A Framework for Age Estimation of Fish from Otoliths: Synergy Between RANSAC and Deep Neural Networks

Souleymane KONE, Abdoulaye SERE, Dekpeltakié Augustin METOUALE SOMDA,
José Arthur OUEDRAOGO
Laboratoire d'Algèbre de Mathématiques Discrètes et Informatique (LAMDI),
Université Nazi BONI, Burkina Faso Equipe Signal, Image et Communications (SIC)

***Abstract*—This study represents a significant advancement in fish ecology by applying deep learning techniques to automate and improve the counting of growth rings in otoliths, which are essential for determining the age and growth patterns of fish. Traditionally, manual methods have been used to analyze these rings, but these approaches are time-consuming, require significant expertise, and are prone to bias. To address these limitations, we propose a novel methodology that combines convolutional neural networks (CNNs) with the RANSAC algorithm, enhancing the accuracy and reliability of ring detection, even in the presence of noise or natural image variations. Unlike manual techniques, which depend on observer expertise and subjective interpretation, our approach improves performance, often surpassing human experts while reducing analysis time. The results demonstrate the potential of deep learning and RANSAC in otolith research, offering powerful tools for sustainable fish population management and transforming research practices in marine ecology by providing faster, more reliable, and accessible analytical methods, setting new standards for more rigorous research.**

***Keywords*—Otoliths; deep learning; pattern recognition; RANSAC; automated counting**

I. Introduction

The advent of deep learning, a subset of artificial intelligence, offers a promising solution to the limitations of traditional methods for counting growth rings in otoliths. These growth rings, critical for estimating fish age and growth, have long been analyzed using manual methods that are time-consuming, reliant on human expertise, and subject to observer bias [1], [2], [3], [4], [5]. By employing convolutional neural networks (CNNs) trained on extensive datasets of annotated otolith images, researchers can now develop models capable of automatically identifying and counting growth rings with high precision and efficiency. This approach represents a major advancement, building on traditional methods documented in works such as the Manuel de sclérochronologie des poissons [6].

Automated models not only improve the accuracy of ring counting but also drastically accelerate the analysis of large datasets, making it possible to study fish populations over broader geographic areas and longer time periods. Such scalability enhances our understanding of aquatic ecosystems, enabling better monitoring and management of fishery resources, and contributing to the conservation of fish populations. These advancements align with findings from the Project DEMERSTOCK, which emphasize the need for innovative methods to improve fish age estimation [7].

CNNs, when trained on large collections of annotated otolith images, achieve accuracy levels that rival, and often surpass, those of human experts. Furthermore, the computational efficiency of CNN-based analysis far exceeds traditional methods, particularly in terms of the time required to process large datasets. These technological advancements are critical for addressing challenges in fish population studies and have been supported by the application of deep learning in analyzing otolith striations, as explored in studies on anchovy (Engraulis encrasicolus) [5].

To fully capitalize on the potential of deep learning, it is essential to establish standardized procedures for otolith image collection, preparation, and annotation. Rigorous cross-validation protocols are also needed to ensure that the models generalize across different species and environmental conditions, as highlighted in studies emphasizing the importance of consistency in fish age estimation methodologies [6], [7].

Beyond otolith analysis, deep learning techniques hold promise for a wide range of applications in fish biology, including species identification, food web analysis, and population health monitoring. Integrating these computational techniques into fish ecology represents a significant leap forward, offering unprecedented opportunities for scaling and precision in ecological studies. As highlighted by Gonzalez [8], the adoption of deep learning in aquatic research is paving the way for more efficient, accurate, and large-scale studies, which are crucial for sustainable marine resource management and biodiversity conservation.

In our study, we propose an innovative method for automating the counting of growth rings in otoliths using deep learning. Our approach employs an enhanced CNN architecture trained on a curated dataset of tilapia otolith images. The model's performance was rigorously evaluated by comparing its predictions to manual counts performed by expert specialists. The results demonstrate that the model achieves accuracy comparable to, or surpassing, that of human experts, while significantly reducing the time required for analysis.

This deep learning-based framework offers a reliable and efficient alternative to traditional manual methods for otolith analysis. By addressing key limitations of existing approaches,

it provides a powerful tool for large-scale ecological studies and fishery management, contributing to improved practices in aquatic resource conservation and sustainable marine management [5], [6], [7].

II. Preliminaries

The analysis of otolith images for determining fish age and growth has experienced renewed interest in recent years, largely attributed to advancements in deep learning and image processing methodologies. Convolutional Neural Networks (CNNs) have demonstrated exceptional effectiveness in the automated identification and quantification of growth rings within otoliths, thereby offering a viable alternative to conventional manual techniques.

