Myotubes Images Segmentation

Cynthia Cristal Quijas Flores[1] A01655996, Alejandro Sanchez Flores[2] A01662783, Carlos Adrián Palmieri Álvarez[3]A01635776, and Dabria Camila Carrillo Meneses[4]A01656716

Instituto Tecnológico y de Estudios Superiores de Monterrey - Campus GDL

Abstract. This study focuses on the automated segmentation of myotubes using artificial intelligence to address the inefficiencies of manual methods. Leveraging U-Net and YOLO, we trained models with a dataset of over 200 manually segmented images, expanding it using data augmentation techniques. The results demonstrate significant improvements in segmentation accuracy and processing time. Our solution integrates seamlessly into laboratory environments, offering benefits such as time and cost savings, improved team productivity, and reliable results. This report outlines the methodology, model performance, and future steps to refine and implement this technology effectively in biomedical research labs.

Keywords: Myotube · Image · Segmentation · Analyze · Neural network · Convolutional

1 Introduction

Myotubes are multinucleated cells essential for muscle development, repair, and regeneration. These structures are critical in understanding muscular diseases and treatments, making their segmentation in microscopic images a significant task for biomedical research **?**. Manual segmentation, however, is prone to errors, time-consuming, and increases rework costs. This report discusses the development of a myotube segmentation tool using semi-trained artificial intelligence models like U-Net and YOLO. By automating this process, we aim to enhance efficiency, reduce errors, and lower costs in laboratory workflows (Schultz and McCormick, 1994).

2 Methodology

The development of our solution followed a structured process to ensure efficiency and accuracy. Initially, we gathered the images shared with us via Dropbox, which were then transferred to a Network Attached Storage (NAS) system. This storage device facilitated continuous and collaborative access to the data across our team. Once the images were accessible on our computers, we used LabelMe, a software tool for image annotation, to segment the images. LabelMe provided JSON files containing the coordinates of the polygons representing myotubes.

With the annotated dataset of images and JSON files, we proceeded to select and implement machine learning models suitable for the task, specifically YOLO and CNN, to optimize segmentation accuracy and performance.

2.1 Business understanding

Accurate segmentation of myotubes is a very important component in the area of biomedical research and its applications in healthcare, in this particular case, especially for the study of muscle diseases and the development of effective treatments, among others. Nowadays, in manual myotube segmentation, several limitations can be found such as loss of time, propensity to errors and high segmentation costs.

Manual segmentation consumes valuable time of personnel such as laboratorians or interns, reducing productivity in laboratories, when they could dedicate that time to other pending or more relevant activities. As another problem, human variability and the tedious nature of the work increase the risk of errors, affecting the quality of the results. Also, the time spent on these manual tasks increases reanalysis and rework, making the way of working neither efficient nor optimal.

The main objective of this project is to develop an automated artificial intelligence-based tool that addresses these limitations. The goal is to optimize processing times, improve accuracy, decrease manual work time and reduce costs associated with corrections.This solution offers a great impact, allowing to focus on complex analyses and the advancement of scientific research.

2.2 Understanding of the data

The dataset used for this tool is key to ensure the performance of the model. More than 200 manually segmented images were collected, representing real laboratory myotube structures, each image was segmented by team members previously trained by experts in the field, ensuring that the data labels are reliable. There are images in the dataset with different experimental conditions, which increases the model's ability to generalize to varied scenarios without problems. Data augmentation techniques, such as rotations, scaling and brightness changes, were also applied, increasing the variability of the training set and improving the visibility for myotubes recognition.

2.3 Data preparation

For the data preparation phase, we selected images that did not show the presence of alcohol and in which the development of myotubes was at a more mature stage. These images underwent several noise reduction techniques to enhance clarity and prepare them for the segmentation process. Techniques such as salt and pepper noise removal, smoothing, and binarization were applied to minimize noise interference. In addition, edge detection was facilitated through the use of

filters, including bilateral filtering and dilation, to make the identification of boundaries more accurate. These preprocessing steps were crucial for improving the quality of manual segmentation, which was carried out using the Labelme tool.

