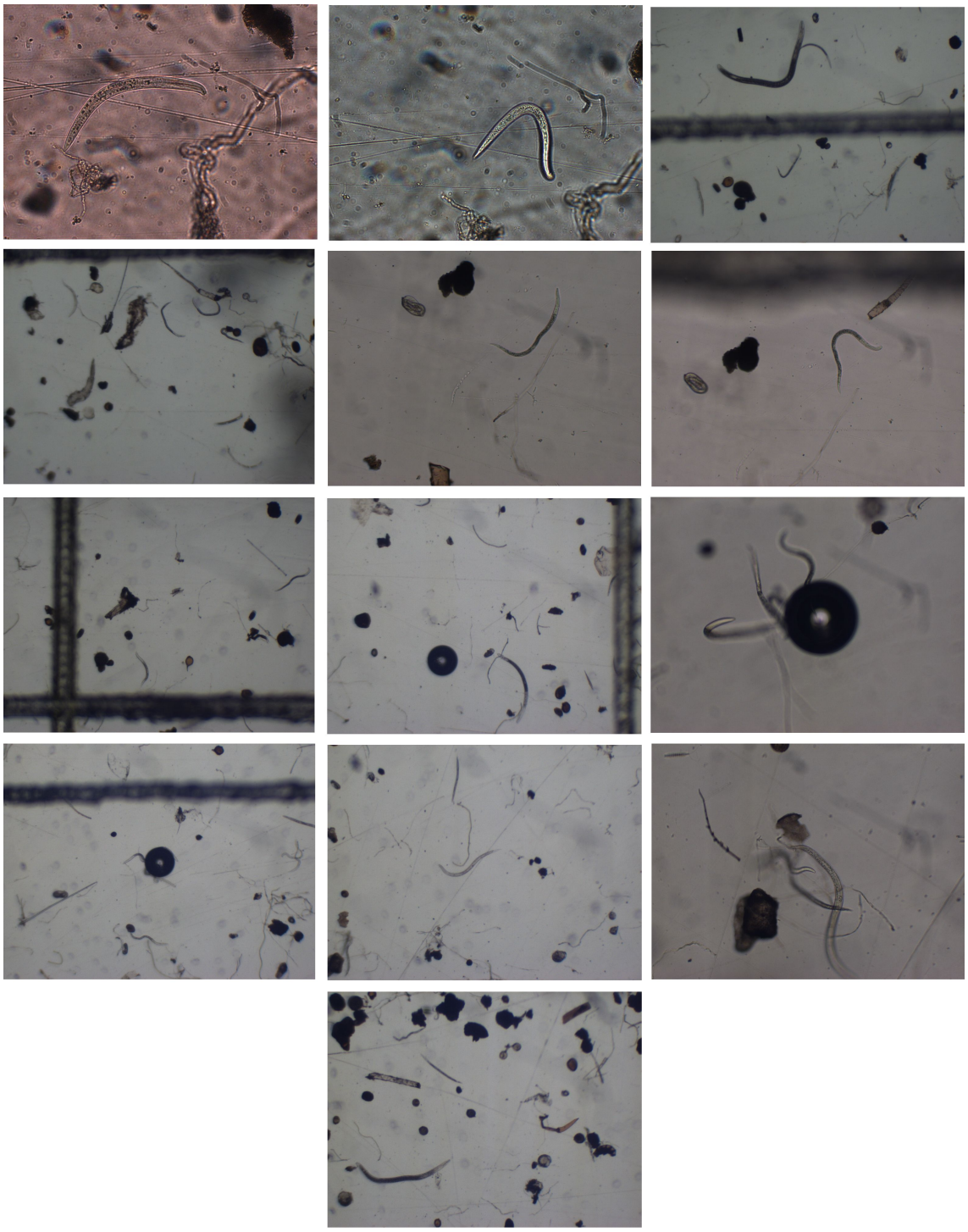


Figure 4.2: Examples from each of the 13 image sequences available for training, test and validation

4, 3 and 2 *SegNet* encoders/decoders respectively. In the case of Figure 4.3 (a), the divergence between train and validation errors indicate that the model overfits; however, the loss stabilizes at 0.08. This means that there is not enough data to train the parameters using 4 encoders. In case of Figure 4.3 (b) the model is able to reach a similar accuracy with the training and testing sets and avoids overfitting by using less parameters. Figure 4.3 (c) shows a higher loss than the previous cases. Given these results, the model with 3 encoders and 3 decoders is selected.

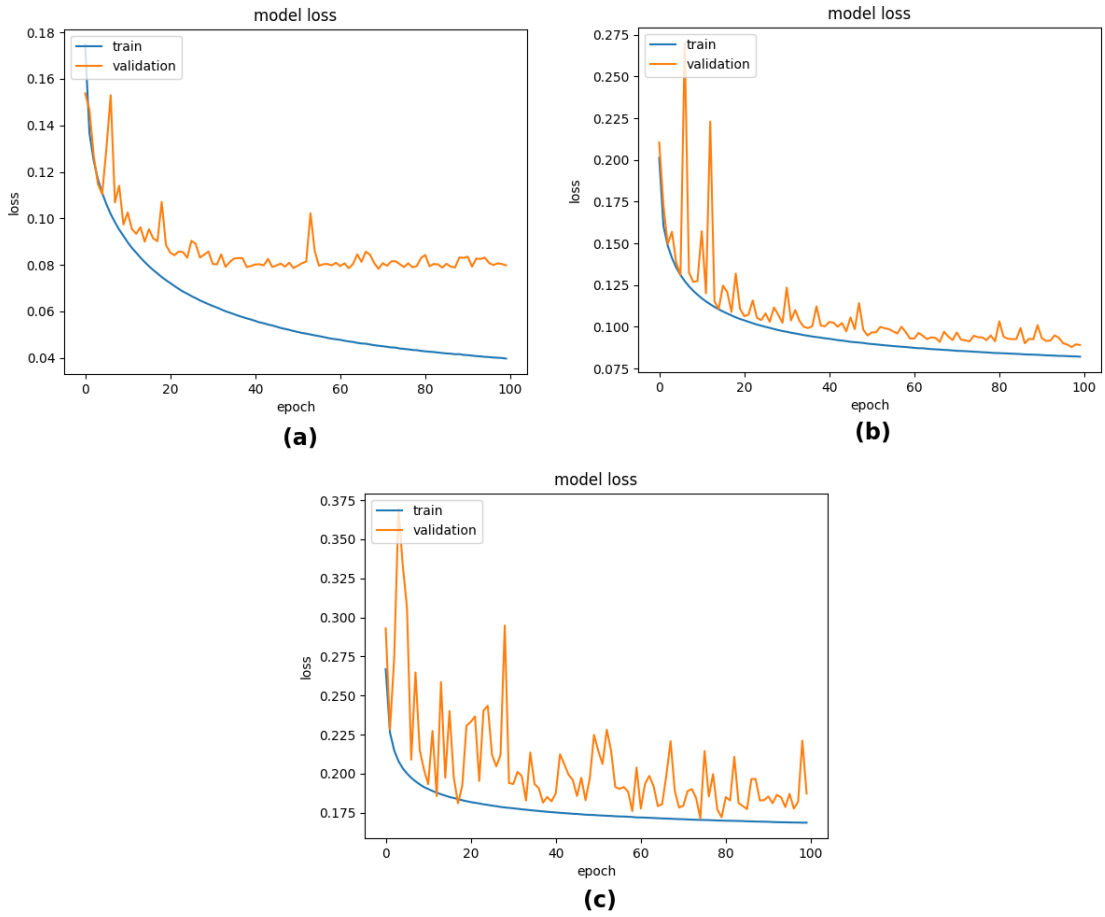


Figure 4.3: Metrics for cross entropy loss on validation and train sets during *SegNet* training process for the selected model configurations. (a) Model loss for *SegNet* configuration using 4 encoders and 4 decoders (b) Model loss for *SegNet* configuration using 3 encoders and 3 decoders (c) Model loss for *SegNet* configuration using 2 encoders and 2 decoders.

