UNIVERSITY OF CALIFORNIA,
IRVINE

Video Segmentation for Cardiac Analysis in Embryonic Zebrafish Using Deep Learning

THESIS

submitted in partial satisfaction of the requirements
for the degree of

Doctor of Philosophy
in Electrical Engineering and Computer Science

by

Amir mohammad Naderi

Thesis Committee:
Professor Hung Cao, Chair
Professor Yanning Shen
Professor Lee Swindlehurst

2024

DEDICATION

To

my parents

in recognition of their worth

TABLE OF CONTENTS

Page

LIST OF FIGURES

Page

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my committee chair and principal investigator, Professor Hung Cao, for his unwavering support and guidance throughout my graduate studies in electrical engineering at the University of California, Irvine. Professor Cao has provided a comfortable environment for me to conduct my research, and his insightful advice has been invaluable in helping me navigate through difficult situations.

I would also like to thank my committee members, Professors Yanning Shen, Zhou Li, Lee Swindlehurst, and Xiaohui Xie for their valuable contributions to my research. Their expertise and constructive feedback have been instrumental in shaping the direction of my work.

I would like to acknowledge that portions of this dissertation are derived from my previously published master’s thesis, titled "Deep learning-based framework for cardiac function assessment in embryonic zebrafish from heart beating videos” as it was the continuation of my research. This thesis was submitted to and published by the University of California, Irvine, in 2023.

Moreover, I am grateful to the Department of Electrical Engineering and Computer Science at University of California Irvine for providing me with a world-class education and research opportunities. I also extend my appreciation to the Department of Education for their financial support of my research through Graduate Assistance in Areas of National Need fellowship.

Finally, I would like to acknowledge my family and friends for their unwavering support and encouragement throughout my academic journey. Their love and encouragement have kept me motivated and inspired to achieve my goals.

ABSTRACT OF THE THESIS

Video Segmentation for Cardiac Analysis in Embryonic Zebrafish Using Deep Learning
by

Amir mohammad Naderi

Doctor of Philosophy in Electrical Engineering and Computer Science

University of California, Irvine, 2023

Professor Hung Cao, Chair

Deep learning-based models have revolutionized biomedical image and video segmentation, enabling precise and automated analysis of complex structures. This advancement is particularly critical in the study of zebrafish cardiovascular videos, where accurate segmentation of the heart is essential for understanding cardiac function and development. In this work, we focus on the Zebrafish Automated Cardiac Analysis Framework (ZACAF), utilizing the U-net architecture to achieve high-accuracy segmentation of the zebrafish heart from video frames to calculate important cardiovascular parameters.

After multiple collaborations with researchers studying different phenotypes in zf using ZACAF, to enhance the generalizability of our model across varying datasets and conditions, we employed transfer learning techniques, leveraging pre-trained models to adapt to new data efficiently. Additionally, we incorporated Test-Time Augmentation (TTA) to further improve model robustness and accuracy by applying various transformations to the input data during inference. This approach proposed a systematic solution for adopting ZACAF in broader genetic studies using deep learning algorithms.

Recognizing the importance of temporal dynamics in video data, we extended our work to integrate temporal features into the segmentation model. By analyzing changes between consecutive frames, we aim to capture the heartbeat dynamics more effectively, providing a comprehensive tool for cardiac analysis in zebrafish embryos. Our approach not only advances the field of biomedical video segmentation but also contributes to the broader understanding of cardiac function in developmental biology.

INTRODUCTION

Examining a bright field microscopic image of a zebrafish (zf) embryo yields a wealth of information. Figure 1 presents such an image, showcasing the intricate details of a zf embryo. In the study of zebrafish embryos, the measurement of blood flow velocities serves as a key determinant of cardiovascular function. This task is achieved by tracking the movements of red blood cells (RBCs) within the embryo's body, facilitated by its transparent skin. The acceleration, deceleration, and peak velocity of RBC movements, indicative of blood flow dynamics, can be precisely quantified for analysis. To this end, the motions of RBCs within the dorsal aorta and the cardinal vein, two primary blood arteries, are observed. Individual cell positions are determined using consecutive frames, and RBC

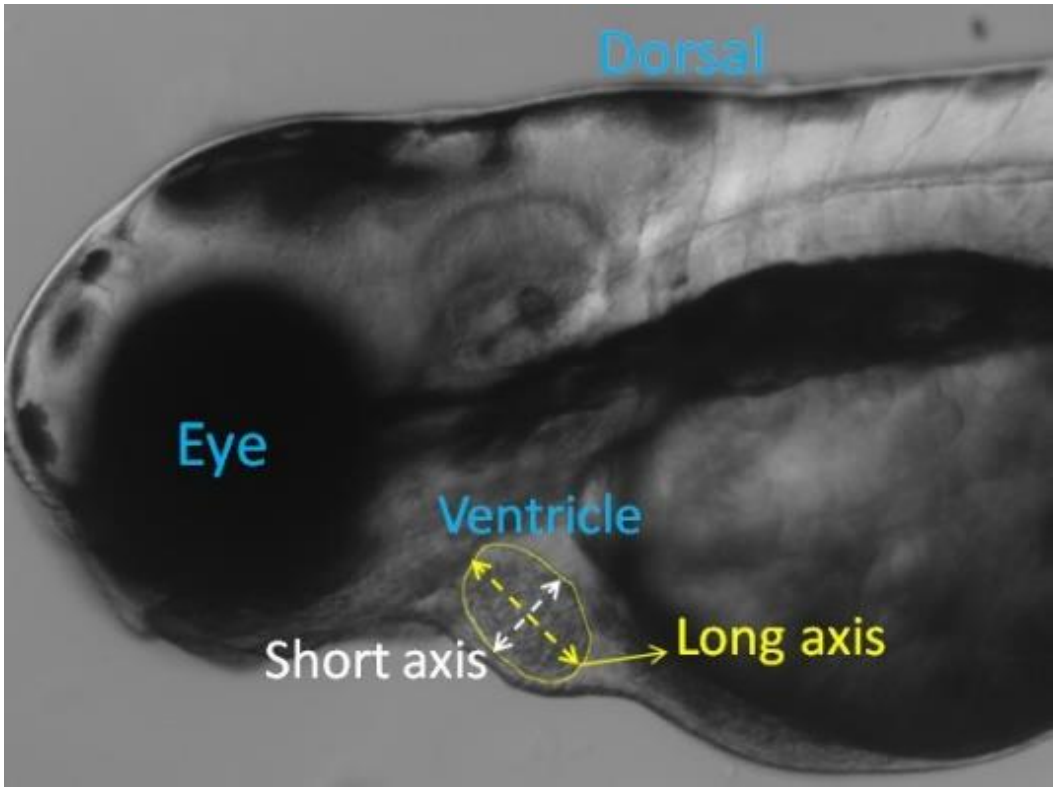

Figure 1:A frame in a video recorded from a 3-dpf zebrafish with segmentation for ventricle border and long and short axes.

velocity is calculated by considering the coordinates of the cell's location and the time interval between frames, as outlined below:

$$RBC\ velocity = \frac{\sqrt{(x_2 - x_1)^2 - (y_2 - y_1)^2}}{\Delta t} \qquad (1)$$

Embryonic zebrafish, particularly those up to three days post-fertilization (dpf), boast transparency, allowing for clear observation of internal organs, such as the heart and circulatory system. During this developmental stage, bright field microscopic videos serve as valuable tools for quantifying both the mechanism and morphology of the heart. Typically, two-dimensional (2D) movies are captured to facilitate cardiovascular analysis. Throughout the cardiac cycle, the continual shifts in ventricular wall position are meticulously monitored, beginning with the selection of a linear region of interest defining the ventricle's borders.

The measurement of myocardial thickness holds significance, especially in assessing the extent of induced defects in hypertrophic cardiomyopathy. In zebrafish embryos, fractional area change (FAC) stands as a well-established metric for evaluating ventricular function and contractility. This metric can be estimated utilizing 2D still frames of the ventricle captured at end-diastole (ED) and end-systole (ES). The fully dilated ventricle corresponds to ED, while the fully contracted ventricle corresponds to ES. At these distinct positions, the ventricular areas (EDA and ESA) are determined, subsequently allowing for the calculation of FAC through the following formula:

$$FAC = \frac{(EDV - ESV)}{EDV} \times 100 \qquad (2)$$

Another measure of ventricular contractility is fractional shortening (FS), which can be calculated using the ventricular diameters at ED and ES (Dd and Ds) as follows:

$$FS = \frac{(D_d - D_s)}{D_d} \qquad (3)$$

To determine stroke volume, ejection fraction, and cardiac output, ventricular volumes must be computed. The long- and short-axis diameters (DL and DS) are initially measured from 2D still images. The following volume formula can be employed if the ventricle has a prolate spheroidal shape:

$$Valume = \frac{1}{6} \times \pi \times D_L \times {D_S}^2 \qquad (4)$$

However, if we consider that the shape of the ventricle is unknown while having the 2D shape of the ventricle, the volume can be calculated using the formula bellow:

$$Volume = \frac{8}{3\pi D_L} \times A^2 \ (5)$$

In this formula A is the area of the segmented 2D ventricle [1].

The blood volume pumped from the ventricle for each beat is called stroke volume (SV), and it is easily determined using the ventricle volumes at ED (EDV) and ES (ESV):

$$SV = (EDV - ESV) \qquad (6)$$

The fraction of blood evacuated from the ventricle with each heartbeat is known as ejection fraction (EF), and it may be computed using the formula:

$$EF(\%) = \frac{(EDV - ESV)}{EDV} \times 100 = \frac{SV}{EDV} \times 100 \qquad (7)$$

The following formula can be used to compute cardiac output (CO) from SV and heart rate (HR):

$$CO\left(\frac{nanoliter}{min}\right) = SV \times HR \qquad (8)$$

The time between two identical subsequent points (i.e., ED or ES) in the captured images is used to calculate HR.

Chapter 1: Overview

1.1. Different image processing methods

Various methods have been discussed in the literature for calculating the heart rate (HR) of zebrafish from videos. Among these, frequency transforms such as fast Fourier transform, filtering techniques, and tracking pixel intensity changes are commonly utilized for quantifying both HR and heart rate variability. These methods can be broadly categorized into three groups: time domain analysis, frequency domain analysis, and blind source separation techniques. Ling et al, has a review paper on Quantitative measurements of zebrafish heartrate and heart rate variability.[2] However, most of the methods for HR measurement cannot measure the other cardiovascular like heart contractibility measures like EF and FS. On the other hand, most methods used for quantification of heart contractibility can be used to measure HR. Hence, those methods are more general and here we focus on them.

The general idea for quantification of ventricular function metrics for evaluating contractility is sematic segmentation of the heart and more specifically ventricle. By segmenting the ventricle in a series of consecutive frames, ES and ED frames can be found and after that the metrics discussed earlier can all be calculated easily. In colored microscopic videos, segmenting the heart becomes considerably easier by filtering the red color, given its heightened intensity in the heart region. However, in black and white recordings, more intricate approaches are required. Upon initial examination of a beating heart video, two distinct features stand out: the identification of borders and the periodic movement of the heart. These features are pivotal for manual segmentation of the ventricle. Background subtraction can effectively capture movement features, while a range of methods aimed at enhancing and automatically segmenting borders can be employed to tackle border features.

1.2. Background subtraction

Background subtraction is a widely used technique for segregating the moving elements of a scene in a stationary camera setup by distinguishing between background and foreground elements. This method involves subtracting continuous frames from a video to identify moving objects. In the case of zebrafish videos, static pixels representing the immobile parts of the fish body can be removed, as the primary moving components are typically blood cells and the heart. Several background subtraction approaches exist, including frame difference, Gaussian mixture model, kernel density estimation, and codebook methods. Each method offers varying degrees of accuracy in identifying moving objects. In scenarios where the fish remains completely static with minimal noise, background subtraction can be particularly useful for segmenting the ventricle. However, it's important to note that dynamic features such as blood cells, vessels, respiratory-induced gale movement, and noisy pixels may also be detected in the output of this method. Nevertheless, these extraneous elements can be filtered out using different techniques, ensuring accurate segmentation of the ventricle.

Large moving objects detected that are not a part of ventricle can be removed using specifying a region of interest (ROI) and thresholding the size of the object. Small, detected particles and noise can be removed by filtering. For example, arithmetic mean filter can be used for smoothening and geometric mean filter can be used for removing salt and paper noise. Morphological filters are useful for smoothing binary images, especially for removing small structures and border detection. The morphological filter's concept is a shrink and let grow procedure. The term "shrink" refers to the use of a median filter to round off large structures and remove small structures, with the surviving structures being grown back by the same amount during the grow process[3]. In zebrafish videos, employing background subtraction often leads to detecting the ventricle as a region containing a group of multiple objects that form the shape which represents ROI. Applying morphological filters can be helpful because we need to have a single object to represent the ventricle. Nevertheless, background subtraction can result in inaccurate results that cannot be guaranteed.

On the other hand, if we want to detect borders there are several algorithms that can enhance or detect the edges. Enhancing the image in a way that ventricular border can be more visible can be beneficial to researchers for manual and automatic segmentation.

1.3. High pass filter

High-pass filters operate by extracting the derivative of a signal in the time domain. When applied to images, these filters accentuate rapid changes, such as edges, earning them the moniker "sharpening filters." Common examples of high-pass filters include the Laplacian and Sobel filters, both renowned for their ability to sharpen images. Furthermore, Gaussian and Butterworth filters can also be tailored to function as high-pass filters, providing additional flexibility in image enhancement techniques.

1.4. Thresholding

Histogram-based thresholding stands as one of the fundamental techniques in image segmentation. This method entails using thresholding to convert a grayscale image into a binary representation. In its simplest form, each pixel's intensity is altered to black if it falls below the predefined constant threshold or white if it surpasses it. Various histogram thresholding models exist, each employing distinct approaches to ascertain the threshold value for segmentation.

A. Global thresholding:

Global thresholding entails the selection of a threshold value that effectively separates the foreground and background regions across the entire image. This method operates under the assumption that the image's histogram exhibits a bimodal distribution, implying that pixel intensities can be segregated into two discernible groups representing the foreground and background. The threshold value is commonly determined through an iterative process, with Otsu's method being a notable example. Otsu's method aims to maximize the inter-class variance between these two groups, thus ensuring optimal threshold selection.

B. Adaptive thresholding:

Adaptive thresholding is a variation of global thresholding that adjusts the threshold value locally based on the intensity values of the neighboring pixels. This approach is useful for images with non-uniform illumination or shading, as it can adapt to changes in the local intensity values. The threshold value is typically computed for each pixel using a local region, such as a square or circular neighborhood, and can be based on methods such as the mean, median, or Gaussian distribution of the local intensity values.

C. Iterative thresholding:

Iterative thresholding is a method that involves iteratively adjusting the threshold value based on the intensity values of the pixels within the foreground and background regions. This approach can be used to segment images with complex histograms that do not have clear separations between the foreground and background regions. The threshold value is typically initialized using a global or adaptive thresholding method and then iteratively refined based on the mean or median intensity values of the pixels within each region.

D. Edge-based thresholding:

Edge-based thresholding is a method that involves detecting the edges or boundaries between the foreground and background regions and using these edges to determine the threshold value. This approach can be useful for images with complex structures or textures that do not have clear separations between the foreground and background regions. The threshold value is typically selected based on the gradient magnitude or edge strength of the image and can be determined using methods such as the Canny edge detector or the Laplacian of Gaussian filter.

There are algorithms like Otsu that find the best threshold automatically. For example, Otsu finds a threshold that segments the background and foreground classes of the histogram by minimizing intra-class intensity variance.

1.5. Histogram equalization

Discussing histogram equalization is indeed pertinent in the context of image enhancement techniques. This method is particularly effective in enhancing the global contrast of an image, especially when the original image is represented by a narrow range of intensity values. By redistributing pixel intensities across the histogram, histogram equalization ensures a more uniform utilization of the entire intensity range, thereby improving contrast throughout the image. This process is especially beneficial in zebrafish videos, where the region of interest often appears excessively dark due to various factors such as microscope positioning or tissue characteristics. The increased dynamic range of contrast facilitated by histogram equalization significantly aids both manual and automated segmentation of the heart. However, it's important to note that histogram equalization works best when the distribution of pixel values is relatively uniform across the image. In cases where the image contains regions with significantly lighter or darker intensities compared to the rest of the image, the improvement in contrast may not be as pronounced. This is particularly true in zebrafish videos captured using light sheet microscopy, where the background often exhibits the highest intensity and different sections of the fish may have varying levels of transparency, leading to uneven contrast enhancement. Hence, Adaptive Histogram Equalization (AHE) solves the problem by transforming each pixel with a transformation function consequent to a neighborhood region. Finally, Contrast Limited AHE (CLAHE) is a variant of adaptive histogram equalization, which doesn't have the issue of over amplification of noise in regular AHE. [4]

1.6. Edge detection

Edge detection serves as a foundational technique in image processing, crucial for identifying the boundaries or edges delineating objects within an image. The primary objective of edge detection is to differentiate between regions of uniform intensity or color and those characterized by sharp transitions or gradients. Various approaches exist for edge detection, but a common strategy involves identifying abrupt changes in intensity or color indicative of object boundaries. These changes can be discerned by analyzing the first derivative of the image along different directions or by employing filters designed to accentuate high-frequency components of the image.

One of the most prevalent filters utilized for edge detection is the Sobel filter, a convolution filter adept at estimating horizontal and vertical gradients within the image. The Sobel filter employs two 3x3 kernels to compute derivatives along the x and y axes, respectively.

These derivatives are then combined to yield estimations of gradient magnitude and direction at each pixel location.

Another common approach to edge detection is to use the Canny algorithm, which was introduced by John Canny in 1986. The Canny algorithm involves several steps, including smoothing the image with a Gaussian filter, calculating the gradient magnitude and direction of the smoothed image, performing non-maximum suppression to thin the edges, and applying hysteresis thresholding to connect weak edges to strong edges. The Canny algorithm, which is one of the most prominent edge detection methods, has a multi-stage algorithm to detect a wide range of edges in images[5]. For fully automated heart segmentation in zebrafish microscopic videos, edge detection algorithms like Canny are usually not robust. The most important problem is that in the zebrafish videos have numerous edges and tissues therefore the canny algorithm detects many different edges next to each other. This makes it imposable to tell which edge belongs to the heart. However, edge detection can be used as preprocessing or one of the steps in an automatic segmentation framework.

1.7. Color filtering

Color filtering can indeed be a valuable tool in processing zebrafish videos, even if not all recordings are in color. Given that blood is red and the heart along with most vessels exhibit red coloration in colored microscopic videos, leveraging this characteristic can aid in segmentation. A straightforward method involves setting a threshold for red intensity, wherein pixels falling outside of a specified range for red are assigned to black, while those within the range are assigned to white. This approach effectively highlights the heart, blood vessels, and some accompanying noise in a binary image.

In existing literature, transgenic animals expressing myocardial-specific fluorescent reporters have been extensively utilized. These videos often involve manual feature selection, which enhances both manual and automated segmentation processes. However, for fully automated quantification of cardiovascular metrics such as ejection fraction (EF), precise segmentation of the ventricle is essential. In such cases, a simple color filtering method targeting the specific coloration of the heart can effectively segment the entire heart, providing a viable solution for automated analysis. Akerberg, et al proposed a Convolutional Neural Network (CNN) framework that automatically segments the chambers from the videos and calculates the EF [6]. In this paper a CNN architecture has been used to segment the ventricle and atrium individually in a video where the heart is highlighted. In conclusion, color filtering alone can only be employed as a feature selection method to increase the accuracy of segmentation.

1.8. Machine learning

Indeed, traditional methods like edge detection, color filtering, and background subtraction may lack robustness across different zebrafish videos, especially when ventricle edges exhibit varying shades of gray. To address this challenge, researchers have turned to machine learning approaches to develop fully automated frameworks.

Unsupervised learning segmentation methods, such as K-means clustering and Gaussian mixture models (GMM), have been explored. These methods aim to partition the image into distinct clusters based on pixel intensities without requiring labeled training data. K-means clustering iteratively assigns pixels to clusters based on their proximity to cluster centroids, while GMM models the distribution of pixel intensities as a mixture of Gaussian distributions.