For example, Sert in [9] introduced a technique that employs CNNs for the automatic estimation of fish age from otolith imagery. The authors developed a model that can detect and count growth rings with an accuracy comparable to that of human specialists. In addition, Wang in [10] investigated the use of deep learning for the automatic estimation of fish age, creating an optimized CNN model that exceeds traditional methods in both accuracy and efficiency.

Moreover, Liu in [11] suggested an approach based on transfer learning for estimating fish age. By applying a CNN model pre-trained on natural images, they tailored it for the analysis of otoliths, resulting in encouraging outcomes. Furthermore, Zhang in [12] introduced a multi-scale CNN model that utilizes convolutions at varying scales to capture features across different resolutions, thus enhancing the accuracy of age estimation.

Finally, a study by Sun in [13] presents a method that combines deep learning with data augmentation to improve the robustness and precision of fish age estimation. Data augmentation is instrumental in expanding the training dataset, thereby enhancing the model's resilience to variations in images.

In conclusion, the recent progress in applying CNNs and deep learning techniques for otolith analysis presents promising opportunities for automated fish age estimation, yielding significant results in terms of both precision and efficiency.

III. Methodology

The methodology proposed in this study is a framework. In the following sections, we will provide a detailed description of the implementation steps.

A. Framework

In our proposed framework, we primarily have three components that we will describe. Fig. 1 illustrates the representation of our proposed framework.

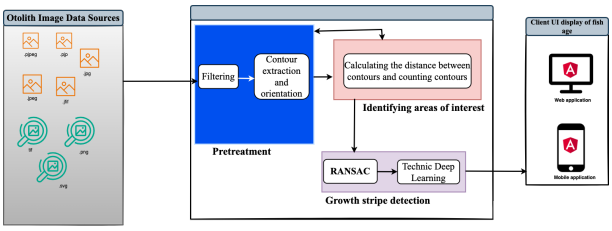

Fig. 1. Framework.

- First component: Otolith Image Data Sources.
  In this section, we gather otolith image data in various formats from seanoe.org.
- Second component: Architecture for Otolith Image Analysis.

  Preprocessing: The first step is preprocessing, which is essential for enhancing image quality by reducing noise and sharpening the contours. Gonzalez, in his seminal work in [8], provides an overview of image processing techniques, including important filtering methods crucial for this stage. We employ the Canny edge detection algorithm, developed by Canny [14], which is well-regarded for its efficiency in detecting a wide variety of edges in images.
  This step allows us to precisely identify the regions of interest by calculating the distances between contours and counting them. To trace and extract contours, we rely on a method for structural analysis of binary images, as described by Suzuki in [15].

  Detection of Regions of Interest: Contour Counting and Distance Measurement: The integration of traditional image processing techniques with cutting-edge deep learning methods provides a robust framework for analyzing otolith images. Previous studies highlight the critical role of thorough preprocessing, precise contour detection, and sophisticated pattern recognition approaches in achieving accurate classification of otolith structures.
  Growth Band Detection: Deep Learning and RANSAC: The third step involves detecting growth bands in otoliths, which requires advanced techniques. For this purpose, we opted to use deep learning in conjunction with the RANSAC method. RANSAC, introduced by Fischler and Bolles in [16], is highly effective for detecting geometric shapes even when the data is noisy. It is particularly useful for identifying linear patterns in otolith images.
  Deep learning has significantly advanced image analysis. For instance, Krizhevsky in [17] demonstrated that convolutional neural networks (CNNs) are highly powerful for image classification and can be adapted to identify complex patterns in otoliths. Long in [18] proposed fully convolutional networks (FCNs) for image segmentation, an effective method for distinguishing growth bands. Finally, recent advancements such as the YOLO model and Faster R-CNN, developed by Redmon in [19] and Ren in [20], allow for fast and accurate detection of patterns in images.

- Third Component: Client UI display of fish age.
  The third component of our system focuses on the development of a user-friendly interface for displaying the predicted age of fish based on otolith image analysis. This component is critical as it bridges

the gap between complex backend computations and end-user interactions, providing a clear and intuitive representation of the results.

The user interface (UI) is designed to display the predicted fish age, alongside other relevant data, such as growth band patterns and confidence intervals for the predictions. The UI integrates seamlessly with the image processing and deep learning models, ensuring real-time feedback for users such as marine biologists, fisheries managers, and researchers.

