Deep Learning approaches for Head and Operculum Segmentation in Zebrafish Microscopy Images

Navdeep Kumar[1], Alessio Carletti[2], Paulo J. Gavaia[2], Marc Muller[3], Leonor Cancela[2], Pierre Geurts[1], and Raphaël Marée[1]

[1] Montefiore Institute, Dept. EE & CS, University of Liège, Belgium
[2] Centre of Marine Sciences (CCMAR), University of Algarve, Portugal
[3] Laboratory of Organogenesis and Regeneration (GIGA Research), University of Liège, Belgium

Abstract. In this paper, we propose variants of deep learning methods to segment head and operculum of the zebrafish larvae in microscopic images. In the first approach, we used a three-class model to jointly segment head and operculum area of zebrafish larvae from background. In the second, two-step, approach, we first trained binary segmentation model to segment head area from the background followed by another binary model to segment the operculum area within cropped head area thereby minimizing the class imbalance problem. Both of our approaches use a modified, simpler, U-Net architecture, and we also evaluate different loss functions to tackle the class imbalance problem. We systematically compare all these variants using various performance metrics. Data and open-source code are available at https://uliege.cytomine.org.

Keywords: Image segmentation · Deep Learning · Zebrafish images

1 Introduction

Biomedical research heavily uses Zebrafish (*Danio rerio*) as a model to study developmental processes. In the earlier stage of their lifecycle, zebrafish embryos and larvae are completely transparent, which greatly facilitates monitoring of their developmental organs such as operculum and vertebral column using microscopy techniques [5, 10, 3]. Biomedical researchers also rely on microscopy to study the effects of various chemical compounds on the developing parts of the fish model in toxicological studies [1]. Such analyses often involve segmenting different categories of regions of interest (ROI) within images in order to quantify their morphological changes. For example, the analysis of *Head* and *Operculum* (a series of bone) regions of Zebrafish larvae and the quantification of the operculum-to-head ratio is considered as a good marker of increased bone formation and mineralization and it is a validated method to screen for bioactive compounds which have effects on bones [12][7]. It also gives an additional information on the possible toxicity of a compound at the organism level. However,

the visual examination and area quantification are a bottleneck and prevent applying such a workflow at high throughput.
In this paper, supervised deep learning strategies are proposed and evaluated to segment head and operculum regions, as evaluation of such approaches has not been proposed previously. We describe image acquisition settings, our dataset, and our methods in Section 2, and we present our results in Section 3.

2 Methodology

In the section, we describe image acquisition procedure and dataset description followed by two deep learning strategies and provide more details about convolutional neural network (CNN) architectures that have been used to segment head and operculum areas.

2.1 Image Acquisition and Dataset Description

Zebrafish larvae stained with alizarin red S were imaged using a MZ 7.5 fluorescence stereomicroscope (Leica, Wetzlar, Germany) equipped with a green light filter (λex=530–560 nm and λem=580 nm) and a black-and-white F-View II camera (Olympus, Hamburg, Germany). Images were acquired using the following parameters: exposure time 1 s, gamma 1.00, image format 1376x1032 pixels, binning 1x1. For morphometric analysis, color channels of the RGB images were split. Red channel (8-bit) images were used for further analyses.
We follow a supervised deep learning approach that requires original images and corresponding head and operculum ground-truth masks to design and validate the approach. Our dataset consists of 8-bit single channel (red channel) fluorescence images of zebrafish larvea at 6 days post fertilization (dpf). Red channel fluorescence images were first transformed into greyscale images (with contrast and brightness enhancement) to ease the manual annotations by experts of head and operculum areas. Manual annotations (illustrated in Figure 1) consist of contours of head area and operculum area as the main objective is to compute the operculum-to-head ratios for different experimental conditions. A total of 2293 zebrafish images of 1376×1032 resolution have been collected and manually annotated over a period of one year. The dataset consists of 28 different sets of experiments using 5 different compounds, to analyse their effect on the operculum of the zebrafish larvea. Each set has been acquired with the same acquisition settings. Manual annotations were imported into Cytomine open-source software [9] to centralize data and ease binary masks creation to be further used as inputs of deep learning algorithms.

2.2 Two deep learning strategies

One-step segmentation with a three-class model. Following this strategy, original size images without cropping are used. Since typical CNN networks require input images of small size (see below), original sized images are first

downsized to the size required by the network, keeping their original aspect ratio to avoid any kind of distortions in the predictions, while upsizing the predicted masks. Since our images are rectangular but network require square images, we padded the rectangular images with zeros to make them square. A three-class output segmentation model is then trained to segment both head and operculum from background areas as illustrated in Fig. 1 (top).