Additionally, supervised deep learning methods have gained traction in automated segmentation tasks. Deep learning models, particularly convolutional neural networks (CNNs), are trained on labeled data to learn intricate patterns and features directly from the images. These models can efficiently capture complex relationships between image pixels and their corresponding labels, making them highly effective for tasks such as ventricle segmentation in zebrafish videos.

Overall, both unsupervised and supervised machine learning approaches offer promising avenues for building robust and fully automated frameworks for zebrafish video analysis.

1.8.1. K-means

Clustering algorithms, including the unsupervised K-Means algorithm, play a crucial role in data analysis by uncovering hidden structures, clusters, and groupings within datasets. Unlike classification algorithms, clustering algorithms do not require predefined categories or labels, making them particularly useful for exploring unlabeled data.

In the context of image processing, the K-Means clustering algorithm can effectively separate the region of interest from the background by partitioning the image into K clusters based on similarity. Each cluster represents a distinct color range in colored images or grayscale intensity range in black and white videos. The K-Means algorithm proceeds in two phases: first, it determines K centroids, and then it assigns each data point to the cluster with the nearest centroid based on Euclidean distance. After grouping the data points, the algorithm recalculates the centroids and iteratively assigns data points to clusters until convergence. The centroid of each cluster represents the center point where the sum of distances between all data points in the cluster is minimized. This iterative process aims to minimize the overall sum of distances between each data point and its assigned cluster centroid. To further refine the segmentation results and generate a noise-free image, median filtering is commonly employed as a noise removal technique. Median filtering replaces each pixel value with the median value of its neighboring pixels, effectively reducing noise and preserving image details. In summary, the K-Means clustering algorithm, coupled with median filtering, offers an effective approach for segmenting regions of interest in images and removing noise, facilitating clearer and more accurate analysis of image data. The segmented image may still have some undesired regions or noise after it has been segmented. As a result, the median filter is applied to the segmented image to improve its quality.[7]

1.8.2. Gaussian mixture model

Gaussian mixture-based segmentation is a prominent image processing technique rooted in histogram thresholding, a widely used method for image segmentation. In histogram thresholding, an image is partitioned into two distinct regions or classes: the target region and the background region, each exhibiting its own uni-modal gray level distribution. Consequently, the challenge in segmentation lies in selecting an appropriate threshold to delineate the image into these regions effectively. Gaussian mixture-based segmentation extends this concept by modeling the distribution of image pixels as a mixture of Gaussian distributions. Each pixel is then classified into different segments based on the Gaussian distribution it most closely aligns with. The fundamental premise of Gaussian mixture-based segmentation is that each segment within an image can be characterized by a probability distribution, reflecting the statistical properties of the pixels within that segment.

By leveraging Gaussian mixture modeling, this segmentation technique offers a more nuanced and probabilistic approach to image segmentation, enabling finer delineation of regions based on their underlying statistical properties. Specifically, the pixel intensities within each segment are assumed to be normally distributed, and the goal of the segmentation algorithm is to estimate the parameters of the Gaussian distributions that best fit the observed data [8]. To accomplish this, the Gaussian mixture-based segmentation algorithm first initializes a set of Gaussian distributions with random parameters and then iteratively refines these parameters to better fit the observed data. During each iteration, the algorithm computes the likelihood that each pixel belongs to each of the Gaussian distributions and then assigns each pixel to the segment corresponding to the Gaussian distribution with the highest likelihood. The parameters of the Gaussian distributions are then updated based on the pixels assigned to each segment. This process is repeated until the parameters of the Gaussian distributions converge to a stable solution. Once the segmentation is complete, each pixel in the image is assigned to a specific segment, which can be used for further analysis or processing. The accuracy of model parameter estimations and how closely the histogram of an image approximates a Gaussian mixture determine the effectiveness of Gaussian-mixed-based segmentation techniques.[9]

Here, the mentioned methods have been implemented not only to compare with the approach using deep learning described in the next section but also to use for preprocessing. All these are shown in Figure 2, a-d panels.

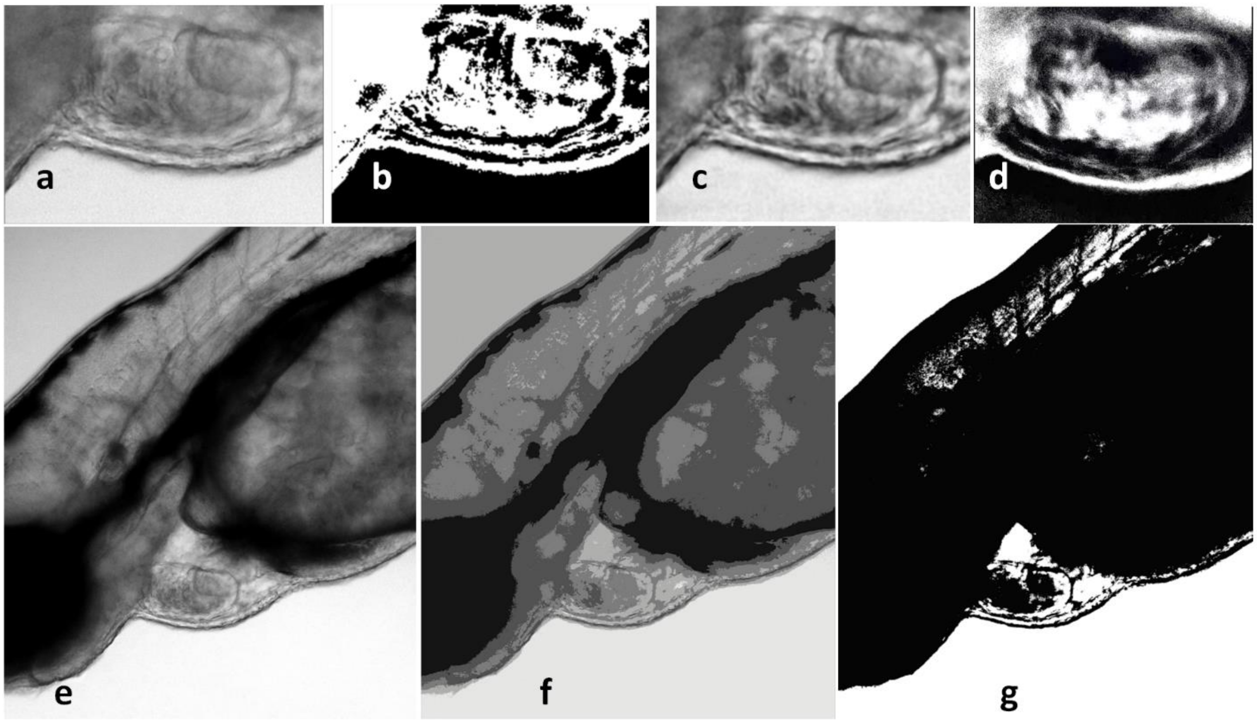

Figure 1: Ventricle segmentation using different methods. Panel a-d: A frame from the video of a 3 dpf zebrafish with 40X zoom undergoing different HBS algorithms. a. Original frame. b. Manual histogram thresholding. c. CLAHE. d. Otsu thresholding. Panel e-g: A frame from the video of a 3 dpf zebrafish with 10X zoom undergoing GMM and K-means approaches. e. Original frame f. GMM. g. K-means.

The abovementioned methods, namely edge detection, color filtering, and background subtraction, are not robust with different videos since ventricle edges might have multiple shades of gray. Therefore, we also attempted to use machine learning approaches to compare. First, unsupervised learning segmentation methods like K-means and Gaussian mixture model (GMM) were applied to the videos. As shown in Figure 2e, f, and g, although these methods improve the visibility of the ventricle borders, the heart's automatic segmentation is not possible. Moreover, much of the unnecessary information (pixels) in the image, particularly in the image generated using K-means, remains.

These methods improve the visibility of the ventricle borders, while the heart's automatic segmentation is not possible. Moreover, much of the unnecessary information (pixels) in the image, particularly in the image generated using K-means, is remaining. However, manual segmentation is extremely tedious work and in most practical research scenarios there are numerus videos recorded and manual segmentation can be time consuming as well. In conclusion, for a fully automated framework a more robust method is required. For achieving this goal, a few recent papers proposed using deep learning methods.

1.8.3. Semantic image segmentation

Semantic segmentation, also known as image segmentation, involves clustering portions of an image that belong to the same object class together, thus categorizing every pixel in the image. This pixel-level prediction task is often supervised, where a mask of the object portions serves as ground truth data.

Supervised classification algorithms for pixel-level prediction include fuzzy measures, decision trees, support vector machines, and artificial neural networks (ANNs). Among these, ANNs have demonstrated remarkable performance and accuracy, particularly in biomedical images. In the context of zebrafish ventricle segmentation, semantic-level image classification aims to assign distinct semantic classes to each scene image, with the ventricle as the object of interest and the background as the other class.

To represent these classes in the mask, a specific color scheme is employed, typically with black representing the background class and white representing the ventricle class. This color choice is crucial for aligning with evaluation metrics like the Dice coefficient and Intersection over Union (IoU) coefficient. The Dice coefficient and IoU coefficient are widely used metrics for evaluating the performance of image segmentation algorithms or classification models. These metrics measure the overlap between the predicted segmentation or classification result and the ground truth mask. By convention, the background is assigned the value of 0 (black), and the object of interest is assigned the value of 1 (white) when computing these metrics. Following this convention allows for an accurate assessment of the algorithm's accuracy in capturing the ventricle and discriminating it from the background. Higher coefficients indicate better performance in accurately segmenting the ventricle.

Chapter 2: ZACAF model and its method

2.1. Semantic image segmentation validation metrics

In the quantification of cardiovascular metrics from videos using deep learning methods, our primary goal is to accurately predict the geometrical shape of the ventricle, including its position, size, and shape, in alignment with the manually created masks serving as ground truth. The predicted shape should ideally closely resemble or match the ground truth mask. To validate the performance of automatic segmentation, it is essential to employ metrics that effectively evaluate the accuracy of the segmentation results. In semantic image segmentation, the most used metrics comprise pixel-wise accuracy, Dice coefficient, and Intersection over Union (IoU).

a. *Pixel-wise Accuracy*

In segmentation of the ventricle, since the mask indicating the ventricle is either white or black, there are only two classes so that we can use the binary case of pixel accuracy. The accuracy is defined as the percent of pixels classified correctly as

$$pixel - wise\ Accuracy = \frac{pixels\ classified\ correctly}{All\ pixels} \qquad (9)$$

In these videos the ventricle has a much smaller area compared with the rest of the frame so this metric alone can be misleading. However, the correct identification of white pixels (which are the pixels creating the background) is essential because they ensure the position, and the shape of the ventricle is also correct.

b. *Dice coefficient*

The dice coefficient is a widely used metric for determining how similar two objects are. It has a scale of 0 to 1, with 1 indicating perfect match or complete overlap. For a binary case, the coefficient is calculated as

$$Dice = \frac{2|(A \cap B)|}{|A| + |B|} \qquad (10)$$

where A is the predicted image and B is the ground truth (manually created mask).

c. *Intersection over union*

It's also known as the Jaccard Index, which is just the area of overlap between the predicted segmentation and the ground truth divided by the area of union between both. This measure runs from 0 to 1, with 0 indicating no overlap and 1 indicating complete overlap. For the binary case, it can be calculated as:

$$J = \frac{|A \cap B|}{|A \cup B|} \qquad (11)$$

2.2. Literature review

The heart of a framework like ZACAF is its deep learning-based segmentation model that inputs the video frames and outputs corresponding binary masks of the ventricle. Among the most commonly utilized segmentation architectures are U-net, FCN (Fully Convolutional Network), and SegNet. U-net, known for its symmetric encoder-decoder architecture with skip connections, has gained widespread adoption in biomedical image segmentation tasks due to its ability to capture fine-grained spatial details effectively [10]. FCN replaces fully connected layers with convolutional layers, enabling end-to-end pixel-wise predictions and providing flexibility in handling images of variable sizes [11]. SegNet, similar to U-net in its symmetric structure, utilizes an encoder-decoder architecture without skip connections, relying on max-pooling indices for up-sampling [12]. Each of these architectures offers distinct advantages in different segmentation tasks, catering to specific requirements such as spatial detail capture, flexibility, or precise localization.

To date, the majority of existing studies have primarily focused on basic heart rate detection methods, such as edge tracing [13]. Nasrat et al. introduced a semi-automatic approach for quantifying fractional shortening (FS) in zebrafish embryo heart video recordings [14]. Their software offers automated visual insights into end-systolic (ES) and end-diastolic (ED) stages by displaying color-coded lines on a motion-mode display. However, the manual marking of ventricle diameters during ES and ED stages, followed by FS calculation, proves to be highly laborious, time-intensive, and prone to inconsistencies when dealing with a large number of frames. Akerberg et al. proposed a SegNet beased framework to automatically segment chambers from videos and calculate ejection fraction (EF) [6]. Nonetheless, their approach relies on specific transgenic animals expressing myocardial-specific fluorescent reporters and high-end fluorescence microscopes, limiting its widespread applicability within the research community, especially for those lacking access to such resources. Furthermore, Huang et al. highlighted potential issues with transgenic fluorescence protein expression leading to dilated cardiomyopathy [15], underscoring concerns regarding the use of foreign proteins that may impact myocardial function. Additionally, Akerberg et al. utilized frames from only four videos, raising concerns about potential overfitting when video features such as fish position, lighting conditions, or lens focus on the ventricle differ from the training set. Zhang et al. proposed a U-net based framework similar to ZACAF however, similar to Akerberg's study, they used a transgenic zf line expressing reporters and high-end fluorescence microscopes [16]. Suryanto et al used DeepLabCut for labeling the ventricle to facilitate automatic zf cardiac assessment. DeepLabCut is a software toolbox that employs deep learning techniques to enable markerless pose estimation and tracking in videos of animals or humans [17]. Nonetheless, this framework only marks 8 points on the ventricle which limits the resolution and accuracy of the predicted ventricle. This will significantly limit the robustness of the framework, especially with mutant types.

For a more Inclusive and available example of a fully automatic cardiovascular segmentation for zf, Naderi et al. proposed a framework using U-net to segment monochromic light sheet microscopy videos. [18] In this framework, after preprocessing using sharpening filter and CLAHE, 50 videos of wild and mutant type zf have been manually segmented to be used for the training dataset. The U-net was then trained and validated using the dataset and the deep learning model showed 99.1% for pixel-wise accuracy, 95.04% for Dice coefficient, and lastly 91.24% and for the IoU. They created a graphical user interface to provide an end-to-end platform so researchers can use it conveniently. The framework inputs raw videos and segments the ventricle in each frame of it. The output for each frame is a binary mask of the ventricle. From there diameters of the ventricle in each frame can be calculated. The frames with the largest and smallest area are going to represent ED and ES respectively. Having the ED and ES frames important cardiovascular parameters namely EF, FS, CO, and SV can be quantified. The EF quantification has been validated using 8 videos that haven't been included in the training set. The averages of absolute errors and standard deviations for the automatically calculated EF of the 8 wild type test videos compared to the expert's manual calculation were 6.13% and 3.68%, respectively.

2.3. U-net architecture

The U-net architecture, a convolutional neural network (CNN) originally designed for biomedical image segmentation tasks, has seen widespread adoption in various areas of computer vision due to its effectiveness. Its architecture is specifically tailored to address the challenges inherent in semantic segmentation, where each pixel in an image is assigned, a label based on its context and relationships with neighboring pixels.

U-net comprises two primary components: an encoder network and a decoder network. The encoder network is responsible for extracting high-level features from the input image, while the decoder network utilizes these features to generate a segmentation map. These networks are linked by skip connections, facilitating direct information flow between them.

The encoder network typically consists of multiple convolutional layers that apply filters to the input image, extracting features at varying levels of abstraction. Each convolutional layer's output undergoes a nonlinear activation function, such as ReLU, introducing nonlinearity into the network. Additionally, pooling layers within the encoder downsample the feature maps, reducing spatial dimensions to capture abstract features efficiently while minimizing computational complexity.

Conversely, the decoder network generates the segmentation map using features extracted by the encoder. It comprises transposed convolutional layers, also known as deconvolutional layers, which upsample the feature maps to their original dimensions. Like the encoder, each layer's output is passed through a nonlinear activation function to introduce nonlinearity.

The skip connections in U-net facilitate direct information transfer from the encoder to the decoder, bypassing intermediate layers. This helps retain spatial information and enhances segmentation accuracy, particularly in regions with intricate foreground or background elements. Overall, U-net's architecture enables robust and accurate semantic segmentation in diverse image datasets.

To summarize, U-net is a powerful architecture for image segmentation that is designed to capture high-level features of an image and retain spatial information through the use of skip connections. It has been widely used in biomedical image analysis and has also shown promising results in other areas of computer vision, such as natural language processing and robotics. [10] A similar architecture has been employed by Decourt *et al.* to segment the human left ventricle from magnetic resonance imaging (MRI) images [19]. The main idea of the U-net is to complement a traditional contracting network by successive layers, where pooling operations are replaced by up-sampling operators. Besides, a subsequent convolutional layer can then be trained to assemble a precise output based on this information. The training of the network uses the original image as an input and the mask of the corresponding image as the output, and the objective is to minimize the error of the estimation and the mask. In the next part ZACAF details will be discussed.

2.4. Experimental animals

Zebrafish (Danio rerio; WIK strain) were maintained under a 14 h light/10 h dark cycle at 28.5°C. All animal study procedures were performed in accordance with the Guide for the Care and Use of Laboratory Animals published by the U.S. National Institutes of Health (NIH Publication No. 85-23, revised 1996). Animal study protocols were approved by the Mayo Clinic Institutional Animal Care and Use Committee (IACUC #A00002783-17-R20).

2.5. Video imaging of beating zebrafish hearts at the embryonic stage

Zebrafish in the embryonic stages were anesthetized using 0.02% buffered tricaine methane sulfonate (MS222 or Tricaine) (Ferndale, Washington, US) for 2 minutes and then placed lateral side up with the heart facing the lower-left corner. The specimens were held in a chamber with 3% methylcellulose (Thermo Fisher Scientific, Massachusetts, US). The videos were recorded using a Zeiss Axioplan 2 microscope (Carl Zeiss, Oberkochen, Germany) with a 10X lens and differential interference contrast (DIC) capacity. The used Zeiss' Axiocam 702 mono Digital Camera 426560-9010-000 records videos with 60 fps; however, using the Zeiss computer software, videos get stored in 5 fps, 10 fps, and 20 fps. Video clips were processed using ImageJ for manual quantification of cardiac functional indices, including heart rate and fraction shortening, as detailed in the following sections.

2.6. ZACAF

Figure 3 illustrates the architecture of the proposed U-net model with details.