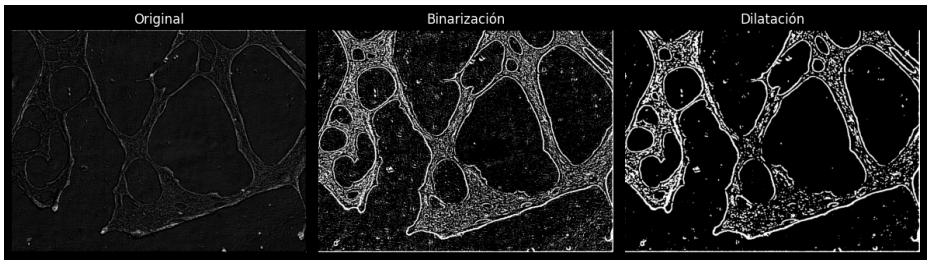

Fig. 1: Filters and techniques used

2.4 Modeling

Two separate models were constructed and trained for the task of segmenting cellular structures: a convolutional neural network (CNN) with a U-Net architecture (DataScientest, 2024) and a YOLOv11 segmentation model. The U-Net model was specifically designed to handle the complex task of segmenting myotube boundaries, using its encoder-decoder framework to learn both spatial and contextual information from the annotated data. Similarly, the YOLOv11 segmentation model was implemented independently, focusing on pixel-level segmentation of myotubes and cellular structures. YOLOv11's advanced segmentation capabilities allow it to identify and delineate regions of interest with high accuracy while maintaining fast processing speeds. Both models were trained independently, with U-Net concentrating on fine-grained segmentation and YOLOv11 segmentation providing an additional approach for pixel-wise predictions, enabling more robust results for complex segmentation tasks.

2.5 Assessment

The U-Net model, using the coordinates generated from manual segmentation, progressively refined its understanding of myotube boundaries, leading to more accurate and automated segmentation of subsequent images. The YOLOv11 segmentation model, trained for pixel-level segmentation, was assessed for its ability to identify and segment myotubes and other cellular structures in images. By leveraging YOLOv11's efficient segmentation capabilities, the model was able to detect and segment boundaries quickly, even in complex or dense regions. While both models addressed the segmentation task, U-Net focused on high-precision

segmentation of detailed structures, while YOLOv11 segmentation offered an alternative approach with strong performance in segmentation tasks that required faster processing.

2.6 Deployment

A comprehensive deployment strategy was developed to ensure the smooth integration and execution of both models. This included setting up the environment and necessary software, such as Python and relevant libraries (TensorFlow, PyTorch, OpenCV, etc.), to facilitate the efficient execution of both the U-Net and YOLOv11 segmentation models independently. An intuitive user interface was also developed to allow users to easily interact with the models. Through this interface, users could input new images and receive segmentation results without requiring technical knowledge of the underlying processes. This ensured that both models could be utilized effectively for various research and diagnostic applications in a user-friendly environment.

3 Models

3.1 Convolutional Neural Network (CNN)

3.1.1 Architecture The segmentation model was built using a convolutional neural network (CNN) with a sequential architecture, leveraging multiple convolutional layers and transposed convolutional layers to achieve precise image segmentation. The input to the model consists of images with a shape of (128, 128, 3), representing the height, width, and color channels, respectively. The key components of the model are as follows (DataScientest, 2024):

Convolutional Layers: The model starts with three convolutional layers, each followed by a MaxPooling layer for dimensionality reduction. The first convolutional layer applies 32 filters of size (3, 3), while the subsequent layers apply 64 and 128 filters, respectively. Each layer uses the ReLU activation function and employs L2 regularization ($\lambda = 0.001$) to prevent overfitting. Padding is set to 'same' to maintain the spatial dimensions of the feature maps.

Dropout Regularization: After each MaxPooling operation, a Dropout layer with a rate of 0.3 is applied to further reduce overfitting by randomly setting a fraction of input units to zero during training.

Transposed Convolutional Layers: The downsampled feature maps are upsampled using transposed convolution layers (Conv2DTranspose). These layers mirror the downsampling path, gradually reconstructing the spatial dimensions of the feature maps. Each transposed convolution layer uses the ReLU activation function and the same (3, 3) filter size, with strides of (2, 2) for upsampling.

