Quantification Platform for Touch Response of Zebrafish Larvae using Machine Learning

Yanke Wang[1], Christian Pylatiuk[1], Ralf Mikut[1], Ravindra Peravali[2], Markus Reischl[1]

[1]Institute for Automation and Applied Informatics, Karlsruhe Institute of Technology
Hermann-von-Helmholtz-Platz 1, 76344 Eggenstein-Leopoldshafen
E-Mail: yanke.wang@kit.edu

[2]Institute for Biological and Chemical Systems, Karlsruhe Institute of Technology
Hermann-von-Helmholtz-Platz 1, 76344 Eggenstein-Leopoldshafen

Abstract

A touch-evoked response of zebrafish larvae provides information of the mechanism of the gene functional expressions. Recently, an automated system has been developed for precise and repeated touch-response experimentation with minor human intervention. The data collected by the system are analyzed with regard to an automated quantification pipeline for scientific conclusions, including five quantification criteria: latency time, C-Bend curvature maximum, C-Bend peak time, response time, and moving distance. To quantify these criteria, we propose a larva tracking based automatic quantification pipeline by using a U-Net for initialization of tracking, a particle filter as tracking strategy, and region growing for the segmentation of larvae. Experimental data with different treatments are analyzed by using the proposed quantification platform for demonstration, and the result proves that this platform can generate comparable touch-response behavior quantification readouts in an efficient and automatic way. This platform provides an alternative to automatically screening the drugs for knowledge discovery according to the pattern of the touch-response behaviors of zebrafish larvae mutated by chemicals.

1 Introduction

Zebrafish larvae are commonly used animal models for the organism-based screenings due to small size, high fecundity and short reproductive cycle [8]. Their specific (repeatedly and obvious) behaviors indicate certain functional mechanisms of mutants by the treatments [1, 7], making it possible to do the large-scale high-throughput screening of chemicals or drugs. Automated experimental systems to acquire the data of these behaviors have been developed so far [4, 5, 6, 7, 20], so the automated high-throughput quantification of the data from the systems is also becoming in a higher demand, as manual quantification is time-consuming and not statistically comparable. In particular, the touch-response experimental data (videos) are in a high frame rate [2, 19], so the automated quantification is more essential in this case. During the touch-evoked response of zebrafish larvae, the larvae form into a series of C-Bends and swim away after touching, and it is important to quantify the time that the larvae take to respond as well as the strength of the response (such as the latency time, C-Bend curvature maximum, C-Bend peak time, response time, and moving distance). However, it is difficult to generate a precise number of C-Bend curvature and moving distance manually [20]. Furthermore, the operators cannot keep the same criteria all the time for each video, as the video has more than ten thousand frames in average. Thus, we proposed a touch-response quantification pipeline for single zebrafish larva in [2], but as for the multi-larvae case, we face more challenges: i) multiple larvae need to be tracked and segmented at the same time; ii) which larva is touched should be decided; iii) the quantification of multiple larvae has higher computational costs. To solve these problems, we optimized the pipeline to an automatically customized touch-response quantification platform in this work.

In this proposed quantification platform of touch-response experimental data, the tracking procedure plays the vital role, especially for the tracking of multiple larvae [19]. Recently, machine learning or deep learning based tracking methods have emerged to promote the accuracy of the tracking procedure [3, 17], and many previous works focused on the tracking and segmentation of single or multiple adult zebrafish [9, 10, 11, 15]. To make the best of the deep learning methods, we used a U-Net based segmentation method for the initialization of tracking. However, those high-computational methods are difficult to be used in the tracking procedure of our high-frame-rate videos. In order to make the quantification pipeline much less complex, we proposed a optical flow based needle tracking procedure and particle filter based larvae tracking procedure. Besides, the segmentation for each larva is also of importance to the analysis of the movements. In [16], a Gaussian Mixture Model (GMM) based segmentation is used to detect the moving objects, and the noise is filtered according to the region size by using a global Otsu method. However, in our platform, considering global information makes the procedure more computational expensive. Therefore, a local region growing based segmentation method is used for each larva according to the result of tracking procedure. Based on the tracking and segmentation results, we proposed a pipeline to find the touched larvae and generate the behavior quantification according to the proposed experiment criteria. In order to test the performance of the proposed platform, we conducted six sets of experiments with different drugs and analyzed the experiment criteria and detected errors (failure cases). With the verification of the experiment results, this platform shows a high efficiency for analyzing the touch-response experimental data, and releases the operators. The methods used in this platform can make contributions to the application in the field of video analysis. As well, the platform can be also transformed to the quantification pipeline of other organisms (like medaka) and can be also added with more quantification criteria.