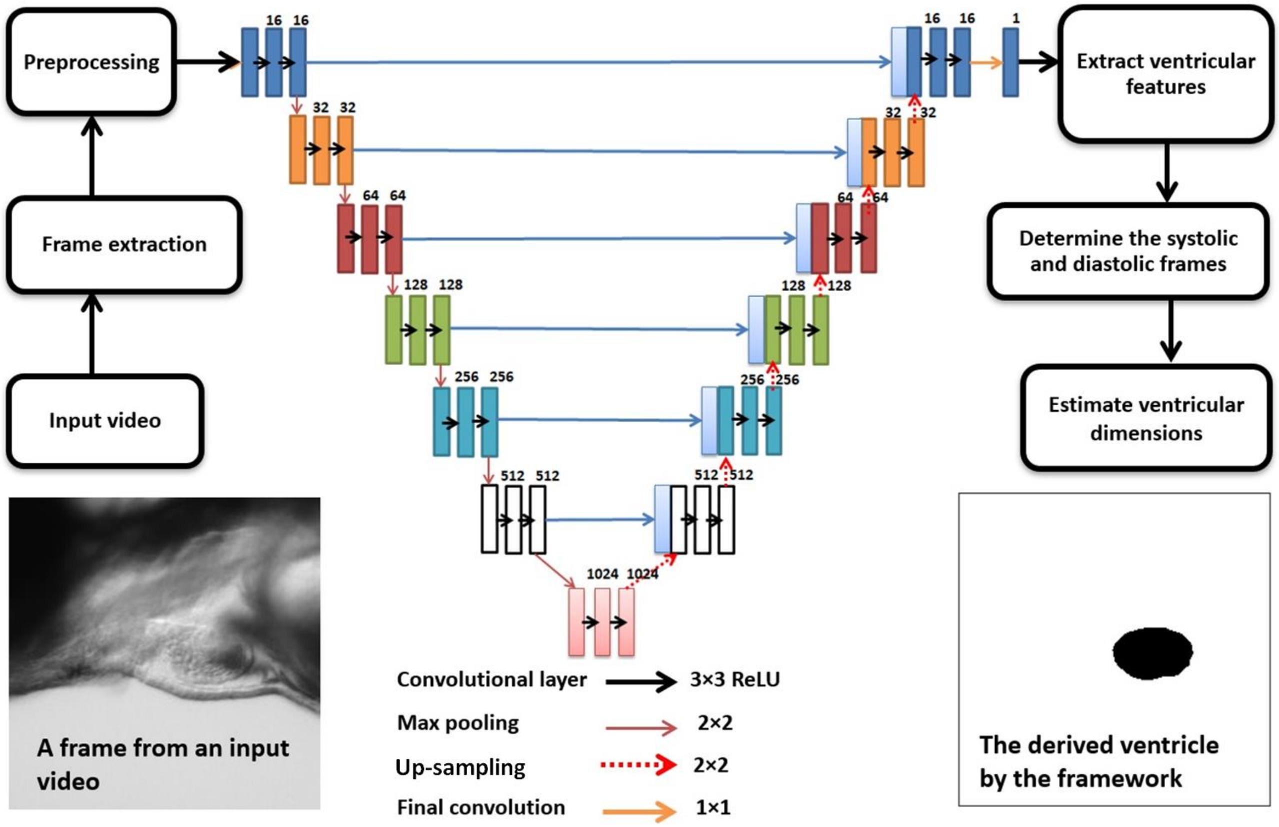
Input video
Frame extraction
Preprocessing
16 16
32 32
64 64
128 128
256 256
512 512
1024 1024
16 16
1
Extract ventricular features
Determine the systolic and diastolic frames
Estimate ventricular dimensions
A frame from an input video
The derived ventricle by the framework
Convolutional layer 3×3 ReLU
Max pooling 2×2
Up-sampling 2×2
Final convolution 1×1

Figure 2: The process flow and the U-net architecture. Each rectangle represents a layer and the number above it shows the number of neurons inside. A trained model can estimate a mask of the ventricle from all the extracted frames of the input video. When all the frames have a predicted mask, by determination of ES and ED frames, important cardiac indices like EF, FS, and stroke volume can be automatically calculated and saved in a desired format.

The network consists of a contracting path and an expansive path, which gives it a U-shaped architecture.

The contracting path is a typical convolutional network that consists of repeated convolutions, each followed by a rectified linear unit (ReLU) and a max-pooling operation. Dropouts have been used to prevent overfitting. The architecture has been optimized to obtain the best result. For training, NVidia's T4 GPU from Google Collaboratory was employed. The most commonly used loss functions for semantic image segmentation were deployed to evaluate the model, namely Binary Cross-Entropy and Dice loss function.

Cross-entropy can be defined as a measure of the difference between two probability distributions for a given random variable or set of events. It is extensively used for classification problems, and since segmentation is the classification at a pixel level, cross-entropy has been widely used. Binary Cross-Entropy is defined as:

$$Loss_{BCE}(y,\hat{y}) = -(y\log(\hat{y}) + (1-\mathrm{y})\log(1-\hat{y})) \qquad (12)$$

where y is the true value and $\hat{y}$ is the predicted outcome.

The Dice coefficient is a commonly used metric in computer vision problems for calculating the similarity between two images. In 2016, it was also adapted as a loss function, namely Dice Loss [20].

$$Loss_{Dice}(y,\hat{y}) = 1 - \frac{2y\hat{y}+1}{y+\hat{y}+1} \qquad (13)$$

The U-net model has been trained with both models, and the performance has been assessed using validation and test sets. Further, the calculation of EF has also been evaluated using both loss functions.

2.7. Titin truncated Mutants

Dilated cardiomyopathy (DCM) is a hereditary, progressive disease, which eventually leads to heart failure [21]. Thus, it is essential to evaluate the early cardiac functions associated with DCM. Dozens of pathogenic genes have been found in the genetic studies of cardiomyopathy, and the incidence rate of DCM is about 1/250 [22]. Titin truncated variants (TTNtv) are the most common genetic factor in DCM, accounting for 25% of DCM cases [23]. Therefore, we have recently restated the allelic heterogeneity in zebrafish segments and established a stable mutation system to assess mutant zebrafish's cardiac functions systematically and accurately. In order to study the mechanobiology of induced defects of these disease models, heart functions need to be reliably evaluated [24].

2.8. Dataset

A training dataset was created employing raw microscopic videos of zebrafish containing 800 pixel-wise annotated images. 50 videos of the lateral view from 50 different 3-dpf zebrafish were analyzed for creating the dataset. 10 of these videos are from the TTNtv mutant line. From each video, 10 to 30 frames were extracted. A total number of 850 frames were extracted for the training set. Each training set has a frame from the video, and a mask manually created showing only the ventricle with ImageJ software. After making the masks, all image and mask sets have been organized into folders. Each set has two folders inside, one for the original extracted frame and the other for its corresponding mask. Finally, all sets were shuffled to avoid overfitting. The validation set with the 10% of the data's size has been split from the dataset before training.

2.9. Preprocessing

In the preprocessing stage, a region of interest is defined, knowing all recordings have the same positioning for the zebrafish. Although this cropping improves the accuracy by removing unnecessary information, it can be avoided to make the framework robust to different video types. Additionally, a sharpening filter accomplished by performing a convolution between a custom weighed kernel and an image is used to make edges more visible. After training, the U-net architecture was able to predict the ventricle segment. The model has been trained several times by applying the mentioned image processing methods to the training images. The method with the best results was CLAHE thresholding which was added to the preprocessing section.

2.10. Quantification of the diameters of the predicted ventricle

The ventricle's diameters are measured for all extracted frames automatically with the contour tool from OpenCV (an open-source computer vision library). The maximum and minimum measured areas of the ventricle in different frames show the ES and ED stages, respectively. Using the measurement of ES and ED frames, we can calculate the ejection fraction (EF), fractional shortening (FS), and stroke volume (SV). Also, the time between two ES (or ED) frames could be used to derive heart rate (HR). The predicted ventricle is assumed to be an ellipsoid. For quantification of EF, the ventricle area can be used (**Eq (5)**) by counting the pixels inside the predicted shape. Since the frames are 2D, we are estimating the ventricle volume to its area. For FS, measurements of the short axis in ES and ED frames are needed. As the ventricle is not a perfect ellipsoid, estimation of the short and long axes can be carried out in two different ways. In the first method, an ellipsoid could be fitted in the predicted shape, and then the axis of the fitted ellipsoid would be measured. The second way is to find the longest line as the long axis of the estimated ellipsoid, which could be found in the geometrical shape; then, the short axis of the ellipsoid is the short axis of the ventricle. In this framework, the 2-D area of the ventricle directly measured from the mask has been used for EF since it is more accurate.

2.11. Graphical User Interface (GUI)

This framework was developed in Python, and thus, for researchers who are not familiar with programming, working with it can be challenging. To address this, a Graphical User

Interface (GUI) has been designed to provide a user-friendly interface to facilitate researchers' process. Moreover, after training the U-net, the trained model can be saved, which means the most computationally heavy part could be done only once. The GUI saves the output files in the CSV format, along with information about EF, FS, diameter readings of the area, short and long axis, and frame numbers. Therefore, each video's data can be easily accessed at anytime and anywhere with the expandable cloud feature. Our ZACAF provides an end-to-end interface to researchers to automatically calculate, classify, and record various cardiac function indices reliably. ZACAF can work with multiple videos simultaneously and output the results in a fraction of the time compared to that of manual segmentation. The deep learning model in the ZACAF can easily be updated and optimized with a new model and data.

2.12. Assessment of the accuracy of the framework with the defined metrics

The model’s performance can be seen in Figure 4. The model has been trained with two loss functions discussed in section **2.6,** and the best results with parameter tuning are illustrated. The metrics mentioned above resulted in 99.1% for pixel-wise accuracy, 95.04% for Dice coefficient, and lastly 91.24% and for the IoU.

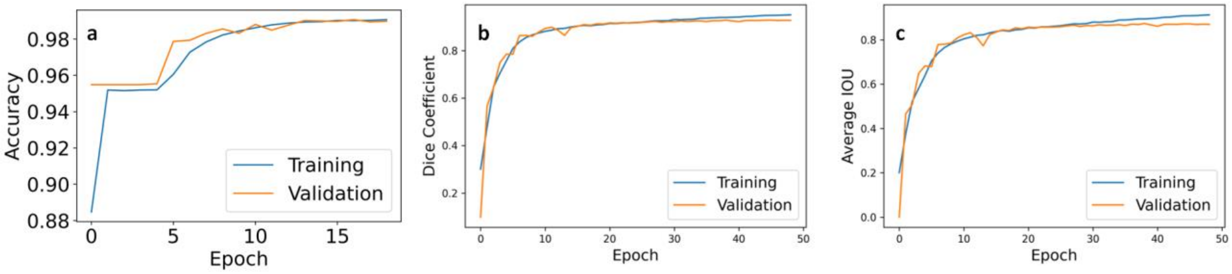

Figure 3: ***The proposed model’s performance plotted with the metrics commonly used in semantic image segmentation. a.*** *Pixel-wise accuracy* ***b.*** *Dice coefficient* ***c.*** *IoU metric. This plot shows the performance of the framework with the training and validation sets during the process of training of the deep learning model.*

All mentioned metrics are evaluating the best performing model that had a Dice loss function with an Adam optimizer and a 0.001 learning rate along with decay steps of 240 and a decay rate of 0.95. The validation split was 10% which means 80 sets. Following the training, we visually assessed the framework's ability to correctly segment ventricular chambers and the periodic pulsating movement of it within series of frames of a test video. This process was used in parameter tuning for the deep learning model.

2.13. Assessment of the performance of the framework for EF

The framework was evaluated by comparing the results obtained by manual assessment of EF from an experienced biologist with those using the software since one of the primary purposes of this framework is EF calculation. In this calculation, finding the area in all frames of a video is important because we want to find the ED and ES areas. Hence, assessment should involve the series of frames in a test video rather than having random images in a validation set. For this reason, we assess the performance of ZACAF with EF calculation. First, 8 videos of wildtype zebrafish embryos and another 8 from TTNtv mutant embryos were used as the framework's input. These videos are the test set and have not been used in training. Second, manual processing and estimation were performed for each video to derive EF by an expert to use as the ground truth. The program saves the predicted ventricle masks for every frame of a video, and the ED and ES frames are simply the frames with a maximum and minimum area of the segmented ventricle, respectively. After automatically finding ES and ED frames, the EF of the fish in the input video would be calculated and saved in a CSV file along with other indices calculated. The averages of absolute errors and standard deviations for the calculated EF of the 8 wild type test videos compared to the expert's manual calculation were 6.13% and 3.68%, respectively.

As ED and ES frames are the most important parameters to quantify cardiovascular indices, we plotted the correlation of the automated and manual measurements (Figure 5 a,b). Moreover, Bland–Altman analysis was then used to assess the agreement in manual and automatic ventricle segmentation. Bland–Altman demonstrates the difference that were measured at the same time plotted against the average of the EF with two methods. Larger differences would specify larger disagreement between the two calculations [25]. From 16 test videos two different sets of ES and ED frames (meaning 4 frames from each video) have been manually and automatically segmented. As it can be seen in Figure 6 c the Bland-Altman has been plotted for all pairs of measurements in the same figure with blue and red dots representing mutant and wild fishes respectively.

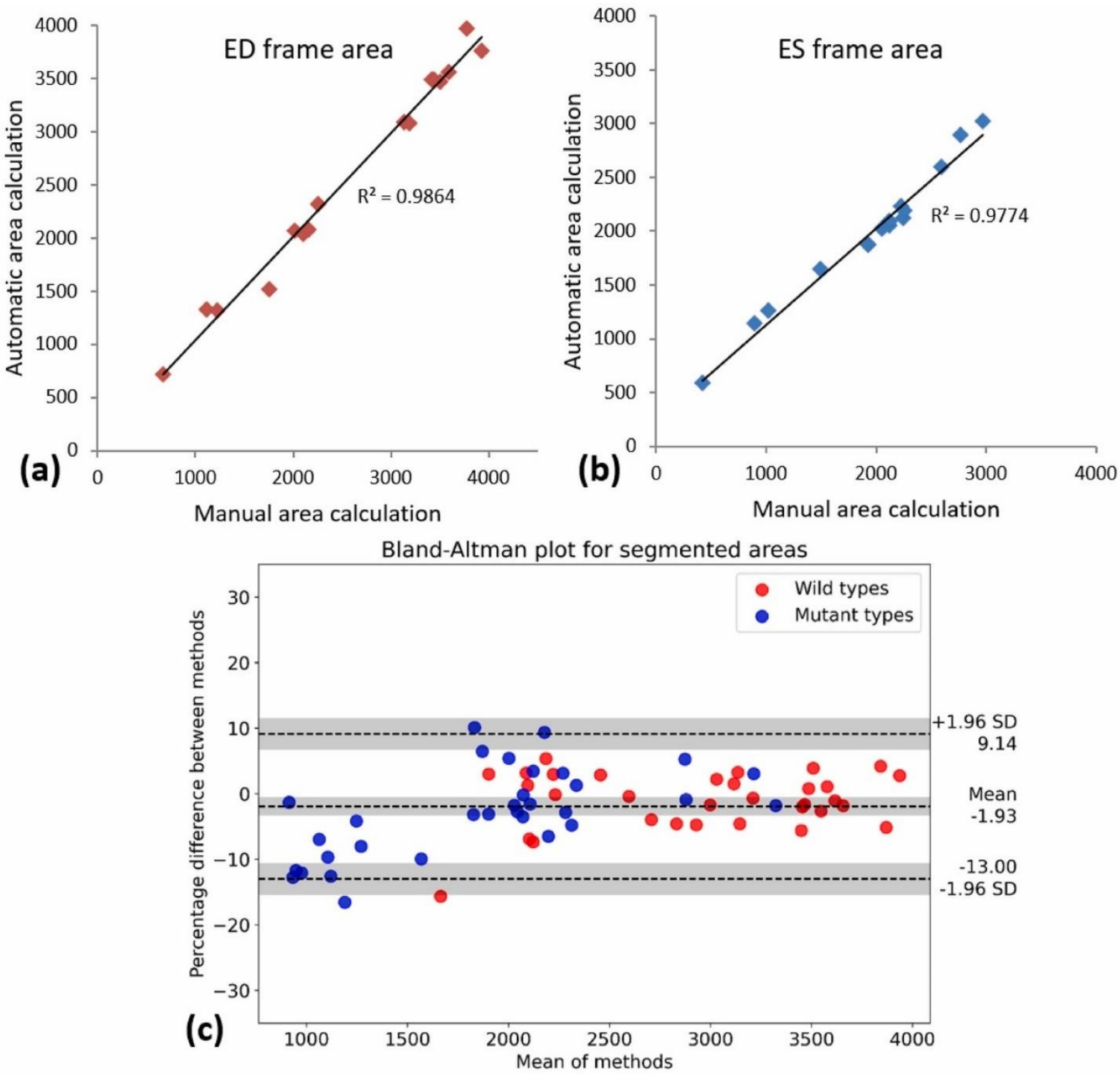

*Figure 4: After finding and measurement of the ventricle area in ED and ES frames of 8 wild type and 8 TTNtv mutant fish with both manual and automated methods, the results are demonstrated in a correlation plot while the calculated EF for the wild and mutant types is plotted in Bland-Altman to demonstrate the agreement of measured values. Linear relation of the measurements with slopes close to 1 shows the accuracy of the ZACAF. (**a**) ED frame area. (**b**) ES frame area. (**c**) Bland-Altman plot for 64 sets of measurements of the segmented ventricle using manual and ZACAF methods. Both mutant and wild have 32 pairs each represented in the plot. Red and blue dots represent wild and mutant fishes respectively. The measurements are in pixels.*

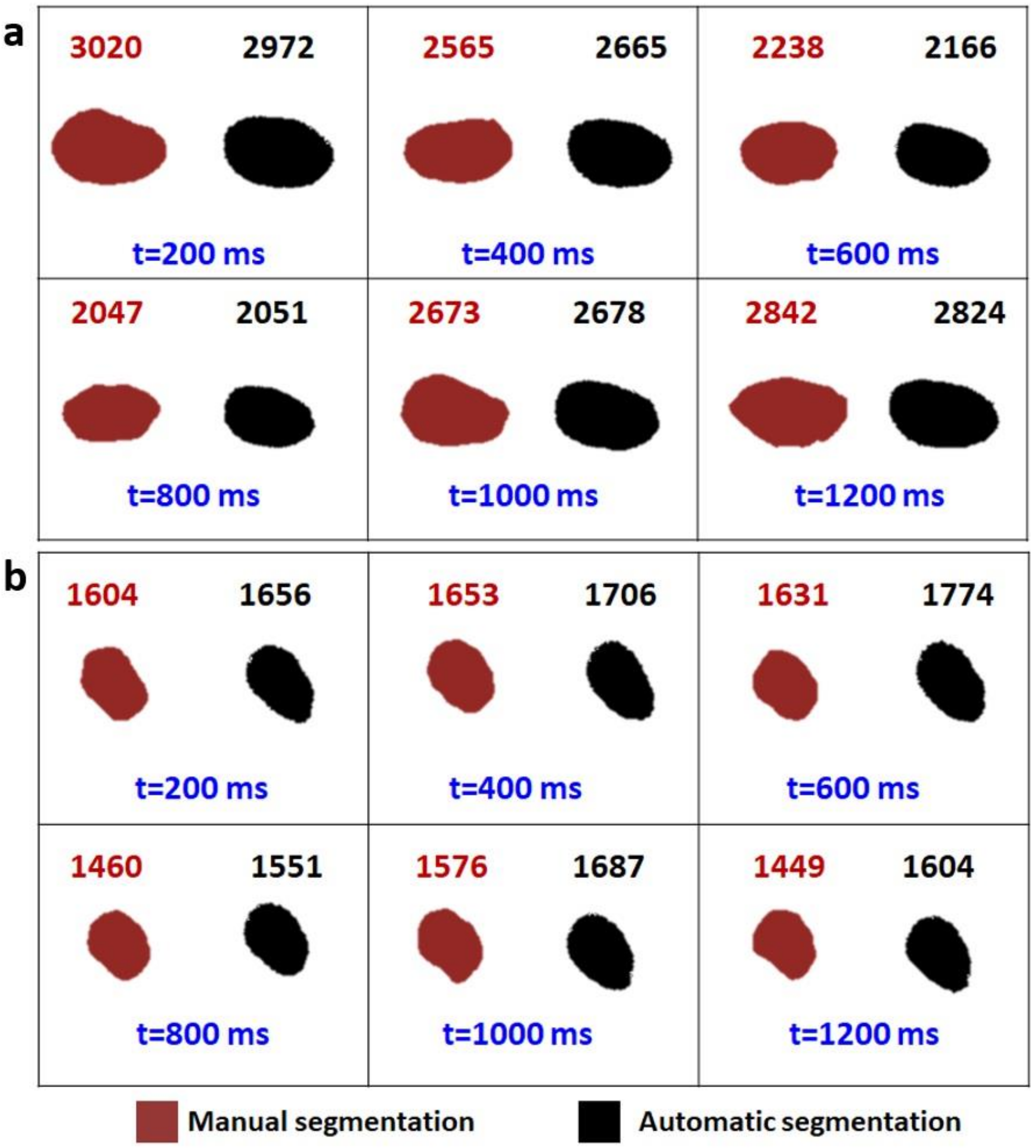

*Figure 5: **Validation of U-net image segmentation framework.** The sequential frames from a wild type zebrafish recorded video with fps of 5 are extracted. The respective ventricle mask of each frame is shown in each panel via manual and automatic segmentation. The area of each ventricle is measured and written above its own box. Considering the fps of the videos and the average heart rate of the zebrafish, 6 consecutive frames have been shown in this figure to ensure having at least one full cycle.*

Figure 6 presents the comparison of manual and automatic segmentation of the ventricle in 6 continuous frames to cover an entire cardiac cycle for both wild type (**a)** and TTNtv (**b)**. In manual segmentation, measures were done using the freehand selection tool in the ImageJ software.