Output Layer: The final layer is a 1x1 convolution that outputs a single channel using the sigmoid activation function. This layer produces pixel-wise binary predictions, classifying each pixel as either belonging to the myotube or the background.

Data Preparation: The dataset consisted of 200 manually segmented images. The dataset was split into training and validation sets using an 80-20 ratio. Specifically, 80% of the images (X_train) and their corresponding masks (y_train) were used for training, while 20% of the data (X_val, y_val) was set aside for validation. This split was performed using $train_test_split$, with a test size of 0.2 and a random state of 42 to ensure reproducibility. The images and masks were shuffled prior to splitting.

Compilation and Training: The model was compiled with the Adam optimizer, using a learning rate of 0.001, and binary cross-entropy as the loss function. The accuracy metric was used to monitor performance during training. The model was trained for 300 epochs using a set of training images (X_train) and their corresponding masks (y_train), with validation data (X_val, y_val) provided to evaluate performance at each epoch.

3.1.2 Performance The U-Net model demonstrated strong performance during training and validation, achieving high accuracy and low loss by the end of the training process. After 300 epochs, the following results were obtained:

- Training Accuracy: 98.09
- Training Loss: 0.0599
- Validation Accuracy: 97.14
- Validation Loss: 0.0894

These results indicate that the model was able to generalize well to the validation set, with only a slight increase in validation loss compared to the training loss, suggesting that the model was not overfitting.

Additionally, the model's architecture summary shows that it contains a total of 999,365 parameters, of which 333,121 are trainable, occupying approximately 1.27 MB. The remaining 666,244 parameters correspond to optimizer-related values, amounting to 2.54 MB. The compactness of the model, with no non-trainable parameters, ensures that it efficiently learns from the data while maintaining a relatively small memory footprint (approximately 3.81 MB in total).

Overall, the model exhibited both high accuracy and efficient resource utilization, making it well-suited for the task of image segmentation in this specific application.

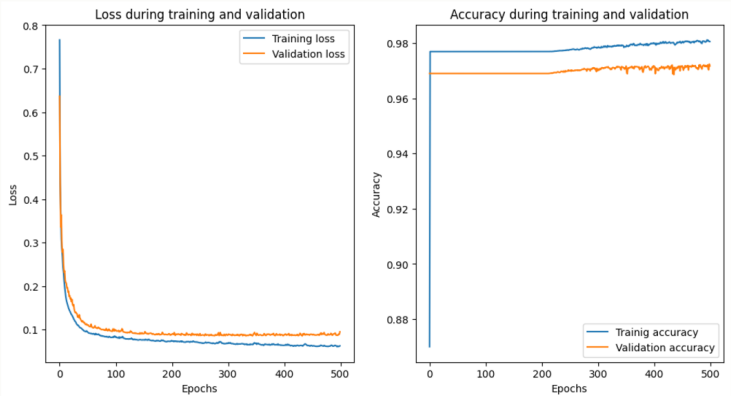

Fig. 2: Model Performance

3.1.3 Results The predictions made by the U-Net model were highly accurate, even surpassing manual segmentation in some cases. Specifically, the model was able to predict regions where myotubes were likely to be present, including areas that were not detected during manual segmentation.

In the image displayed above, the results of the model's predictions are shown across four panels:

1. Original Image (First Column): This panel represents the raw microscopic image of the cells, which was used as input for the model.
2. Manual Segmentation Mask (Second Column): This is the ground truth mask, manually created using the Labelme tool. It identifies the regions where myotubes were annotated by the human expert.
3. Model Prediction (Third Column): The model's predicted mask is displayed here. It highlights areas where the model identified the presence of myotubes, based on the patterns learned during training.
4. Final Results (Fourth Column): In this panel, the model's predictions are overlaid on the original image. "Bounding boxes" have been drawn around areas where the model predicted the presence of myotubes, enabling easy visualization of the regions of interest. Notably, some of these bounding boxes include areas that were missed during manual segmentation, underscoring the model's ability to capture subtle patterns that may be overlooked by human annotators.

Overall, these results demonstrate the robustness of the model in identifying the development of myotubes, and its potential to serve as a valuable tool for enhancing the accuracy and efficiency of manual segmentation tasks.