Organization of the article is as follows. Section 3 describes the tracking procedures, local segmentation for the larvae and the quantification pipeline of the proposed platform. Section 3 provides the setup of the experiments, the quantification criteria and results as well as the discussion. According to the above results, conclusions are drawn in Section 4.

2 Methodology

As the videos collected by the automated system are in a high frame rate (1000 frames per second), an efficient tracking procedure is required. The initial positions obtained from the first frame is vital to accuracy of the whole tracking procedure, so a U-Net based segmentation (Step 1 in Fig. 1) is used to generate the binarization of the larvae and the needle for the initialization of the tracking. However, we cannot use the U-Net for the tracking of the following frames, as deep learning inference is computationally expensive, causing the high temporal costs for one single video. Therefore, optical flow (Step 2 in Fig. 1) is used for the tracking of the needle, and particle filter (Step 2 $\sim$ 4 in Fig. 1) is used for the predictions of the positions of the larvae in the following frames. Based on the predictions, region growing (Step 5 in Fig. 1) is applied for the local segmentation of each larva. The output of the tracking procedure includes images patches of each larva as well as the positions of the needle and the larva in all frames.

2.1 U-Net for initialization

Ahead of the tracking procedure, the positions of the larvae need to be initialized, which is usually done manually, but in order to make this procedure fully automated, we used a U-Net [18] to do the segmentation of the first frame of the video for initialization. The U-Net mainly consists of down-sampling blocks - two convolutional layers (Conv) and one max pooling (Max-pool) - for feature extraction and up-sampling blocks - one deconvolutional layer (deConv) and two convolutional layers. As shown in Step 1 in Fig. 1, the U-Net based segmentation inputs the image within the well area cropped by Hough transform and generates two binary images with larvae and needle, followed by the needle tracking and larva tracking strategies respectively.

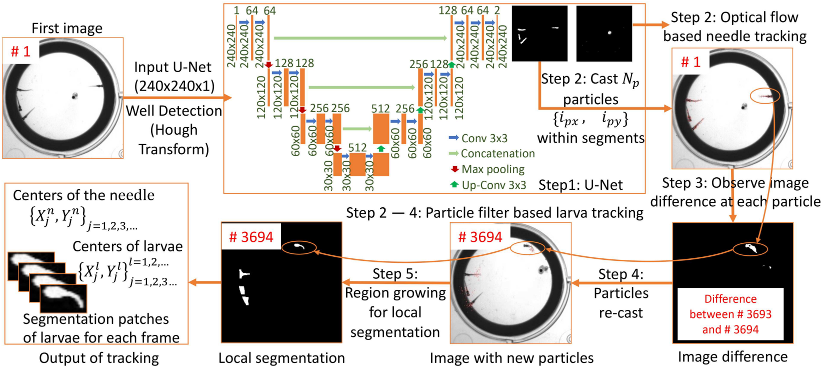


Figure 1: Overall architecture of the proposed tracking procedure. Step 1: Network architecture of a U-Net used to segment the larvae and needle for initialization. The outputs are two binary images for the larvae and needle respectively. Step 2: Two tracking strategies for the needle and larvae respectively. The optical flow tracking method is used for the positions of the needle as it moves slightly between two frames. For the particle filter based larva tracking method, particles are cast in this step within the segmented larvae areas. Step 3: According to the position of each particle, the image difference between two frames (with an example of the image difference between Frame #3693 and Frame #3694) is observed for the binary probability of the corresponding particle. Step 4: The particles with binary probability 0 are re-cast around the larvae center, details in Section 2.3. Step 5: For each larva, the segmentation is achieved by local region growing, discussed in Section 2.4. The outputs of the tracking contain the image patches of all larvae as well as the series of the centers of the larvae and the needle.

2.2 Optical flow based needle tracking

In the optical flow tracking procedure, the tracking target is assumed to move slightly between two frames [12, 13], and the movement of the needle meets this requirement. Thus, the optical flow based needle tracking strategy is used. Let $\{X_j^n, Y_j^n, t_j\}$ be the old needle (n) center at frame t_j, so the new needle center at frame t_{j+1}, $\{X_{j+1}^n, Y_{j+1}^n, t_{j+1}\}$, is estimated according to the gradients as described in [2].