Chapter 3: Expanding the work to new datasets

Similar to ZACAF, several image processing frameworks have been proposed to automate the process of automatic quantification of zf cardiovascular parameters. However, most of these works rely on supervised deep learning architectures. However, supervised methods tend to be overfitted on their training dataset. This means that applying the same framework to new data with different imaging set up and mutant types can result in severe decrease of the performance. Here, we take Nrap genotype, and Zebrafish Automatic Cardiovascular Assessment Framework (ZACAF) as an example to demonstrate a modified framework. In this modification we apply data augmentation, Transfer learning, and test time augmentation to ZACAF to improve the general performance and propose a protocol for other researchers to be able to apply the available frameworks for their own data.

Nebulin Related Anchoring Protein (NRAP) is a protein coding gene expressed in cardiac and skeletal muscle. NRAP is a member of the Nebulin family of proteins and promotes myofibril assembly by colocalization with actin during myoblast fusion in early stages of development in skeletal muscle [26]. Previous studies in zebrafish have shown that overexpression of Nrap results in severe skeletal muscle myopathy. Additionally, reducing levels of Nrap in a Klhl41 deficient zebrafish model resulted in a less severe phenotype, as klhl41 is a regulator of NRAP ubiquitination [27]. In cardiac muscle, NRAP plays a role in myofibril assembly and is localized at cardiac intercalated discs [28]. In mouse models of dilated cardiomyopathy, NRAP is overexpressed early in development [28]. Thus, downregulation of NRAP suggests therapeutic advantages in multiple model organisms. Interestingly, a homozygous truncating mutation of *NRAP* was found in a human dilated cardiomyopathy patient. However, the variant was not detected in a cohort of 231 dilated cardiomyopathy patients, and the patient's unaffected brother carries the same mutation, suggesting a low penetrance and allele risk [29]. Due to the potential therapeutic advances of NRAP downregulation found in Zebrafish and Mice, as well as a report of an NRAP truncating variant in a dilated cardiomyopathy patient, we aimed to investigate the cardiac effects, specifically ejection fraction and fractional shortening, of NRAP downregulation in embryonic zebrafish with ZACAF deep learning model. Despite a report of NRAP downregulation associated with cardiomyopathy in a human patient, we see no significant differences in ventricle shape, ejection fraction and fractional shortening between genotypes in embryonic Zebrafish.

3.1. Considerations for zebrafish age and use of anesthesia

The use of anesthesia is necessary for fish greater than 2 days post fertilization, as the fish become mobile at 3dpf as their swim bladder inflates. Tricaine anesthesia, commonly used in zebrafish, may impact the animal's cardiac system. Thus, it is necessary to record the animal no later than 2dpf or deliver anesthesia in a standardized way where each fish receives the same dose. For 5dpf zebrafish, we placed the zebrafish in a separate dish of 16mg/L of Tricaine anesthesia in egg water for exactly 5 minutes prior to transferring the zebrafish to a glass slide emended in 3% methyl cellulose for image acquisition. This method proved to be time consuming, so we opted to record animals at 2dpf instead. At 2dpf, the zebrafish cardiac system is fully developed and functional (5). Additionally, at 2dpf, the zebrafish begin to hatch from their chorion, allowing for their bodies to become straight. It is important that the animal is in this stage before recording, so there is a narrow window for acquisition – the fish must have hatched from the chorion but are not yet mobile. To appropriately time this, dividers should be used at the time of crossing. Additionally, movement of egg water in the dish via transfer pipet will stimulate chorion hatch in the morning and recordings should be made shortly after in the afternoon. An additional consideration besides the use of anesthesia for the age of fish is the pigmentation of the animal. At 2dpf, the animal is transparent, while at 5dpf, skin pigmentation may hinder the ability to acquire a clear image of the ventricle.

3.2. Considerations for image acquisition

Accurate calculations of ejection fraction require a frame of the organism’s ventricle in a true systolic and diastolic position. Assuming a resting heart rate of 60 to 100 beats per minute, a high-speed recording camera is necessary. Initially, videos were recorded on a Leica K3 camera (Leica, Germany) with a maximum frame rate of 30 frames per second attached to a Leica S9D microscope at 5x magnification. Videos were processed with the Leica LASX software. However, with this speed, we were unable to visualize the heart in a true diastolic and systolic position. Thus, on the improved setup, we used a FastCam-PCI high-speed digital camera (Photron, USA) with a frame rate of 250fps attached to a Zeiss upright microscope at 10X and processed with the FastCam-PCI image capture board. Additionally, at 5X, we were unable to visualize the boundaries of the ventricle, while at 10X, there is a clear visualization of the ventricle boundaries and red blood cells which can be seen in figure 3. In order to maximize video quality while reducing file size, videos were acquired in greyscale with a resolution of 512 × 480 pixels and recorded for only roughly 4.35 seconds, enough time to capture roughly 8 cardiac cycles and 1,088 frames with a 0.004 shutter speed (5). An additional consideration for recording is that fluorescent lights illuminating the microscope room (or the microscope light bulb itself) have a specific frequency. Fluorescent lights can result in horizontal lines, banding, or flickering in the video, depending on the shutter speed that was utilized to capture the footage. Although it is advised to shoot in varied lighting conditions, changing the camera's shutter speed can also help resolve this problem.

One other important factor that needs to be considered for imaging is the placement of the fish under the microscope. Firstly, in most literature for quantification of cardiac function using 2D imaging the ventricle is assumed to be an ellipsoid. Hence, during image acquisition it is important to make sure the fish is properly positioned to its side. improper placement of the fish under the microscope can result in the ventricle having a pear-shaped structure which can be observed in figure 3 on the left. This will eventually result in inaccuracy in the quantification of EF.

Furthermore, it is important to know that the placement of the fish under the microscope is a feature in deep learning models. Uniform placement of the fish across the videos recorded for the data set can affect the model into only responding accurately to test videos with similar placement. To train a robust framework Data augmentation must be used in the process of training the deep learning model. Using data augmentation will make the framework less prone to placement of the fish.

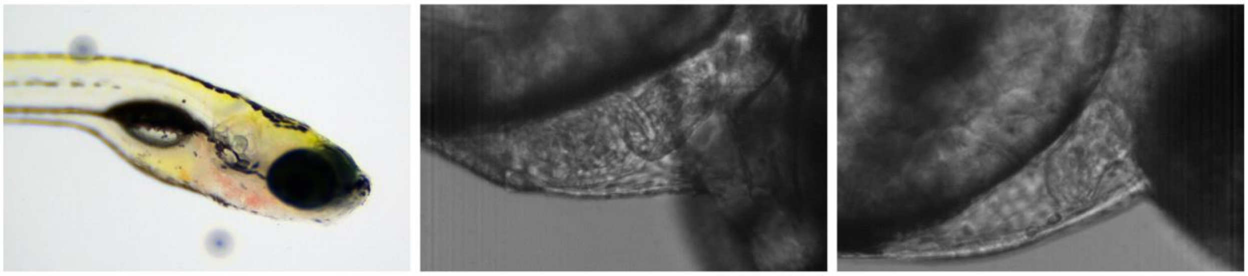

Figure 6: Comparing the visibility of the zf ventricle using 10x and 5x zoom and a pear-shaped ventricle, respectively from left to right. As can be seen, in the 5x zoom image the borders of the ventricle cannot be identified. However, in the 10x image in the middle both chambers can be seen. The rightest image is an example of a pear-shaped ventricle.

3.3. Dataset

In this study, a training dataset was constructed using raw microscopic videos of zebrafish, comprising a total of 410 pixel-wise annotated images. To create this dataset, 41 videos of the lateral view from 41 different 2-dpf zebrafish were analyzed. Specifically, 9 of these videos were obtained from the Nrap mutant line, 19 from the Heterozygous line, and the remaining 9 from the wild type variant. It is noteworthy that during the process of manual segmentation, the experts responsible for the task were provided with videos with randomly generated names which hide the label identifying the fish's genotype. This will make sure that they would not have a bias while performing the segmentation. From each video, 10 sequential frames were extracted, resulting in a total of 410 frames for the training set. Each training set consisted of an original frame extracted from the video and a manually created mask showing only the ventricle, using ImageJ software. Following mask creation, all image and mask sets were organized into folders, with each set containing two folders: one for the extracted original frame and the other for its corresponding mask. For the validation set, two end-systolic (ES) and two end-diastolic (ED) frames were extracted and manually segmented from each video, resulting in a total of 82 images with their corresponding masks. This will make sure that the validation set is independent from the training set.

3.4. Data augmentation

In semantic segmentation, data augmentation is a technique used to increase the size of the training dataset by creating new training examples from the existing ones [30]. This is achieved by applying a range of transformations to the original training images, resulting in new images that are still representative of the same underlying scene or object, but with variations in appearance. Data augmentation is commonly used in deep learning-based computer vision tasks, including semantic segmentation, to prevent overfitting and improve the generalization capability of the model. By augmenting the training data, the model is exposed to more variations in the input data, leading to improved performance on new data. Some common image transformations used for data augmentation in semantic segmentation include horizontal and vertical flipping, rotation, scaling, cropping, and color jitter [31]. These transformations can be applied randomly or systematically during the training process to generate a diverse set of training examples.

In the original ZACAF implementation no augmentation was used since all the videos were recorded with a uniform placement under the camera. However, in this dataset the orientation of the fish was random. This augmentation assures that the framework is less prone to the manner that the fish is placed under the microscope and cameras setup, which gives the user more freedom while recording. Additionally, data augmentation is one of the methods used for improving the performance with limited data and reducing overfitting. Lastly, using data augmentation enables Test Time Augmentation (TTA) which is discussed in the next section. Here only horizontal and vertical flipping transformations have been used to imitate all the possible fish placements during the recording process.

3.5. Transfer learning

Transfer learning is a machine learning technique where a pre-trained model, typically trained on a large dataset, is used as a starting point for a new task or problem [32]. The pre-trained model has already learned a set of feature representations that are applicable to a wide range of problems, and these learned features can be transferred and fine-tuned to the new problem with a smaller dataset. This allows for faster training and better performance than training a model from scratch on the new dataset. If transfer learning is not utilized, the model will be initialized with random weights. In this case, the original ZACAF model was trained on a dataset of zebrafish recorded by a different group using a different microscope setup. The dataset included less mature fish and different mutant types. However, the features learned during the original model's training for ventricle segmentation from zebrafish are expected to improve the training for the new dataset. Therefore, we employed the pre-trained weights from the original model for the new model.

3.6. Test Time Augmentation

Test-time augmentation (TTA) is a technique used in computer vision, including semantic segmentation, to improve the accuracy of models during inference. In semantic segmentation, TTA involves applying various image transformations or augmentations to the test images and feeding them through the model multiple times to obtain an ensemble of predictions. These predictions are then combined, typically by averaging or voting, to obtain a final segmentation map. TTA helps to account for variability in the test data and reduces the risk of overfitting to the training data. By using TTA, a model can be more robust and accurate on previously unseen data. Some common image augmentations used in TTA for semantic segmentation include flipping, rotating, scaling, and cropping. These transformations create multiple versions of the same image, which can be fed through the model to obtain a diverse set of predictions.

It is important to note that TTA can increase the computational cost of inference since the model needs to be run multiple times for each image. However, the benefits in terms of accuracy improvement can often outweigh this cost. Moshkov et al incorporated TTA in the task of semantic segmentation of single-cell analysis of microscopy images based on U-net and Mask R-CNN deep learning models which showed improvement in the prediction accuracy [33]. A set of test time augmentation techniques was then applied to the test image to generate multiple predictions, which were subsequently combined to obtain a final prediction. Specifically, horizontal flipping, vertical flipping, and a combination of both were utilized to create three additional variations of the original test image. For each of these variations, a prediction was obtained using the semantic segmentation model. The four predictions were then combined by taking their element-wise average and thresholding the result with a value of 0.2.

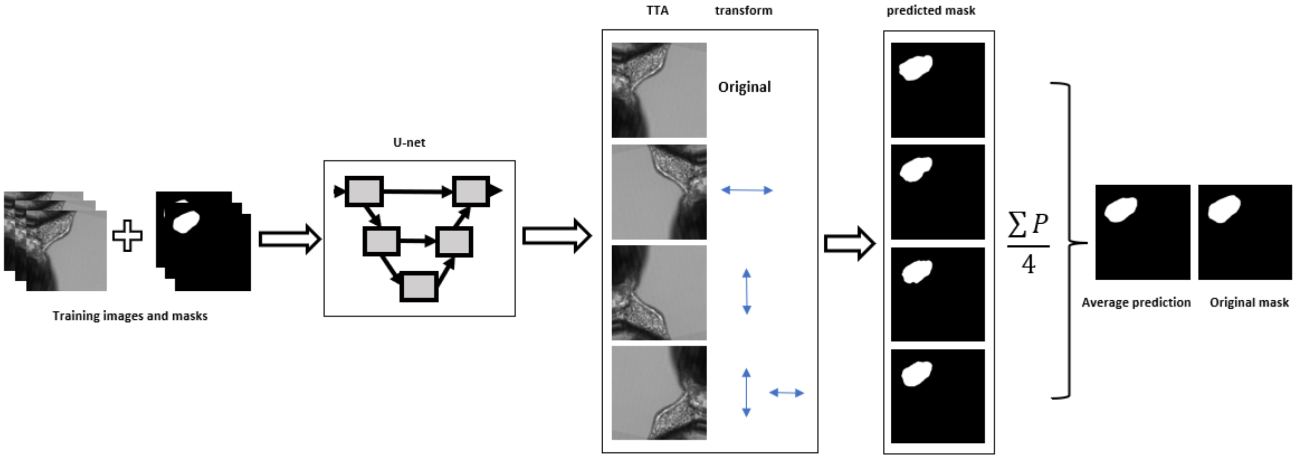

Figure 7: Implementation of the test time augmentation techniques. The U-net architecture is trained by augmented dataset composed of images and their corresponding masks and the TTA will be applied to the test set's output. The transforms used in TTA are horizontal and vertical flipping and their combination. These transformations along with the original prediction make 4 images which then would be subjected to an element-

wise average. On the far left of the figure the final prediction resulted from the TTA can be compared with the original manually segmented mask.

3.7. Assessment of the EF in Nrap deficient zebrafish

N=41 Zebrafish were recorded and genotyped via qPCR with TaqMan Custom SNP Genotyping Assays, resulting in 19 heterozygotes (46%), 9 wild type (22%), and 13 mutants (31%). After measurement of EF using the modified ZACAF a one-way ANNOVA test revealed no significant differences between Ejection Fraction and Fractional Shortening in all three genotypes. (Figure 9) Observations of "pear shaped" ventricles were made at the time of image processing (figure 7) and we found no significant correlation. Out of n=41 zebrafish, 11 abnormal ventricles were observed yielding a frequency of 0.27. Within the group of abnormal ventricles, 18% were mutant, 36% were heterozygous and 45% were wild type.

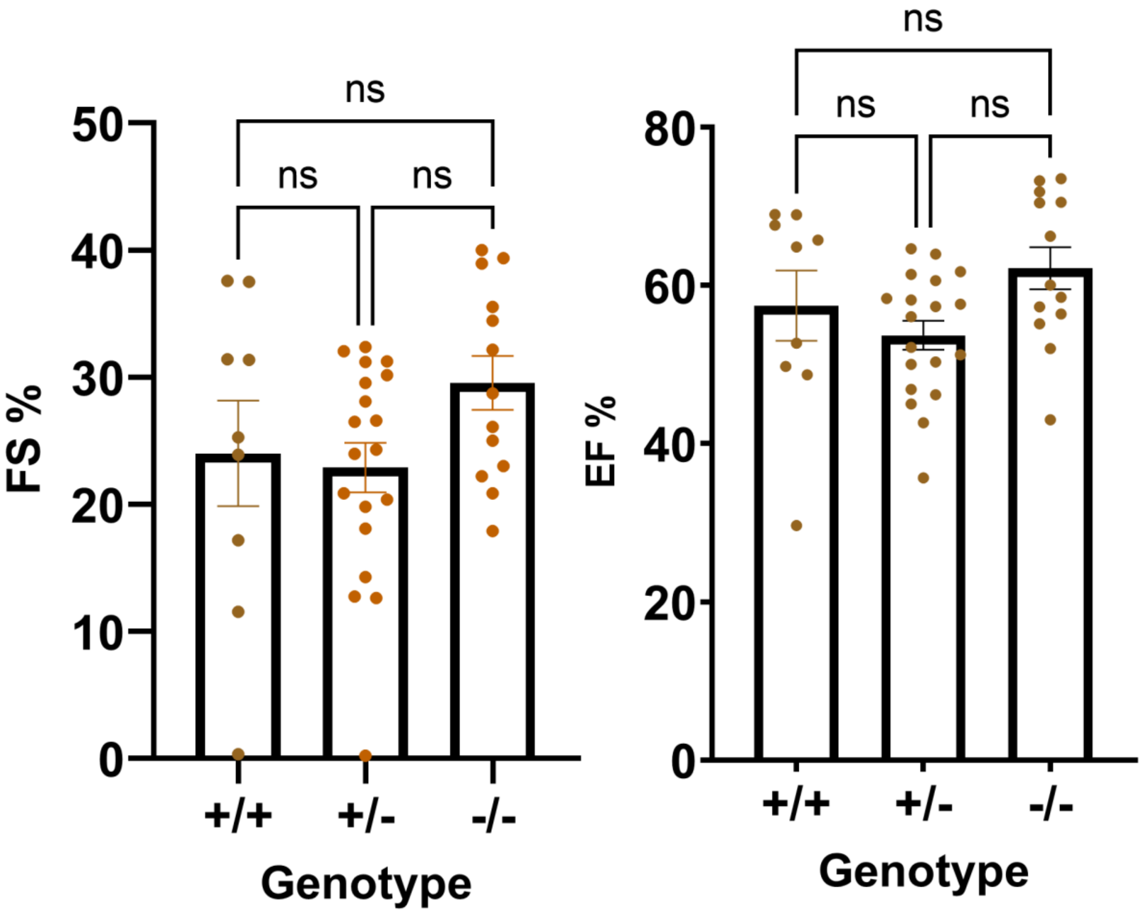

Figure 8: NRAP Zebrafish Model 2dpf Ejection Fraction and Fractional Shortening From 9.29.22 with one-way ANOVA

3.8. Assessment of the performance of the model with the defined metrics

The IoU metric serves as a performance assessment for the best-performing model, trained utilizing a Dice loss function, an Adam optimizer, and a learning rate of 0.001 with decay steps of 240 and a decay rate of 0.95. The validation split encompassed 20% or 80 sets. The identical model underwent training on both ZACAF original data and the new nrap dataset, with and without the application of TL. For TL, the training continued the original ZACAF model using only half of the nrap data, demonstrating the efficacy of TL in the context of a limited dataset. Figure 10 depicts the training and validation Dice loss and IoU metric for five different training regimens: ZACAF trained on its original data with and without data augmentation, trained on the nrap data, and ZACAF original fine-tuned on half of the nrap dataset. To ensure model integrity, a model checkpoint was implemented as a callback to retain the model with the highest validation IoU coefficient. In Figure 10, the training and validation IoU rates for the original ZACAF were 88.1% and 85.1%, respectively, increasing to 92.4% and 91.8%, respectively, after applying data augmentation to the training data. Further, a minor overfitting is observed with the metrics being close. This underscores the positive impact of data augmentation on the model. Additionally, training the model on half of the nrap data resulted in IoU rates of 91.9% and 87.6% for training and validation, respectively. Comparatively, employing TL on half of the data with the original ZACAF model yielded rates of 91.6% and 85.9%, respectively. Notably, the model trained solely with the complete nrap dataset exhibited similar performance. The figure 10 highlights that utilizing a small dataset (around 200 sets) with TL yielded comparable results to a much larger nrap dataset, emphasizing the potential of TL as a framework for easy access to ZACAF while simultaneously enhancing its performance.

