Computer Vision for Zebrafish

To automate detection of anomalies in Zebrafish larvae

Non-sensitive

Volume 1
Report No 1
August 28th, 2025
Rishabh Sharma

National Research Council Canada | Conseil national de recherches Canada

Canada

Table of contents

Introduction

Background

Zebrafish *(Danio Rerio)* are some of the most widely used vertebrate in developmental biology and toxicology due to genetic similarity in humans (*Government of Canada*), rapid maturation, transparent embryos, and small size. The Zebrafish Research Facility at the National Research Council routinely performs 'phenotype checking', providing insight on how genetic of chemical factors impact early development, organ formation, and overall viability of these zebrafish. Effectively, a well filled with one zebrafish egg in cell is injected with a certain chemical or has a certain environmental element placed on them. Then, after a set amount of time (the fish used in this experiment were developed between 96 to 120 hours past fertilization (hpf)), phenotypes can be observed from this set of larvae. By categorizing phenotypic outcomes, researchers can identity sublet abnormalities, quantify developmental toxicity, and link biological mechanisms to observable traits.

In this report, we will use the following phenotypes in our processes, with the appropriate abbreviation attached to each respective one:

Note hpf → hours past fertilization

- ***NOE (No observable effect):*** *Embryos appear normal for their developmental stage, showing pigmentation ≥48 hpf and hatching ≥72 hpf.*
- ***H (Hatched):*** *Embryos that hatched early at 24 hpf; considered "affected" at this stage.*
- ***UH (Unhatched):*** *Embryos that fail to hatch ≥48 hpf; only scored as "affected" when still unhatched at ≥72 hpf.*
- ***LR (Loss of lateral recumbency):*** *Loss of the normal resting side posture, seen >72 hpf; only counted as "affected" at 120 hpf.*
- ***TSU (Trouble staying upright):*** *Difficulty maintaining upright position >72 hpf; tracked but not scored as "affected."*
- ***SB- (Swim bladder absence):*** *Failure to inflate swim bladder at 120 hpf, impacting buoyancy.*
- ***PCE/YSE (Pericardial / Yolk surface edema):*** Abnormal swelling around the heart or yolk.
- ***HB (Heartbeat abnormality):*** Noticeably slow or irregular heart rhythm.
- ***LC (Lighter colour):*** Reduced pigmentation compared to normal embryos.

- ***MA/DM (Melanocyte aggregation / Donut melanocyte):*** Pigment clusters or ring-shaped melanocytes at edges.
- ***DP (Dark pigment):*** Abnormally high pigmentation levels.
- ***CM (Constant movement):*** Hyperactive, continuous movement not typical of normal embryos.
- ***Nec (Necrotic/dead):*** Cell death or dead embryo.
- ***ABN (Abnormal embryo):*** General malformations beyond specific categories.
- ***GH/CY (Grey head / Cloudy yolk):*** Opaque head or yolk, suggesting developmental issues.
- ***ST (Stubby tail):*** Shortened or truncated tail.
- ***SH (Small head):*** Head size smaller than expected for stage.
- ***BP (Blood pooling):*** Accumulation of blood in localized regions.
- ***L (Lordosis):*** Abnormal inward spinal curvature.
- ***Sc (Scoliosis):*** Lateral spine curvature.
- ***Ky (Kyphosis):*** Outward spinal curvature (hunchback).
- ***Cu (Curled body):*** Body remains curled as if inside the chorion.
- ***LY/LL/DY (Large yolk / Large lumen / Disrupted yolk):*** Enlarged yolk sac, abnormal intestinal lumen, or disorganized yolk structure.
- ***DL/EL/DN (Dark liver / Enlarged liver / Dark nose/mandible):*** Liver abnormalities or necrotic pigmentation in nose/jaw.
- ***Ot (Malformed otolith):*** Defects in otolith (inner ear structure).
- ***Jw (Malformed jaw):*** Abnormal jaw development.
- ***RF (Ruffled fin):*** Fins appear uneven or ragged.
- ***/ (Missing embryo):*** Embryo not present in the well.