2.3 Particle filter based larva tracking

For the tracking procedure of the larvae, optical flow does not meet the assumption as the larvae move significantly, so we used a particle filter based tracking strategy. Particle filter, not like Kalman filter, has no constrained assumptions [3], and the tracking result is dependent on the score of each particle cast randomly according to the prior knowledge (the previous positions of the larvae in our case). As shown in Step 2 in Fig. 1, the particles (with number N_p) are cast randomly within the segments of each larva to do the following tracking procedure. Defined that ${}_iP^l_j = \{{}_ix^l_j, {}_iy^l_j, t_j\}$ be the particle i at position $\{{}_ix^l_j, {}_iy^l_j\}$ of frame t_j for larva l, the binary probability $b\{{}_ix^l_j, {}_iy^l_j, t_j\}$ indicates whether the l-th larva exists, shortened as ${}_ib^l_j$. The new center of the larva l at frame t_{j+1} $\left(\{X^l_{j+1}, Y^l_{j+1}, t_{j+1}\}\right)$ is estimated as follows

$$X^l_{j+1} = \frac{1}{N_p}\sum_{i=1}^{N_p}({}_ix^l_j \cdot {}_ib^l_{j+1}), \quad Y^l_{j+1} = \frac{1}{N_p}\sum_{i=1}^{N_p}({}_iy^l_j \cdot {}_ib^l_{j+1}) \tag{1}$$

where ${}_ib^l_{j+1}$ is the binary probability at $\{{}_ix^l_j, {}_iy^l_j\}$ of the lth larva in frame t_{j+1}. The binary probability is computed according to the image difference as follows

$${}_ib^l_{j+1} = \begin{cases} 1 & \text{if } {}_id^l_{j+1} > T_d \\ 0 & \text{else} \end{cases}, \quad {}_id^l_{j+1} = |f_{j+1}({}_ix^l_j, {}_iy^l_j) - f_j({}_ix^l_j, {}_iy^l_j)| \tag{2}$$

where ${}_id^l_{j+1}$ is the pixel difference at $\{{}_ix^l_j, {}_iy^l_j\}$ between frame t_{j+1} and frame t_j, $f_{j+1}({}_ix^l_j, {}_iy^l_j)$ is the pixel value at $\{{}_ix^l_j, {}_iy^l_j\}$ of frame t_{j+1}, $f_j({}_ix^l_j, {}_iy^l_j)$ of frame t_j, and T_d is the threshold for the image difference for the moving pixels. The particles ${}_iP^l_j$ with ${}_ib^l_{j+1} = 0$ are re-cast in a Gaussian distribution as follows

$$\begin{aligned} {}_i\hat{P}^l_{j+1} = \{{}_ix^l_{j+1}, {}_iy^l_{j+1}, t_{j+1}\}, \{{}_ix^l_{j+1}, {}_iy^l_{j+1}\} \sim \mathcal{N}(\boldsymbol{\mu}^l_{j+1}, \boldsymbol{\Sigma}) \\ \boldsymbol{\mu}^l_{j+1} = \begin{bmatrix} X^l_{j+1} \\ Y^l_{j+1} \end{bmatrix}, \boldsymbol{\Sigma} = \begin{bmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{bmatrix} \end{aligned} \tag{3}$$

where ${}_i\hat{P}^l_{j+1}$ is the updated (re-cast) particle at $\{{}_ix^l_{j+1}, {}_iy^l_{j+1}\}$ of the new frame t_{j+1}, $\boldsymbol{\mu}^l_{j+1}$ is the new center of the larva estimated by (1), and $\boldsymbol{\Sigma}$ is the heuristic variance for the range of re-casting the particles. The retained particles are used for storing the previous information of the positions of the larvae, and

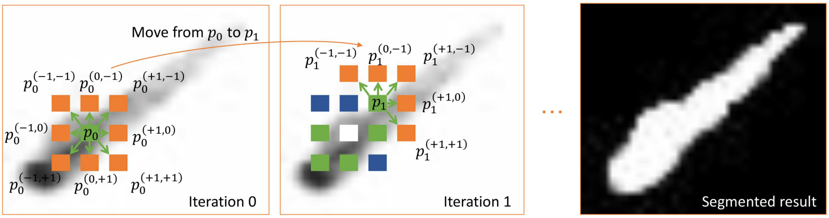


Figure 2: The principle of region growing based local segmentation of the larva. Region growing begins with an initialized point and grows according to the adjacent pixel values until no more new pixels meet the requirements, details in Section 2.4. The adjacent points of p_k are denoted as $p_k^{(*,*)}$, e.g. $p_k^{(-1,-1)}$ is the top left point of p_k. The pixels in green are the next centers for iterations with the adjacent points in orange, and the pixels that meet the requirements are labeled as white and otherwise as blue.

the re-cast particles are used for searching for new potential positions of the larvae.