Two-step segmentation with two binary models. In this approach, a first binary segmentation model is trained to segment the head from the background in original full images downsized appropriately (as in the three-class approach). A second binary segmentation model is trained to segment the operculum area using resized cropped images (rectangular box around the head). At prediction phase, the first model is applied to segment the head, then a rectangular bounding box is automatically extracted. Using these box coordinates, we apply the second model to the resized cropped images (around the head) to segment the operculum area. The two-step approach is illustrated in Fig. 1 (bottom).

2.3 U-Net Implementation

For both approaches, the U-Net architecture [11] has been adapted to segment areas of the zebrafish larvea. The main idea of U-Net is its two parts: the convolution (encoder) or contracting operations, and deconvolutional (decoder) or expanding operations. In the first part, convolutional operations are applied in successive layers with the max pooling operations at the end of each layer, thereby contracting the input resolution. In the second part, an expanding resolution path is adopted using upsampling or deconvolutional layers. The first part is considered as a traditional stack of convolutional and max pooling layers to capture context information within the image. In the second part, deconvolutional operations are applied along a symmetric expanding path to capture the precise localized information. One more important thing about this architecture is its symmetric concatenation of the previous activations from the first part to the activations of the second part.
As preliminary results with the original U-Net architecture on the training set were unsatisfactory (including a tendency to predict only the majority class, i.e. the background), we implemented some modifications in U-Net architecture including the input size and output size of the network and number of layers and filters. Fig. 2 shows our "modified U-Net" network architecture.

In our experiments, we used two versions of modified U-Net, one that accepts 512×512 images as input and another that accepts 256×256 images. In both the cases, the output size of the masks is same as the input size whereas in [11], authors used 572×572 inputs and 388×388 outputs. The reason behind using two variants of the network is to assess whether using less parameters will negatively impact recognition performance. Using smaller networks indeed reduces execution times which can be useful in real-time applications. With the small size variant of the U-Net architecture (with 256×256 input size), we used fewer filters in

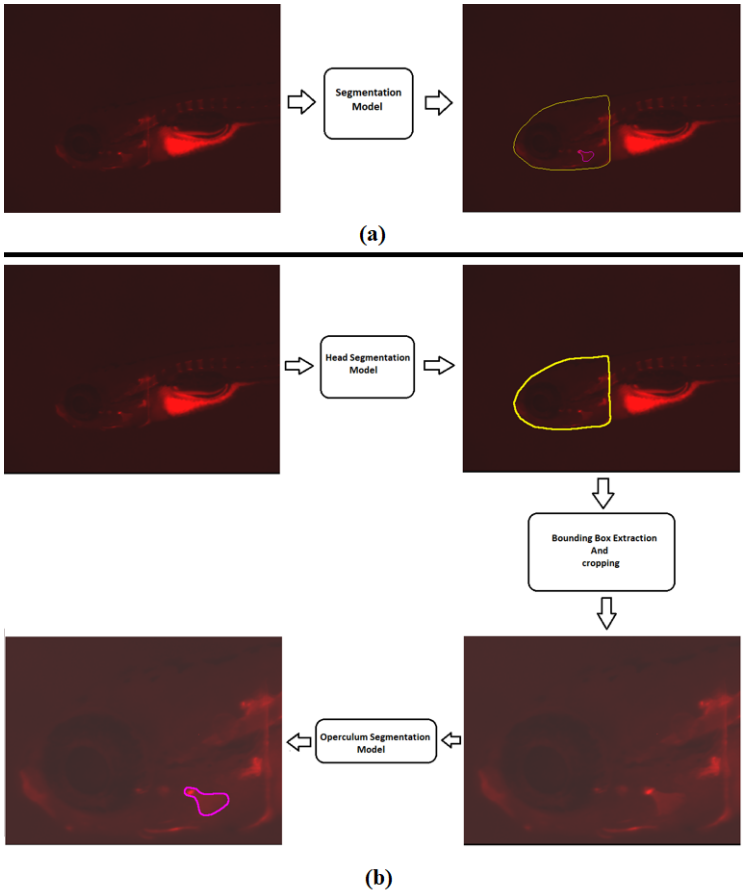

Fig. 1. (a) One-step segmentation approach with three classes: head (yellow contour), operculum (pink contour), background. (b) Two-step binary segmentation approach with a first binary model (head vs background) followed by a second binary model (operculum vs other).

each convolutional block as compared to the larger network thereby reducing the network size and the number of parameters by 5 folds. For better optimization, we used "Adam" [6] optimizer and batch normalization in each convolutional block before max pooling. Adam uses *gradient descent with momentum* combined with an adaptive learning rate using exponential moving averages, which makes it more computationally and memory efficient than "Stochastic Gradient Descent" used in the original *U-Net* paper. During training, we also used data augmentation (random flips and rotations, brightness, and contrast changes). We implemented these networks in Python using Tensorflow and Keras [2].