To ensure usability, the UI follows best practices in human-computer interaction (HCI) design, with a focus on clarity, simplicity, and responsiveness. Key features include:

Age Display: The predicted age is prominently shown in the UI, allowing users to quickly interpret the results. The age is calculated based on the detected growth bands in the otolith images, using a combination of RANSAC and deep learning models.

Visual Representation of Growth Bands: A graphical overlay of the detected growth bands is displayed alongside the age prediction, helping users to understand the visual cues from the otolith images that contribute to the age determination. This visual aid enhances transparency and user trust in the automated process.

Confidence and Uncertainty Metrics: The system includes confidence intervals or uncertainty metrics derived from the deep learning model to indicate the reliability of the age predictions. This is particularly important in scientific research and decision-making processes, where understanding the model's confidence can influence further analysis or actions.

Interactive Features: Users are provided with interactive tools to adjust parameters, view detailed otolith images, and explore different stages of the preprocessing and analysis pipeline. This empowers users to gain deeper insights into the fish aging process and make informed decisions.

In conclusion, this third component serves as a critical interface, translating complex analytical outputs into actionable insights for users. By focusing on intuitive design and real-time interaction, the Client UI enhances the accessibility and practical use of the otolith age estimation system.

B. Implementation Architecture for Otolith Image Analysis

The architecture is structured into three main stages: preprocessing, region of interest identification, and growth band detection. The model was trained using a dataset of tilapia otolith images. Below, we showcase a sample image from the thousands of tilapia otoliths used in each phase of the process.

Algorithm 1 Preprocess Image

Input: $image\_path$: path of the image
Output: $filtered\_image$: filtered image,
$contours$: contours extracted from the image
$image \leftarrow cv2.imread(image\_path, cv2.IMREAD\_GRAYSCALE)$
if $image$ is None **then**
  raise FileNotFoundError **with the message**
  "Image not found at location: $image\_path$"
end if
$filtered\_image \leftarrow cv2.GaussianBlur(image, (5,5), 0)$
$edges \leftarrow cv2.Canny(filtered\_image, 100, 200)$
$contours, \_ \leftarrow cv2.findContours(edges, cv2.RETR\_TREE, cv2.CHAIN\_APPROX\_SIMPLE)$
emit($filtered\_image$, $contours$)

1) Preprocessing: Filtering and Edge Extraction: Algorithm 1 performs image preprocessing by reducing noise through Gaussian blur and extracting contours, which are critical for applications such as object detection and shape analysis. Noise reduction and contour enhancement are achieved by applying filters. To ensure accurate contour detection, we utilize the Canny edge detection technique, allowing us to identify contours and their orientation. Through the application of Algorithm 1, a preprocessed image is produced. Fig. 2 illustrates the preprocessing for filtering and edge extraction.

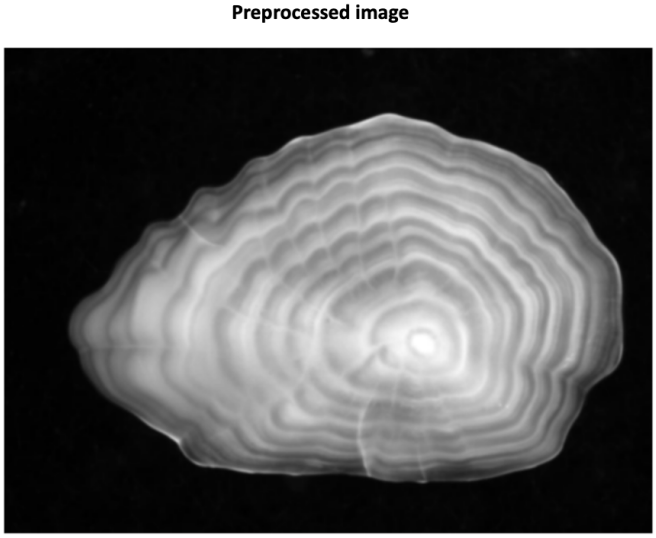

Fig. 2. Tilapia otolith preprocessed.

2) Identification of Regions of Interest: Distance Calculation and Contour Counting: We employ filtering techniques to minimize noise and enhance contour clarity. To optimize contour detection, we implement the Canny edge detection method, which allows for precise identification and orientation of contours. By executing Algorithm 1, we generate a preprocessed image ready for further analysis.