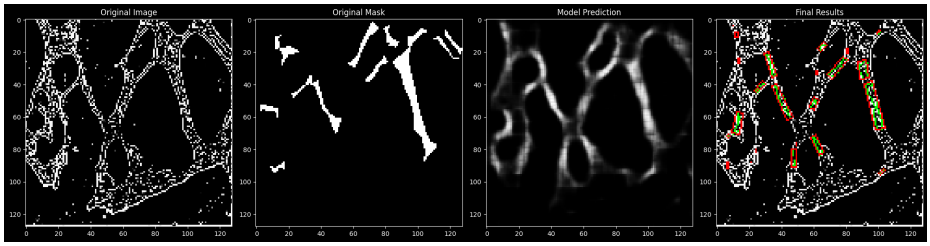

Fig. 3: Results

3.2 You Only Look Once model (YOLO)

3.2.1 Architecture 1The backbone of the YOLOv11-N model is responsible for extracting features from the input images. Using a series of convolutional layers and specialized blocks, it efficiently processes the data (Ultralytics, 2024c).

C3k2 Block: This block replaces the previous C2f block used in earlier YOLO versions. The C3k2 block is designed for improved computational efficiency by using two smaller convolutions instead of one large convolution. This approach helps maintain high performance while speeding up processing times. The "k2" indicates a kernel size of 2, which optimizes the model for faster inference without compromising accuracy (Roboflow, 2024b).

Convolutional Layers: The initial convolutional layers downsample the image while progressively increasing the number of channels, creating multi-scale feature maps. These multi-scale feature maps are crucial for detecting objects of varying sizes in the input images (Roboflow, 2024b).

The neck serves as an intermediate processing stage that aggregates and refines feature representations across different scales. It typically includes the following components (Roboflow, 2024b):

SPPF (Spatial Pyramid Pooling - Fast): This layer improves the model's ability to capture contextual information at multiple scales by pooling features from various spatial resolutions. This enhancement increases overall detection performance by providing a more comprehensive understanding of the image at different scales (Roboflow, 2024b). C2PSA (Convolutional Block with Parallel Spatial Attention): This component applies attention mechanisms to help the model focus on the most important features in the image. It allows the model to prioritize different regions of the input more effectively, improving the model's ability to detect and classify objects based on relevant spatial information (Roboflow, 2024b).

The head component generates the final predictions based on the refined feature maps produced by the backbone and neck. It generally includes (Roboflow, 2024b):

Prediction Mechanism: This mechanism outputs bounding boxes, class probabilities, and segmentation masks (for instance, segmentation tasks). These predictions are derived from the feature maps processed, allowing for precise local-

ization, classification, and segmentation of objects within the image (Roboflow, 2024b).

3.2.2 Performance The YOLOv11n-seg model was re-trained using transfer learning to improve its segmentation performance on myotube images. Transfer learning allowed the model to leverage pre-trained weights from a previous model, significantly speeding up the training process and improving its ability to generalize from limited data (Ultralytics, 2024b). By freezing the early layers of the model, which had already learned general features from the initial training, the focus was placed on fine-tuning the later layers for the specific task of myotube segmentation. This approach helped the model learn the unique features of myotubes without requiring extensive training from scratch, making the process more efficient and effective (Ultralytics, 2024a).

To further improve the model, several key parameters were used: - Learning Rate: A small learning rate was set to prevent the model from overshooting the optimal solution and to allow for fine adjustments in the later layers (Roboflow, 2024a).

- Batch Size: A moderate batch size was chosen to balance memory usage and training efficiency, allowing stable updates to the model weights (Roboflow, 2024a).

- Epochs: The model was trained for a sufficient number of epochs to ensure adequate learning from the data without overfitting (Roboflow, 2024a).

- Freezing Layers: The first 16 layers were frozen, ensuring that only the later layers were fine-tuned, which speeds up training and helps the model focus on task-specific features. These layers correspond to the neckbone structure of the model (Roboflow, 2024a).

- Early Stopping: Patience for early stopping was used, allowing the model to stop training if no improvements were seen for a set number of epochs, preventing unnecessary training and reducing overfitting (Roboflow, 2024a).