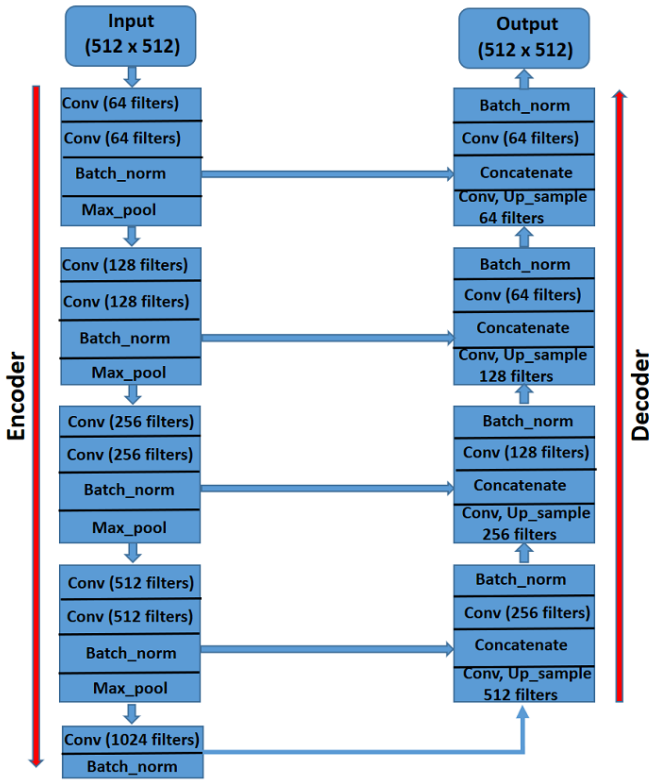

Fig. 2. Modified U-Net architecture used in experiments for 512×512 sized images.

2.4 Experimental protocol

We first split the dataset into 2105 images for training and 188 images for final evaluation. To assess variability, the set of 2105 images is split into five equally sized folds. Each fold is used in turn as the validation set and the remaining folds as the training set. Five models are trained independently on each training set for 1000 epochs and the five best models on their corresponding validation set across the epochs are finally retained as the final models. In addition, we used early stopping, which forces the training to stop when there is no improvement in the training loss for 100 consecutive epochs. We report below the average performance and standard deviation of these five models estimated on the test set.

2.5 Model Training with different loss functions

In both approaches, we used deep learning based semantic segmentation approach in which a model predicts the class of every pixel in the image (dense

predictions). In such a setting, we are faced with a problem of class imbalance as less than 2% of the image area is occupied by operculum region while around 90% is background region. In such situation, the contribution of the majority class (in our case, the background) in the loss during training is more important than that of the minority class, which biases the model in favor of the majority class while ignoring minority class. While the two-step approach tends to reduce this phenomenon (by cropping then predicting operculum only within the head region), a certain class imbalance still persists. Therefore, for both approaches we propose to evaluate different loss functions during training to handle class imbalance. Namely, we evaluated the Cross Entropy Loss, Dice Loss, Tversky Loss, Focal Loss and Jaccard Loss [8].

3 Results and Discussion

Tables 1 and 3 show the results of the first (*three-class* segmentation) approach whereas tables 2 and 4 of the second (*two-step binary class segmentation*) approach using 256×256 input size and 512×512 input size networks, respectively. In both the variants, we report several performance metrics that take into account class imbalance, namely *Precision, Recall*, *F1 Score* and *Dice score*, computed at the pixel level and averaged over the 5 models. To get a single score with which to compare the models, the Dice score is further averaged over head and operculum. Its standard deviation over the 5 models is also provided to assess variability.

Table 1. Segmentation results with the one-step, three-class approach using different loss functions for input size 256.