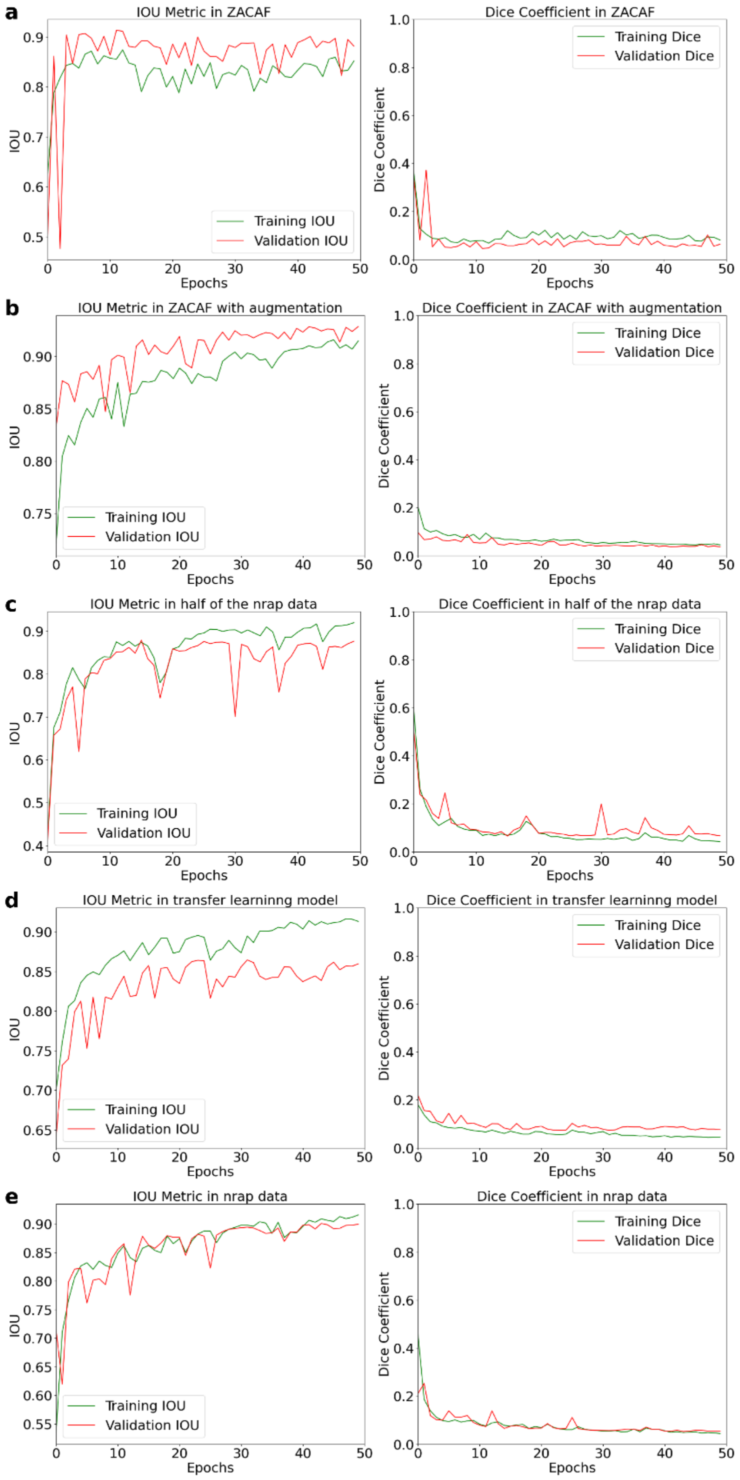

Figure 9 Comparison of the performance of the models in their dice loss and IoU metric during training and validation. a) ZACAF model trained on its original data. b) ZACAF model trained on its original data with data augmentation. c) model trained on half of the nrap data only. d) model trained on half of the nrap data using TL by taking the ZACAF model pretrained weights. e) model trained only on the complete nrap dataset.

3.9. Assessment of the performance of the framework for EF in the test set

To evaluate the framework's effectiveness in calculating EF, we compared the results obtained from manual assessment by an expert with those generated by the software. As EF calculation requires finding the area in all frames of a video to determine the ED and ES areas, the framework's performance was assessed using a series of frames from a test video, rather than random images from a validation set. To this end, we evaluated the framework's performance with EF calculation, using 2 wild-type zebrafish embryos and 2 nrap mutant embryos as inputs for the test set, without using them in training. We first performed manual processing and estimation for each video to derive EF as the ground truth. Then, the model predicted ventricle masks for each frame of the input video, and the frames with the maximum and minimum area of the segmented ventricle were identified as the ES and ED frames, respectively. It is worth mentioning that the outer edge of the ventricles was identified in the manual segmentation process that created the training dataset, thus the model evaluation and results were also based on that fact. The framework subsequently computed EF and saved it, along with other indices, in a CSV file. Among the five models discussed in the preceding section, three were selected for testing on videos. These models were trained on half of the nrap data, trained on the full nrap data, and the TL model using half of the nrap data on the original ZACAF weights. Each of these models was independently employed to calculate three sets of EF and FS. Additionally, test-time augmentation (TTA) was applied to all three model predictions in a separate experiment, resulting in a total of six sets of results. In Figure11, two distinct plots illustrate the cumulative error in four test videos during the calculation of EF and FS.

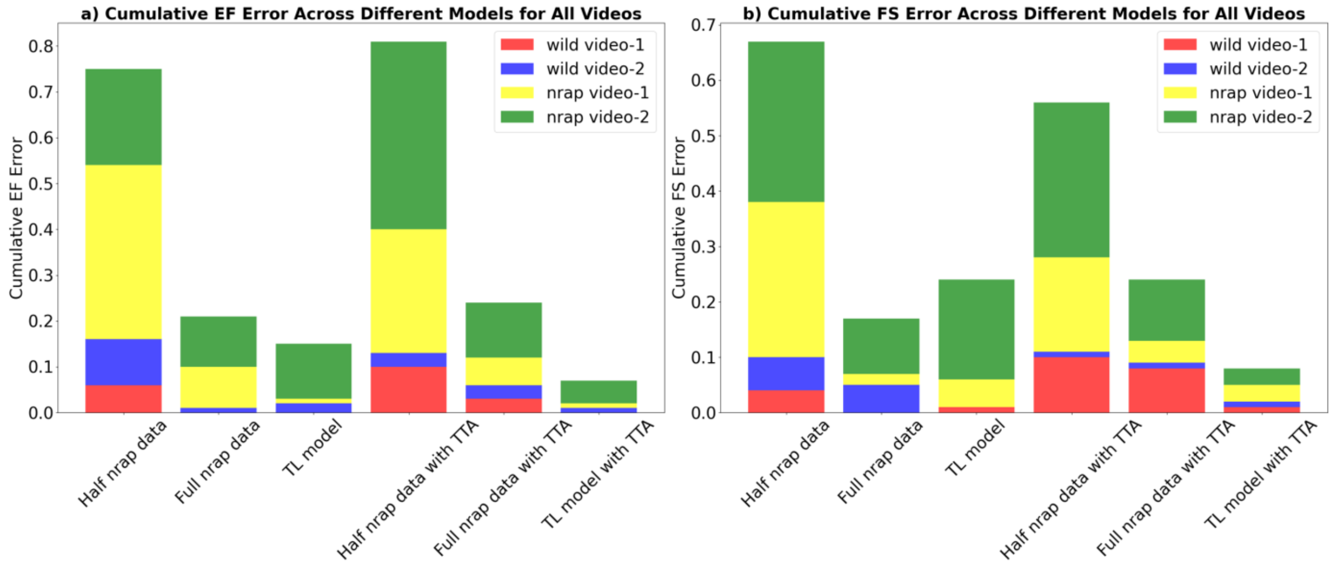

Figure 11: EF (a) and FS (b) were calculated from 4 test videos and the error with the manual calculation from all videos was added. This error can be seen within different models and the colors show the test video so the performance of each model can be seen for each video. model trained on half of the nrap data only, model trained on only on the complete nrap dataset, model trained on half of the nrap data using TL, with and without TTA can be seen on the X axis of the plot, respectively. The legend shows the contribution of each video to the cumulative error compared to the manual calculation.

Firstly, the model trained with half of the nrap data exhibited the highest error, which was expected given the limited training data (200 sets) for a complex convolutional neural network (CNN). Conversely, the model trained on the complete nrap dataset demonstrated significantly lower error. Secondly, the TL model, utilizing the original ZACAF model as pretrained weights and continuing training with half of the nrap dataset, showcased the effectiveness of TL and the feasibility of utilizing such a small dataset. This finding suggests that new users of ZACAF can leverage it by creating a small dataset. Additionally, test videos from the original ZACAF dataset were tested using all models, with only the TL model successfully segmenting the ventricle. Notably, Figure 11 highlights that TTA has substantially reduced the error. TTA not only enhances edge accuracy but also proves beneficial when the model mistakenly detects the atrium or other tissues. Maintaining focus on the ventricle is challenging in the microscopic recording of zebrafish heart. TTA can address scenarios where both chambers are partially in focus, ensuring only the ventricle is detected. Figure 12 provides an example of TTA's application in improving segmentation in videos where both chambers are in focus. Therefore, a notable advantage of employing TTA is that it facilitates the recording of future datasets with greater ease, reducing the need for extensive cleaning and supervision. The model incorporating TL and TTA demonstrated the best performance in the test set, with average errors of 2% and 1.7% in EF and FS, respectively, in test videos. Additionally, an intriguing observation from Figure 11 indicates that mutant nrap fishes exhibit higher error.[34]

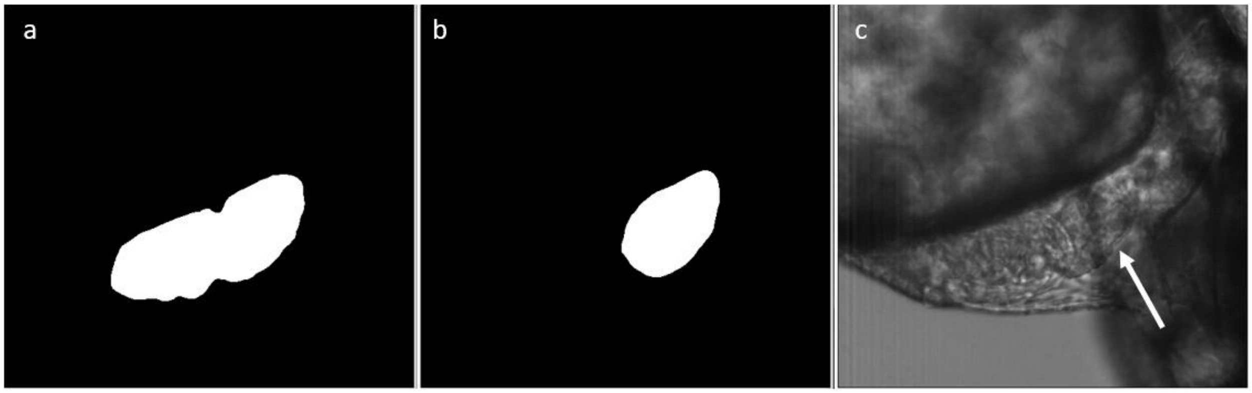

Figure 12: a frame from nrap fish video recording where the ventricle and atrium are both visible. a) prediction without TTA. b) prediction after TTA. c) original video frame with an arrow showing the ventricle

Chapter 4: Video segmentation and temporal information

Zebrafish has emerged as a prominent model organism in various biological studies, particularly in cardiovascular research and genetic screenings. However, the current methods for quantifying and monitoring cardiac functions in zebrafish embryos often entail laborious manual efforts and yield inconsistent estimations. In response to this challenge, the development of automated assessment frameworks holds significant promise. Leveraging video segmentation methods and harnessing temporal information can offer valuable enhancements to the Zebrafish Automatic Cardiovascular Assessment Framework (ZACAF). This fact particularly shows itself during manual creation of the dataset, where it becomes clear that looking at the image in context of the neighboring frames significantly helps the manual segmentation task. Considering virtually all deep learning-based models proposed for zf heart function assessment analyze the problem in a single image based method.

Heart beating videos captured from zebrafish embryos exhibit periodicity, making them inherently rich in temporal information. Utilizing video segmentation methods enables the extraction of dynamic spatial and temporal features, which are crucial for accurate assessment of cardiovascular indices such as ejection fraction (EF) and fractional shortening (FS). While manual segmentation techniques require expert annotation and meticulous attention to temporal features, automated segmentation frameworks empowered by deep learning models offer a more efficient and consistent alternative.

By incorporating temporal information into the segmentation process, ZACAF can capitalize on the inherent rhythmic nature of heart beating videos, facilitating precise delineation of cardiac structures across frames. This not only streamlines the assessment process but also ensures robustness and reliability, particularly when dealing with large volumes of video data. Furthermore, the versatility of ZACAF, capable of operating with black and white microscopic recordings at varying frame rates, underscores its applicability across diverse laboratory settings and research infrastructures.

In this chapter, we explore the potential synergies between video segmentation methods and temporal information in enhancing ZACAF's capabilities for automated cardiovascular assessment in zebrafish embryos. The methods for capturing and learning temporal features in deep learning models proposed here can be beneficial not only for zf videos but for all video segmentation or object recognition tasks. Through systematic validation and comparison with manual processing, we demonstrate the efficacy and accuracy of our approach, laying the foundation for efficient, consistent, and reliable analysis of temporal features in video segmentation.

4.1. Literature review for video segmentation

Temporal consistency in video segmentation is a crucial aspect of advancing applications such as self-driving cars, robotics, and augmented reality. The process of creating a 3D semantic map from video sequences starts with generating 2D semantic maps for each frame, where each pixel is assigned a label that defines its semantic category. While the accuracy of individual frame predictions is essential, the consistency of these predictions across consecutive frames is equally important. Consistent semantic labeling across frames facilitates more accurate and efficient fusion of these maps, leading to a more reliable 3D representation of the environment.

Deep learning models have made significant improvements in improving the temporal consistency of video segmentation. Traditionally, optical flow has been employed to capture the movement of pixels across frames, thus maintaining coherence in the segmentation results. However, computing optical flow is computationally demanding and its accuracy directly impacts the segmentation model's performance. To overcome these limitations, researchers have developed sophisticated methods that leverage temporal information from past frames without solely depending on optical flow. These methods include using temporal consistency losses, integrating memory modules, and designing architectures that can inherently handle temporal information. By incorporating these advanced techniques, the goal is to ensure that the segmentation models produce stable and coherent results over time, thereby enhancing the overall effectiveness of video-based applications.

Below is a literature review for temporal consistency in video semantic segmentation. In each subsection below one of the recent papers addressing temporal consistency in video segmentation is discussed along with its downside.

1) An Unsupervised Temporal Consistency (TC) Loss to Improve the Performance of Semantic Segmentation Networks

This study[35] focuses on enhancing the performance of semantic segmentation networks by introducing an Unsupervised Temporal Consistency (TC) Loss. Leveraging a sequential and unlabeled dataset comprising video sequences, they employ optical flow functions to gauge the stability of network predictions and estimate apparent motion within the video sequence. Optical flow enables the estimation of pixel displacement between consecutive frames, providing crucial insights into temporal dynamics essential for improving segmentation network performance.

Downsides: This framework uses optical flow which is susceptible to computational complexity and to errors in scenarios with fast motion, occlusions, or texture less regions, which can negatively impact the accuracy of temporal consistency in video segmentation. Also, temporal consistency (TC) is derived from a separate task and dataset, resulting in video segmentation and optical flow estimation being treated as distinct processes rather than an integrated solution.

2) Every Frame Counts: Joint Learning of Video Segmentation and Optical Flow

This paper[36] introduces a pioneering framework for concurrent video semantic segmentation and optical flow estimation. The model takes a pair of sequential images, randomly selected from adjacent video frames, as input. It employs a shared encoder for both decoders: one dedicated to segmentation and the other to optical flow. Consequently, two predictions are generated for segmentation, alongside two pseudo-predictions obtained by applying flow features to the encoded feature maps in each segmentation decoder block—essentially wrapping feature maps with flow information. Subsequently, a temporal consistency loss is defined, leveraging the segmentation feature map and the feature map resulting from the fusion of flow and current frame subtraction. Additionally, an occlusion map is utilized to prevent penalizing occluded pixels, enhancing the model's robustness and accuracy.

Downsides: This paper again uses optical flow which is vulnerable to the mentioned issues like computationally challenging and non-robustness.

3) Frame Difference-Based Temporal Loss for Video Stylization

Neural style transfer models have proven effective in stylizing regular videos according to specific styles. However, maintaining temporal consistency between frames has been a challenge. In addressing this issue, a simpler temporal loss, termed the frame difference based (FDB) loss, has been proposed[37]. This loss function aims to mitigate temporal inconsistencies by quantifying the disparity between stylized frames and original frames. It computes the distance between the differences observed in both pixel space and feature space, as defined by convolutional neural networks, thereby ensuring more coherent stylization across successive frames.

Downsides: a separate video stylization model is needed. Also, it is not suitable for when stylization is not needed.

4) AuxAdapt: Stable and Efficient Test-Time Adaptation for Temporally Consistent Video Semantic Segmentation

The approach[38] incorporates a small auxiliary segmentation network, referred to as AuxNet, which fine-tunes the decisions made by the original segmentation network (Main-Net) by incorporating its own estimations alongside those of MainNet. During each frame iteration, only AuxNet undergoes updates via back-propagation, while MainNet remains fixed. The rationale behind this strategy lies in addressing temporal inconsistency attributed to uncertainty. By training the network based on its own challenging decisions, it strengthens its confidence in predictions, particularly for image regions resembling those encountered previously. Furthermore, AuxNet is trained via test-time adaptation while Main-Net remains frozen, facilitating improved adaptability and robustness in real-world scenarios.

Downside: In this network an additional network is needed. Also, two separate training stages are needed.

5) Efficient Semantic Video Segmentation with Per-Frame Inference

Here a novel approach is proposed[39] for transferring temporal consistency knowledge from large models to smaller ones through Temporal Consistency Knowledge Distillation (TCKD). This method leverages previous predictions as supervised signals to assign consistent labels to each corresponding pixel along the time axis. Termed Motion Guided Temporal Consistency, this approach utilizes a temporal loss mechanism, subtracting predictions at time from wrapped predictions on a motion estimation network. This process aligns segmentation maps between two input frames by leveraging motion guidance. Additionally, an occlusion mask was introduced to mitigate noise stemming from warping errors. In this implementation, a pre-trained optical flow prediction network as the motion estimation net was proposed. Moreover, attention operators to assess similarity between pairs and multiple frames were integrated, thereby facilitating the training of a compact student network. This comprehensive framework aims to distill temporal consistency knowledge effectively, enabling smaller models to achieve comparable performance to larger counterparts in dynamic environments.

Downside: This has the mentioned issues of using optical flow. Here, the optical flow network is pretrained and might show errors with new domains of video.

6) Domain Adaptive Video Segmentation via Temporal Consistency Regularization

This paper [40]introduces DA-VSN, a domain adaptive video segmentation network designed to bridge domain gaps in videos through Temporal Consistency Regularization (TCR) across consecutive frames in the target domain. DA-VSN comprises two innovative and complementary components: cross-domain TCR and intra-domain TCR.

In the cross-domain TCR framework, the network leverages adversarial learning to guide predictions of target frames towards achieving temporal consistency akin to that observed in source frames, learned from annotated source data. This involves employing a GAN architecture where the generative model learns to predict two consecutive frames from both target and source domains. Dual discriminator structures are employed: one focuses on spatial alignment of single video frames across different domains, while the other concentrates on temporal alignment of consecutive video frames from different domains. To ensure consistency across spatial losses, a divergence loss is introduced between the two losses.

Conversely, the intra-domain TCR framework propagates predictions from previous frames forward using frame-to-frame optical flow estimates. It then enforces consistency in unconfident predictions of the current frame with confident predictions propagated from the previous frame.

In summary, the paper presents distinct frameworks for enhancing temporal consistency: one for cross-domain adaptation and another for intra-domain refinement. These frameworks collectively contribute to the robustness and adaptability of DA-VSN in addressing domain gaps and enhancing the accuracy of video segmentation tasks.

Downsides: In the cross-domain segment, the superiority of using a dual discriminator network over transfer learning remains ambiguous. Considering the computational intensity and complexity involved, it begs the question of whether the marginal gains justify such an investment. Additionally, the absence of metrics showcasing enhancement in temporal features, apart from mIoU, leaves room for improvement. Intra-domain analysis employs optical flow yet encounters conventional challenges. The absence of temporal metrics to demonstrate improvements further underscores this concern.