Algorithm 2 Calculate Contour Distances

Input: $contours$: contours extracted from the image
Output: $distances$: distances between contours,
$count$: number of contours
$distances \leftarrow$ empty list
$count \leftarrow 0$
for i from 0 to $len(contours) - 2$ **do**
  for j from $i + 1$ to $len(contours) - 1$ **do**
    $min\_distance \leftarrow \infty$
    for each $point1$ in $contours[i]$ **do**
      for each $point2$ in $contours[j]$ **do**
        $dist \leftarrow$ norm of $np.linalg.norm(point1 - point2)$
        if $dist < min\_distance$ **then**
          $min\_distance \leftarrow dist$
        end if
      end for
    end for
    $distances.append(min\_distance)$
  end for
end for
$count \leftarrow$ length of $contours$
emit($distances$, $count$)

Algorithm 2 evaluates the spatial proximity between contours within an image, a key step for tasks such as analyzing the spatial distribution of objects or performing precise segmentation of elements of interest. Upon applying the second layer of our model, the output generated by Algorithm 2 is as follows. Fig. 3 illustrates the contour detection of the Tilapia otolith.

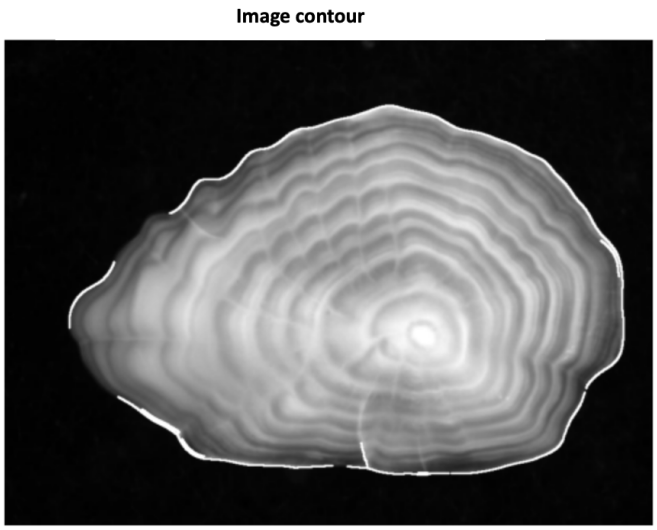

Fig. 3. Tilapia otolith detected contour.

3) Detection of Growth Bands: Deep Learning and RANSAC: This stage utilizes the RANSAC algorithm to detect circular growth bands in the otoliths. The application of a CNN further refines the detection process, enhancing the classification of otoliths.

Algorithm 3 RANSAC for Circle Detection

Input: $image$: image to process
Output: $circles$: circles detected in the image
Initialize $circles \leftarrow []$
$keypoints \leftarrow extract\_keypoints(image)$
for each subset of $keypoints$ **do**
  $model \leftarrow fit\_model(subset)$
  $inliers \leftarrow identify\_inliers(model, keypoints)$
  if count($inliers$) ¿ threshold **then**
    $circles \leftarrow update\_circles(circles, model)$
  end if
end for
emit($circles$)

We employed the RANSAC algorithm to identify circular growth bands in otoliths, as detailed in Algorithm 3. This process involves the detection of circular shapes, with the results being stored in the variable 'circles,' and ultimately, the detected circles are emitted.

We carry out feature extraction, focusing on key metrics such as the number of contours, the average distance between contours, and the number of detected circles. This analysis is essential for identifying growth patterns within the images, as outlined in Algorithm 4.

Algorithm 4 Extract Features

Input: $image\_path$: path of the image
Output: $features$: features extracted from the image
$preprocessed\_image, contours \leftarrow preprocess\_image(image\_path)$
$distances, contour\_count \leftarrow calculate\_contour\_distances(contours)$
$circles \leftarrow hough\_transform(preprocessed\_image)$
if $distances$ is empty **then**
  $mean\_contour\_distance \leftarrow 0$
else
  $mean\_contour\_distance \leftarrow$ mean of $distances$
end if
if $circles$ is None **then**
  $circle\_count \leftarrow 0$
else
  $circle\_count \leftarrow$ length of $circles[0]$
end if
$features \leftarrow \{$
  "$image\_path$" : $image\_path$,
  "$contour\_count$" : $contour\_count$,
  "$mean\_contour\_distance$" : $mean\_contour\_distance$,
  "$circle\_count$" : $circle\_count$,
  "$growth\_stripe\_count$" : $contour\_count$
 # Hypothesis: each contour represents a growth stripe
$\}$
emit($features$)

By putting into practice Algorithm 3 and Algorithm 4, we obtain a detection of the growth bands. Fig. 4 and 5 illustrate

the detection of growth streaks in Tilapia otoliths as well as the calculation of the distance between these growth streaks. Upon