Avg. scores with three-class output based segmentation					
Loss function	Class	Precision	Recall	F1 Score	Dice Score±Std.
Cross Entropy	Head	0.9806	0.9796	0.9801	0.9412±0.0043
	Operculum	0.8780	0.9263	0.9014	
Tversky loss	Head	0.9779	0.9806	0.9792	0.9470±0.0017
	Operculum	0.9086	0.9190	0.9136	
Dice loss	Head	0.9819	0.9806	0.9813	0.9462±0.0024
	Operculum	0.9120	0.9092	0.9106	
Jaccard Loss	Head	0.9678	0.9789	0.9733	0.49±0.0002
	Operculum	0.0	0.0	0.0	
Focal loss	Head	0.9820	0.9798	0.9809	0.9442±0.0046
	Operculum	0.9060	0.9076	0.9066	

In the three-class approach, the Tversky Loss seems to better cope with the strong class imbalance in both 512×512 and 256×256 settings. The worst performer in the three-class approach is Jaccard loss as it only predicts the majority class (90% background) and no operculum area. This loss leads however to good predictions with the two-step binary approach in both input size settings.

Table 2. Segmentation results with the two-step, binary approach using different loss functions for input size 256.

Avg. scores with two binary-class output based segmentation					
Loss function	Class	Precision	Recall	F1 Score	Dice Score±Std.
Cross Entropy	Head	0.9832	0.9805	0.9819	0.9540±0.0015
	Operculum	0.9196	0.9340	0.9267	
Tversky loss	Head	0.9824	0.9806	0.9815	0.9524±0.0024
	Operculum	0.9104	0.9374	0.9237	
Dice loss	Head	0.9828	0.9826	0.9827	0.9511±0.0046
	Operculum	0.9175	0.9276	0.9225	
Jaccard Loss	Head	0.9782	0.9826	0.9804	0.9513±0.0012
	Operculum	0.9124	0.9355	0.9238	
Focal loss	Head	0.9835	0.9815	0.9825	0.9516±0.0018
	Operculum	0.9213	0.9261	0.9236	

Table 3. Segmentation results with the one-step, three-class approach using different loss functions for input size 512.

Avg. scores with three-class output based segmentation					
Loss function	Class	Precision	Recall	F1 Score	Dice Score±Std.
Cross Entropy	Head	0.9815	0.9747	0.9781	0.9358±0.0064
	Operculum	0.8992	0.8934	0.8953	
Tversky loss	Head	0.9812	0.9789	0.9800	0.95±0.0011
	Operculum	0.9090	0.9308	0.9196	
Dice loss	Head	0.9822	0.9744	0.9783	0.9428±0.0043
	Operculum	0.9085	0.9066	0.9074	
Jaccard Loss	Head	0.9678	0.9789	0.9733	0.49±0.0002
	Operculum	0.0	0.0	0.0	
Focal loss	Head	0.9817	0.9768	0.9792	0.9364±0.007
	Operculum	0.9078	0.8846	0.8946	

Table 4. Segmentation results with the two-step, binary approach, using different loss functions for input size 512.

Avg. scores with two binary-class output based segmentation					
Loss function	Class	Precision	Recall	F1 Score	Dice Score±Std.
Cross Entropy	Head	0.9840	0.9780	0.9810	0.9189±0.0159
	Operculum	0.9114	0.8428	0.8747	
Tversky loss	Head	0.9832	0.9785	0.9808	0.9505±0.0024
	Operculum	0.9223	0.9245	0.9234	
Dice loss	Head	0.9828	0.9797	0.9812	0.9424±0.0057
	Operculum	0.9256	0.8947	0.9097	
Jaccard Loss	Head	0.9818	0.99796	0.9807	0.9516±0.002
	Operculum	0.9178	0.9311	0.9244	
Focal loss	Head	0.9841	0.9732	0.9786	0.9490±0.0031
	Operculum	0.9207	0.9227	0.9217	