2.4 Region growing for local segmentation

The particle filter might lose the larva as it only considers the moving pixels. Thus, the segment for each larva is required for a more precise center of the larvae as well as for the analysis of the behaviors. We used a region growing to do the local segmentation for each larva, as shown in Fig. 2. The region starts at the estimated center $\left(\{X^l_{j+1}, Y^l_{j+1}\}\right)$ of the larva (the initialized point) and label each pixel in a 3×3 adjacent area according to the pixel value and image gradient. Assumed that $p_k = \{p_{kx}, p_{ky}\}$ is the center of the adjacent area in each iteration (starting iteration: $p_0 = \{X_{j+1}, Y_{j+1}\}$), the label of each adjacent point $p_k^{(*,*)} = \{p_{kx}^{(*,*)}, p_{ky}^{(*,*)}, * = -1, 0, +1\}$ is calculated as

$$L(p_k^{(*,*)}) = \begin{cases} 1 & \text{if } g(p_{kx}^{(*,*)}, p_{ky}^{(*,*)}) < T_g \\ & \quad \text{and } T_l < f(p_{kx}^{(*,*)}, p_{ky}^{(*,*)}) < T_h \\ 0 & \text{else} \end{cases}, \tag{4}$$
$$g(p_{kx}^{(*,*)}, p_{ky}^{(*,*)}) = |f(p_{kx}^{(*,*)}, p_{ky}^{(*,*)}) - f(p_{kx}, p_{ky})|$$

where $L(p_k^{(*,*)})$ is the labelled (segmented) result for the position $p_k^{(*,*)}$ at k-th iteration, $g(p_{kx}^{(*,*)}, p_{ky}^{(*,*)})$ is the image gradient at $p_k^{(*,*)}$, $f(p_{kx}^{(*,*)}, p_{ky}^{(*,*)})$ is the pixel value at $p_k^{(*,*)}$, and T_g, T_l, T_h are the heuristic thresholds chosen for the binarization. The position $p_k^{(*,*)}$ labelled as 1 is the next center of the adjacent area at iteration $k+1$ for the growing of the region, until all new centers are labelled as 0. As the larva area is connected with other objects or noise, the iteration may not stop even if the area covers the larva in a larger scale. Thus, we set another size threshold (T_s) to end the iterations. As the growing of the regions only occurs in the local areas of the larvae, the computation is much faster than the global binarization methods or deep learning based methods [15, 16].

2.5 Quantification pipeline based on tracking procedure

When zebrafish larvae are touched, they exhibit characteristic (or specific) behaviors [1]. In this work, five typical quantification indices are considered, three of which were considered in a previous work (latency time t_l, response time t_r, and moving distance d_m) [2]. As for the quantification of the response strength, in this work, we consider to use the maximum of the C-Bend curvature that the larvae shaped (C-Bend curvature maximum, c_m), as the average cannot quantify the peak value of the response strength. Additionally, we propose to use another parameter, C-Bend peak time (t_{cp}), to describe the time that the larvae took to have the peak response strength.

The tracking procedure for the needle and larvae outputs: i) sets of image patches for each larva in each frame; ii) the centers of the larvae; iii) the centers of the needle in each frame, shown in Fig. 1 and Fig. 3. The touched larva is decided by comparing the final position (at t_f) of the needle $X_{t_f}^n$ and the initialized positions of the larvae $\{X_0^l\}^{l=1,2,\cdots}$. In order to compute t_1, t_2, t_3, and t_4, another two thresholds are defined: i) T_{nl} for the distance between the needle and the larva deciding the touch is successfully applied; ii) T_m for the movement of the larvae, with details in Fig. 3. According to the time points above, the quantification indices are computed as follows: i) the latency time is computed as $t_l = t_2 - t_1$; ii) the C-Bend peak time is computed as