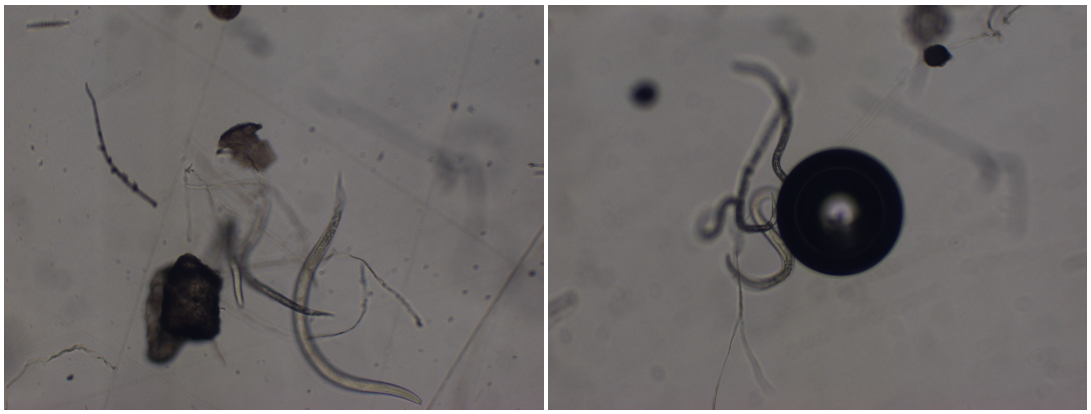
The accuracy of the network is also evaluated using the 321 images of the test set on the selected model configuration, and generating 40×40 patches following the same strategy used to generate training data. The loss and mean absolute error (MAE) were computed using the prediction of the network and compared to the expected binary result. This information is detailed in Table 4.2.

The network performed worst on sequences 9 and 12. An example of these sequences is shown in Figure 4.4. In these sequences there are overlaps between nematodes and on

Table 4.2: Loss computed for SegNet using patches from the test set

Sequence	Eval data samples	Loss	Mean absolute error
1	30479	0.1108	0.0450
2	8010	0.1171	0.0673
3	20996	0.1104	0.0567
4	7860	0.1628	0.0649
5	2400	0.1558	0.0662
6	21390	0.1152	0.0584
7	5457	0.1844	0.0673
8	28060	0.1927	0.0706
9	6570	0.4857	0.1292
10	510	0.1488	0.0708
11	6845	0.1696	0.0656
12	18626	0.3391	0.1037
13	5476	0.2100	0.0718

each case a different individual is tagged. This causes a loss penalty since the score for these cases has to be lowered to keep the total loss at a minimum. The other sequences show a loss that ranges from 0.11 to 0.21 with the trained model. This indicates that the model achieved an acceptable degree of generalization over the whole data range. These results are an indicator on the expectations on the performance of the subsequent stages. In terms of accuracy, the model obtained from 12.9% to 4.5% of mean absolute error.

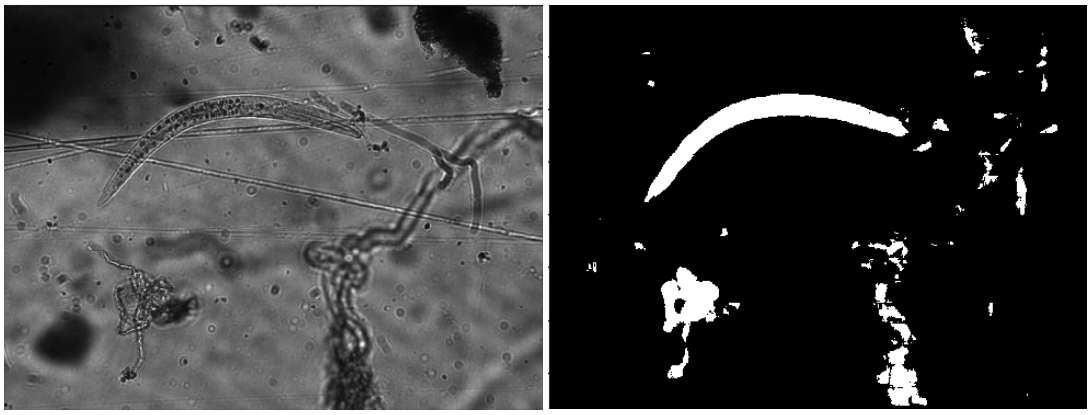


(a) **(b)**

Figure 4.4: Worse case scenario for the training data on the segmentation task using deep learning with only one labeled nematode (a) Example of sequence *9*; several nematodes overlap during the sequence and only one per image is manually segmented as ground truth (b) Example of sequence *12*; nematodes overlap during the sequence and there are occlusions with other objects

Figure 4.5 shows a result of the binary pixel-wise classification map for an image, using Algorithm 4 with a certainly of 97% accuracy for the nematode prediction as used in

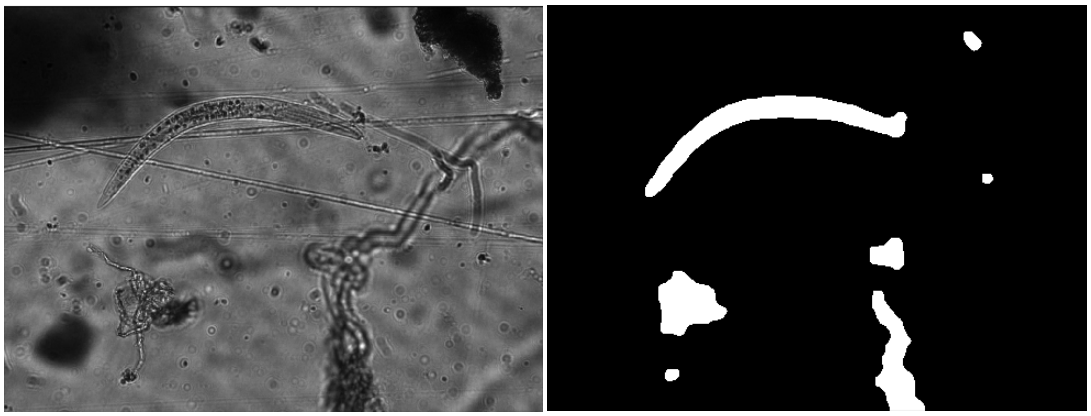
[68]. This means that only predictions with a score above 0.97 are classified as nematode and pixels with scores below are assigned to the background. The 0.97 threshold is applied on every test sequence. The criteria for selecting the training data for background information fails to generalize all non-nematode information, since some regions in the image background are classified as nematode even when the shape is not vermiform.



(a) **(b)**

Figure 4.5: Binary classification mask generated from a test image sample without applying any morphology operations (a) Original image of the test sequence (b) Binary mask generated using proposed algorithm without any morphology operations

Figure 4.6 shows the same binary pixel-wise classification map for the image in Figure 4.5(a) but after applying opening and closing operations. Blobs close to each other are joined and edges are smoothed out for the contour analysis. Also, disturbances such as overlaps with objects or wrongly classified background regions caused unwanted deformations in the nematode binary blobs (see for instance the head of the nematode in figure Figure 4.6(b))



(a) **(b)**

Figure 4.6: Binary classification mask generated from a test image sample applying morphology operations (a) Original image of the test sequence (b) Binary mask reconstruction generated using proposed algorithm with morphology operations

4.3 Contour detection and landmark assignment

The binary mask generated by means of the pixel-wise classification stage can be evaluated against the mask generated from the polygon enclosed by the landmarks of the test data. However, as shown previously, the pixel-wise classification mask may contain regions that do not belong to nematode information or contain information of another nematode that was not labeled in the image. In a first step, the final result of the pixel-wise segmentation generated in the object detection stage has to be compared with the binary mask obtained from the test data, discarding all binary sections or blobs that do not correspond to the ground truth (i.e. the one manually labeled in the image). This analysis allows to estimate the score of the smoothed prediction without penalizing the network for detecting more than one nematode. The proposed criterion for discarding the blobs that do not correspond to nematodes is evaluated in a subsequent test.

Table 4.3 shows the distribution shift of the Jaccard index (Section 2.2) for the 321 test samples, as a way to validate the operation of *SegNema*. The Jaccard index is able to summarize the overlap of the binary blob obtained by pixelwise classification and ground truth achieved for the nematode class per image. The rows in the table represent the IoU score computed from the comparison of the ground truth binary mask with the pixel-wise classification mask generated directly from the object detection stage without discarding any blob in the image. The columns in the table represent the IoU score computed with the ground truth and only the binary blob closer to the reference mask, discarding all remaining elements in the classification mask. Only one connected region is taken into account for this measurement, meaning that if a mask is composed of more than one piece, only one is taken for reference. The regions were separated using an 8-pixel connectivity check available on the *scikit-image* library [73].