Bottleneck in Current Processes

A major bottleneck in this whole process lies in the current manual scoring of phenotypes. Identifying and categorizing abnormalities in larvae requires trained observers to visually inspect 100s to 1000s of larvae, often under microscope. This process in time-consuming, subjective and prone to variability, especially when distinguishing between subtle morphological traits such as pigmentation changes, edema, or spinal curves. The sheer diversity of phenotypes also complicates standardization of these processes.

Purpose

As a result, there is a need for a way to automate image analysis to streamline phenotype detection and improve reliability. With modern machine learning (ML) frameworks, this task

becomes easier than ever. Our main goal in this report is to develop a ML framework that uses computer vision (CV) to be trained on various phenotypes in zebrafish larvae, being able to decipher the phenotypes present in any given larvae with reasonable accuracy.

The end product should be an application where the researcher uploads and image of the 96-cell well of larvae, and each cell is scored and exported to one collected .csv file. This .csv file can then be manually checked by the researcher if required. In doing so, we hope to remove the subjectiveness out of the process and automate the main process of scoring each larva for their respective phenotypes. ***Although this end product was not implemented as of writing, the goal of this report is to detail the progress we made towards the finish line.***

Automated zebrafish phenotype scoring has gained momentum over the past decade, with notable tools such as FishInspector for morphological trait detection (Scholz), KimmelNet for automated staging (Jones et al.), and deep learning frameworks like EmbryoNet, which leveraged millions of images to classify defects tied to signaling pathways with higher accuracy than human experts (Čapek et al.). Similarly, the ZACAF framework has demonstrated strong results in quantifying cardiovascular parameters from live imaging (Naderi et al.). While these advances highlight the promise of automation, they often focus on specific sub-tasks (e.g., heart function, staging, or narrow sets of morphological defects), require highly curated imaging conditions, or lack generalizability across diverse phenotypes. This leaves a gap in building models capable of robust, broad-spectrum classification of multiple developmental abnormalities directly from raw embryo images. Our work aims to fill this gap by developing a computer vision pipeline that generalizes across a wide range of phenotypic outcomes, enabling higher-throughput, consistent, and scalable zebrafish phenotype scoring.

Methods

The following section goes over our methodology, the architecture of our CV models, and the respective results of each model type.

Experimentation

As the following section will discuss, data was a bottleneck here and we only had a very limited dataset to work with. In total, we had three plates with 96 cells in each (data available in link in *Links and Works Cited* section). Hence, we had 96 x 3 = 288 larvae samples to build our model. However, out of our 288 samples, 51 of them were too blurry to make any reasonable guesses (see *Images* section for example), so they were omitted from the training, testing and

validation of the model. *Table 1* below has all the phenotypes found in our samples. When a certain fish had more than one phenotype labelled to it, we went with the

phenotype that allowed us to train the best model. In most cases, that meant when a fish was labelled both *SB* and *KY*, the data only had *SB* as the main phenotype, as other instances of *KY* weren't common in the dataset and it was more of a "one-off" occurrence. The *Images* section has examples of larvae from all sorts of phenotypes.

Phenotype	Number of Fish
Healthy	78
SB	133
Dead	14
Missing	6
Other	6

Table 1; Phenotypes of larvae samples

From above, the *Other* phenotype consists of a variety of different phenotypes that only existed 1-2 times in the whole dataset, so it wasn't worthwhile to train the model on them because no real patterns would be captured.

As the data is quite skewed in terms of the phenotypes we have (most are either *healthy* or *SB*), we decided the best course of action was to make a binary classifier to decipher between these two respective phenotypes. Then, we scale that to a multi-phenotype classifier to decipher between *healthy, SB*, *Dead* or *Missing*.