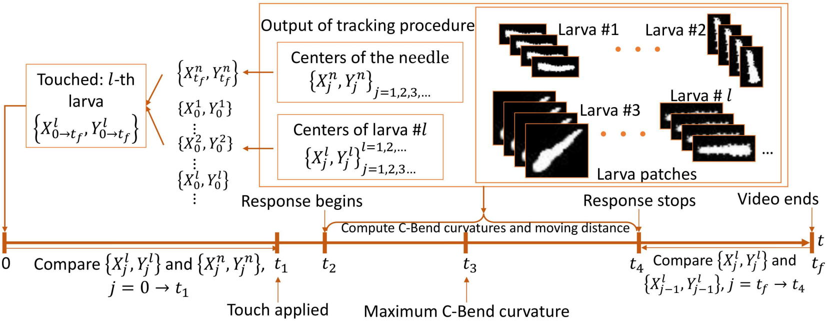

Output of tracking procedure
Centers of the needle
$\{X_j^n, Y_j^n\}_{j=1,2,3,\ldots}$
Centers of larva #l
$\{X_j^l, Y_j^l\}_{j=1,2,3\ldots}^{l=1,2,\ldots}$
Larva #1
Larva #2
Larva #3
Larva # l
Larva patches
...
Touched: l-th larva
$\{X^l_{0\to t_f}, Y^l_{0\to t_f}\}$
$\{X^n_{t_f}, Y^n_{t_f}\}$
$\{X_0^1, Y_0^1\}$
$\{X_0^2, Y_0^2\}$
$\{X_0^l, Y_0^l\}$
Response begins
Response stops
Video ends
Compute C-Bend curvatures and moving distance
0
Compare $\{X_j^l, Y_j^l\}$ and $\{X_j^n, Y_j^n\}$, t_1
$j = 0 \to t_1$
t_2
t_3
t_4
Compare $\{X_j^l, Y_j^l\}$ and $\{X_{j-1}^l, Y_{j-1}^l\}$, $j = t_f \to t_4$
t_f
t
Touch applied
Maximum C-Bend curvature

Figure 3: The pipeline for behavior quantification according to the results of tracking procedure. According to the output of the tracking procedure, important time points (t_1: touch applied, t_2: response begins, t_3: the time point of maximum C-Bend curvature, t_4: response stops) are searched: i) the distance between lth larva and the needle is compared from $t = 0$ until the time point with the distance less than T_{nl}, as t_1; ii) from t_1, the distance of the lth larva between two frames (X_j^l and X_{j+1}^l) is computed until the time point over T_m, as t_2; iii) the time point with maximum C-Bend curvature is t_3; iv) from t_f back to the previous time points, the positions of the lth larva between two frames (X_j^l and X_{j-1}^l) are compared until the time point over T_m, as t_4. More details for each index are described in Section 2.5

$t_{cp} = t_3 - t_2$; iii) the response time is computed as $t_r = t_4 - t_2$; iv) from t_2 to t_4, the moving distance d_m and C-Bend curvatures c_m of the l-th larva in each frame are computed according to the corresponding image patches (methods described in [2]).

3 Experiments and results

3.1 Experiment setup

In order to test the performance of the proposed platform on the experimental data from the automated touch-response system [2], we quantified six sets of experimental data (as Table 1 shows): videos of *wild* type (without treatment), larvae with Dimethyl sulfoxid (*DMSO*)[1], as well as larvae treated by Diazepam (*Dia*) to reduce the movements [14], Isoproterenol hydrochloride (*Iso*) with

[1] As each treatment is prepared with *DMSO*, the experiments on the larvae with only *DMSO* are also conducted as controls.

Table 1: The experimental data (number of videos) to be quantified.

Type	Treatment	Concentration	Age	Number of videos
Wild	Fish water	-	73 hpf	24
DMSO	Dimethyl sulfoxide	1%	73 hpf	27
Dia	Diazepam	100 μmol/mL	73 hpf	38
Iso	Isoprenaline hydrochloride	100 μmol/mL	73 hpf	30
Caffi	Caffeine	100 μmol/mL	73 hpf	24
Saha	Suberoylanilide hydroxamic acid	100 μmol/mL	73 hpf	30

unknown effects, Caffeine (*Caffi*) for also reduction of movements [14], and Suberoylanilide hydroxamic acid (*Saha*) with unknown effects, respectively. Each treatment is in a concentration of 100 μmol/mL for the demonstration. The larvae were dechorionated and treated at 27 hpf, and the experiments were conducted at 73 hpf, as visualized in Table 1.

The parameters used for the quantification platform are outlined in Table 2. The average size of the larvae is 162.66 pixels, computed by 320 images (4 larvae in each), so the threshold (T_s) is set as 200 pixels for safety. The other parameters are selected heuristically.

3.2 Experiment criteria

The experiment criteria (quantification indices) are discussed in Section 2.5 to verify whether the proposed quantification platform can generate corresponding results to the assumptions of the effects of the treatments, as described in Section 3.1. Besides, the detected errors of the quantification pipeline should also be analyzed, e.g. the inaccuracy of the segmentation method and missing objects by the tracking procedure. As well, the videos collected by the automated system contain some unquantifiable ones, such as the larvae were not touched, and the larvae or needle might not be detected. Thus, we also aim

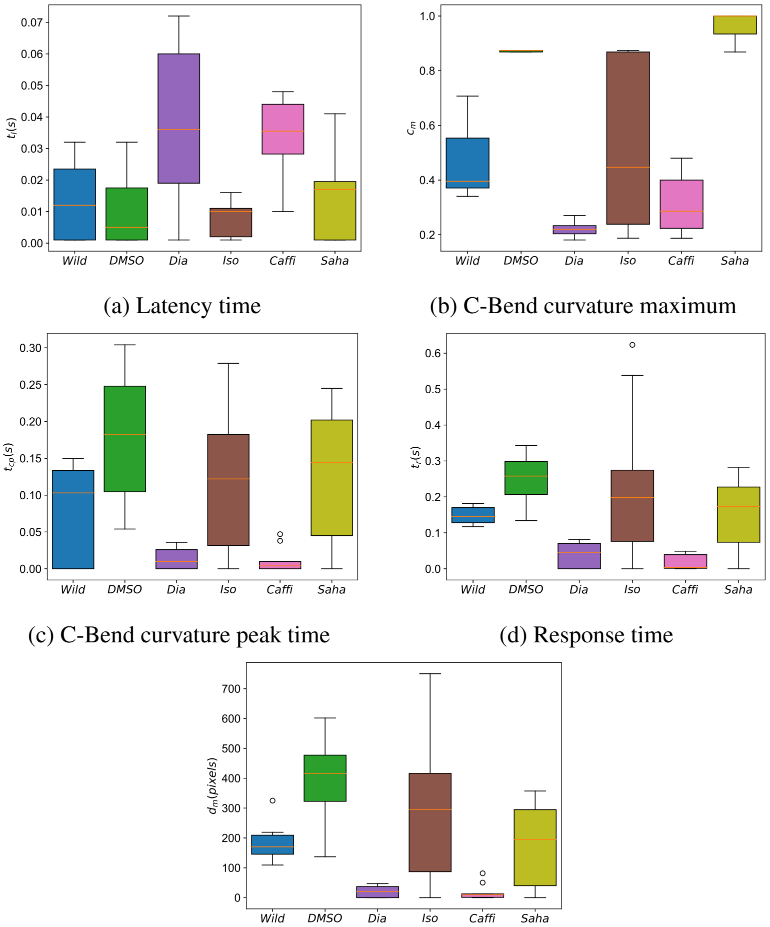
Table 2: The parameters used for the proposed quantification platform.

Symbol	Quantity	Value
N_p	the number of the particles	50
T_d	the threshold for the image difference (pixels)	10
σ_x, σ_y	the standard deviation for the range of re-casting the particles (pixels)	7
T_g	the threshold for the image gradient in local segmentation (pixels)	100
T_l	the lower threshold for the binarization in local segmentation (pixels)	50
T_h	the higher threshold for the binarization in local segmentation (pixels)	220
T_s	the size threshold for the larvae (pixels)	200
T_{nl}	the threshold for the distance between the larva and needle (pixels)	10
T_m	the threshold for the movement of the larvae	50%

to give the analysis of detected errors by using the number of videos with no larvae touched ($\#NT$) as well as those with failure of quantification ($\#QF$), with details in Section 5.

3.3 Results

We applied our quantification pipeline (described in Section 2.5) to the experimental data outlined in Table 1 and visualized the quantification results for the touched larvae in Fig. 4, including latency time (t_l), C-Bend curvature maximum (c_m), C-Bend curvature peak time (t_{cp}), response time (t_r), and moving distance (d_m). The five quantification indices give a consistent output: The larvae with longer latency time have lower response strength (lower c_m), shorter time to shape the C-Bend peak (lower t_{cp}), and less response duration (lower t_r and d_m), examples seen from the cases of Dia and $Caffi$. This



(a) Latency time

(b) C-Bend curvature maximum

(c) C-Bend curvature peak time

(d) Response time

(e) Moving distance

Figure 4: Five quantification indices of six experiment cases (*wild*, *DMSO*, *Dia*, *Iso*, *Caffi*, and *Saha*) generated by the quantification pipeline in Section 2.5, including latency time, C-Bend curvature maximum, C-Bend curvature peak time, response time, and moving distance.

result can also prove that the larvae under the treatments of *Dia* and *Caffi* respond less compared with the *Wild* and *DMSO*, verifying the assumptions in Section 3.1. Additionally, the treatment *Iso* can not change the touch-response behaviors of the larvae significantly as the five indices show similar results to the *Wild* type. As for the treatment *Saha*, the result is similar to the case of *DMSO*, verifying that this treatment cannot change the touch response of zebrafish larvae too much. The results above verified that our proposed platform can generate comparable quantification according to the experimental data acquired by the automated touch-response system and is also potentially useful for drug screening.