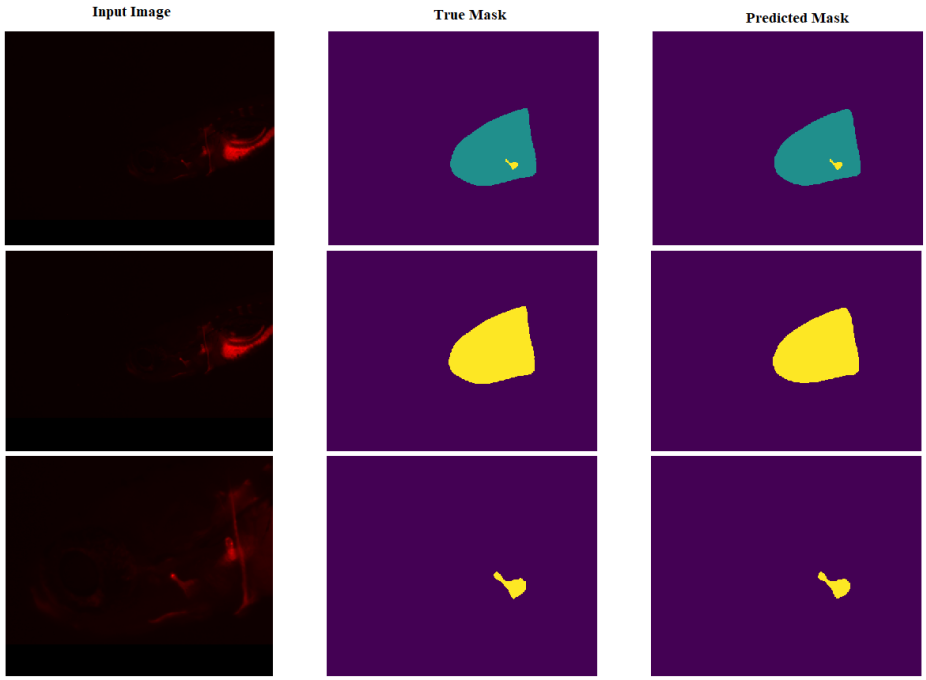

Fig. 3. Sample predictions with best performer on test images with three class (top row) and two-step binary class (last two rows). From the first to third column: input Image, true mask, predicted mask.

In the two-step binary segmentation approach, all losses are very close except cross entropy in 512×512 setting. Overall, the two-step approach for 512×512 inputs with Jaccard loss has a slight edge over other losses. We believe that the improved performance of the two-step approach is due to the fact that the second segmentation model works with a cropped, head-focused, dataset. Because of the cropping, the class imbalance is not as severe and the operculum image is not downscaled as much as with the three-class approach. Predictions are thus more precise and less influenced by the class imbalance. Regarding the two input sizes, we see that they lead to almost identical performance in terms of Dice Score. Sample predictions from the best performing models are shown in Fig. 3.

3.1 Robustness to image acquisition with another microscope

In practice, microscopes with different acquisition settings might be used overtime by biomedical researchers which raises the issue of robustness of segmentation models to such variabilities, an issue known as domain shift [4]. As a first step towards robustness evaluation, we applied our best two-step binary approach on additional, unlabeled, images acquired with another microscope namely Leica

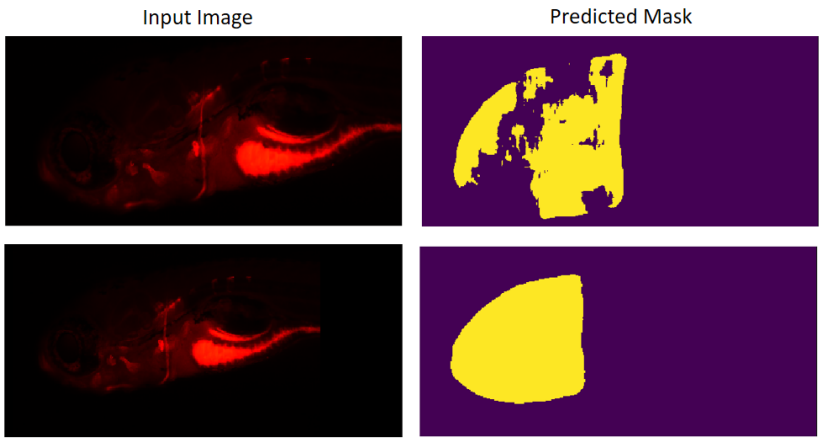

Fig. 4. Robustness evaluation: Predictions from best model using two-step binary class approach on a new image acquired with another microscope before pre-processing (first row), and after pre-processing (second row).