Success of the latter model was questionable from the start due to the lack of data from the *Dead* and *Missing* phenotype, but regardless we implemented it to show how our binary classifier could be scaled to any number of phenotypes in the future, once we get that data.

We also made another model to decipher whether or not a fish is on its side, or upright. The idea was that figuring out the orientation of certain larva might give us useful insights and might change our ultimate diagnosis for the final phenotypes we give said larva. The *Images* section show some examples of fish, both upright and on their side. Of the data points we had, *Table 2* below describes the orientation of the fish found in all 288 samples. As before, we omitted all samples that were too blurry, missing or that we couldn't decipher an actual orientation.

Commented [SR1]: Insert image of upright fish and fish on the side

Orientation	Number of Fish
Upright	166
Side	25

Table 2; Orientation of Samples

The data is skewed quite a bit once more, most samples are ‘upright’. Later, we will show how this impacts the binary classification model for this task in our *Results* section.

Architecture Explanation

All code is provided in the GitHub link in the *Links* section of this report.

First, we turned each plate of wells into its respective .csv file with two columns; *file_name* and *phenotype*. Although certain larvae had multiple phenotype labels, we didn’t have enough data to have a model that can predict multiple labels at once. Most instances of phenotypes like *KY* happened too infrequently to actually learn visual elements of said phenotype. Regardless, each .csv file was made by cross-referencing the sheets in the link in *Links and Works Cited*, resulting in *pageOne.csv, pageTwo.csv,* and *pageThree.*csv. Using *splitdata.py*, we then merged all three .csv files respectively and randomly split them into train, validation, and test sets, named appropriately in the repository.

In order to ensure there were no deviations in the data, all images were resized to 224x224 pixels and normalized using the ImageNet statistics (mean: [0.485, 0.456, 0.406] and standard distribution: [0.229, 0.224, 0.225]). Check out both *resize_and_normalize_images.py* and *define_transform* in *evaluate_efficient_net_multiclass.py*.

The actually train the model, we have two separate model types; *ResNet18* and *EfficientNet*.

ResNet18

This is a convolutional neural network (CNN) that uses residual connections to allow gradients to flow through deeper layers, mitigating the vanishing gradient problem (He et al., 2015). The architecture consists of 18 layers, including convolutional, pooling, and fully connected layers. In *train_pipeline.py*, the final fully connected layer was modified to output two classes (*healthy* and *SB*) for binary classification. Dropout was added for regularization, as shown in our *ModifiedResNet18* class.

EfficientNet-B0

In *train_efficientnet.py*, we can see this new model, which uses ‘compound scaling’ to balance depth, width and resolution. This all allows us to achieve relatively high accuracy with few parameters (Tan & Le, 2019). We use this EfficientNet for both binary (*healthy* and *SB*, as well as *Upright* and *side* orientations) and multi (*healthy*, *SB, dead, missing*) classification. The model was initialized with pre-trained ImageNet weights.

Training Procedure

Both models were trained using the Adam optimizer and cross-entropy loss. Early stopping was implemented to prevent overfitting, as detailed in *train_pipeline.py* and *train_efficientnet.py*. The full training loop is:

1. Forward propagation to compute predictions
2. Backward propagation to calculate gradients.
3. Weight updates using the optimizer.

For ResNet18, the binary classifier was trained on *healthy* and *sb* phenotypes. For EfficientNet-B0, the multiclass classifier was trained on *healthy*, *sb*, *dead*, and *missing*. The best performing models was saves as *best_model.pth* and *best_efficientnet_multiclass.pth*, respectively.

Evaluation

The evaluation scripts (*evaluate_model.py* for ResNet18 and *evaluate_efficientnet_multiclass.py* for EfficientNet-B0) load the trained models and applied them to our test set. Images were preprocessed using the same transformations as during training. Predictions were compared to actual labels, and metrics such as accuracy and loss were calculated. The results for everything are below.