Despite the useful results in Fig. 4, some problems still exist apparently, like the detected errors of the proposed platform. As mentioned in Section 3.2, among the videos collected ($\#T$), we first compared the manually screened $\#NT_g$ (the ground-truth number of the videos with no larvae touched) with the numbers output from the proposed quantification platform ($\#NT_p$), shown in Table 3, with the percentage ($E_{NT} = |\#NT_g - \#NT_p|/\#T$). As well, we also give the number of failure of quantification ($\#QF$) with the percentage ($E_{QF} = \#QF/|\#T - \#NT_g|$). Our proposed platform can with more than 90% in average find the larvae not touched. Besides, no results were generated from around 10% of valid videos ($\#T - \#NT_g$) by our system. In addition, we assume that the larvae under the treatment of *Dia* scarcely have response, so the output of latency time is expected to be infinite, and the other indices are expected to be 0. However, the system cannot generate infinite number, but from Fig. 4a, the latency time is the highest which is still useful to be compared with the other cases. Furthermore, the results in Fig. 4b-4e are not exactly zero, caused by following reasons: i) Some larvae still have slight response; ii) The movements of the needle might push the larvae away (fake response); ii) The tracking procedure generates the movements of the larvae because of the slight environment changes or other inaccuracy. Nonetheless, the results of Fig. 4b-4e are still comparable to the other cases, and in other words, our proposed system verified our assumption on treatment *Dia* even with slight variance. Finally, the proposed platform can achieve the quantification in a higher efficiency with in average 63 ms per frame on CPU, compared with using U-Net for the tracking procedure (in average 2.60 s per frame on CPU).

Table 3: The analysis of the detected errors (failure cases) of the proposed platform.

Type	$\#T$	$\#NT_g$	$\#NT_p$	E_{NT}	$\#QF$	E_{QF}
Wild	24	4	7	12.5%	1	5%
DMSO	27	3	8	18.5%	0	0%
Dia	38	4	1	7.9%	8	23.5%
Iso	30	6	6	0%	2	8.3%
Caffi	24	7	5	8.3%	3	17.6%
Saha	30	5	6	3.3%	2	8%
Average	-	-	-	8.4%	-	10.4%

3.4 Discussion

The results in Section 5 verified that our proposed platform can work as an automated quantification tool for the multi-larvae touch-response experimental data in a high frame rate. This platform has following advantageous strategies: i) The decision of t_1 as well as the touched larva is decided by the last point of the needle and the initialized point of the larvae, as the local segmentation of the larva during tracking procedure is not as accurate as the initialized segmentation by the U-Net; ii) The movement of the larvae is decided by the change of each particle instead of the change of the larva center, as the centers of the larvae might change slightly but constantly during the tracking procedure, even when the larvae do not actually move; iii) The decision of t_4 is done from the last frame to the previous, since the larva might move slowly (no significant changes of pixels) for a moment and start moving strongly again; iv) The design of the quantification pipeline makes it possible to consider the global information for a more reasonable quantification result, as the quantification is achieved after the tracking task of all frames.

However, some drawbacks are still needed to consider carefully when the users apply this platform or pipeline to the customized data. The tracking procedure and local segmentation of the larvae are the keys for this quantification platform, but they cannot be guaranteed for a good result in following cases: i) the larvae overlap with each other when moving; ii) the well edge area has similar brightness with the larvae; iii) the needle overlaps with the larvae.

Nevertheless, these problems can be solved by statistical analysis of a large set of data, so our proposed platform is vital in such case.

4 Conclusion

In this work, we introduced a machine learning based quantification platform for touch response of zebrafish larvae, which can generate five quantification indices (latency time, C-Bend curvature maximum, C-Bend curvature peak time, response time, and moving distance) automatically without human intervention. This platform uses an automated quantification pipeline based on a multi-larvae tracking procedure, with a U-Net for initialization of tracking procedure, a particle filter for tracking, and region growing for local segmentation of larvae. To test the performance of the proposed quantification pipeline, six sets of experiments (2 controls and 4 treatments) were conducted and the results generated from this platform as well as the analysis of the detected errors verified the effectiveness of the platform. A high efficiency is also guaranteed with in average 63 ms per frame for the quantification pipeline on CPU. Our future work will be to apply our proposed platform on more data from other drug screening of touch response of zebrafish larvae.

Acknowledgements

This work was financially funded by China Scholarship Council (CSC). The authors would also like to thank the program of Natural, Artificial and Cognitive Information Processing (NACIP) and the BioInterfaces International Graduate School (BIF-IGS) at the Karlsruhe Institute of Technology (KIT).

References

[1] L. Saint-Amant and P. Drapeau. “Time course of the development of motor behaviors in the zebrafish embryo”. In: *Journal of Neurobiology* 37.4. S. 622-632. 1998.

[2] Y. Wang, D. Marcato, V. Tirumalasetty, N.K. Kanagaraj, C. Pylatiuk, R. Mikut, R. Peravali and M. Reischl. “An automated experimentation system for the touch-Response quantification of zebrafish larvae”. In: *IEEE Transactions on Automation Science and Engineering* S. 1-13. 2021.