Table 4.3: Distribution shift for the intersection over union (Jaccard index) from the raw pixel-wise classification and after discarding all but the region closest to the ground truth. Highlighted the highest shift of the test samples

		Best component vs. Ground truth									
	IoU	0-0.09	0.1-0.19	0.2-0.29	0.3-0.39	0.4-0.49	0.5-0.59	0.6-0.69	0.7-0.79	0.8-0.89	0.9-0.99
All components vs. Ground truth	0-0.09			2		7	**12**	8	2		
	0.1-0.19			1	3	7	17	**27**	8		3
	0.2-0.29			1			9	13	13	**32**	
	0.3-0.39				1	1	1	3	10	**16**	3
	0.4-0.49							5	**48**	29	
	0.5-0.59								6	**27**	
	0.6-0.69							**3**		**3**	
	0.7-0.79										
	0.8-0.89										
	0.9-0.99										

The results show a bias were 92.8% of the blobs generated by the sementation stage have a Jaccard index between 0.50 and 0.99 once the ground truth polygon is compared with the matching blob in the prediction mask. In 100% of the test cases the Jaccard index increased or stayed under the same range once discarding all but the corresponding mask. A low score can be attributed to occlusions in the image or overlaps of nematodes that split one nematode blob into several non-connected sections; discarding one or more of those sections decreases the area for the Jaccard index calculation. Tables 4.4 and 4.5 show the information normalized by rows and columns, respectively.

On Table 4.4, the results show that the largest sample for the data whose Jaccard index is between 0.8 and 0.9 comes from an image where an index was originally between 0.2 and 0.3, for the nematodes for which there are landmarks available. This increase of around 0.5 is found in most cases coming from scores below 0.4 to 0.7; for instance, most 0.5 to 0.6 scores were obtained from 0 to 0.1 cases. The mask with all components evaluated in a score between 0.4 to 0.5 and 0.5 to 0.6 show an increase of 0.4 on the Jaccard index. Table 4.5, shows that most components with a index of 0.2 to 0.29 were given by already low scores. Also it shows that in the range of 0.3 to 0.69 most cases are given by an low scoring case from 0.1 to 0.19. This changes on scores 0.7 to 0.79, for which most cases had an original score between 0.4 to 0.49.

Table 4.4: Distribution shift for the intersection over union (Jaccard index) from the raw pixel-wise classification and after discarding all but the region closest to the ground truth normalized by rows. Highlighted the most common drifts given an initial score.

		Best component vs. Ground truth									
	IoU	0-0.09	0.1-0.19	0.2-0.29	0.3-0.39	0.4-0.49	0.5-0.59	0.6-0.69	0.7-0.79	0.8-0.89	0.9-0.99
All components vs. Ground truth	0-.09			0.06		0.23	**0.38**	0.26	0.06		
	0.1-0.19			0.02	0.04	0.11	0.26	**0.40**	0.12		0.04
	0.2-0.29			0.01			0.13	0.19	0.19	**0.47**	
	0.3-0.39				0.03	0.03	0.03	0.08	0.29	**0.45**	0.08
	0.4-0.49							0.06	**0.58**	0.35	
	0.5-0.59								0.18	**0.82**	
	0.6-0.69							0.5		0.5	
	0.7-0.79										
	0.8-0.89										
	0.9-0.99										

The geodesic projection is used to discard blobs that do not contain a shape corresponding to a nematode in the landmark assignment stage. For each blob in the image, thinning and contour detection is performed using methods in the *OpenCV* [7] and *scikit-image* [73] libraries, respectively. As explained in Section 3.3, an initial landmark assignment relies on a projection onto a manifold embedded in an 80 dimensional shape space. To build the training data for ADM, all 2839 shapes from the training data were used. The

Table 4.5: Distribution shift for the intersection over union (Jaccard index) from the raw pixel-wise classification and after discarding all but the region closest to the ground truth normalized by columns. Highlighted the highest value, which represent the most representative initial score per final comparison.

		Best component vs. Ground truth									
	IoU	0-0.09	0.1-0.19	0.2-0.29	0.3-0.39	0.4-0.49	0.5-0.59	0.6-0.69	0.7-0.79	0.8-0.89	0.9-0.99
All components vs. Ground truth	0-.09			**0.5**		**0.47**	0.30	0.13	0.02		
	0.1-0.19			0.25	**0.75**	**0.47**	**0.43**	**0.45**	0.09		**0.5**
	0.2-0.29			0.25			0.23	0.22	0.14	**0.30**	
	0.3-0.39				0.25	0.06	0.03	0.05	0.11	0.15	**0.5**
	0.4-0.49							0.08	**0.55**	0.27	
	0.5-0.59								0.07	0.25	
	0.6-0.69							0.05		0.03	
	0.7-0.79										
	0.8-0.89										
	0.9-0.99										

final count for the shapes used to create the manifold and train the dictionary is 11729 when applying the data augmentation operations described in Section 3.4.

The 11729 nematode shapes are randomly partitioned in 10 different subsets in order to perform a cross-validation analysis. For each set, the euclidean distance in the 80 dimensional shape space is calculated to the closest neighbor in the remaining 9 subsets, i.e. between each of the 1172 shapes belonging to one subset and the 10557 samples of the 9 remaining subsets. A histogram with the bins of width 0.05 is computed with all 11729 distance (Figure 4.7). The probability distribution of distances is estimated with normalized histogram, and with it the cumulative distribution is derived. Figure 4.7 shows that with a probability of 0.99 all valid nematodes shapes will have an euclidean distance in the shape space to the manifold equal or less than 0.07. This distance is used as a threshold to discard non-vermiform shapes in the pixel-wise classification stage. An example of the separation of the data is shown in Figure 4.8.

Table 4.6 shows the evaluation of this criterion using the manually labeled 321 images of the test set as ground truth. The error percentage is shown for the cases were the labeled nematode was detected (true positive) and the cases were the nematode was not detected (false negative). The worst performing sequences are 9 and 12 since occlusions and overlaps are present during most of the sequence and cause pixel-wise classification mask include shapes that depart from the learned distribution. On the remaining sequences the error is contained within 10% and this can be attributed to irregularities on the classification map, occlusions and overlaps.

The behaviour of the images in sequences 9 and 12, for which the error is 70% and 32% respectively, matches the results expected, given the loss obtained in the model evaluation