Results

The following section will detail results from all four models:

1. Binary Classifier using ResNet

Confusion Matrix	Predicted: Healthy	Predicted: SB
Actual: Healthy	4	6
Actual: SB	2	18

Validation accuracy, as outputted by *evaluate_model.py*, is 78.57% with an average loss of 0.4912.

2. Binary Classifier using EfficientNet

Confusion Matrix	Predicted: Healthy	Predicted: SB
Actual: Healthy	0	10
Actual: SB	0	18

Validation accuracy, as outputted by *evaluate_efficientnet.py*, is 67.86% with an average loss of 0.6427.

3. Angle Classifier using EfficientNet

Confusion Matrix	Predicted: Top	Predicted: Side
Actual: Top	25	0
Actual: Side	7	0

Validation accuracy, as outputted by *evaluate_efficient_angle.py*, is 78.12% with an average loss of 0.5484.

4. Multi-classifier using EfficientNet

Confusion Matrix

True Positives: Healthy: 2 SB: 18 Dead: 0 Missing: 0	False Negatives: Healthy: 6 SB: 2 Dead: 5 Missing: 0
False Positives: Healthy: 2 SB: 11 Dead: 0 Missing: 0	True Negatives: Healthy: 23 SB: 2 Dead: 28 Missing: 33

Validation accuracy, as outputted by *evaluate_efficient_multi.py*, is an abysmal 57.58% with an average loss of 0.9658.

As we can see here, the models that are backed by skewed datasets have resulting biased outputs. For instance, in our binary classifier for the orientation of the larva, we have a lot of fish that have an upright view, so naturally most of the predictions (even if the image was of a larva on its side) was that the fish was upright.

Limitations

The performance of all models was suboptimal, primarily due to the skewed dataset. In order to improve the accuracy across all models, we need to address this data bottleneck.

Data Bottleneck

The dataset was both small (only 288 total larvae, with 51 discarded due to poor image quality) and heavily imbalanced (dominated by healthy and SB phenotypes). This resulted is biased classifiers that over-predicted the majority classes. Rare phenotypes such as *Dead*, *Missing* or *Side* were underrepresented to the point where no meaningful patterns could be learned. Additionally, since multiple phenotypes often co-occurred in a single-label training approach simplified the problem at the code of ignoring important biological variation.

Image Quality and Consistency

Blurry or poorly lit samples limited the amount of usable data and introduced noise into the training process. While images were normalized and resized, this all reduces the total samples we had to train with, and reduced generalizability of the models when applied outside this dataset.

Model Scope

Our current architectures (ResNet18, EfficientNet-B0) were adapted for small-scale binary and multi-class classification but were not optimized for detecting highly subtle features such as pigmentation changes or cardiovascular defects. Moreover, the models were trained on static images, whereas some phenotypes (e.g, heartbeat irregularities, constant movement) inherently require temporal information from video data.

Evaluation Limitations

Due to the limited dataset, train/validation/test splits were small, and metrics like accuracy may not fully capture real-world performance. Models were not benchmarked against human exprt scores, so the comparative advantage of automation remains unquantified.

Next Steps

Expand and Balance the Dataset

The most immediate priority is to collect a significantly larger number of larvae images across all phenotypes. This should include replicates under varying experimental conditions to capture natural variability. Techniques such as oversampling minority classes, synthetic augmentation

(e.g., rotation, contrast adjustments), or generative models (GANs, diffusion models) can be used to reduce imbalance until more data is gathered.

Move Toward Multi-Label Classification

Future models should support multi-label outputs to reflect the biological reality that larvae can exhibit multiple phenotypes simultaneously (e.g., SB and kyphosis). This would improve both accuracy and practical utility for researchers.

Incorporate Temporal Data

For phenotypes like CM (constant movement) or HB (heartbeat abnormalities), still images are insufficient. Integrating video-based models, such as 3D CNNs or transformer-based vision architectures, will allow temporal dynamics to be captured.