7) Learning to Associate Every Segment for Video Panoptic Segmentation

This paper[41] with the objective of simultaneously learning coarse segment-level matching and fine pixel-level matching. A deep Siamese model is designed, employing two identical networks to assess similarities between inputs, trained on pairs of frames where the neighboring reference frame is randomly sampled from a wide range relative to the current target frame. This setup encourages the model to acquire representations conducive to optimal content association across input frames.

For segment-level correspondence, the authors introduce a temporal associative embedding loss, encompassing class-wise and instance-wise contrast mechanisms. This facilitates the learning of meaningful associations between segments across frames. Additionally, at the pixel level, optical flow is utilized to enhance matching precision, aiding in fine-grained alignment between frames.

By integrating both segment-level and pixel-level matching strategies within a unified framework, the proposed method offers a comprehensive solution for video panoptic segmentation, enabling robust and accurate segmentation across consecutive frames.

Downsides: This will not work as suitable for binary classification. For pixel-wise correspondence they are still using optical flow which still has the mentioned problems.

8) Simultaneously Short- and Long-Term Temporal Modeling for Semi-Supervised Video Semantic Segmentation

This paper[42] aims to leverage both short- and long-term inter-frame correlations effectively. Here's how the method accomplishes this:

Spatial-Temporal Transformer (STT) Module: This module is dedicated to handling short-term temporal modeling. It operates on the feature maps of the query frame and its adjacent frames (up to n frames). By integrating spatial and temporal information, the STT module enables accurate modeling of short-term correlations between consecutive frames.

Reference Frame Context Enhancement (RFCE): The RFCE module is designed to capture long-term temporal correlations by optimizing the context for both the query frame and the reference frame. It ensures that frames distant from each other within the same video are appropriately accounted for, enhancing the model's ability to understand temporal dynamics over longer durations.

In addition to RFCE, the paper introduces the Global Category Context (GCC) module to address potential deficiencies in categorical information present in reference frames compared to query frames. By compensating for these discrepancies, the GCC module further enhances the effectiveness of the RFCE module in modeling long-term temporal correlations.

By integrating these components, the proposed approach offers a comprehensive solution for semi-supervised video semantic segmentation, effectively capturing both short- and long-term temporal dependencies for improved segmentation accuracy and robustness.

Downsides: downside of this approach is the extensive use of transformers, which leads to high computational cost and increased complexity of the proposed model. This can make the method less efficient and more challenging to deploy in real-time applications or on resource-constrained devices.

4.2. Approaches for capturing temporal consistency

Here we expand on some of the methods already discussed in the literature for leveraging temporal features in video object segmentation.

4.2.1. Optical Flow

Optical flow is a fundamental technique in computer vision used to estimate the motion of objects between consecutive frames of a video. By capturing the apparent motion of brightness patterns, optical flow provides a dense vector field representing the displacement of pixels over time. This section delves into the principles, algorithms, and applications of optical flow in the context of temporal analysis for heartbeat video segmentation.[43]

Optical flow assumes that the intensity of a pixel remains constant as it moves from one frame to the next. Mathematically, if I (x, y, t) denote the intensity of a pixel at position (x, y) and time t, the optical flow constraint can be expressed as:

$$I\ (x, y, t) = I\ (x+\Delta x, y+\Delta y, t+\Delta t)\ (14)$$

For small displacements, this can be approximated using a Taylor series expansion, leading to the optical flow equation:

$$\frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v + \frac{\partial I}{\partial t} = 0 \ \ (15)$$

Where u and v are the horizontal and vertical components of the optical flow vector, respectively. The partial derivatives $\frac{\partial I}{\partial x}$ and $\frac{\partial I}{\partial y}$ represent the spatial gradients, while $\frac{\partial I}{\partial t}$ represents the temporal gradient. Several algorithms have been developed to solve the optical flow equation, each with its own strengths and limitations. Two widely used methods are the Lucas-Kanade and Horn-Schunck algorithms. Optical flow can be leveraged to capture the dynamic movements of the heart, ensuring temporal consistency, and improving segmentation accuracy.

Additionally deep learning based methods have been also proposed for capturing optical flow. FlowNet [44]is a deep learning architecture designed to estimate optical flow, which represents the apparent motion of objects between consecutive video frames. The architecture uses an encoder-decoder structure: the encoder extracts hierarchical features from the input frames, and the decoder gradually upsamples these features to produce a dense optical flow map. By effectively capturing motion dynamics, FlowNet enhances the accuracy and efficiency of video segmentation tasks, where understanding object movement is essential.

4.2.2. Frame Differencing

Frame differencing is a straightforward and effective technique used in video processing to detect motion by calculating the difference between consecutive frames. This method is particularly useful in scenarios where the objective is to identify changes or movements within the video, such as the beating of a heart in medical imaging. Frame differencing is based on the principle that significant changes in pixel values between consecutive frames indicate motion or activity. By computing the absolute difference between the corresponding pixels of successive frames, regions of change can be highlighted, which often correspond to moving objects or dynamic structures within the scene. Mathematically, let $I_t(x,y)$ represent the intensity of a pixel at position $(x,\ y)$ in frame t. The frame difference $D_t(x,y)$ between frame t and frame t+1 is given by:

$$D_t(x,y) = |I_t(x,y) - I_{t+1}(x,y)| \ (16)$$

Where |·| denotes the absolute value. If $D_t(x, y)$ exceeds a predefined threshold θ, the pixel is considered to be part of a moving region.

4.2.3. Temporal correlation

Temporal correlation is a statistical technique used to measure the similarity between consecutive frames in a video. By quantifying how closely related frames are over time, temporal correlation helps in maintaining the temporal coherence of segmented regions, which is particularly important in the dynamic context of heartbeat videos.

Temporal correlation measures the degree to which pixel values or extracted features in one frame are similar to those in the subsequent frame. A high correlation indicates that the frames are similar, suggesting minimal motion or change, while a low correlation indicates significant differences, corresponding to motion or activity within the scene.

Mathematically, the temporal correlation coefficient ρ between two frames I_t and I_{t+1} at a specific pixel (x, y) can be calculated using the Pearson correlation formula:

$$\rho(I_t, I_{t+1}) = \frac{\sum_{i,j}^{(I_t(x,y)-u_t)(I_{t+1}(x,y)-u_{t+1})}}{\sqrt{\sum_{i,j}^{(It(x,y)-u_t)^{2}} \sum_{i,j}^{(I_{t+1}(x,y)-u_{t+1})^2}}} \quad (17)$$

where u_t and u_{t+1} are the mean pixel values of frames I_t and I_{t+1}, respectively, and (x, y) denotes the pixel coordinates. Temporal correlation helps ensure that segmented regions remain consistent over time by identifying areas with high correlation. This reduces the likelihood of sudden changes or flickering in the segmentation results.

4.2.4. Architecture for conserving temporal consistency

Aside from auxiliary networks designed for learning temporal features, Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks are powerful tools for modeling sequential data and capturing temporal dependencies.[45] Their ability to process sequences of frames makes them particularly suitable for analyzing dynamic phenomena such as the beating of a heart in medical videos. RNNs are designed to handle sequential data by maintaining a hidden state that captures information from previous time steps. This enables RNNs to model temporal dependencies and patterns over time. However, standard RNNs suffer from limitations such as the vanishing gradient problem, which makes it difficult for them to learn long-term dependencies.

LSTM networks are a specialized type of RNN designed to overcome these limitations. LSTMs introduce memory cells and gating mechanisms (input gate, forget gate, and output gate) that regulate the flow of information, allowing them to maintain and update long-term dependencies effectively.

On the other hand, using transformers for learning temporal features in video segmentation involves employing their ability to model relationships across an entire sequence of frames through self-attention mechanisms. Transformers can effectively capture long-range dependencies, allowing the model to consider contextual information from distant frames, which is crucial for maintaining temporal consistency in video segmentation. By processing video frames as sequences, transformers can learn intricate temporal dynamics and spatial-temporal correlations, leading to more coherent segmentation across frames. This can significantly improve the accuracy of the segmentation results, especially in complex scenes where temporal information is critical.

4.3. Frame similarity methods

Here, we proposed an alternative approach to temporal feature extraction. Instead of using a separate deep learning network for extracting temporal connections like optical flow, we try to use frame based statistical correlation to measure similarities between a series of sequential frames like what was proposed for stylized videos and the use of frame subtraction. Using a metric for image similarity our final aim here is to propose a temporal loss function that does not require an auxiliary deep learning model and can be applied to most existing segmentation models.

4.3.1. Correlation of Fast Fourier Transform (FFT)

FFT, a mathematical tool, facilitates the transformation of images from the spatial domain to the frequency domain. By applying FFT to consecutive frames in a video sequence, temporal frequency information can be extracted. This enables the detection of temporal patterns, motion, and dynamics across frames. The comparison of FFT representations between frames reveals similarities and differences in frequency content, which in turn, aids in identifying regions of motion and temporal coherence over time.

Correlation, on the other hand, measures the similarity between two signals or images. When applied to consecutive frames, correlation elucidates the degree of resemblance between corresponding pixels. By computing the correlation between adjacent frames, one can discern regions of similarity or change over time. Utilizing correlation in either the spatial or frequency domain allows for the detection of motion and temporal dependencies, thereby facilitating more robust video segmentation.

The 2D Fast Fourier Transform (FFT) of an image $I\ (x, y)$ can be expressed as:

$$\mathrm{F(u,v)} = \sum_{\mathrm{M-1}}^{\mathrm{x=0}} \sum_{\mathrm{N-1}}^{\mathrm{y=0}} \mathrm{I(x,y)} \cdot \mathrm{e}^{-2\pi\mathrm{i}(\frac{\mathrm{ux}}{M}+\frac{\mathrm{vy}}{\mathrm{N}})} \quad (18)$$

Where, M and N are the dimensions of the image in the horizontal and vertical directions, respectively.

To show if FFT correlation can act as a temporal similarity measure between two images, a python script was written to get FFT of every frame of a video and then calculate the correlation between the first frame to rest of the frames of the video.

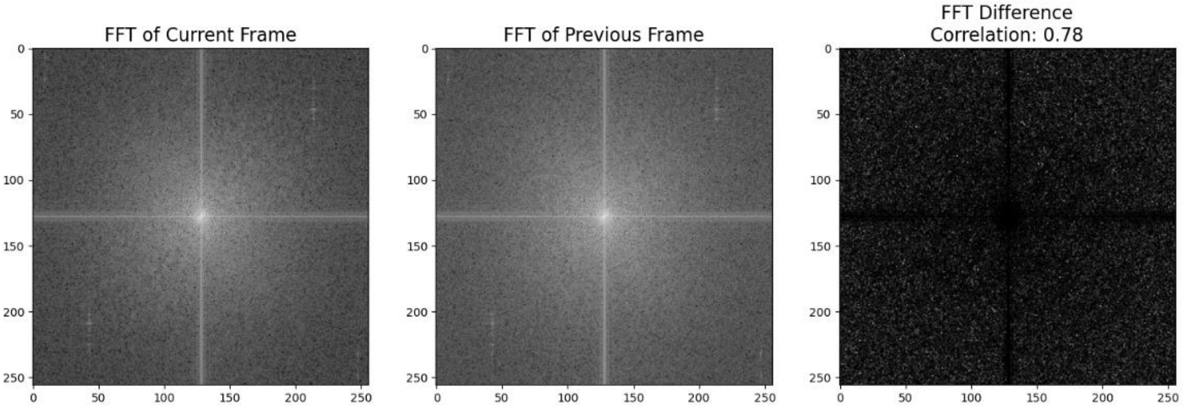

Figure 13 FFT of two consecutive frames from a video along with the difference of the two FFTs and their correlation. The information of edges and objects can be extracted using FFT.

As can be seen in figure 14, a periodic trend can be seen in the plot of FFT correlation of frames with the first frame however, it is not very consistent, and the difference of correlation is not significant.

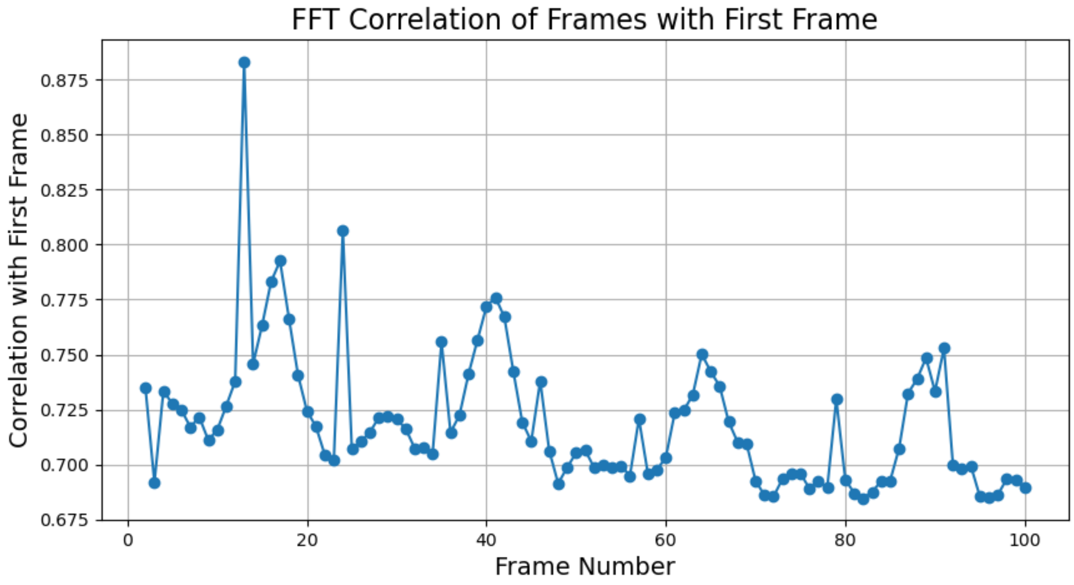

Figure 14 Plot of FFT correlation of the sequence of frames with the first frame. As it can be seen in the figure multiple local maximus are present which have correlation with the heartbeat of the zf in the video. However, there are also multiple outlier points that can be seen which are noise.

4.3.2. The Structural Similarity Index

The Structural Similarity Index metric (SSIM) is a widely used metric for assessing the similarity between two images, aiming to model the human visual perception more closely than traditional metrics such as Mean Squared Error (MSE) or Peak Signal-to-Noise Ratio (PSNR). SSIM evaluates the structural information in an image by comparing three components: luminance, contrast, and structure. These components are designed to capture the perceptual differences that human eyes are more sensitive to.

The SSIM index between two images x and y is defined as follows:

$$SSIM(x, y) = \frac{(2\mu_x\mu_y+C_1)(2\sigma_{xy}+C_2)}{(2\mu_x+\mu_y 2+C_1)(\sigma_x^2+\sigma_y^2+C_2)} \quad (19)$$

C_1 and C_2 are small constants to stabilize the division. By comparing these components, SSIM provides a comprehensive measure that accounts for changes in structural information, luminance, and contrast. The SSIM value ranges from -1 to 1, where 1 indicates perfect similarity and -1 indicates complete dissimilarity.[46]

SSIM is particularly useful in image quality assessment, where it outperforms traditional metrics in capturing perceptual differences. Its application extends to various fields including image compression, transmission, and enhancement, where preserving perceptual quality is crucial. By focusing on structural information, SSIM aligns more closely with the human visual system, making it a valuable tool for evaluating image similarity in a manner that is both robust and perceptually meaningful.

We apply the same process we did with FFT here with SSIM. We get SSIM of every frame of a video and the first frame. Then we plot it through all frames in a video sequence. In figure 15 it can be seen that SSIM worked much better than FFT correlation for our intended task of measuring temporal consistency.

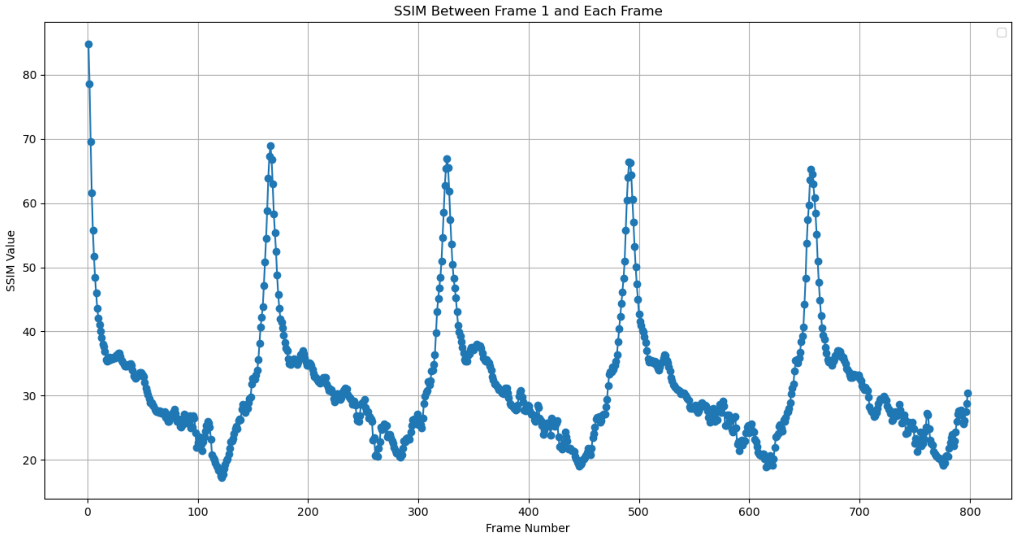

Figure 15 SSIM of the sequence of frames with the first frame. As it can be seen in the figure multiple local maximus are present which have correlation with the heartbeat of the zf in the video. Compared to the similar figure that was achieved by FFT correlation this signal shows more correlation with the heartbeat signal and less noise is observed.

4.4. Video based measurement using SSIM

As it can be seen in figure 15, the plot has a direct relation with the area of the ventricle. The background subtraction-based occlusion mask focuses on the movement in the video, and since the ventricle has the majority of the pixels involved in the movement, the plot represents the measurement of the area changes in the video. This shows that this framework using SSIM can be used for measuring movement changes in a video. However, the zf videos are very complicated in terms of noise and borders of the moving ventricle. Here, to show the ability of SSIM based framework we create an animation. In this animated video, we have a 256×256 video that has a black background with a white circle in the middle. This circle changes size periodically and considering that we created the animation we have the area of the circle over frames. Two frames of this animation are brought in figure 16 to demonstrate how the video looks.

The same SSIM based framework was applied to the created animation and figure 17 shows the correlation of the known area of the circle to SSIM of the first frame to all other

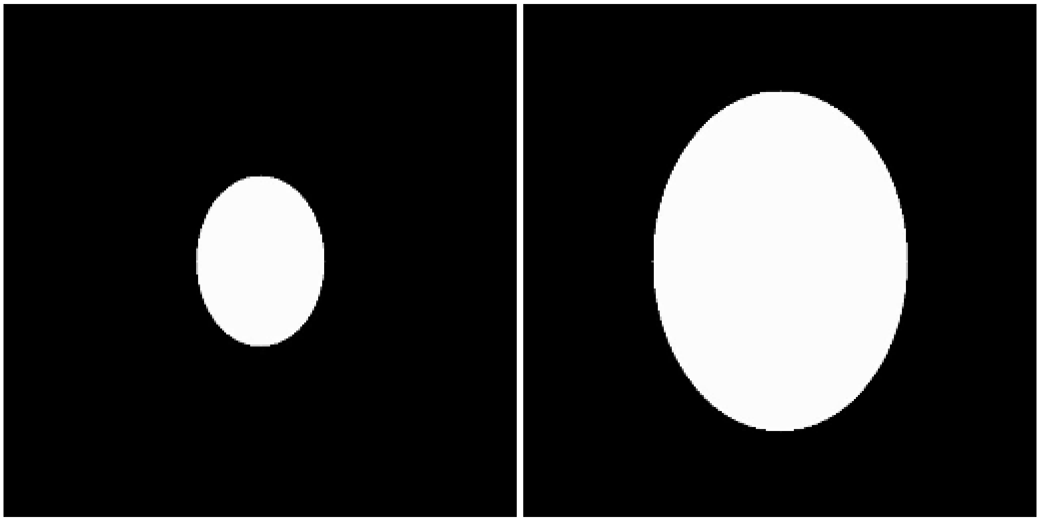

Figure 16 animation of the binary circle that changes in size periodically. We created this animation, so the size of the object is known.

frames.

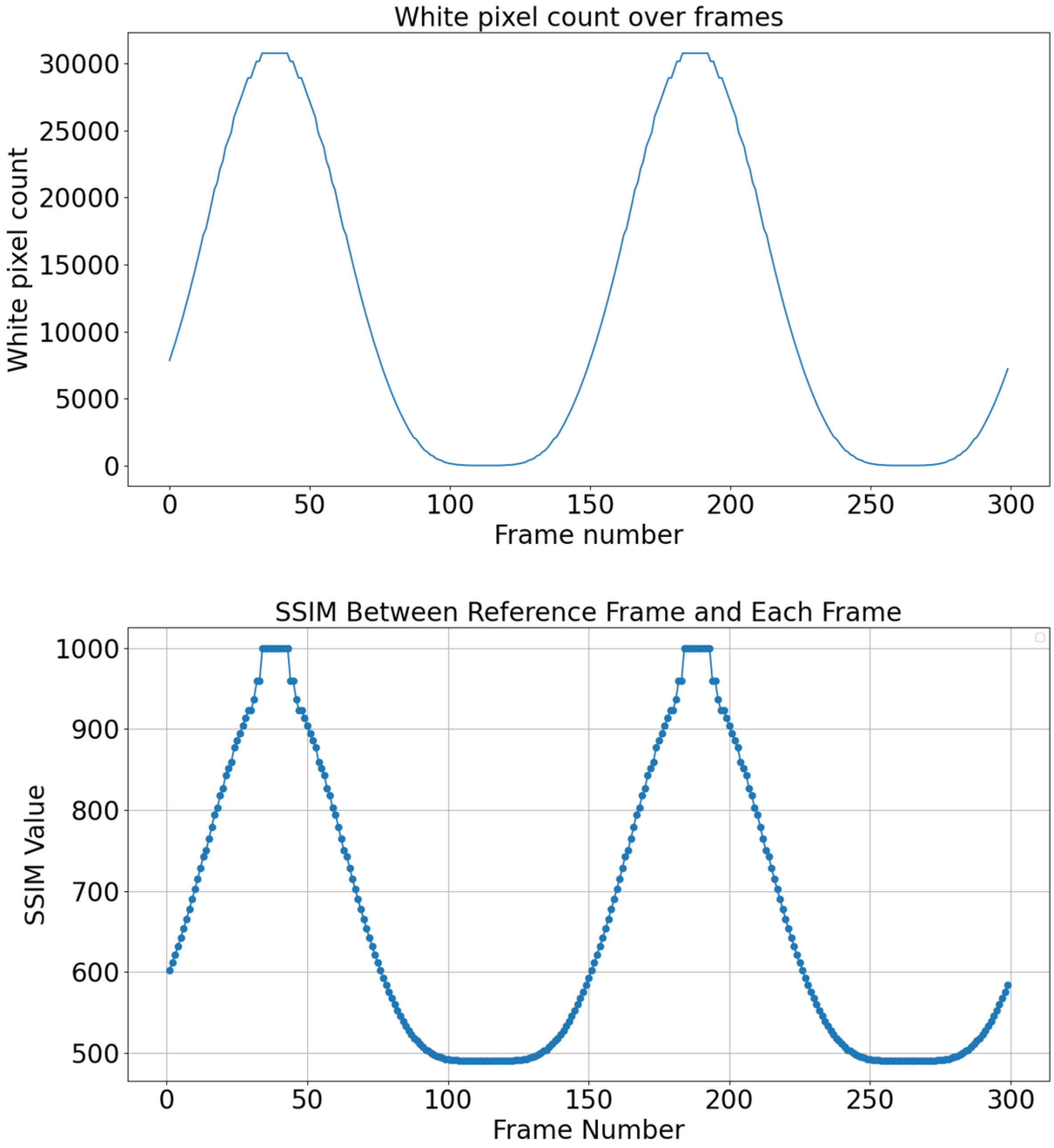

Figure 17 SSIM of the frames compared to the actual area of the circle. In the first panel the area of the circle throughout the video is plotted. The second panel shows the same SSIM method applied to the frame with the largest circle. As can be seen, the second panel shows correlation to the area of the circle. The top of the peaks is saturated when the circle being too large compared to the background.

4.5. Detection of Occlusions and Artifacts

Background subtraction with occlusion masks can help identify and mitigate occlusions or artifacts in heart beating videos. Occlusions such as debris or bubbles can obscure cardiac structures, leading to inaccuracies in segmentation. By applying background subtraction as occlusion masks, we can suppress the effects of occlusions, revealing the underlying cardiac structures more clearly. This improves the accuracy of segmentation and ensures that the deep learning model receives cleaner input data, leading to more reliable cardiovascular quantification results.

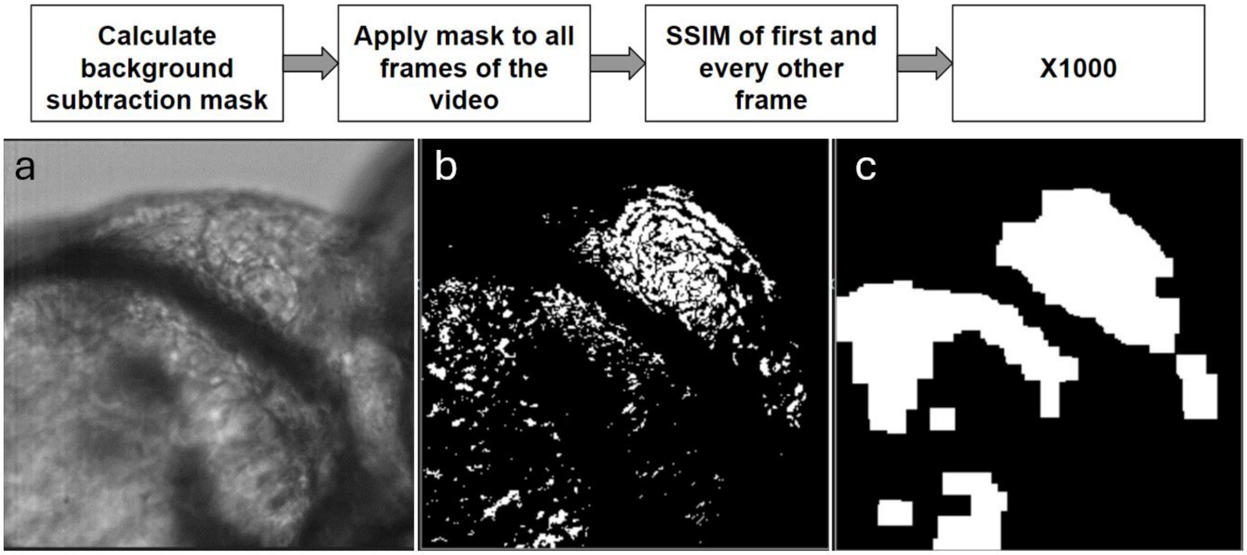
Calculate background subtraction mask
Apply mask to all frames of the video
SSIM of first and every other frame
X1000
a
b
c

Figure 18 Workflow of the SSIM based framework for area measurement where we apply background subtraction as an occlusion mask to minimize the noise. The result is multiplied by 1000 arbitrary to make the resulting numbers easier to read. On the bottom section, panel (a) shows a frame of a video. panel (b) shows the mask resulting from applying background subtraction to the video. panel (c) shows the result of applying morphological filters to the mask in panel (b).

Enhanced Feature Extraction: Background subtraction with occlusion masks enables the extraction of dynamic spatial and temporal features from heart beating videos. By subtracting the background and focusing only on the foreground (i.e., cardiac structures), we can highlight the motion and changes occurring in the video over time. These dynamic features can be valuable inputs for the deep learning model, allowing it to learn the temporal dynamics of cardiac activity and make more accurate predictions of cardiovascular indices such as ejection fraction (EF) and fractional shortening (FS).

4.6. Improving Model’s temporal consistency using SSIM

SSIM can be used to measure the structural similarity between consecutive frames in heart beating videos captured from zebrafish embryos. By calculating SSIM values between frames, we can assess the temporal consistency of the video sequence. High SSIM values indicate that the frames are similar, implying stable temporal characteristics such as regular heartbeat intervals. This information can be utilized to ensure the robustness and reliability of the segmentation process across frames.

By pre-processing heart beating videos with background subtraction with occlusion masks, we can provide the deep learning model with cleaner, more consistent input data. This can improve the model's training process by reducing noise and variability in the training data, leading to better generalization and performance on unseen data. Additionally, incorporating temporal information through SSIM and background subtraction can help the model capture the temporal dependencies inherent in heart beating videos, enhancing its ability to accurately quantify cardiovascular parameters.

Overall, leveraging SSIM and background subtraction with occlusion masks in the ZACAF framework can enhance the extraction of temporal information from heart beating videos, leading to more accurate and reliable automated cardiovascular assessment in zebrafish embryos. This approach can streamline the analysis process, improve the consistency of results, and facilitate advancements in cardiovascular research and genetic screenings using zebrafish models.

In the last section, we showed that SSIM can be used as a great tool for extracting temporal features. Now a framework is proposed to use SSIM as a temporal loss, to improve temporal consistency.

4.6.1. Dataset with temporal annotation

Researchers often utilize benchmark datasets to evaluate and compare the performance of video segmentation models. Benchmark datasets provide a standardized platform that ensures the consistency and reliability of experimental results. A well-constructed benchmark dataset for video segmentation should possess several key characteristics to effectively demonstrate a model's accuracy and temporal consistency. Firstly, it should include a diverse range of video sequences that cover various scenarios, including different lighting conditions, object motions, and occlusions, to test the model's robustness. Secondly, the dataset should provide high-quality, accurately annotated ground truth labels for each frame, enabling precise evaluation of segmentation accuracy. Temporal annotations are also crucial, as they allow the assessment of the model's ability to maintain consistency across consecutive frames. Additionally, the dataset should be large enough to train deep learning models effectively and validate their performance statistically. By meeting these criteria, a benchmark dataset can serve as a critical tool for advancing video segmentation research, offering a clear measure of how well a model performs in both spatial accuracy and temporal coherence. The zf datasets created in this work are temporally annotated as a sequence of 10 frames were selected and manually annotated during creation of the dataset. However, because we want to show the robustness and accuracy of our proposed model for learning temporal features, we use a popular benchmark dataset.

The DAVIS (Densely Annotated VIdeo Segmentation)[47] dataset is a widely recognized benchmark in the field of video segmentation. Introduced to provide a comprehensive and challenging platform for evaluating video object segmentation algorithms, DAVIS includes high-quality video sequences with densely annotated ground truth masks for each frame. This dataset is specifically designed to test a model's ability to accurately segment moving objects while maintaining temporal coherence across frames.

DAVIS stands out due to its meticulous annotation and diverse range of video sequences. The dataset encompasses various real-world scenarios, including dynamic backgrounds, occlusions, and complex object interactions, which are essential for testing the robustness and generalizability of segmentation models. The annotations provided in DAVIS are pixel-accurate, ensuring that the ground truth masks are of the highest quality, which is crucial for precise performance evaluation. Moreover, DAVIS includes several subsets, such as DAVIS-2016 and DAVIS-2017, each catering to different aspects of video segmentation challenges. DAVIS-2016 focuses on single-object segmentation, while DAVIS-2017 extends to multi-object segmentation, reflecting more complex and realistic scenarios. Researchers utilize the DAVIS dataset not only for benchmarking the accuracy of their models but also to assess temporal consistency, which is the model's ability to produce stable and coherent segmentations across consecutive frames. This makes DAVIS an indispensable resource for developing and validating advanced video segmentation techniques, driving progress in the field through its rigorous and diverse evaluation framework.

To further advance the segmentation capabilities of the U-net framework, we propose training the model using the DAVIS dataset, renowned for its high-quality, densely annotated video sequences. Initially, we will train the U-net model in a conventional manner, optimizing for pixel-wise accuracy using standard loss functions such as cross-entropy or Dice loss. This will establish a baseline performance in segmenting objects within the challenging and diverse video sequences provided by DAVIS.

4.6.2. Training U-net with an additional SSIM based loss function

To enhance temporal consistency in our segmentation results, we will subsequently retrain the U-net model using a novel approach that incorporates Structural Similarity Index Measure (SSIM) based loss. Our strategy involves feeding two consecutive frames from the video into the U-net model simultaneously. For each pair of consecutive frames, we will calculate the SSIM between the ground truth segmentations to capture the temporal coherence of the actual object movements. Similarly, we will calculate the SSIM between the predicted segmentations for these frames. The SSIM-based loss will then be designed to minimize the error between the SSIM of the ground truth and the SSIM of the predicted segmentations. By incorporating this temporal consistency loss, we aim to ensure that the segmentation predictions not only remain accurate on a frame-by-frame basis but also exhibit smooth and coherent transitions over time, closely mirroring the true dynamics of the video sequences. This approach leverages the rich temporal information inherent in video data, promoting more stable and reliable segmentation outputs in practical applications. For comparison, we also train the same U-net without using SSIM loss. Figure 19 shows the diagram for the proposed U-net that incorporates Dice coefficient and SSIM loss. We can compare the spatial metrics we used to use like IoU or Dice coefficient, to evaluate the model. However, to show the performance of the method in terms of temporal consistency we must propose new metrics as the aforementioned metrics are image based. Since we used SSIM as a loss we can also propose a way to utilize it a a temporal metric.

One way of showing temporal consistency could be measured by showing the SSIM between every pair of consecutive frames. Averaging the SSIM measured for all pairs, can present us with a score to show temporal consistency for a video. Figure 19 shows the mentioned workflow in a diagram.

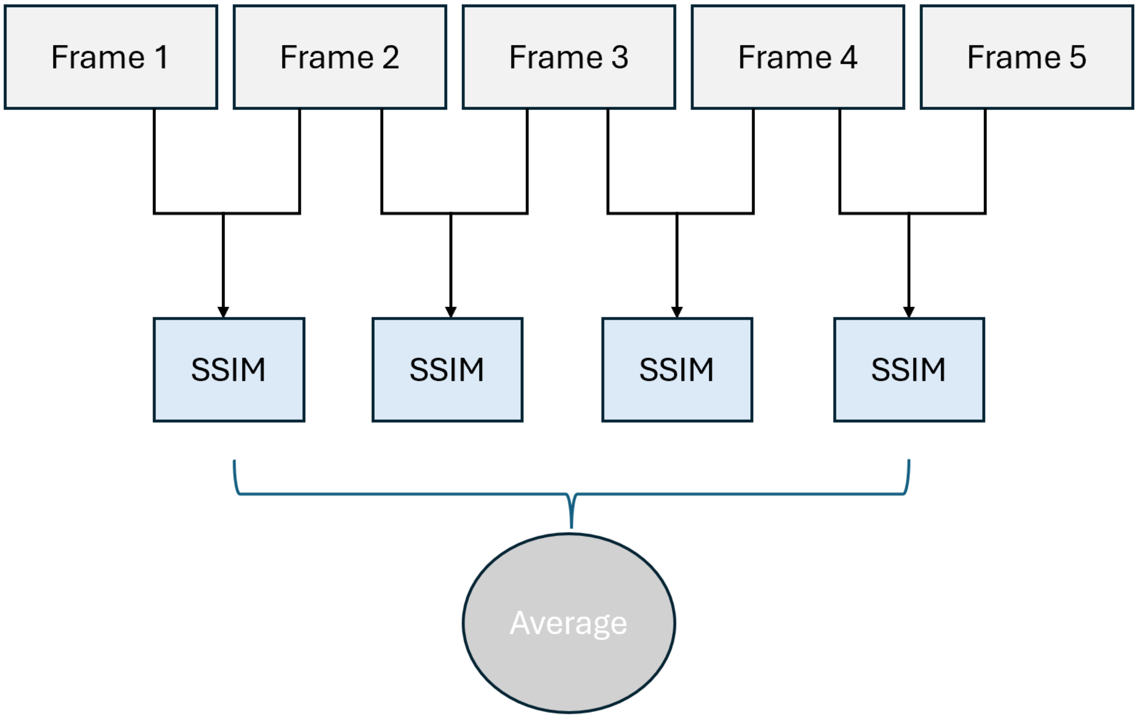

Figure 19 diagram for showing the proposed temporal consistency score for a video prediction. Like SSIM the average score is a number between 0 and 1 and a score closer to 1 shows better temporal consistency. An odd number of frames need to be chosen for this task.

Using the proposed SSIM score, the models trained with and without temporal loss respectively scored 0.96 and 0.92 which shows the benefit gained from using SSIM based temporal loss.

After training the proposed U-net based segmentation model with the DAVIS dataset, once with and once without the SSIM based temporal loss function below we can compare their Dice, IoU, and SSIM score. Also figure 20 demonstrates the inference of a series of frames of the test set. As can be seen in the figure the model trained with the SSIM temporal loss shows better results.

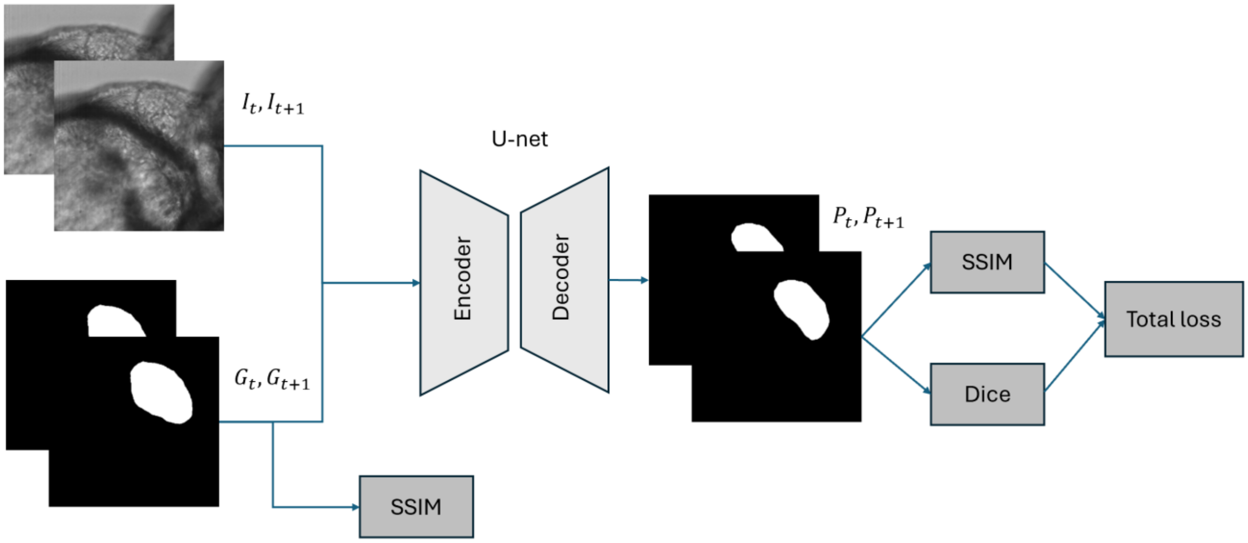

Figure 20 The diagram of the proposed U-net based model where two consecutive frames I_t, $I_{t+1}$and their corresponding ground truth masks G_t, G_{t+1} represent the training phase and the P_t, $P_{t+1}$are the resulted prediction masks. Total loss is calculated by adding the dice coefficient loss and SSIM loss.

Another way of evaluating the models is visually comparing their predicted results. Figure 21 shows the visual comparison between the models and as it can be seen the model that was trained with SSIM based loss has more accurate predictions spatially. The edges are conserved better in these predictions, and it is more consistent. Obviously, these predictions are both not comparable to the state-of-the-art models for video object segmentation that were mentioned in the literature review. That is because those works have used much more complicated and customized for this task. While we used a simple U-net that is usually used for image segmentation. Our purpose here was just to demonstrate the benefits of using SSIM based temporal loss in video segmentation. In the future works we plan to benchmark this loss function's improvement with the state of the art architecture like PSP-net[48] that are specifically proposed for video segmentation task. Also, it is worth noting that for simplification we merged all classes of the ground truth into binary background and foreground.

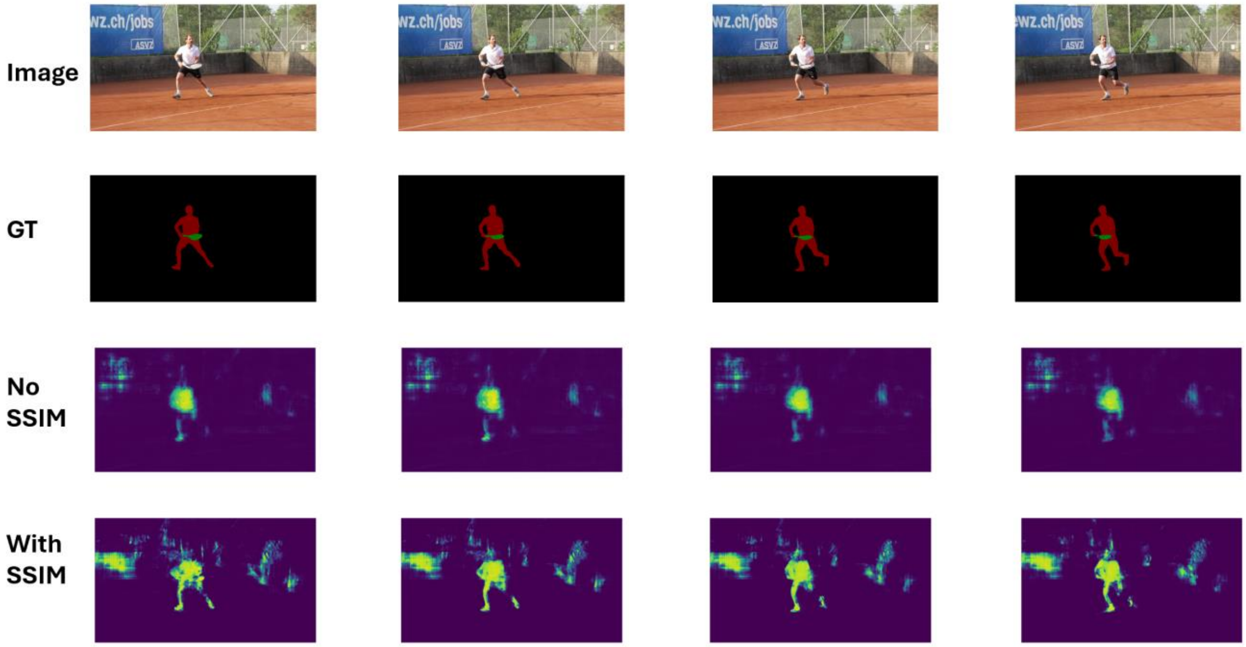

Figure 21 The visual representation of the performance of models that were trained with and without SSIM loss. Four consecutive frames have been shown with their manual ground truth, and two predications from models. As It can be seen the model trained with SSIM loss has captured more details and better edges. Note that GT has been binarized before training for simplification.

In our future work we want to implement this temporal loss in ZACAF as well and show its benefits for periodic heart beating videos. Additionally, we want to use this temporal loss in the latest state of the art for video image segmentation models like PSPnet.

Chapter 5: Discussions and Conclusion

5.1. Estimations in 2-D videos

The approach with microscopic videos relies on 2D videos to derive 3D volume estimation assuming the ventricle as a perfect ellipsoid. This assumption will result in accuracies in the measurements. Especially in mutant type where the shape of the chambers is not close to being ellipses this is going to be influential. However, the only solution to this problem is 3D imaging. In literature there are several studies that have extensively used 3D imaging technics like Z-stack imaging. In segmentation of the chambers using deep learning the third dimension is just going to be an extra layer of input. Granted the model is going to be more complicated but the concept is the same.

5.2. Consistency of measurement

Looking at the two frameworks that used deep learning for automatic segmentation of the zf heart, this method shows to be promising. The fully automated frameworks do the manual quantification of the cardiovascular metrics in a fraction of the time that it takes to do it manually. Additionally, manual segmentation is not consistent. Segmentation of the ventricle in these videos is a challenging task, even manually. The small size, ambiguous edges, and partial obstruction of the heart in the videos can also add complications to manual detection. We have investigated this quantitatively. We asked two experts to segment and measure the ventricle area in single frames of 12 sample videos. They were instructed to do the measurement twice for each frame manually with a short break between each try. The results were 12 frames, each measured 4 times. The standard deviation for each frame measurement was calculated, and the average of standard deviations of the measurements in these 12 frames was about 150 pixels with 50 pixels standard deviation. This is approximately 8% of the average size ventricular area in our setting’s scale. This shows the inconsistency in the manual segmentation. This could be especially significant with mutant embryos whose EF is usually very small. However, due to the nature of neural networks, trained models like ZACAF are consistent, which means that the measurement of a frame multiple times will always result in only one consistent measurement.

5.3. Frame rate issue

It is noteworthy to mention since the ground truth is created using the same frames for segmentation of the ventricle, the frame rate isn’t assessable. The ES and ED frames are the most important frames when it comes to the quantification of parameters like HR, EF, and FS. While recording the videos, the camera shutter takes a sequence of images with a certain fps. The higher the video's fps, the higher chance for exact ES and ED stages being recorded. This fact cannot be proved using the metrics because the prediction is only being compared with the existing manually segmented ground truth, and if the low fps causes the loss of ED or ES frames, there is no way to show it with the metrics.

5.4. Mutant fish lines challenges

From the segmentation point of view, there are two significant differences between the mutant and wildtype fish. The ventricle and the heart, in general, have abnormal shapes in several mutant types. In TTNtv case here, EF is much lower in the TTNtv model as the shape as well as the contractility are significantly affected. Thus, the ventricle area difference in ES and ED frames in TTNtv mutants is very low. Figure 22 provides examples to compare wild and TTNtv zebrafish. In some cases, the ventricle is barely beating so that the area difference in ED and ES frames is lower than the segmentation error.

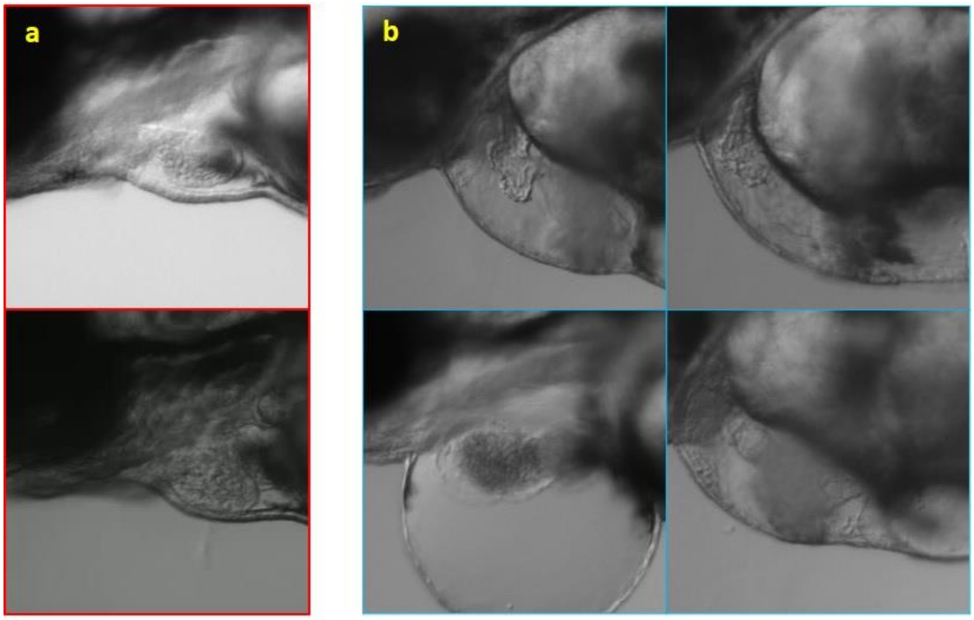

Figure 22: ***Comparison of the shape and size of wildtype (a) and TTNtv mutant zebrafish (b)****. Besides the abnormal shape of the heart with the swollen ventricular wall, the smaller size of the ventricle is also found with TTNtv mutants. Further, the swollen chest can be also noticed.*

In other words, the ventricle area hardly changes to the point that occasionally, the nominator of the formula of EF is lower than the estimation error. That is the primary source for the inaccuracies with the TTNtv mutants, and further improvements of preprocessing or optimization of the framework will not affect the result with the mutant significantly. The videos used in this work have low resolution in order to demonstrate the capability of our framework. Although this is beneficial for researchers to reduce required storage capacity, higher resolution would help resolve this issue, thus improving the robustness and accuracy for TTNtv mutant and wildtype fish in general.

Here, our framework can help researchers quantify the cardiac functions and parameters of studied zebrafish with minimum manual engineering efforts. In EF derivation, counting the pixels is more relevant and accurate than finding the long axis, which can be complicated since the ventricle is not a perfect ellipse. Further, the tool that most researchers use in the ImageJ software is a freehand ruler, which could introduce inaccuracy, especially with the small size of heart chambers.

5.5. Pear shape of the ventricle in some videos

The pear-shaped ventricles showed no significant correlation to the genotypes. We speculate that this shape is observed due to the improper placement of the fish under the microscope. Hence it might be caused by the fish not being placed perfectly on its side. Considering that the ventricle is not a perfect ellipsoid, this shape could be the result of the perspective of the camera from the ventricle which can be imaged as the 2D shape similar to a pear instead of being closer to an ellipse.

5.6. Comparison between EF formulas

In formula (3), A is the 2D area calculated directly from the segmented ventricle and D_L is the long axis. This way of calculation of the volume does not assume the shape of the ventricle to be a prolate spheroidal unlike formula (2). This formula is useful specifically for mutant fish where the long and short axis might not change significantly however the abnormal shape of the heart might contribute to an abnormal EF measurement. The results were calculated using both formulas. The average difference between the EF measurements using the two formulas was 3.34% which is negligible. However, some videos showed significant differences of up to 19.5%.

5.7. Discussion on transfer learning

The use of transfer learning in the ZACAF model has proved to be a successful technique for improving the model's performance for ventricle segmentation in zebrafish. By utilizing pre-trained weights from the original model, the new model was able to benefit from the features learned during the previous training, resulting in faster training and better performance than training the model from scratch on the new dataset. Additionally, the use of callbacks such as the model checkpoint helped to ensure that the best-performing model was saved and used for further analysis, allowing for the model to continue improving its performance.

5.8. Discussion on TTA

In this case, TTA was beneficial because it increased the variability of the test data and helped to reduce the effect of any biases in the original dataset. Since the ZACAF model was trained on a different dataset and microscope setup, there could be some differences in the characteristics of the new dataset that were not present in the original dataset. By applying TTA, the model was able to generate additional test data that had different characteristics, which helped to reduce the impact of any biases in the original dataset. Another benefit of TTA is that it can help to increase the robustness of the model to variations in the input data. Since the model is exposed to a larger variety of test data during inference, it is less likely to overfit to a particular type of input and more likely to generalize well to new data. However, TTA also has some potential drawbacks. One of the main drawbacks is that it can increase the computational cost of making predictions since the model needs to process multiple versions of the test data. Depending on the complexity of the model and the number of test data versions generated, the computational cost can be significant. Another potential drawback of TTA is that it can introduce some variability into the predictions, which can make it difficult to interpret the results. Since the final prediction is based on an average of multiple predictions, it may not be clear which version of the test data was responsible for a particular prediction. This can make it challenging to identify specific areas of the input data that the model is struggling with.

5.9. Discussion of temporal consistency

The integration of video segmentation techniques and temporal information significantly enhances the Zebrafish Automated Cardiovascular Assessment Framework (ZACAF). By leveraging deep learning models for automated segmentation, we address the limitations of manual methods, providing more efficient and consistent results. The rhythmic nature of heart beating in zebrafish embryos makes them ideal for temporal analysis, enabling precise delineation of cardiac structures across frames. The incorporation of temporal information not only streamlines the assessment process but also ensures robustness and reliability, crucial for handling large volumes of video data. This chapter underscores the potential of combining spatial and temporal features to improve cardiovascular function assessment, highlighting the broader applicability of these methods to various video segmentation and object recognition tasks.

5.10. Conclusion

This thesis presents a comprehensive exploration of deep learning-based frameworks for video segmentation, particularly in the context of cardiac function assessment in embryonic zebrafish. Through the development and enhancement of the Zebrafish Automated Cardiac Analysis Framework (ZACAF), this work demonstrates the effectiveness of U-net architecture in achieving high-accuracy segmentation of zebrafish hearts. By incorporating transfer learning and test-time augmentation (TTA), the model's robustness and generalizability across different datasets were significantly improved. Furthermore, the integration of temporal features into the segmentation model enabled a more precise capture of the dynamic nature of cardiac function. This research not only advances the field of biomedical video segmentation but also provides valuable insights into the developmental biology of cardiac function, paving the way for future applications in both research and clinical settings.

References

1. Wisneski, J., et al., *Left ventricular ejection fraction calculated from volumes and areas: underestimation by area method.* Circulation, 1981. **63**(1): p. 149-151.
2. Ling, D., et al., *Quantitative measurements of zebrafish heartrate and heart rate variability: A survey between 1990–-2020.* Computers in Biology and Medicine, 2021: p. 105045.
3. Maragos, P., *Chapter 13 - Morphological Filtering*, in *The Essential Guide to Image Processing*, A. Bovik, Editor. 2009, Academic Press: Boston. p. 293-321.
4. Pizer, S.M., et al., *Adaptive histogram equalization and its variations.* Computer Vision, Graphics, and Image Processing, 1987. **39**(3): p. 355-368.
5. Canny, J., *A Computational Approach to Edge Detection.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 1986. **PAMI-8**(6): p. 679-698.
6. Akerberg, A.A., et al., *Deep learning enables automated volumetric assessments of cardiac function in zebrafish.* Disease models & mechanisms, 2019. **12**(10): p. dmm040188.
7. Dhanachandra, N., K. Manglem, and Y.J. Chanu, *Image Segmentation Using K -means Clustering Algorithm and Subtractive Clustering Algorithm.* Procedia Computer Science, 2015. **54**: p. 764-771.
8. Bishop, C.M. and N.M. Nasrabadi, *Pattern recognition and machine learning*. Vol. 4. 2006: Springer.
9. Gupta, L. and T. Sortrakul, *A gaussian-mixture-based image segmentation algorithm.* Pattern Recognition, 1998. **31**(3): p. 315-325.
10. Ronneberger, O., P. Fischer, and T. Brox. *U-net: Convolutional networks for biomedical image segmentation*. in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*. 2015. Springer.
11. Long, J., E. Shelhamer, and T. Darrell. *Fully convolutional networks for semantic segmentation*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015.
12. Badrinarayanan, V., A. Kendall, and R. Cipolla, *Segnet: A deep convolutional encoder-decoder architecture for image segmentation.* IEEE transactions on pattern analysis and machine intelligence, 2017. **39**(12): p. 2481-2495.
13. Wessells, R.J. and R. Bodmer, *Screening assays for heart function mutants in Drosophila.* Biotechniques, 2004. **37**(1): p. 58-66.
14. Nasrat, S., et al., *Semi-automated detection of fractional shortening in zebrafish embryo heart videos.* Current Directions in Biomedical Engineering, 2016. **2**(1): p. 233-236.
15. Huang, W.-Y., et al., *Transgenic expression of green fluorescence protein can cause dilated cardiomyopathy.* Nature medicine, 2000. **6**(5): p. 482-483.
16. Zhang, B., et al., *Automatic segmentation and cardiac mechanics analysis of evolving zebrafish using deep learning.* Frontiers in cardiovascular medicine, 2021. **8**: p. 675291.
17. Suryanto, M.E., et al., *Using DeepLabCut as a real-time and markerless tool for cardiac physiology assessment in zebrafish.* Biology, 2022. **11**(8): p. 1243.
18. Naderi, A.M., et al., *Deep learning-based framework for cardiac function assessment in embryonic zebrafish from heart beating videos.* Computers in biology and medicine, 2021. **135**: p. 104565.
19. Decourt, C. and L. Duong, *Semi-supervised generative adversarial networks for the segmentation of the left ventricle in pediatric MRI.* Computers in Biology and Medicine, 2020. **123**: p. 103884.
20. Jadon, S. *A survey of loss functions for semantic segmentation*. in *2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*. 2020.

21. Merlo, M., et al., *Evolving concepts in dilated cardiomyopathy.* Eur J Heart Fail, 2018. **20**(2): p. 228-239.
22. Hershberger, R.E., D.J. Hedges, and A. Morales, *Dilated cardiomyopathy: the complexity of a diverse genetic architecture.* Nat Rev Cardiol, 2013. **10**(9): p. 531-47.
23. Wheeler, F.C., et al., *QTL mapping in a mouse model of cardiomyopathy reveals an ancestral modifier allele affecting heart function and survival.* Mamm Genome, 2005. **16**(6): p. 414-23.
24. Hoage, T., Y. Ding, and X. Xu, *Quantifying cardiac functions in embryonic and adult zebrafish.* Methods Mol Biol, 2012. **843**: p. 11-20.
25. Bland, J.M. and D.G. Altman, *Statistical methods for assessing agreement between two methods of clinical measurement.* Lancet, 1986. **1**(8476): p. 307-10.
26. Lu, S., D.E. Borst, and R. Horowits, *Expression and alternative splicing of N-RAP during mouse skeletal muscle development.* Cell Motil Cytoskeleton, 2008. **65**(12): p. 945-54.
27. Jirka, C., et al., *Dysregulation of NRAP degradation by KLHL41 contributes to pathophysiology in nemaline myopathy.* Hum Mol Genet, 2019. **28**(15): p. 2549-2560.
28. Lu, S., et al., *Cardiac-specific NRAP overexpression causes right ventricular dysfunction in mice.* Exp Cell Res, 2011. **317**(8): p. 1226-37.
29. Truszkowska, G.T., et al., *Homozygous truncating mutation in NRAP gene identified by whole exome sequencing in a patient with dilated cardiomyopathy.* Sci Rep, 2017. **7**(1): p. 3362.
30. Shorten, C. and T.M. Khoshgoftaar, *A survey on Image Data Augmentation for Deep Learning.* Journal of Big Data, 2019. **6**(1): p. 60.
31. Cossio, M., *Augmenting Medical Imaging: A Comprehensive Catalogue of 65 Techniques for Enhanced Data Analysis.* arXiv preprint arXiv:2303.01178, 2023.
32. Pan, S.J. and Q. Yang, *A Survey on Transfer Learning.* IEEE Transactions on Knowledge and Data Engineering, 2010. **22**(10): p. 1345-1359.
33. Moshkov, N., et al., *Test-time augmentation for deep learning-based cell segmentation on microscopy images.* Scientific Reports, 2020. **10**(1): p. 5068.
34. mohammad Naderi, A., *Deep Learning-Based Framework for Cardiac Function Assessment in Embryonic Zebrafish from Heart Beating Videos*. 2023: University of California, Irvine.
35. Varghese, S., et al. *An unsupervised temporal consistency (TC) loss to improve the performance of semantic segmentation networks*. in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021.
36. Ding, M., et al. *Every frame counts: Joint learning of video segmentation and optical flow*. in *Proceedings of the AAAI conference on artificial intelligence*. 2020.
37. Xu, J., Z. Xiong, and X. Hu, *Frame difference-based temporal loss for video stylization.* arXiv preprint arXiv:2102.05822, 2021.
38. Zhang, Y., et al. *Auxadapt: Stable and efficient test-time adaptation for temporally consistent video semantic segmentation*. in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2022.
39. Liu, Y., et al. *Efficient semantic video segmentation with per-frame inference*. in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part X 16*. 2020. Springer.
40. Guan, D., et al. *Domain adaptive video segmentation via temporal consistency regularization*. in *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021.
41. Woo, S., et al. *Learning to associate every segment for video panoptic segmentation*. in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021.
42. Lao, J., et al. *Simultaneously short-and long-term temporal modeling for semi-supervised video semantic segmentation*. in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023.

43. Horn, B.K. and B.G. Schunck, *Determining optical flow.* Artificial intelligence, 1981. **17**(1-3): p. 185-203.

44. Dosovitskiy, A., et al. *Flownet: Learning optical flow with convolutional networks*. in *Proceedings of the IEEE international conference on computer vision*. 2015.

45. Hochreiter, S. and J. Schmidhuber, *Long short-term memory.* Neural computation, 1997. **9**(8): p. 1735-1780.

46. Wang, Z., et al., *Image quality assessment: from error visibility to structural similarity.* IEEE transactions on image processing, 2004. **13**(4): p. 600-612.

47. Perazzi, F., et al. *A benchmark dataset and evaluation methodology for video object segmentation*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

48. Zhao, H., et al. *Pyramid scene parsing network*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.