[3] Q. Zhu, Y. Wang, Y. He and X. Hong. “Object tracking with particles weighted by region proposal network”. In: *Multimedia Tools and Applications* 78.9. S. 12083–12101. 2019.

[4] D. Marcato: “An automated and high-throughput photomotor response platform for chemical screens”. In: *37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)* (Alshut, R.; Breitwieser, H.; Mikut, R.; Strähle, U.; Pylatiuk, C.; Peravali, R.), S. 7728-7731. 2015.

[5] G. Audira, B.P. Sampurna, S. Juniardi, S.T. Liang, Y.H. Lai and C.D. Hsiao. “A simple setup to perform 3D locomotion tracking in zebrafish by using a single camera”. In: *Inventions* 3.1. S. 11. 2018.

[6] M. Schutera, T. Dickmeis, M. Mione, R. Peravali, D. Marcato, M. Reischl, R. Mikut and C. Pylatiuk. “Automated phenotype pattern recognition of zebrafish for high-throughput screening”. In: *Bioengineered* 7.4. S. 261-265. 2016.

[7] R.M. Basnet, D. Zizioli, S. Taweedet, D. Finazzi and M. Memo. “Zebrafish larvae as a behavioral model in neuropharmacology”. In: *Biomedicines* 7.1. S. 23. 2019.

[8] A.A. Popova, D. Marcato, R. Peravali, I. Wehl, U. Schepers and P.A. Levkin. “Fish-microarray: a miniaturized platform for single-embryo high-throughput screenings”. In: *Advanced Functional Materials* 28.3. S. 1703486. 2018.

[9] X. Wang: “Crowdsourced generation of annotated video datasets: a zebrafish larvae dataset for video segmentation and tracking evaluation”. In: *IEEE Life Sciences Conference (LSC)* (Cheng, E.; Burnett, I.S.; Huang, Y.; Wlodkowic, D.), S. 274-277. 2017.

[10] X. Wang: "Automatic tracking of multiple zebrafish larvae with resilience against segmentation errors". In: *IEEE 15th International Symposium on Biomedical Imaging (ISBI)* (Cheng, E.; Burnett, I.S.; Wilkinson, R.; Lech, M.), S. 1157-1160. 2018.

[11] Y.X. Bai, S.H. Zhang, Z. Fan, X.Y. Liu, X. Zhao, X.Z. Feng and M.Z. Sun. "Automatic multiple zebrafish tracking based on improved HOG features". In: *Scientific Reports* 8.1. S. 1-14. 2018.

[12] B.K. Horn and B.G. Schunck. "Determining optical flow". In: *Artificial Intelligence* 17.(1-3). S. 185–203. 1981.

[13] T. Senst, V. Eiselein and T. Sikora. "Robust local optical flow for feature tracking". In: *IEEE Transactions on Circuits and Systems for Video Technology* 22.9. S. 1377-1387. 2012.

[14] H. Richendrfer, S.D. Pelkowski, R.M. Colwill and R. Creton. "On the edge: pharmacological evidence for anxiety-related behavior in zebrafish larvae". In: *Behavioural Brain Research* 228.1. S. 99-106. 2012.

[15] F. Romero-Ferrero, M.G. Bergomi, R.C. Hinz, F.J. Heras and G.G. de Polavieja. "Idtracker. ai: tracking all individuals in small or large collectives of unmarked animals". In: *Nature Methods* 16.2. S. 179-182. 2019.

[16] X. Wang, E. Cheng, I.S. Burnett, Y. Huang and D. Wlodkowic. "Automatic multiple zebrafish larvae tracking in unconstrained microscopic video conditions". In: *Scientific Reports* 7.1. S. 1-8. 2017.

[17] S. Ren, K. He, R. Girshick and J. Sun. "Faster r-cnn: Towards real-time object detection with region proposal networks". In: *Advances in Neural Information Processing Systems* 28. S. 91-99. 2015.

[18] O. Ronneberger: "U-net: Convolutional networks for biomedical image segmentation". In: *International Conference on Medical image computing and computer-assisted intervention* (Fischer, P.; Brox, T.), S. 234-241. 2015.

[19] Y. Wang, N.K. Kanagaraj, C. Pylatiuk, R. Mikut, R. Peravali, M. Reischl. “High-throughput data acquisition platform for multi-larvae touch-response behavior screening of zebrafish”. In: *IEEE Robotics and Automation Letters*. submitted in 2021.

[20] V. Bedell, E. Buglo, D. Marcato, C. Pylatiuk, R. Mikut, J. Stegmaier, W. Scudder, M. Wray, S. Züchner, U. Strähle and R. Peravali. “Zebrafish: a pharmacogenetic model for anesthesia”. In: *Methods in Enzymology*, 602. S. 189-209. 2018.