MZ10F fluorescence stereomicroscope equipped with a green fluorescence filter (λex= 546/10 nm), a barrier filter (λem =590 nm) and a DFC7000T camera (Leica, Wetzlar, Germany) with a different output image size (1920 $\times$ 1440). When run on these unprocessed new images, we observe that the performance of our model declines, as illustrated by Fig. 4 (first row). We hypothesized that this is due to the fact that, in the new microscope setting, ROIs (fish head and operculum) are larger in proportion to the size of the full image as compared to ROIs in the original training images. To address this issue, we applied a very simple *pre-processing* step to reduce the scale proportion of ROIs in the image. First, we downscaled the new images to the resolution of the original images (i.e., from 1920×1440 down to 1376×1032) keeping the same aspect ratio. We then centered the resulting 1376×1032 image into a 1920×1440 image, filling the new pixels with zeros. Figure 4 (second row) illustrates the positive effect of this pre-processing on the prediction. Note that downscaling further the image in the first step does not seem to affect the performance. We hypothesized that this is due to the use of pooling layers that makes network features somewhat scale invariant (in the direction of a decrease of resolution at least). In practice, this scale calibration step would require a human expert to manually draw a rectangle around the head within a single image when a new microscope is used to initiate the automatic rescaling for the whole set of new images (so that the bounding box is rescaled down to the average size of the head in the learning set images). We consider this manual intervention to be acceptable.

4 Conclusions

We have evaluated deep learning based semantic segmentation variants on a new dataset of more than two thousands fluorescent microscopy images of Zebrafish larvae where the goal is to quantify operculum-to-head ratio. The dataset and prediction code compatible with Cytomine open-source web platform [9] is available to foster further research and to enable biomedical experts to routinely use our developments and proofread predictions. We plan to use such developments as the basis of large-scale morphological studies where the effects of different concentrations of many compounds on bone formation and mineralization will be evaluated thoroughly using various statistics (such as operculum-to-head ratio) derived from predicted masks. In the future, it may be necessary to investigate more advanced approaches for other image variations due to change of acquisition setting but ours was sufficient on the new microscope used by our collaborators.

Acknowledgments. This work, as well as N.K. and A. C. are supported by the EU MSCA-ITN project BioMedAqu (766347). R.M. was partially supported by ADRIC Wallonia Grant and EU IMI BIGPICTURE grant. M.M. is a "Maître de Recherche" at the Fund for Scientific Research (F.R.S.–FNRS) .

References

1. Cassar, S., Adatto, I., Freeman, J.L., Gamse, J.T., Iturria, I., Lawrence, C., Muriana, A., Peterson, R.T., Van Cruchten, S., Zon, L.I.: Use of zebrafish in drug discovery toxicology. Chemical research in toxicology **33**(1), 95–118 (2019)
2. Chollet, F., et al.: Keras (2015), https://github.com/fchollet/keras
3. Evans, J.G., Matsudaira, P.: Linking microscopy and high content screening in large-scale biomedical research. High Content Screening pp. 33–38 (2007)
4. Guan, H., Liu, M.: Domain adaptation for medical image analysis: A survey. arXiv:2102.09508 (2021)
5. Hill, A.J., Teraoka, H., Heideman, W., Peterson, R.E.: Zebrafish as a model vertebrate for investigating chemical toxicity. Toxicological sciences **86**(1), 6–19 (2005)
6. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv:1412.6980 (2014)
7. Lessman, C.A.: The developing zebrafish (danio rerio): A vertebrate model for high-throughput screening of chemical libraries. Birth Defects Research Part C: Embryo Today: Reviews **93**(3), 268–280 (2011)
8. Ma, J., Chen, J., Ng, M., Huang, R., Li, Y., Li, C., Yang, X., Martel, A.L.: Loss odyssey in medical image segmentation. Medical Image Analysis **71**(102035) (2021)
9. Marée, R., Rollus, L., Stévens, B., Hoyoux, R., Louppe, G., Vandaele, R., Begon, J.M., Kainz, P., Geurts, P., Wehenkel, L.: Collaborative analysis of multi-gigapixel imaging data using cytomine. Bioinformatics **32**(9), 1395–1401 (2016)
10. Mikut, R., Dickmeis, T., Driever, W., Geurts, P., Hamprecht, F.A., Kausler, B.X., Ledesma-Carbayo, M.J., Marée, R., Mikula, K., Pantazis, P., et al.: Automated processing of zebrafish imaging data: a survey. Zebrafish **10**(3), 401–421 (2013)
11. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. pp. 234–241 (2015)

12. Tarasco, M., Laizé, V., Cardeira, J., Cancela, M.L., Gavaia, P.J.: The zebrafish operculum: A powerful system to assess osteogenic bioactivities of molecules with pharmacological and toxicological relevance. Comparative Biochemistry and Physiology Part C: Toxicology & Pharmacology **197**, 45–52 (2017)