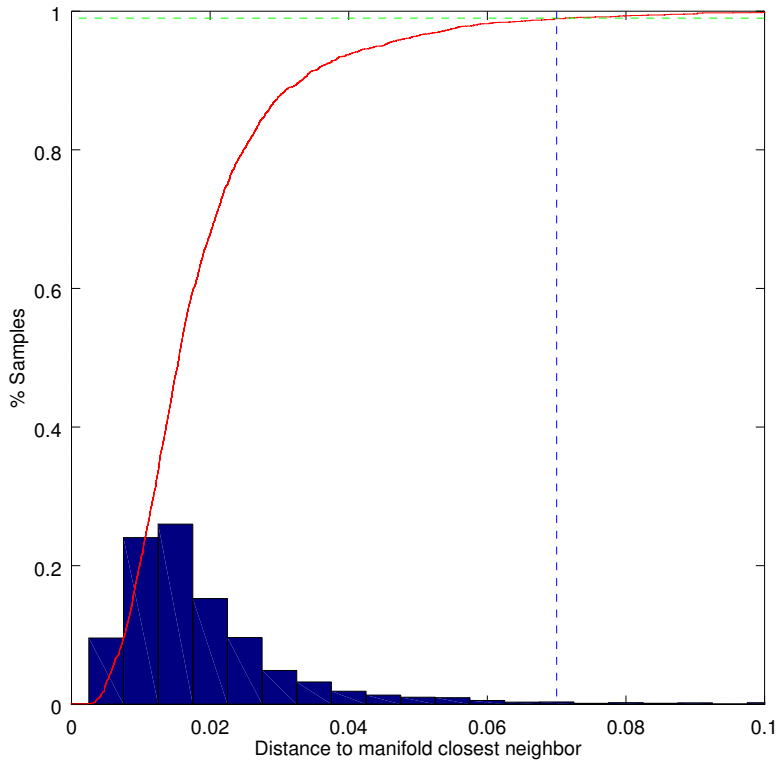
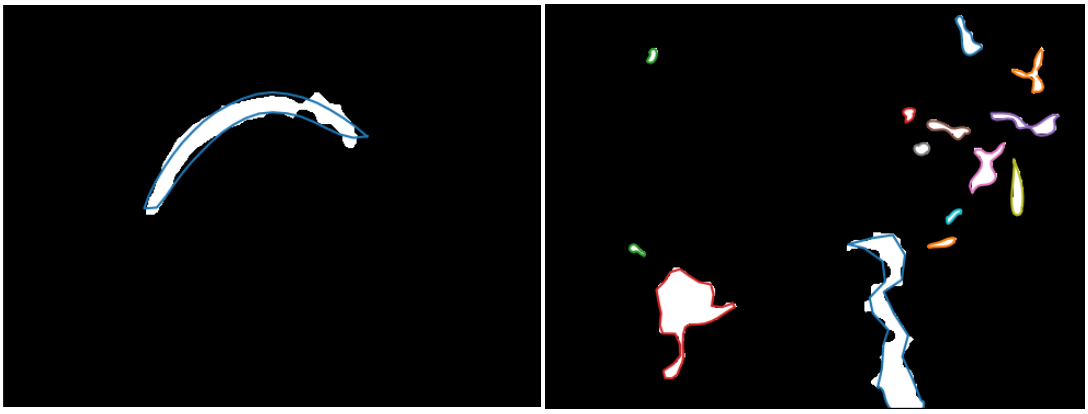


Figure 4.7: Histogram of distances to the closest neighbor in the manifold for validation samples of nematodes shapes using all distances in the cross validation analysis. In red the cumulative distribution function shows that 99% (shown as a dashed green line) of the samples are with a distance below 0.07 (shown as a dashed blue line)



(a) **(b)**

Figure 4.8: Example of binary blob selection using the distance to the closest neighbor in the manifold after performing the geodesic projection. Shapes with a distance to the manifold (a) below the threshold (b) exceeding the threshold.

Table 4.6: Evaluation of the shape selection method based on shapes using as ground-truth the manually labeled data of the validation set

Sequence	Number of Samples	True Positives (TP)	False Negatives (FN)	Error
1	58	52	6	10%
2	15	14	1	6.6%
3	41	37	4	10%
4	16	15	1	6.6%
5	5	5	0	0%
6	43	43	0	0%
7	11	7	0	0%
8	57	51	6	10%
9	13	4	9	70%
10	2	2	0	0%
11	13	13	0	0%
12	37	25	12	32%
13	10	9	1	10%

stage. These results show a limitation of the system in detecting overlapping nematodes, given the training data used for the pixel-wise classification stage.

4.4 Landmark adjustment

The measurement of the deviation between predicted and ground-truth landmarks is performed as shown in Figure 4.9. In principle, the aim is to determine the deviation between the points assigned to the nematode and the manually labeled and interpolated data.

The measurement of the distance d between a landmark of the projected nematode and the ground truth is handled as a point-to-line-segment distance. Let p_i be the i-th landmark of the nematode described by $\mathbf{p}$ positioned by *SegNema*, and let r_j be the closest landmark of the reference nematode $\mathbf{r}$ to p_i. Note that p_i,$r_j \in \mathbb{R}^2$ while $\mathbf{r}$,$\mathbf{p} \in \mathbb{R}^{80}$. The shortest distance from p_i to the reference $\mathbf{r}$ is taken as the shortest distance to either segment $\overline{r_j r_{j+1}}$ or $\overline{r_j r_{j-1}}$. An example of this distance is shown in Figure 4.10

In each sequence there are nematodes of different width. This information is taken into account to determine the most suitable adjustment strategy with respect of the nematode dimensions. The average width for each of the nematodes is shown in Table 4.7.

For each image in the 13 sequences, the statistics of the distance between each landmark and the closest line segment are computed for three cases:

1. After the landmark generation from the binary mask contour and the geodesic projection described in the previous section.

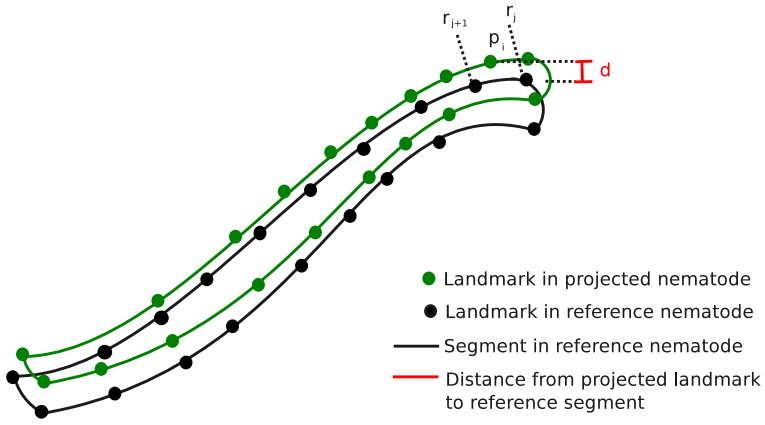


Figure 4.9: Landmark evaluation strategy to measure distance between landmarks predicted made by *SegNema* and landmarks set by an operator. The distance d is calculated from the projected landmarks (in green) and the segment generated from the reference nematode (in black)

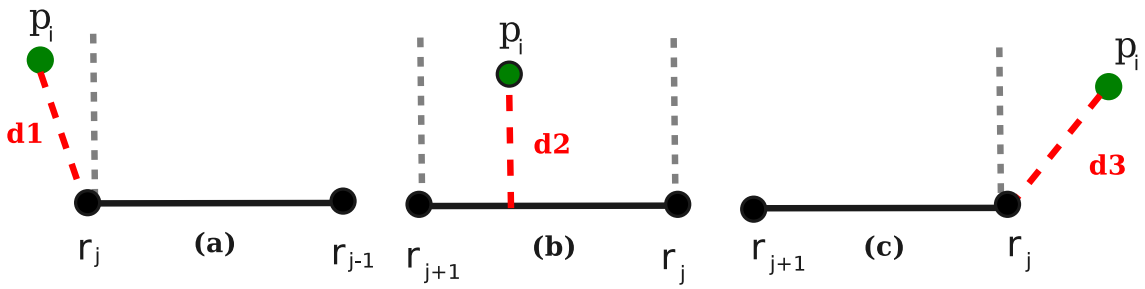


Figure 4.10: Point to line segment shortest distance. (a) and (c) are outside their respective starting and ending perpendiculars and consequently the shortest distance must be calculated from either the start or the end of the line segment. (b) p_i **r** is closest to a point within the segment.

Table 4.7: In each sequence there are nematodes of different width. In the measurements made for the assignment of points this information must be taken into account to determine the possible variations in the assignment of landmarks.

Sequence	Average width (pixels)
1	43.63
2	40.77
3	26.22
4	9.71
5	15.98
6	17.89
7	7.99
8	9.58
9	23.64
10	9.65
11	9.47
12	23.40
13	9.68

2. After performing the landmark adjustment by means of the OGD applied on the image.

3. After performing a landmark adjustment by means of the result of a 1D CNN nematode/background regressor of 37 pixels width.

The average, standard deviation (STD), median, maximum and minimum of the 80 distances from the estimated landmarks to the closest ground-truth line segment are gathered for each image in the validation set. Table 4.8 shows the summary of these statistics for sequence 1 using the alignment in case 1. Meta-statistics (average, STD, median, minimum and maximum) are calculated from average, STD, median, minimum and maximum statistics of all images in the first sequence.

Table 4.8: Meta-statistics of the distances from the estimated landmark positions to the closest ground-truth line segments computed for each image on sequence 1.

		Data Statistics				
		Average	STD (σ)	Median	Max	Min
Meta Statistics	Average	6.21	6.15	4.68	26.21	0.17
	STD (σ)	3.60	4.63	2.65	19.87	0.20
	Median	5.05	5.00	3.97	21.42	0.09
	Max	19.64	27.01	11.18	107.86	0.93
	Min	1.80	1.78	1.34	7.04	0.01

From the computed meta-statistics, the average of average, the median of average, the

average of standard deviation, the median of maxima and median of minima are selected as a criteria to determine the performance of the adjustment strategies. The average of the averages is selected for the estimation of how far the landmarks are from the line segments. As estimate of the deviation of the average distance the average of the standard deviation is used. The average of the median is useful to detect if there are any peaks affecting the average measurement. The median of the maxima indicates a value of the highest distances in the worst case for the landmark assignment prediction. Finally the median of minima allows to compare if the evaluated approach decreases the accuracy on the sequences between adjustment strategies. This information is synthesized on Tables 4.9, 4.10 and 4.11 for the landmark assignment using GP, OGD and 1D CNN approach as a criteria for adjustment respectively.

Table 4.9: Results of pixel distance of landmarks assigned using GP to border segment

Sequence	Average of Average	Median of Average	Average of STD	Median of Max	Median of Min
1	6.21	5.05	6.15	21.42	0.09
2	6.51	6.40	4.95	19.91	0.28
3	3.07	3.11	2.21	8.97	0.09
4	1.90	1.74	1.45	4.16	0.06
5	2.92	3.02	2.08	7.78	0.18
6	2.34	2.32	1.66	6.57	0.07
7	1.64	1.58	1.08	4.26	0.07
8	2.03	2.07	1.33	4.84	0.04
9	4.38	4.20	3.25	10.80	0.11
10	2.15	1.84	4.68	5.19	0.11
11	1.68	1.70	1.05	3.90	0.03
12	4.69	3.98	2.80	10.53	0.20
13	1.97	1.86	1.32	4.86	0.05

Table 4.12 summarizes the best adjustment strategies per sequence given the selected criteria. In most sequences, for both adjustment strategies, the improvement from a first approximation of landmarks based on binary regions and GP is below one pixel in the selected meta statistics. The results are further summarized given the average nematode width per sequence in Table 4.13. It is showed that in the cases were the average nematode width is between 45 and 40 pixels, the 1D CNN adjustment stragegy has the better performance. In sequences were the average nematode width is 26 to 15 pixels, the best adjustment strategy is OGD and between 15 to 9 pixels the aproximation using GP has the best performance. According to the information obtained, in the 13 sequences analized both OGD and GP scored better in 5 sequences, and 1D-CNN scored better in 4.

Figure 4.11 shows examples of the final segmentation results of *SegNema*. Figures 4.11 (b) and 4.11(c) show the capability of *SegNema* of detecting nematode of various sizes on the

Table 4.10: Results of distance of landmarks assigned using gradients as an adjustment criteria to border segment

Sequence	Average of Average	Median of Average	Average of STD	Median of Max	Median of Min
1	6.22	5.32	6.24	22.17	0.10
2	6.34	6.21	4.98	19.11	0.26
3	2.97	2.92	2.27	8.79	0.07
4	1.88	1.77	1.46	3.99	0.08
5	2.87	3.04	2.13	7.86	0.03
6	2.38	2.40	1.72	6.52	0.04
7	1.66	1.57	1.10	4.08	0.07
8	1.95	2.02	1.30	4.74	0.04
9	4.38	4.17	3.40	10.83	0.12
10	2.17	1.39	1.77	5.26	0.12
11	1.77	1.78	1.08	4.27	0.06
12	4.53	3.75	2.79	10.46	0.21
13	1.98	1.91	1.37	5.33	0.04

Table 4.11: Results of distance of landmarks assigned using 1D CNN regressor as an adjustment criteria to border segment

Sequence	Average of Average	Median of Average	Average of STD	Median of Max	Median of Min
1	5.34	4.31	6.03	20.20	0.11
2	6.12	5.93	4.71	16.62	0.19
3	3.20	3.27	2.51	9.63	0.04
4	3.51	2.36	2.85	5.94	0.07
5	3.36	3.16	2.41	8.30	0.05
6	3.21	3.20	2.27	8.51	0.06
7	2.17	2.08	1.42	4.75	0.12
8	2.21	2.16	1.61	5.20	0.04
9	4.57	4.62	2.80	10.77	0.23
10	2.08	1.37	1.80	5.20	0.03
11	2.20	2.13	1.38	4.72	0.10
12	4.06	4.07	2.86	11.65	0.07
13	3.21	2.35	2.34	5.30	0.10

Table 4.12: Best performance adjustment strategy per sequence for the selected criteria. The lowest value for distance in the criteria is selected. If a tie happens more decimal values are used.

Sequence	Best strategy for average of average	Best strategy for median of average	Best strategy for average of STD	Best strategy for Median of Max	Best strategy for Median of Min
1	1D CNN	1D CNN	1D CNN	1D CNN	1D CNN
2	1D CNN	1D CNN	1D CNN	1D CNN	1D CNN
3	OGD	OGD	GP	OGD	1D CNN
4	OGD	GP	GP	OGD	GP
5	OGD	GP	GP	GP	OGD
6	GP	OGD	GP	OGD	OGD
7	GP	OGD	OGD	OGD	OGD
8	OGD	OGD	OGD	OGD	1D CNN
9	GP	OGD	1D CNN	1D CNN	GP
10	1D CNN	1D CNN	OGD	GP	1D CNN
11	GP	GP	GP	GP	GP
12	1D CNN	OGD	OGD	OGD	1D CNN
13	GP	GP	GP	GP	OGD

Table 4.13: Summary of best performance adjustment strategy per sequence for the selected criteria given the width of nematodes in the sequences

Sequence	Average nematode width	Best adjustment strategy
1	43.63	1D CNN
2	40.77	1D CNN
3	26.22	OGD
4	9.71	GP
5	15.98	GP
6	17.89	OGD
7	7.99	OGD
8	9.58	OGD
9	23.64	GP or 1D-CNN
10	9.65	1D CNN
11	9.47	GP
12	23.40	OGD
13	9.68	GP

same execution using the adjustment strategies proposed. Also, it is shown that in some cases such as the red example on Figure 4.11(c), the nematode is partially segmented.

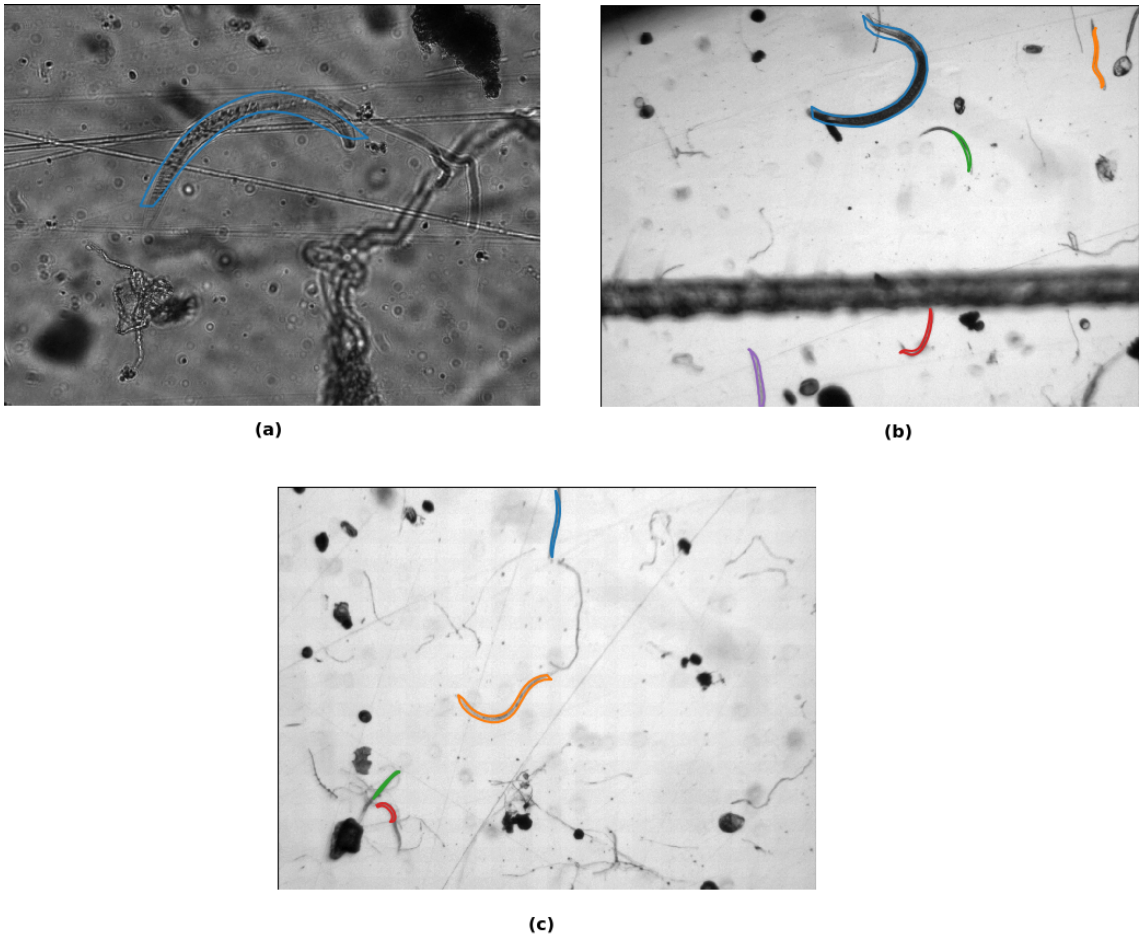


Figure 4.11: *SegNema* results when evaluating a series of images from the test set. Nematodes shapes found are highlighted. (a) nematode detection using 1D-CNN adjustment strategy. (b) nematode detection using OGD adjustment strategy. (b) nematode detection using GD adjustment strategy.

Chapter 5

Conclusions

This work proposes a novel strategy for the segmentation of nematodes in microscopy images, which combines the result of convolutional neural networks with shape models. Landmarks that describe the location of a nematode are positioned based on the network prediction and adjusted to the image contents. They are also restricted to follow a nematode shape. The major challenges of the problem reside in creating a nematode semantic segmentation stage and an initial landmark positioning. In *SegNema*, a *SegNet* based network is used for the segmentation stage, and ADM is used for landmark positioning. With this approach the system was able to detect 87.5% of nematodes marked by an operator on a test set of 321 images taken from 13 microscopy video sequences.

The training strategy, for pixel-wise nematode segmentation consists of using patches of 40×40 pixels taken from the images and binary masks generated from landmarks set by an human operator. This strategy allows to train deep neural networks having under consideration the limited size of the data set available for training. Results shows that in 98% of the 321 images in the test set, the Jaccard index increased after incorporating shape models to discard blobs in the binary classification mask that do not correspond to a nematode shape. In the remaining 2% the Jaccard index remains the same. This is a clear improvement over an segmentation approach consisting only of deep learning encoder/decoder pixel-wise classification stage.

Once the inference results are processed by the shape model stages, on 92.8% of classification results the blobs generated by the segmentation stage have a Jaccard index between 0.50 and 0.99 once the ground truth polygon created by landmarks added by a human operator is compared with the matching blob on the prediction mask. This means that in 92.8% of the cases the classification stage had a similarity score of over 50% with the mask created by a human operator.

The system was able to associate a shape model with the information of an image from a mask generated by means of a deep learning model using sections of the image. This process can be applied to detect other objects besides nematodes. Using an approximation of the skeleton from a thinning process on binary regions, the system was able to assign

a shape model to 87.5% of the nematodes on the test image set. The main causes of error were conditions such as occlusions and overlaps. This limitation seems to be a consequence of the training strategy used and could be resolved if sufficient training data were available that are capable of representing nematodes at additional conditions.

In the cases where the width of the nematode was between 45 and 40 pixels, the best adjustment strategy for ADM corresponds to a 1D CNN. In sequences where the average nematode width is 26 to 15 pixels, the best adjustment strategy is the oriented gradient derivative approach; and between 15 to 9 pixels an approximation using the GP has the best performance. ADM's adjustment strategy was able to maintain shapes within the nematode domain. This is visible in the results obtained in the 1D CNN evaluation where each landmark is given the freedom to move 37 pixels. The landmark adjustment strategies proposed did not improve considerably the nematode classification. Once those landmarks were positioned in the image, in most sequences the improvements oscillate around 1 and 3 pixels. This behavior seems to occur because the geodesic projection makes a truncation that limits the possible representations of nematodes outside the sample taken as a representative manifold. It is recommended to use other truncation strategies that are capable of performing wider representations of the landmark data. The system was not able to detect nematodes in conditions such as occlusions and overlaps. This limitation seems to be a consequence of the training strategy used, and could be resolved with additional training data. The system was able to detect 87.5% of the nematodes marked by an operator on a test image set over 13 different sequences.

Future work

The system was able to detect more than one nematode in images despite their different sizes. This classification information could not be quantitatively evaluated due to the absence of segmented ground truth data of all the nematodes in the images and it is recommended to carry out an evaluation with corresponding additional ground data with the necessary technical criteria.

Due to the modular approach of *SegNema* it is possible to use other networks for the pixel-wise segmentation. Geometric deep learning should be evaluated as a replacement of ADM, in order to explore alternative approaches of the shape generalization. In the proposed approach, the use of morphology operations was prone to cause that nematode pixels were combined with erroneous classifications, which caused the shapes represented by the binary blobs to fall outside the shape manifold. It is recommended to replace morphology operations by MRF or CRF as an alternative strategy that could improve the resolution of the classification edges.

Bibliography

[1] A. Akintayo, G. L. Tylka, A. K. Singh, B. Ganapathysubramanian, A. Singh, and S. Sarkar, "A deep learning framework to discern and count microscopic nematode eggs", in *Scientific Reports*, 2018.

[2] G. Alcantara, "Empirical analysis of non-linear activation functions for deep neural networks in classification tasks", *CoRR*, vol. abs/1710.11272, 2017. arXiv: 1710.11272. [Online]. Available: http://arxiv.org/abs/1710.11272.

[3] K. Alex, S. Ilya, and H. Geoffrey, "ImageNet classification with deep convolutional neural networks", in *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, ser. NIPS'12, Lake Tahoe, Nevada: Curran Associates Inc., 2012, pp. 1097–1105. [Online]. Available: http://dl.acm.org/citation.cfm?id=2999134.2999257.

[4] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation", *CoRR*, vol. abs/1511.00561, 2015. arXiv: 1511.00561. [Online]. Available: http://arxiv.org/abs/1511.00561.

[5] F. Bernard, F. R. Schmidt, J. Thunberg, and D. Cremers, "A combinatorial solution to non-rigid 3D shape-to-image matching", *arXiv preprint arXiv:1611.05241*, 2016.

[6] A. F. Bird and J. Bird, "12 - nematode pathology", in *The Structure of Nematodes (Second Edition)*, A. F. Bird and J. Bird, Eds., Second Edition, San Diego: Academic Press, 1991, pp. 274–300, ISBN: 978-0-12-099651-3. DOI: https://doi.org/10.1016/B978-0-12-099651-3.50018-6. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780120996513500186.

[7] G. Bradski, "The OpenCV Library", *Dr. Dobb's Journal of Software Tools*, 2000.

[8] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, "Geometric deep learning: Going beyond euclidean data", *CoRR*, vol. abs/1611.08097, 2016. arXiv: 1611.08097. [Online]. Available: http://arxiv.org/abs/1611.08097.

[9] J. Canny, "A computational approach to edge detection", *IEEE Transactions on pattern analysis and machine intelligence*, no. 6, pp. 679–698, 1986.

[10] T. Chai and R. R. Draxler, "Root mean square error (RMSE) or mean absolute error (MAE)?", *Geoscientific Model Development Discussions*, vol. 7, pp. 1525–1534, Feb. 2014. DOI: 10.5194/gmdd-7-1525-2014.

[11] S. Chang, Y. Zhang, W. Han, M. Yu, X. Guo, W. Tan, X. Cui, M. Witbrock, M. A. Hasegawa-Johnson, and T. S. Huang, "Dilated recurrent neural networks", in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., Curran Associates, Inc., 2017, pp. 77–87. [Online]. Available: http://papers.nips.cc/paper/6613-dilated-recurrent-neural-networks.pdf.

[12] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs", *CoRR*, vol. abs/1606.00915, 2016. arXiv: 1606.00915. [Online]. Available: http://arxiv.org/abs/1606.00915.

[13] L. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation", *CoRR*, vol. abs/1706.05587, 2017. arXiv: 1706.05587. [Online]. Available: http://arxiv.org/abs/1706.05587.

[14] L. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation", *CoRR*, vol. abs/1802.02611, 2018. arXiv: 1802.02611. [Online]. Available: http://arxiv.org/abs/1802.02611.

[15] F. Chollet *et al.*, *Keras*, https://keras.io, 2015.

[16] G. E. Christensen, "Deformable shape models for anatomy", 1994.

[17] D. C. Ciresan, U. Meier, L. M. Gambardella, and J. Schmidhuber, "Convolutional neural network committees for handwritten character classification", in *2011 International Conference on Document Analysis and Recognition*, 2011, pp. 1135–1139. DOI: 10.1109/ICDAR.2011.229.

[18] T. F. Cootes and C. J. Taylor, "Active shape models-'smart snakes'.", in *BMVC*, vol. 92, 1992, pp. 266–275.

[19] W. Duch and N. Jankowski, "Survey of neural transfer functions", *Neural Computing Surveys*, vol. 2, pp. 163–213, 1999.

[20] Z. Emersic, L. L. Gabriel, V. Struc, and P. Peer, "Pixel-wise ear detection with convolutional encoder-decoder networks", *CoRR*, vol. abs/1702.00307, 2017. arXiv: 1702.00307. [Online]. Available: http://arxiv.org/abs/1702.00307.

[21] M. Fey, J. E. Lenssen, F. Weichert, and H. Müller, "Splinecnn: Fast geometric deep learning with continuous b-spline kernels", *CoRR*, vol. abs/1711.08920, 2017. arXiv: 1711.08920. [Online]. Available: http://arxiv.org/abs/1711.08920.

[22] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, and J. G. Rodrıguez, "A review on deep learning techniques applied to semantic segmentation", *CoRR*, vol. abs/1704.06857, 2017. arXiv: 1704.06857. [Online]. Available: http://arxiv.org/abs/1704.06857.

[23] F. A. Gers and J. Schmidhuber, "Long short-term memory learns context free and context sensitive languages", in *Artificial Neural Nets and Genetic Algorithms*, V. Kůrková, R. Neruda, M. Kárný, and N. C. Steele, Eds., Vienna: Springer Vienna, 2001, pp. 134–137, ISBN: 978-3-7091-6230-9.

[24] O. E. Gomez, "Segmentación de nematodos en imágenes digitales usando redes neuronales", Magister Scientiae, Instituto Tecnológico de Costa Rica, 2009.

[25] A. Greenblum, R. Sznitman, P. Fua, P. Arratia, and J. Sznitman, "Caenorhabditis elegans segmentation using texture-based models for motility phenotyping", vol. 61, pp. 2278–2289, Aug. 2014.

[26] M. Grüner, "Active dictionary models: A framework for non-linear shape modeling", Tesis de maestría, Instituto Tecnológico de Costa Rica, 2015.

[27] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition", *CoRR*, vol. abs/1512.03385, 2015. arXiv: 1512.03385. [Online]. Available: http://arxiv.org/abs/1512.03385.

[28] L. He, X. Ren, Q. Gao, X. Zhao, B. Yao, and Y. Chao, "The connected-component labeling problem: A review of state-of-the-art algorithms", *Pattern Recognition*, vol. 70, pp. 25–43, 2017, ISSN: 0031-3203. DOI: https://doi.org/10.1016/j.patcog.2017.04.018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0031320317301693.

[29] R. J. Hyndman and A. B. Koehler, "Another look at measures of forecast accuracy", *International Journal of Forecasting*, vol. 22, no. 4, pp. 679–688, 2006, ISSN: 0169-2070. DOI: https://doi.org/10.1016/j.ijforecast.2006.03.001. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0169207006000239.

[30] P. Imbach, M. Beardsley, C. Bouroncle, C. Medellin, P. Läderach, H. Hidalgo, E. Alfaro, J. V. Etten, R. Allan, and D. Hemming, *Climate change, ecosystems and smallholder agriculture in Central America: an introduction to the special issue*, 2017.

[31] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift", *CoRR*, vol. abs/1502.03167, 2015. arXiv: 1502.03167. [Online]. Available: http://arxiv.org/abs/1502.03167.

[32] L. C. Jain and L. R. Medsker, *Recurrent Neural Networks: Design and Applications*, 1st. Boca Raton, FL, USA: CRC Press, Inc., 1999, ISBN: 0849371813.

[33] D. Jung, W. Jung, B. Kim, S. Lee, W. Rhee, and J. H. Ahn, "Restructuring batch normalization to accelerate CNN training", *CoRR*, vol. abs/1807.01702, 2018.

[34] N. JW, "Movement and toxicity of nematicides in the plant root-zone", *Fact sheet ENY-041(formerly RF-NG002) U. S. Department of Agricultural, Cooperative Extension Service, University of Florida. Available at: http://edis.ifas.ufl.edu/NG002*, 2002. [Online]. Available: http://edis.ifas.ufl.edu/NG002.

[35] N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang, "On large-batch training for deep learning: Generalization gap and sharp minima", *CoRR*, vol. abs/1609.04836, 2016. arXiv: 1609.04836. [Online]. Available: http://arxiv.org/abs/1609.04836.

[36] A. Khan, I. Khan, and T. Aziz, "A survey on parametric spline function approximation", *Appl. Math. Comput.*, vol. 171, no. 2, pp. 983–1003, Dec. 2005, ISSN: 0096-3003. DOI: 10.1016/j.amc.2005.01.112. [Online]. Available: http://dx.doi.org/10.1016/j.amc.2005.01.112.

[37] J. J. Koenderink, "The structure of images", *Biological cybernetics*, vol. 50, no. 5, pp. 363–370, 1984.

[38] K. Kreutz-Delgado, J. F. Murray, B. D. Rao, K. Engan, T.-W. Lee, and T. J. Sejnowski, "Dictionary learning algorithms for sparse representation", *Neural Computation*, vol. 15, no. 2, pp. 349–396, 2003. DOI: 10.1162/089976603762552951. eprint: https://doi.org/10.1162/089976603762552951. [Online]. Available: https://doi.org/10.1162/089976603762552951.

[39] J. Lafferty, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data", Morgan Kaufmann, 2001, pp. 282–289.

[40] S. Lai, L. Xu, K. Liu, and J. Zhao, "Recurrent convolutional neural networks for text classification", in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, ser. AAAI'15, Austin, Texas: AAAI Press, 2015, pp. 2267–2273, ISBN: 0-262-51129-0. [Online]. Available: http://dl.acm.org/citation.cfm?id=2886521.2886636.

[41] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning", *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[42] Y. LeCun, B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard, and L. D. Jackel, "Handwritten digit recognition with a back-propagation network", in *Advances in neural information processing systems*, 1990, pp. 396–404.

[43] Y. Ma and Y. Fu, *Manifold Learning Theory and Applications*, 1st. Boca Raton, FL, USA: CRC Press, Inc., 2011.

[44] F. Mallard, V. Le Bourlot, and T. Tully, "An automated image analysis system to measure and count organisms in laboratory microcosms", *PLOS ONE*, vol. 8, no. 5, pp. 1–10, May 2013. DOI: 10.1371/journal.pone.0064387. [Online]. Available: https://doi.org/10.1371/journal.pone.0064387.

[45] J. E. Marín, "Ubicación de un nematodo en imágenes digitales utilizando modelos activos de forma", Tesis de licenciatura, Instituto Tecnológico de Costa Rica, 2009.

[46] J. Masci, E. Rodolà, D. Boscaini, M. M. Bronstein, and H. Li, "Geometric deep learning", in *SIGGRAPH ASIA 2016 Courses*, ser. SA '16, Macau: ACM, 2016, 1:1–1:50, ISBN: 978-1-4503-4538-5. DOI: 10.1145/2988458.2988485. [Online]. Available: http://doi.acm.org/10.1145/2988458.2988485.

[47] M. D. Mathew, N. D. Mathew, and P. R. Ebert, "Wormscan: A technique for high-throughput phenotypic analysis of caenorhabditis elegans", *PLOS ONE*, vol. 7, no. 3, pp. 1–6, Mar. 2012. DOI: 10.1371/journal.pone.0033483. [Online]. Available: https://doi.org/10.1371/journal.pone.0033483.

[48] F. Monti, D. Boscaini, J. Masci, E. Rodolà, J. Svoboda, and M. M. Bronstein, "Geometric deep learning on graphs and manifolds using mixture model cnns", *CoRR*, vol. abs/1611.08402, 2016. arXiv: 1611.08402. [Online]. Available: http://arxiv.org/abs/1611.08402.

[49] Oficina IICA Costa Rica, "La agricultura de Costa Rica: Situación al 2010, su evolución y prospectiva", *ICAA: Política de Estado para el Sector Agroalimentario y el Desarrollo Rural costarricense 2010-2021*, 2010.

[50] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition", in *in Conference Record of The Twenty-Seventh Asilomar Conference on Signals, Systems and Computers*, 1993, pp. 1–3.

[51] L. Perez and J. Wang, "The effectiveness of data augmentation in image classification using deep learning", *CoRR*, vol. abs/1712.04621, 2017. arXiv: 1712.04621. [Online]. Available: http://arxiv.org/abs/1712.04621.

[52] G. Priya M.S and K. Nawaz, "Effective morphological image processing techniques and image reconstruction", *IJTRD*, vol. abs/1703.10593, 2017. [Online]. Available: http://www.ijtrd.com/ViewFullText.aspx?Id=8373.

[53] P. Ramachandran, B. Zoph, and Q. V. Le, "Searching for activation functions", *CoRR*, vol. abs/1710.05941, 2017. arXiv: 1710.05941. [Online]. Available: http://arxiv.org/abs/1710.05941.

[54] R. Real and J. M. Vargas, "The probabilistic basis of jaccard's index of similarity", *Systematic Biology*, vol. 45, no. 3, pp. 380–385, 1996, ISSN: 10635157, 1076836X. [Online]. Available: http://www.jstor.org/stable/2413572.

[55] R. Rubinstein, T. Faktor, and M. Elad, "K-svd dictionary-learning for the analysis sparse model", in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Mar. 2012, pp. 5405–5408. DOI: 10.1109/ICASSP.2012.6289143.

[56] D. Scherer, A. Müller, and S. Behnke, "Evaluation of pooling operations in convolutional architectures for object recognition", in *Proceedings of the 20th International Conference on Artificial Neural Networks: Part III*, ser. ICANN'10, Thessaloniki, Greece: Springer-Verlag, 2010, pp. 92–101. [Online]. Available: http://dl.acm.org/citation.cfm?id=1886436.1886447.

[57] C. E. Shannon, "A mathematical theory of communication", *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 5, no. 1, pp. 3–55, Jan. 2001, ISSN: 1559-1662. DOI: 10.1145/584091.584093. [Online]. Available: http://doi.acm.org/10.1145/584091.584093.

[58] D. I. Shapiro-Ilan, E. E. Lewis, J. F. Campbell, and D. B. Kim-Shapiro, "Directional movement of entomopathogenic nematodes in response to electrical field: Effects of species, magnitude of voltage, and infective juvenile age", *Journal of Invertebrate Pathology*, vol. 109, no. 1, pp. 34–40, 2012, ISSN: 0022-2011. DOI: https://doi.org/10.1016/j.jip.2011.09.004. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0022201111002035.

[59] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation", *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 640–651, Apr. 2017, ISSN: 0162-8828. DOI: 10.1109/TPAMI.2016.2572683. [Online]. Available: https://doi.org/10.1109/TPAMI.2016.2572683.

[60] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition", *CoRR*, vol. abs/1409.1556, 2014. arXiv: 1409.1556. [Online]. Available: http://arxiv.org/abs/1409.1556.

[61] P. Soille, *Morphological Image Analysis: Principles and Applications*, 2nd ed. Berlin, Heidelberg: Springer-Verlag, 2003, ISBN: 3540429883.

[62] S. Soni, D. R. Chadha, and S. Kaur, "Paper on thinning of image using zhang and suen algorithm and neural network", 2016.

[63] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting", *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, Jan. 2014, ISSN: 1532-4435. [Online]. Available: http://dl.acm.org/citation.cfm?id=2627435.2670313.

[64] M. B. Stegmann and D. D. Gomez, "A brief introduction to statistical shape analysis", *Informatics and Mathematical Modelling, Technical University of Denmark, DTU*, p. 15, Mar. 2002. [Online]. Available: http://www2.imm.dtu.dk/pubdb/p.php?403.

[65] M. Sun, Z. Song, X. Jiang, J. Pan, and Y. Pang, "Learning pooling for convolutional neural network", *Neurocomput.*, vol. 224, no. C, pp. 96–104, Feb. 2017, ISSN: 0925-2312. DOI: 10.1016/j.neucom.2016.10.049. [Online]. Available: https://doi.org/10.1016/j.neucom.2016.10.049.

[66] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions", *CoRR*, vol. abs/1409.4842, 2014. arXiv: 1409.4842. [Online]. Available: http://arxiv.org/abs/1409.4842.

[67] F. Al-Tam, A. Dos Anjos, S. Bellafiore, and H. Shahbazkia, "Detection of root knot nematodes in microscopy images", *BIOIMAGING 2015 - 2nd International Conference on Bioimaging, Proceedings; Part of 8th International Joint Conference on Biomedical Engineering Systems and Technologies, BIOSTEC 2015*, pp. 76–81, 2015.

[68] M. Taylor, "Estrategia de ajuste de formas explícitas a imágenes con estructuras vermiformes.", Tesis de Licenciatura, Instituto Tecnológico de Costa Rica, 2017.

[69] J. B. Tenenbaum, V. d. Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction", *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000, ISSN: 0036-8075. DOI: 10.1126/science.290.5500.2319. eprint: http://science.sciencemag.org/content/290/5500/2319.full.pdf. [Online]. Available: http://science.sciencemag.org/content/290/5500/2319.

[70] M. Thoma, "A survey of semantic segmentation", *CoRR*, vol. abs/1602.06541, 2016. arXiv: 1602.06541. [Online]. Available: http://arxiv.org/abs/1602.06541.

[71] I. Tosic and P. Frossard, "Dictionary learning: What is the right representation for my signal?", *IEEE Signal Processing Magazine*, vol. 28, no. 2, pp. 27–38, 2011.

[72] L. J. Van Vliet, I. T. Young, and P. W. Verbeek, "Recursive gaussian derivative filters", in *Pattern Recognition, 1998. Proceedings. Fourteenth International Conference on*, IEEE, vol. 1, 1998, pp. 509–514.

[73] S. van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, T. Yu, and the scikit-image contributors, "Scikit-image: Image processing in Python", *PeerJ*, vol. 2, e453, Jun. 2014, ISSN: 2167-8359. DOI: 10.7717/peerj.453. [Online]. Available: http://dx.doi.org/10.7717/peerj.453.

[74] A. Witkin, "Scale-space filtering: A new approach to multi-scale description", in *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'84.*, IEEE, vol. 9, 1984, pp. 150–153.

[75] W. Zaremba, I. Sutskever, and O. Vinyals, "Recurrent neural network regularization.", *CoRR*, vol. abs/1409.2329, 2014. [Online]. Available: http://dblp.uni-trier.de/db/journals/corr/corr1409.html#ZarembaSV14.

[76] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks", *CoRR*, vol. abs/1311.2901, 2013. arXiv: 1311.2901. [Online]. Available: http://arxiv.org/abs/1311.2901.

[77] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning requires rethinking generalization", *CoRR*, vol. abs/1611.03530, 2016. arXiv: 1611.03530. [Online]. Available: http://arxiv.org/abs/1611.03530.

[78] T. Y. Zhang and C. Y. Suen, "A fast parallel algorithm for thinning digital patterns", *Commun. ACM*, vol. 27, no. 3, pp. 236–239, Mar. 1984, ISSN: 0001-0782. DOI: 10.1145/357994.358023. [Online]. Available: http://doi.acm.org/10.1145/357994.358023.

[79] J. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks", *CoRR*, vol. abs/1703.10593, 2017. arXiv: 1703.10593. [Online]. Available: http://arxiv.org/abs/1703.10593.