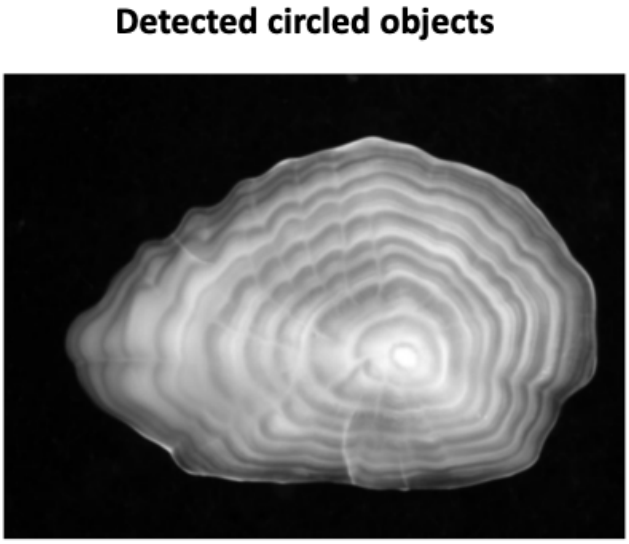

Fig. 4. Detection of growth streaks in tilapia otoliths.

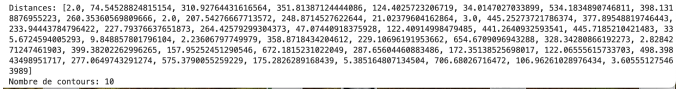

Fig. 5. Calculation of the distance between growth streaks in tilapia otoliths.

implementing our architecture, we move forward by loading the CSV data, constructing the CNN model, and training it specifically on tilapia fish otolith images. The subsequent algorithms outline the steps involved in training the model.

Algorithm 5 Load and Process Data

Input: $images\_folder$: folder containing the images,
csv_path: path of the CSV file
Output: $features\_df$: dataframe containing the features extracted from the images
$data \leftarrow$ read CSV file at csv_path
$data['image\_path'] \leftarrow$ apply lambda function x: join($images\_folder$, x) to $data['image\_path']$
$all\_features \leftarrow$ empty list
for each row, $index$ in $data$ **do**
  $image\_path \leftarrow$ image path from row
  $features \leftarrow extract\_features(image\_path)$
  append $features$ to $all\_features$ Exception as e
  print "Error processing $image\_path$: e"
end for
$features\_df \leftarrow$ create a dataframe from $all\_features$
emit($features\_df$)

Algorithm 5 automates the process of loading images and corresponding data, extracting essential features, and organizing them into a structured format. This step is critical for efficiently managing the extensive image dataset used in pattern detection.

Algorithm 6 Create CNN Model

Input: $optimizer$: optimizer to use for training (default 'adam'),
$init\_mode$: weight initialization mode (default 'uniform')
Output: $model$: created and compiled CNN model
$model \leftarrow Sequential([$
$Dense(64, activation =' relu', kernel\_initializer = init\_mode,$
$input\_shape = (input\_shape,)),$
$Dense(128, activation =' relu', kernel\_initializer = init\_mode),$
$Dense(1, activation =' linear')$
$])$
$model.compile(optimizer = optimizer, loss =' mean\_squared\_error',$
$metrics = ['mae'])$
emit($model$)

Our model is prepared for training on the dataset to facilitate predictions. To define and initialize our convolutional neural network (CNN), we specified the layers, activation functions, weight initialization methods, and compilation parameters. This approach enabled us to develop a flexible and efficient model, well-suited for a range of supervised learning tasks (Algorithm 6).

Algorithm 7 Plot Learning Curves

Input: $history$: model training history
Output: plot of the learning curves
Create a new figure of size $(14, 6)$
Add a subplot $(1, 2, 1)$
Plot $history.history['loss']$ with label 'Training Loss'
Plot $history.history['val\_loss']$ with label 'Validation Loss'
Set the title of the plot to 'Learning Curve (Loss)'
Set the x-axis label to 'Epochs'
Set the y-axis label to 'Loss'
Add a legend to the plot
Add a subplot $(1, 2, 2)$
Plot $history.history['mae']$ with label 'Training MAE'
Plot $history.history['val\_mae']$ with label 'Validation MAE'
Set the title of the plot to 'Learning Curve (MAE)'
Set the x-axis label to 'Epochs'
Set the y-axis label to 'MAE'
Add a legend to the plot
Display the plot

Utilizing Algorithm 7, we visualize the learning curve of our neural network model by leveraging its training history. This facilitates the assessment of the model's performance through the examination of the loss and Mean Absolute Error (MAE) curves for both the training and validation datasets. By

employing this algorithm, we were able to identify potential issues related to overfitting or underfitting and subsequently adjust the model's hyperparameters, thereby enhancing its overall performance.

Algorithm 8 Define Data Augmentation Generator

Output: $datagen$: configured data augmentation generator
$datagen \leftarrow ImageDataGenerator($
$rotation\_range = 20,$
$width\_shift\_range = 0.2,$
$height\_shift\_range = 0.2,$
$shear\_range = 0.2,$
$zoom\_range = 0.2,$
$horizontal\_flip = True,$
$fill\_mode =' nearest'$
$)$

To enhance the robustness of our deep learning model through the creation of variations in the training images, we employ Algorithm 8, which facilitates the definition and configuration of a data generator for image augmentation.

Algorithm 9 Create and Compile Regularized CNN Model

Input: $optimizer$: optimizer for the model
$init\_mode$: initializer mode for the model weights
$dropout\_rate$: rate for the dropout layers
Output: $model$: compiled Keras model
$model \leftarrow Sequential()$
$model.add(Dense(64, activation =' relu', kernel\_initializer = init\_mode, input\_shape = (input\_shape,)))$
$model.add(BatchNormalization())$
$model.add(Dropout(dropout\_rate))$
$model.add(Dense(32, activation =' relu', kernel\_initializer = init\_mode, kernel\_regularizer =' l2'))$
$model.add(BatchNormalization())$
$model.add(Dense(1, activation =' linear'))$
$model.compile(optimizer = optimizer, loss =' mean\_squared\_error', metrics = ['mae'])$
return $model$

Algorithm 9 outlines the process of creating and compiling our regularized convolutional neural network (CNN) model using Keras, incorporating dropout and batch normalization layers.

IV. Results and Discussions

A. Results

In this section, we present the performance of our model, highlighting the key metrics that reflect its learning progress and generalization ability. The training loss begins at 16.6852 in epoch 1 and steadily decreases across subsequent epochs, indicating the model's effective learning process. Similarly, the Mean Absolute Error (MAE) starts at 2.6336 and progressively reduces, ranging between 2.5 and 3.2, suggesting that the model's predictions are becoming more accurate as training continues. These trends demonstrate the model's capacity to learn and capture patterns in the training data.

However, the validation metrics tell a more complex story. Initially, both the validation loss and MAE show significant improvement, with validation loss dropping to 3.8696 and MAE to 1.7366 by the end of the first epoch. In the following epochs, we observe fluctuations, with notable peaks (e.g. at epoch 10), indicating instability in the model's performance on unseen data. The validation loss and MAE spike at epoch 10, suggesting overfitting as the model starts to memorize the training data rather than generalize. This is further corroborated by the rising validation loss, even as training loss continues to decrease.

Overall, the results indicate that while the model shows solid progress in learning, overfitting remains a challenge. More robust regularization techniques, such as reducing the learning rate, using learning rate decay, or applying early stopping, could be beneficial to stabilize the model's performance and improve generalization.

The model achieved a better score of 0.93071, with a final loss of 13.2722 and an MAE of 2.1432, demonstrating its ability to fit the data well when using optimal parameters obtained from grid search. It is important to mention that the datasets were sourced from seanoe.org. The total dataset consists of 10,000 images in "tif" format, of which 7,000 were used for the training set, 1,500 for the validation set, and 1,500 for the test set. In order to obtain results concerning the methodology we proposed, we employed a computer with the following specifications:

- Processor: 2.6 GHz 6-Core Intel Core i7;
- Graphics card: AMD Radeon Pro 5300M 4 GB, Intel UHD Graphics 630 1536 MB;
- Usable memory: 16 GB 2667 MHz DDR4;
- Operating system: macOS Sonoma 14.2.1.

Fig. 6 illustrates the learning curve.

TABLE I. Model Training Results

Epoch	Training Loss	Training MAE	Validation Loss	Validation MAE
1	16.6852	2.6336	3.8696	1.7366
2	20.4612	3.1568	2.0225	0.9299
3	23.0268	3.3713	3.0153	1.2721
4	19.0235	3.1381	2.7854	1.4029
5	15.3980	2.7273	2.3449	0.8462
6	16.1999	2.5265	1.3399	0.8039
7	13.2360	2.7347	0.8052	0.5607
8	13.9841	2.5284	1.3026	0.7432
9	15.0812	2.6757	1.8000	0.6429
10	21.7000	3.1507	3.1815	1.2456

B. Discussions

The model's performance during training is promising, with a steady decline in training loss and MAE. However, the fluctuations observed in the validation metrics suggest issues with generalization, potentially due to overfitting. While dropout was employed to mitigate overfitting, the validation

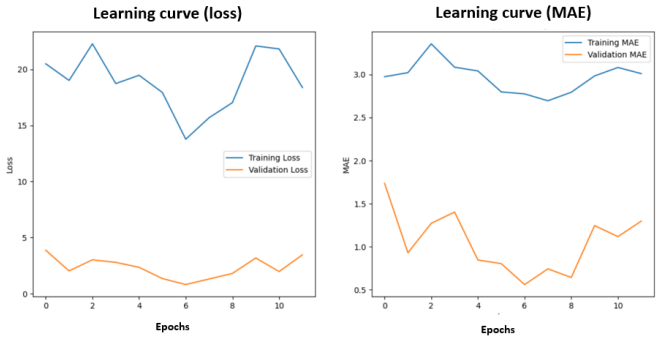

Fig. 6. Learning curve.

performance remains inconsistent, with spikes in both validation loss and MAE, especially noticeable in epoch 10. These spikes suggest that the model is learning noise and specific features of the training data, rather than generalized patterns.

The grid search provided the optimal parameters that resulted in a high score of 0.93071, with a final loss of 13.2722 and an MAE of 2.1432, indicating the model's capability to fit the data effectively under these conditions. Despite this, the presence of outliers and fluctuations in validation performance signals room for improvement in terms of both robustness and consistency. Fine-tuning hyperparameters and further exploring regularization methods may enhance the model's stability and overall performance.

The model's final evaluation, presented in the grid search results and model evaluation table (Table I), shows that although the model performs well during training, achieving optimal scores with the selected parameters, some discrepancies remain, especially for certain data points. Further analysis of these outliers, particularly those where the model overestimates the values, could lead to better model refinement. The methodology employed here demonstrates strong predictive performance for the majority of the dataset, but refinements in error handling could further boost its accuracy.

In summary, the model demonstrates strong performance during training, with consistent results for the majority of the data. There is a high level of agreement between the predicted and actual values. However, fluctuations in the validation metrics, especially with the presence of a few outliers, suggest areas for further improvement. These outliers highlight potential challenges in model generalization, which may require additional fine-tuning. To address this, a more detailed error analysis and adjustments to the model could enhance its precision and robustness. Furthermore, the method employed allows for real-time fish age estimation, showing promise for applications in fishery management and ecological studies. Table II illustrates the model's performance, while Fig. 7 provides a visual comparison of predicted versus actual values. Future work should focus on refining the model to minimize these discrepancies and further optimize its applicability in practical, real-world scenarios.

V. Conclusion and Perspective

This study introduces a hybrid method combining RANSAC and deep learning for counting growth rings in

TABLE II. Grid Search Results and Model Evaluation

Parameters	Values
Best Score	0.9307
Batch Size	20
Epochs	50
Dropout Rate	0.3
Init Mode	Uniform
Optimizer	Adam
Loss	13.2722
MAE	2.1432

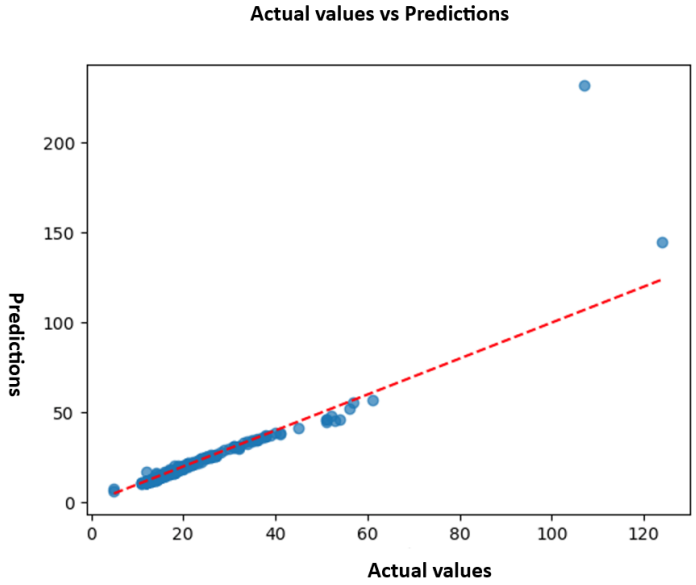

Fig. 7. Actual values vs Predictions.

tilapia, demonstrating effective learning during the training phase, as evidenced by the consistent reduction in training loss and Mean Absolute Error (MAE). The model achieved strong performance, with a peak score of 93,071% during grid search, indicating its ability to fit the data effectively using optimal parameters. However, despite these promising results, the model's performance on unseen data revealed signs of overfitting, as indicated by fluctuations in validation metrics, particularly a significant spike in validation loss and MAE at certain epochs. This suggests the model's limited generalization capacity, which poses a challenge for reliable predictions.

Looking forward, several future directions and research opportunities arise from this work. First, extending the method to other fish species is essential to assess its generalizability and robustness across different ecological contexts. Second, integrating advanced regularization techniques could help mitigate overfitting, such as adopting learning rate decay or early stopping strategies. Additionally, incorporating environmental data could provide insights into the factors influencing growth ring formation, improving the accuracy of age estimations. On a practical level, scaling this model for large-scale applications in fishery management could significantly enhance the sustainable monitoring of aquatic populations. Finally, future

efforts should focus on expanding datasets, refining the model architecture, and exploring its potential in other areas of marine ecology, thereby opening up new avenues for both fundamental and applied research in resource management.

REFERENCES

[1] HÜSSY K., LIMBURG K.E., DE PONTUAL H., THOMAS O.R.B., COOK P.K., HEIMBRAND Y., BLASS M. & STURROCK A.M., 2021. – Trace element patterns in otoliths: the role of biomineralization. Rev. Fish. Sci. Aquacult., 29: 445-477. https://doi.org/10.1080/23308249.2020.1760204

[2] CADRIN S., KERR L. & MARIANI S., 2013. – Stock Identification Methods: Applications in Fishery Science. Second Edition. Elsevier Academic Press, Amsterdam.

[3] CADRIN S.X. & DICKEY-COLLAS M., 2015. – Stock assessment methods for sustainable fisheries. ICES J. Mar. Sci., 72: 1-6. https://doi.org/10.1093/icesjms/fsu228

[4] CHUNG M.T., TRUEMAN C.N., GODIKSEN J.A., HOLMSTRUP M.E. & GRØNKJÆR P., 2019. – Field metabolic rates of teleost fishes are recorded in otolith carbonate. Comm. Biol., 2: 1-10. https://doi.org/10.1038/s42003-018-0266-5

[5] Stries journalières dans les otolithes d'anchois (Engraulis encrasicolus) d'âge 1 et lien avec l'environnement. Disponible en ligne : https://archimer.ifremer.fr/doc/00393/50406/51121.pdf

[6] Manuel de sclérochronologie des poissons. Disponible en ligne : https://horizon.documentation.ird.fr/exl-doc/pleins_textes/divers11-02/010030267.pdf

[7] Projet DEMERSTOCK - Mise au point de la méthode d'estimation de l'âge des poissons. Disponible en ligne : https://www.researchgate.net/publication/378801599_Projet_DEMERSTOCK_-_Mise_au_point_de_la_methode_d%27estimation_de_l%27age_des_poissons

[8] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Prentice Hall, 2002.

[9] SERT, S., et al. (2016). Automatic age estimation of fish using convolutional neural networks and otolith images. Journal of Fish Biology, 89(4), 1978-1985.

[10] WANG, R., et al. (2017). Automated fish age estimation using deep learning and otolith image analysis. Fisheries Research, 188, 125-135.

[11] LIU, Y., et al. (2018). Fish age estimation from otolith images using deep learning and transfer learning. Ecological Informatics, 45, 39-46.

[12] ZHANG, J., et al. (2019). Automated fish age estimation from otolith images using a multi-scale convolutional neural network. Aquatic Living Resources, 32, 202-211.

[13] SUN, X., et al. (2020). Fish age estimation from otolith images using a deep learning framework with data augmentation. Marine Ecology Progress Series, 635, 173-185.

[14] J. Canny, "A Computational Approach to Edge Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679-698, 1986.

[15] S. Suzuki and K. Abe, "Topological Structural Analysis of Digitized Binary Images by Border Following," *Computer Vision, Graphics, and Image Processing*, vol. 30, no. 1, pp. 32-46, 1985.

[16] M. A. Fischler and R. C. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981, doi: 10.1145/358669.358692.

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Advances in Neural Information Processing Systems*, vol. 25, pp. 1097-1105, 2012.

[18] J. Long, E. Shelhamer, and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3431-3440, 2015.

[19] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 779-788, 2016.

[20] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *Advances in Neural Information Processing Systems*, vol. 28, pp. 91-99, 2015.