Improve Imaging and Preprocessing Pipelines

Developing standardized imaging protocols (consistent lighting, magnification, positioning, etc.) will minimize noise. Automated quality control filters should also be implemented to flag blurry or occluded samples before model training.

Benchmark Against Human Experts

Establishing a baseline by comparing model predictions to human expert scorers will highlight areas where automation provides the most value and where further refinement is required. This would also help build trust among researchers considering adoption of the tool.

Broaden Model Architectures

Exploring architectures beyond ResNet/ EfficientNet, such as Vision Transformers, ConvNet Models, or hybrid CNN-transformers approaches, may yield better generalization across diverse phenotypes. Transfer learning from biomedical imaging datasets should also be considered.

Integration Into Research Workflows

In order to make what we built useful, we must also build a user-friendly interface, with potential for live integration into the NRC Zebrafish Research Facility workflow. A deeper dive into the Human-Computer Interaction required here will ne needed.

Conclusion

This project demonstrated the feasibility of using computer vision models to automate phenotype scoring in zebrafish larvae, a process traditionally reliant on time-consuming manual inspection. While our initial models showed only modest accuracy due to small, imbalanced datasets and variability in image quality, the pipeline established here provides a foundation for scaling. By expanding and balancing the dataset, adopting multi-label and temporal modeling approaches, and benchmarking against human experts, the framework can evolve into a robust, generalizable tool for zebrafish research.

Ultimately, integrating this system into researcher workflows as an end-to-end application—where raw well images are automatically scored and exported into structured outputs—has the potential to transform the throughput, consistency, and reproducibility of phenotype assessment. This work marks a first step toward that goal, laying the groundwork for a scalable digital platform that reduces subjectivity and accelerates biological discovery.

Links and Works Cited

GitHub Repository (Public, Open Access): GitHub - ri3habh/zebrafish: CV models for the Canadian Zebrafish Research Facility at the National Research Council of Canada.

Link to Data Images (Restricted SharePoint Link):

Link to Scoring Sheet of Data (Restricted SharePoint Link) : CV - Rishabh

Canada, National Research Council. “Government of Canada.” *National Research Council Canada*, / Gouvernement du Canada, 17 July 2024, nrc.canada.ca/en/research-development/nrc-facilities/zebrafish-research-facility.

Jones, Rebecca A, et al. “Automated Staging of Zebrafish Embryos with Deep Learning.” *Life Science Alliance*, Life Science Alliance, 1 Jan. 2024, www.life-science-alliance.org/content/7/1/e202302351

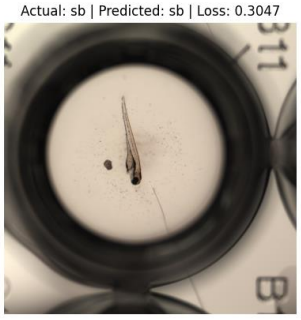
Naderi, Amir Mohammad, et al. “Deep Learning-Based Framework for Cardiac Function Assessment in Embryonic Zebrafish from Heart Beating Videos.” *arXiv.Org*, 24 Feb. 2021, arxiv.org/abs/2102.12173

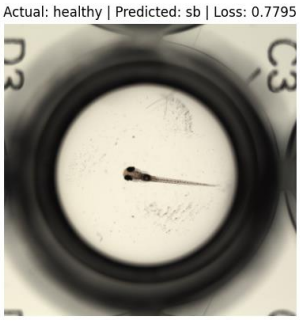
Scholz, Dr. rer. nat. Stefan. *Fishinspector - Helmholtz-Zentrum Für Umweltforschung Ufz*, www.ufz.de/index.php?de=44460. Accessed 21 Aug. 2025.