Through these parameter choices, the model was able to achieve high accuracy while using fewer resources and less training time, effectively segmenting myotubes with improved precision.

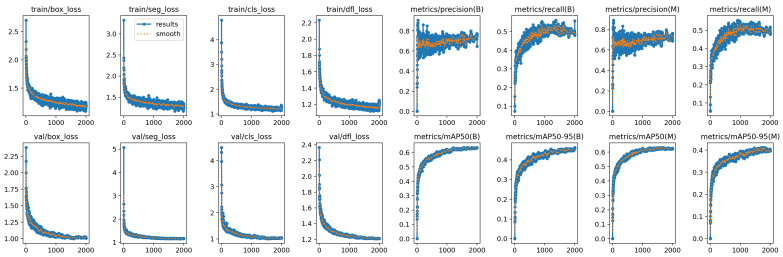

Fig. 4: Training metrics for the YOLOv11n-seg model

3.2.3 Results The predictions made by the YOLO model were accurate. Specifically, the myotubes that are not mature.

In the Fig 5 image, the results of the model's predictions are shown, using the segmentation predicted by the model and the oriented bounding box generated by the polygon of the prediction:

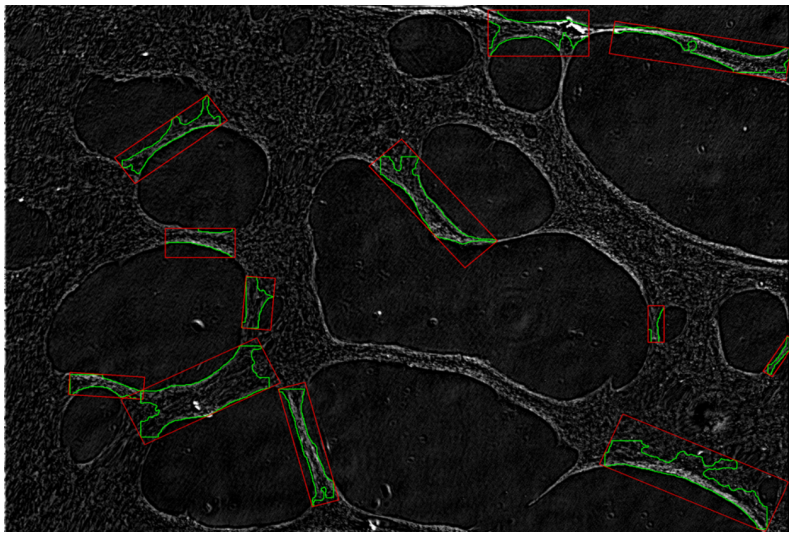

Fig. 5: Predicted Image of the model

4 Conclusion

In conclusion, this study successfully demonstrated the potential of artificial intelligence for automated myotube segmentation. By leveraging U-Net and YOLO models, trained on a dataset of over 200 manually annotated images, we achieved significant improvements in both segmentation accuracy and efficiency compared to manual methods. The results underscore the robustness of the models in identifying subtle patterns and regions that are often missed by human annotation. Furthermore, the development of a user-friendly interface ensures that this solution is accessible to researchers with varying technical expertise, enhancing its applicability in laboratory settings.

This project not only addresses the inefficiencies of manual segmentation, but also highlights the broader impact of integrating machine learning into biomedical workflows. The ability to save time and resources while improving the reliability of results positions this technology as a valuable tool for advancing research on muscular diseases and treatments. Future work could focus on refining model performance and exploring its scalability to larger datasets and diverse cell structures, further solidifying its role in the biomedical field.

Bibliography

DataScientest (2024). U-net: Todo lo que tienes que saber sobre la red neuronal de computer vision.

Roboflow (2024a). How to train yolov11 instance segmentation on a custom dataset.

Roboflow (2024b). What is yolov11? an introduction.

Schultz, E. and K. M. McCormick (1994). Skeletal muscle satellite cells. *Review of Physiology, Biochemistry and Pharmacology 123*, 213–257.

Ultralytics (2024a). Segmento.

Ultralytics (2024b). Yolo11 new - ultralytics yolo docs.

Ultralytics (2024c). Yolo11 nuevo.