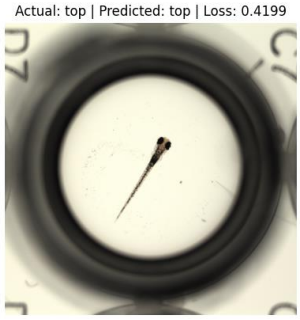
Čapek, Daniel, et al. “EmbryoNet: Using Deep Learning to Link Embryonic Phenotypes to Signaling Pathways.” *Nature News*, Nature Publishing Group, 8 May 2023, www.nature.com/articles/s41592-023-01873-4

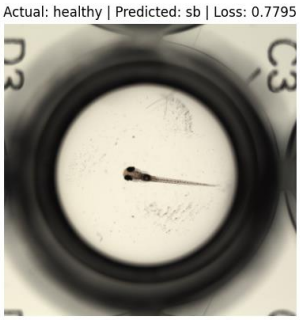
Images

The following are examples of output from our models, showcasing the actual label, our predicted label, and the corresponding loss score:



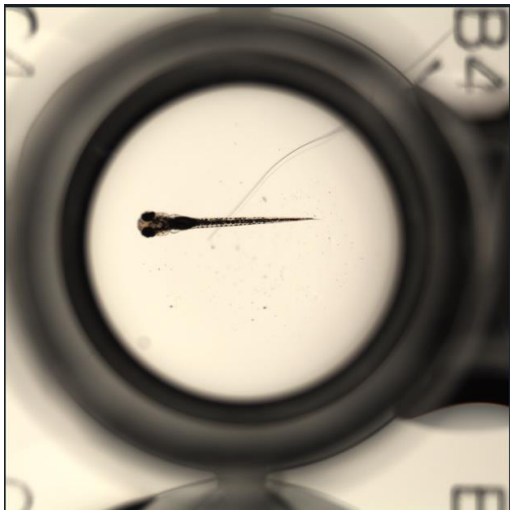






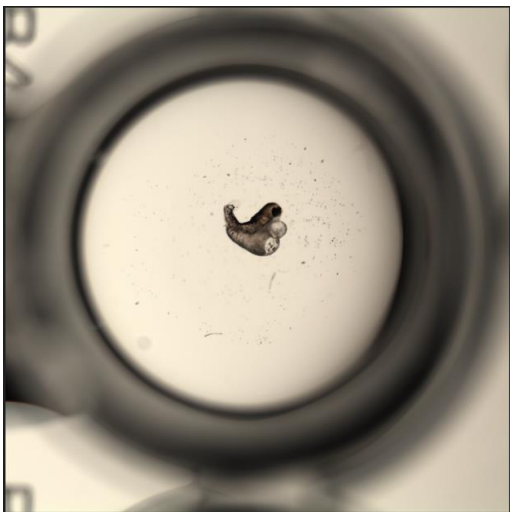
And these are examples of larvae with a variety of phenotypes and orientations:

Health and Upright:

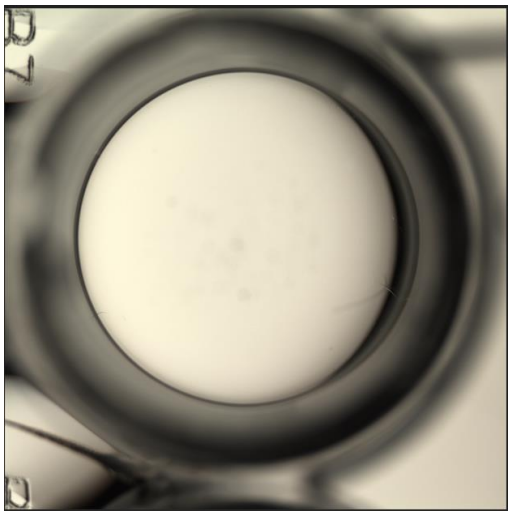


SB and Upright:

Dead and Upright:



Missing:



SB and Side:

Dead and Side:

Too Blurry: