Mutation detection in zebrafish hearts using 3D point cloud classification

Hyung Jip LEE

June 2021

Abstract: Accurately determining nuclei locations and density in fluorescence images is important in many biomedical research projects. Nuclei density and location are both highly involved in the developmental process of the heart. Studies in developmental biology showed that nuclei density and location are useful for the detection of various mutations in the heart. Zebrafish are used as a vertebrate model in many studies. The reasons behind the popularity of zebrafish models are: 1. their genome is easy to manipulate 2. their developmental process is well documented 3. they are suitable for various experimental techniques. Their heart morphology, heart rate and action potential duration are very similar to human hearts [1].

Therefore, understanding the morphology of mutant zebrafish hearts can help fine-tuning mutation detection in human hearts. In this work, a classification model will be built to classify control/mutant images based on the geometric features of the nuclei locations. Hand-operated classification based on nuclei density of these hearts can be time consuming and erroneous. Hence it is necessary to build an automatic classification system.

For this binary classification problem, past works rely on traditional convolutional models and segmentation [2], [3]. A different approach is proposed in this work. That is to study the point clouds representing the nuclei locations: 3D Point cloud feature extraction and classification using PointNet [4]. Due to the lack of data, augmentation will be applied to the training dataset.

It is found that an accuracy of maximum 100% can be achieved using this process. However, note that the size of the validation set is 12 samples.

Despite a good classification accuracy, limitations and possibilities of improvement are found in this work. An even more optimal learning rate can be found using cyclical learning rate while training. The range of point jittering can be further modified and the optimal range of it can be studied. The training size is also very small which makes the evaluation of validation loss and accuracy difficult.

1 Introduction

Mutation detection in hearts is an important problem in the field of biomedical research. It is common to approach this problem as an image classification problem.

However, 3D image classification with multiple channels require excessive amounts of computation, time and memory. Past studies involved image data, using typical convolutional models which require regular input data formats. Some of the examples of CNN for 3D Data consist of Volumetric CNNs [2] or spectral CNNs [3]. However these approaches are believed to experience constraints on low resolution due to data sparsity or organic objects, respectively. Such image data are commonly transformed into voxel grids or multiple images in order to make them regular. However, this is known to yield voluminous input data and other artefacts [4]. Therefore in this work, it is proposed to convert the voluminous image data into a much simpler format of data: point clouds. This approach requires less data pre-processing and modification than with traditional convolutional models. Past studies involve hand-craft features of point clouds which are known to ignore semantic and spatial information [5]. This method is not applicable to the classification problem of interest where geometrical features of point cloud is key. Using an architecture that directly takes point clouds as input has advantages over traditional convolutional networks: no regularisation or registration is needed, and less computation is required. On the other hand, learning local features that express the relation between local points [5] is more relevant to the problem.

Given the advantages of point clouds over images, and PointNet over traditional point cloud processing models, the mutation detection problem is approached as a point cloud classification problem. In this work, the aim is to train the mdoel to achieve 100% accuracy on validation dataset which is achievable because of the small size of validation set, while avoiding overfitting. The specific objectives are to: 1. appropriately convert the 3D images into 3D point clouds 2. construct a dataset and augment it and 3. fine-tune the training process with optimal parameter values and regularisation methods.

2 Methods

2.1 Overall workflow

First of all, the dataset consists of a single .lif file, containing 24 3D confocal microscopy images. 10 of those images are of control hearts, and 14 are mutants. Each 3D image is made of z 512x512 2D images (Fig.1). The value of z specifies the location of the 2D image in 3D, typically ranging from 130 to 180. Using 3D Objects Counter from ImageJ, the 3D images are transformed into point clouds of nuclei centroids. These point clouds are stored as excel files with 3 columns (X, Y and Z coordinates) each. The number of rows in this file corresponds to the number of nuclei counted in the image. These table are sampled and fed into the PointNet model for training.

2.2 Nuclear Staining

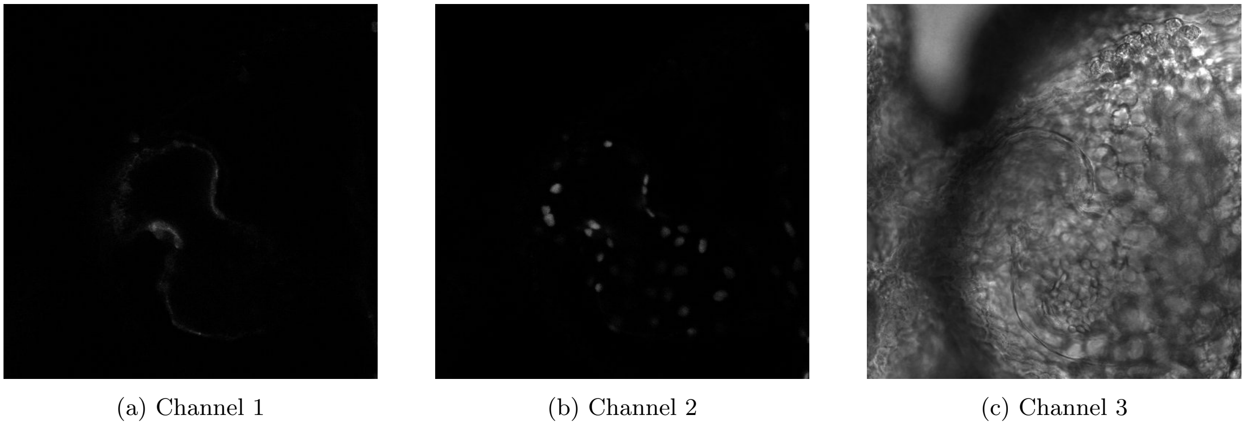
A plasmid is injected to all of the zebrafish eggs at disposal. The plasmid consists of a gene that includes a promotor *fli1* that drives the expression of GFPs in endothelial cells [6]. The promotor is coupled with a nuclear signal, which induces the transport of GFPs into the nucleus of the cell, and hence the fluorescence of the nuclei. In some of the embryos, the injected gene is integrated into the genome and expressed in the cell nuclei (Fig.1b). For RFP expression in other embryos, the *mCherry-LifeAct* [7] couple was used.

2.3 Image Acquisition Protocol

The dataset consists of 24 confocal 3D microscopy images (24 series). There were different steps involved in obtaining the images. For live imaging of the Zebrafish embryo heart, the motion of the Zebrafish must be stopped. To do this, dechorionated Zebrafish embryos were anesthetized using 0.2mg/mL of tricaine (Sigma Aldrich, A5040; stock at 8mg/mL adjusted to pH7-7.5) or 50mM drug 2,3-butanedione monoxime (BDM) (Sigma Aldrich, B0753) for 10 minutes (to stop the heart), and then mounted in 1.2% UltraPure low melting-point agarose (Sigma Aldrich, 16520) that has the same concentration of tricaine or BDM as the media. Confocal imaging was performed either on an up-right Leica SP8-Multiphoton confocal microscope (to image the stopped heart or fixed embryos). A custom-designed mold was used for imaging on the up-right Leica SP8-MP confocal microscope [8]. As a result, images in Fig.1 were obtained (See Ap.1 for an example of 3D visualisations).

2.4 Data Augmentation

First of all, note that only the training set is augmented because the validation set must be representative of the original dataset. The purpose of data augmentation is to improve classification accuracy and robustness [9]. There are various ways of data augmentation. In this project, the training set was augmented simply by transforming the data through randomly jittering points with uniform noise

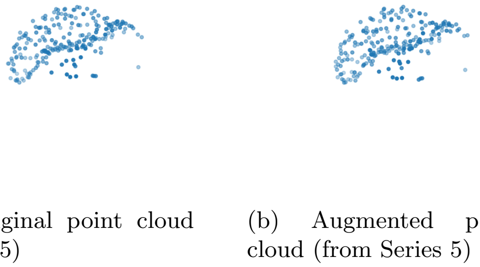

(a) Channel 1

(b) Channel 2

(c) Channel 3

Figure 1: Three channels of Series 1 at $z = 107$

[10] which is to add random numbers (of arbitrary range) to coordinates.


(a) Original point cloud (Series 5)

(b) Augmented point cloud (from Series 5)

Figure 2: Visualisations of point clouds

2.5 PointNet Architecture

PointNet [4] is one of the models that achieved remarkable results in point cloud feature extraction. This model directly takes the point cloud simply consisting of 3D coordinates of the cell nuclei without pre-processing. PointNet takes a point cloud $x_1, ...x_n$ as input where x_n denotes the nth point in the point cloud.

A point cloud is a set and so is invariant to point permutations. Therefore, the output of a model that directly takes a point cloud as input should also be invariant to permutations.

The first MLP in the architecture (Fig.3) maps n points from 3 dimensions (3D space) to 64 dimensions and a single MLP is shared for every n points. This is important to note because it means that the mapping is identical and so independent for all n points. After this mapping, another mapping from 64 to 1024 dimensions is performed for all n points. The permutation invariance design continues with max pooling applied to the $n \times 1024$ matrix to create a global feature of the point cloud. Then the final MLP computes the output scores from this global feature vector. In our case, $k = 1$ because there are two classes, and therefore a single output unit.

The invariance to rotation is implemented in the 'input transform' part with T-Net. The aim of T-Net is to provide pose normalisation for the input. To achieve this, the T-Net applies an appropriate rigid or affine transformation to the $n \times 3$ input point cloud matrix by multiplying it with a transformation matrix. T-Net is a regression network that predicts this input dependant transformation matrix. Just like PointNet, the T-Net itself has point independent feature extraction (MLP), max pooling and fully connected layers. T-Net is used twice in PointNet: one to predict a 3×3 matrix which is multiplied to the $n \times 3$ coordinates matrix and another one to predict a 64×64 matrix which is multiplied to the $n \times 64$

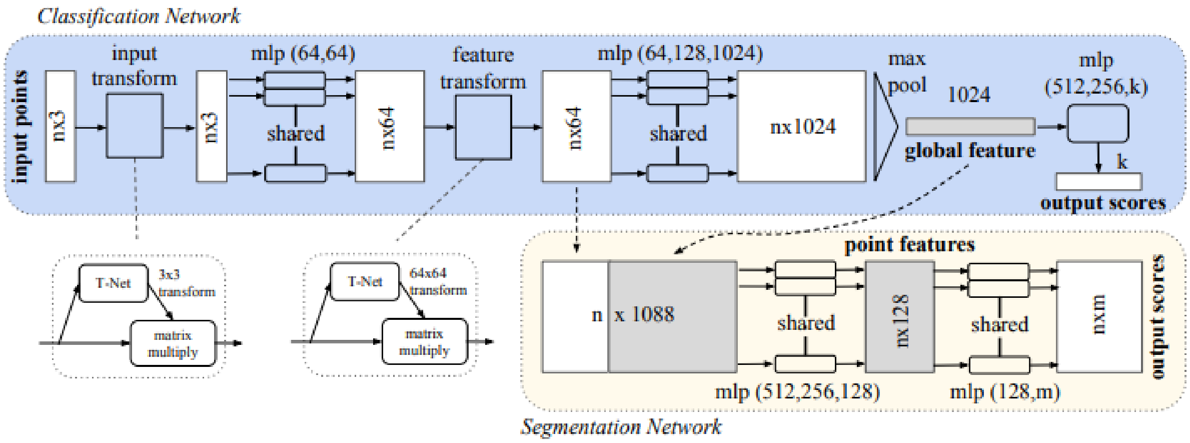
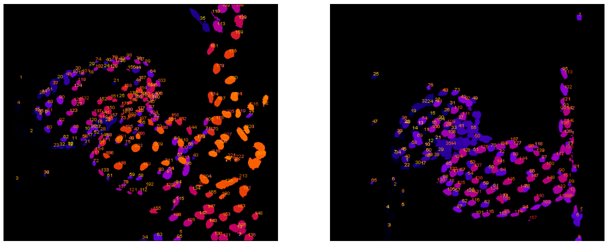


Figure 3: Architecture of the PointNet[4], the segmentation network is not used because segmentation within a single point cloud is not performed.

feature matrix. This achieves pose normalisation for the input [11].

2.6 3D Objects Counter (3D OC)

Each 3D image was acquired and segmented (Fig.4) into objects. The geometrical centers of each object is taken as nuclei centroids. The object map is then transformed into 3D point clouds (Fig.5) using an image processing tool called ImageJ. A table of 3D coordinates of all nuclei centroids found in each .tif images was obtained, and features were extracted from it.



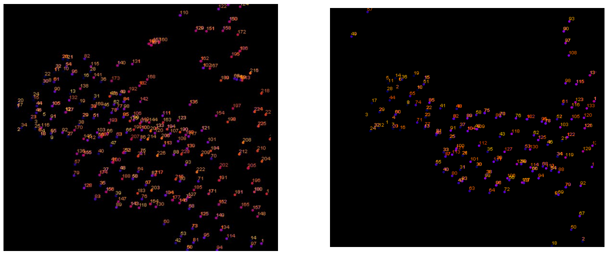
(a) Healthy sample from Series 4

(b) Mutant sample from Series 10

Figure 4: Cell nuclei object maps

3D OC [12] processes all stacks of a 3D image. Pixels deriving from noise are assumed to have lower intensities than pixel deriving from fluorescence signals. For noise removal, each image stack is turned into a grey scale image by applying an adaptive threshold. This means that a pixel value that is above the threshold is taken as part of the nuclei, although full noise removal is rarely achieved. Objects are detected when pixel values suddenly change while performing line scan. A new image that contains changes in pixel values can be obtained using connexity analysis[13]. This method consists of inspecting neighbour pixels (26 voxels in 3D space) of a reference pixel. All neighbour pixels of higher intensity than the reference pixel is labeled as part of the same object as the reference pixel. In this work, all nuclei have similar shapes, so a method called top-hat filtering[13], [14] was also used to segment nuclei.

After segmentation, geometrical centers (centroids) can be computed from the segmented objects (Fig.5). This is done in 3-dimensional space. The centroids of segmented objects are computed using the following formulae:



(a) Control point cloud of nuclei centroids from Series 7 of the .lif dataset

(b) Mutant point cloud of nuclei centroids from Series 19 of the .lif dataset

Figure 5: Visualisations of point clouds

$$C_x = \frac{\Sigma C_{i_x} A_i}{\Sigma A_i}$$

$$C_y = \frac{\Sigma C_{i_y} A_i}{\Sigma A_i}$$

$$C_z = \frac{\Sigma C_{i_z} A_i}{\Sigma A_i}$$

where C_x, C_y, and C_z are the x, y and z coordinates of the geometrical center of the segmented object.

2.7 Training Process

First, the excel files containing the centroid coordinates were shuffled in random order. Then, the 24 files were split in half: the first 12 files make the initial training set and the last 12 make the test set.

In order to clarify the training set construction process, some explanation about data augmentation will be made. While iterating through the files, all point clouds were added to the training set. When the sample is augmented, it means that the non-augmented version is already added to the training set, and the augmented version of the sample is then added. This means that if a sample is augmented once, then 2 samples are added to the training set: the original point cloud and the augmented point cloud.

In this work, the training set was constructed by iterating through the 12 excel files and by augmenting each of the samples. Meaning that if the number of augmentation per sample (this will be called augmentation number) is equal to 3, then each sample would be augmented three times and so the final size of the training set would be 48. In general, $finaltrainingsize = 12 \times (augmentationnumber + 1)$, with an original number of samples of 12 in this case.

Since there are 10 files of healthy hearts and 14 files of mutant hearts, if all samples are equally augmented, then the two classes will not be equally present in the training set. To maximise model performance, every class have to be equally present in the training set. Therefore, different augmentation numbers were used to balance the sample numbers from both classes. The following definition of the relation between the augmentation number on control samples and the augmentation number on mutant samples allows equal representation of each class in the training set (Ap.2):

$$A_m = int(\frac{N}{12 - N}(A_c + 1) - 1)$$

Where A_m is the augmentation number on mutant samples, A_c is the augmentation number on control samples, and N is the number of control samples in the initial training set.

Batch size allows us to control the stability of the model. Indeed, if the batch size is too large, then it negatively affects the model accuracy since it reduces the stochasticity of the gradient descent. A typical batch size is 32. In this work, an augmentation number on control samples A_c of 180 was used, which yields a training size of around $804 - 1004$ point clouds. A batch size of 32 yields 26 iterations per epoch. Any batch size under 80 did not affect the loss and the accuracy, meaning that the model performed equally with batch sizes varying around 32, 48, and 64. The difference in model performance with batch size 32 and 64 was not significant. The smaller batch size was chosen because the loss graph was more smooth, allowing us to obtain a training curve with less spikes, hence the choice of a batch size of 32.

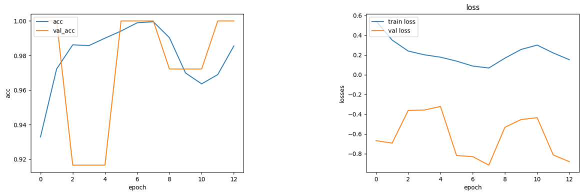
The learning rate is the most important parameter during the training process. Indeed, this parameter plays a crucial role in the training process as it affects the convergence or divergence of the model. A learning rate that is too high can cause overshooting of gradient descent. A learning rate that is too low can cause the gradient descent to be slow which means more computation and training time. With the given dataset, a learning rate higher than 0.07 yielded an infinite validation loss. As learning rate was decreased down to 0.008, the validation loss was significantly lower and validation accuracy was higher on average, maximising at 100%, although it oscillated 90% (Fig.6). The fluctuations in the validation loss also decreased.

While training, validation loss was fluctuating (Ap.5) around 0.6-0.7 while the training loss was converging. This means that the model was overfitting: it performs well on the training set but it randomly guesses the class of samples in the validation set. To reduce overfitting, the layers in the model were simplified and the number of parameters was reduced. DropOut layers were added before the activation layer which helped reduce overfitting. Earlystopping of the training based on the validation loss was also applied as another regularization method. Earlystopping consists of stopping the training process if the validation loss does not improve. This is implemented using a Keras Callback method. The patience was set to 5, meaning that if the validation loss does not decrease in 5 epochs, the training will stop. With these regularization methods and with parameter values that led to best performance, the model was successfully trained.

3 Results

The PointNet model was trained as explained the section 2.5. Under these conditions, the model achieved an accuracy of maximum 100% over the training set (Fig 6). As a final result, validation loss was found to be lower than training loss. The final choice of parameters is:

learning rate	0.008
batch size	32
A_c	180



(a) Validation and Training accuracy

(b) Validation and Training loss, y-axis log scale

Figure 6: Training Results

4 Discussion

The aim in this work is to classify control hearts and mutant hearts. For this, confocal microscopy images of zebrafish hearts were taken, transformed into point clouds, and used to train a PointNet model for binary classification.

These results suggest that with the given parameter values, it is possible to obtain an accuracy of 90%-100% over the validation set using PointNet. This accuracy is similar to what past point cloud classification studies have achieved [4], [5], [9], [15]. However, note that those models were trained on popular, well-documented and organised datasets. If such point cloud dataset is constructed with confocal microscopy images of zebrafish hearts, then the model trained in this work can be compared to literature further in detail. Even though the number of samples was increased a two-hundred folds using data augmentation, more robustness test has to be done with new data in the validation set. It is likely that the model will not be able to generalise well on point clouds with unobserved features.

As seen in the final result, the validation loss is lower than training loss. It is suspected that the validation set turns out to be too 'easy' for the model. This is possible because the validation set is very small, consisting of only 12 samples. Also, uneven sampling from each class can occur in the validation set because originally, there are 10 healthy heart images and 14 mutant images. Indeed, in many cases, the validation set had 5 healthy samples and 7 mutant samples. Data augmentation can also be the cause of higher validation loss as around 99.2% of the training set was made of augmented samples (when augmentation number on healthy samples was 180). On the other hand, high validation accuracy, oscillating above 90%, was obtained. Validation loss and accuracy can be better evaluated with more samples in the validation set. Indeed, as observed in the accuracy plot, the values of the validation accuracy is very discrete due to a limited number of validation samples.

One important limitation of this work is the size of the dataset. It is difficult to rigorously examine the model's performance with an initial training size of 24 images. Augmentation helps overcoming this problem. However, the robustness of the model can only be assessed with more data of the same format. Most importantly, augmentation can only be applied to the training set. Therefore, the size of the validation set cannot be modified, and this negatively affects model performance test. More original samples (non-augmented) are needed to further examine the approach taken in this work.

Different types of point cloud data augmentation such as PointMixUp [9] can be explored. Augmentations other than jittering are also known to increase robustness and boost the performance of the model. Indeed, several studies in the past proved that improving 3D object detection performance is possible by applying translation, random flipping, scaling, rotating and shifting to the KITTI dataset [16]–[18]. Since PointNet is designed to be invariant to translation and rotation, other augmentation methods can be tested on the dataset in this work.

3D OC iterates through every slices of the 3D image. It was observed that no nuclei have been missed out. This was checked by overlapping the thresholded image and the second channel of the image in 3D space. However, for microscopy images with lower resolution, and especially with denser packing of nuclei, note that some nuclei can be missed out after this thresholding step. Although dense packing of nuclei weren't substantially present in the dataset that we were given, considering this can be useful when analysing more data. More advanced segmentation tools such as StarDist3D can be tested.

From the point clouds, points were randomly sampled, as explained in the training process. The reason why points were sampled was because all point clouds in the training and test set had to contain the same number of points due to a fixed input size of the model. The size of the minimal point cloud will be the size of every point clouds in both datasets. Loss of data occuring due to sampling increases bias, and this negatively affects model performance. Increasing the size of the point cloud with the least number of points will help reduce this bias. This means that ideally, the number of nuclei detected in every confocal microscopy images should be constant. In addition, a filter that eliminates isolated nuclei can also be built in order to reduce noise in the point cloud, although PointNet is known to be robust to outliers [4].

Although from experiments, a learning rate of around $0.008 - 0.009$ turns out to yield best performance, it is known that a cyclical learning rate [19] method can be used to enhance model performance, without necessarily determining the optimal learning rate. This method consists of cyclically varying the learning rate while the model trains (Ap.3).

As seen in Section 2.3, augmentation was implemented by jittering points in the point cloud by adding random numbers over a given range. An optimal value for this range could be determined, although other ranges such as $[-10, 10]$, $[-2, 2]$ or $[-3, 3]$ have been tested and did not dramatically change the validation loss and accuracy.

5 Conclusion

In this work, a point cloud analysis approach on heart mutation detection is proposed. Using the PointNet architecture, the classifier achieved an accuracy of 100% over the validation set. The resulting model can be used to automatically detect the mutation type that was present in the given dataset. This approach can be further explored by training this model with more data of the same format. Applying cyclical learning rate and other regularisation methods can also improve the model performance if a bigger validation set can be constructed. This binary classification problem can be extended to a multi-class classification problem to detect multiple types of mutations that affect geometrical features of nuclei centroids if there can be a substantial amount of data that represent each type of mutations. Furthermore, other models that extend on PointNet [15], [20], [21] can be tested on the dataset with more samples collected, so that detailed comparison to current literature can be possible.

6 Acknowledgment

Thanks to Prof. Julien Vermot and Prof. Anil Bharath at Imperial College London for technical and experimental advices. Also thanks to my colleague Rares Dorcioman for data analysis and visualisation-related advices.

7 References

References

[1] C. Genge and E. L. et al., "The zebrafish heart as a model of mammalian cardiac function," *Reviews of Physiology, Biochemistry and Pharmacology*, 2016.

[2] Z. W. et al., "3d shapenets: A deep representation for volumetric shapes," *IEEE*, 2015.

[3] J. B. et al., "Spectral networks and locally connected networks on graphs," *arXiv preprint arXiv:1312.6203*, 2013.

[4] C. Qi and H. S. et al., "Pointnet: Deep learning on point sets for 3d classification and segmentation," *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[5] Y.Tai and Z. et al., "Local feature extraction network for point cloud analysis," *MDPI Symmetry Vol.13 Issue 2*, 2021.

[6] B. R. et al., "Disruption of acvrl1 increases endothelial cell number in zebrafish cranial vessels," *Development and Disease*, 2002.

[7] J. Nicenboim, G. Malkinson1, and T. L. et al., "Lymphatic vessels arise from specialized angioblasts within a venous niche," *Nature*, 2015.

[8] R. Chow and P. L. *et al.*, "Following endocardial tissue movements via cell photoconversion in the zebrafish embryo," *Journal of Visualized Experiments*, 2018.

[9] Y. Chen and V. H. et al., "Point mix up: Augmentation for point clouds," *ECCV*, 2020.

[10] MATLAB, *Point cloud classification using pointnet deep learning*, 2014.

[11] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, "Spatial transformer networks," *NeurIPS*, 2015.

[12] S. Bolte and D. W. F. Cordelieres, *Imagej plugin 3d objects counter*, 2017.

[13] S. Bolte and F. Cordelieres, "A guided tour into subcellular colocalization analysis in light microscopy," *The Royal Microscopical Society*, 2006.

[14] J. Serra, *Mathematical Morphology and Its Applications to Image Processing.* 1982, ISBN: 978-94-011-1040-2.

[15] Qi, C.R., and S. et al., "Pointnet++: Deep learning with sets and point clouds," 2016.

[16] S. Shi and C. G. et al., "Point-voxel feature set abstraction for 3d object detection," 2020, pp. 10 529–10 538.

[17] Z. Wang and K. Jia, "Frustum convnet: Sliding frustums to aggregate local point-wise features for amodal," *IROS*, 2019.

[18] Q. Chen, L. Sun, and Z. W. et al., "Object as hotspots: An anchor-free 3d object detection approach via firing of hotspots," *arXiv preprint arXiv:1912.12791*, 2019.

[19] L. N. Smith, "Cyclical learning rates for training neural networks," *IEEE*, 2017.

[20] R. Li, X. Li, and et al., "Pointcnn: Convolution on x-transformed points," *NeurIPS*, 2018.

[21] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph cnn for learning on point clouds," *TOG*, 2019.

[22] napari contributors, *Napari: A multi-dimensional image viewer for python*, 2019. DOI: `10.5281/zenodo.3555620`.

[23] J. C. et al., “3d finite element electrical model of larval zebrafish ecg signals,” *PLOS ONE*, 2016.

8 Appendix

Ap.1 Additional data visualisation using Napari [22].

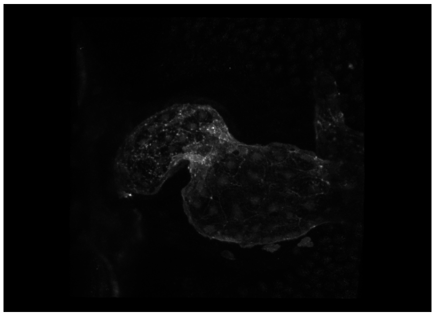


Figure 7: 3D visualisation of Channel 1 of control sample series 2

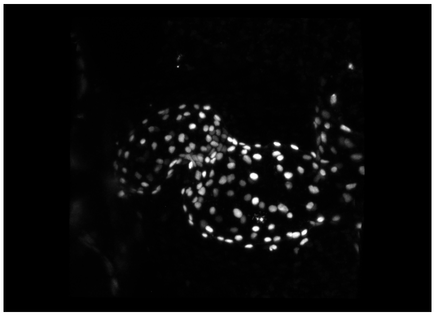


Figure 8: 3D visualisation of Channel 2 of control sample series 2

Ap.2 Augmentation number formula derivation (only applies to the training set):

To balance each class in the training set, there must be an equal number of samples S from each class. Therefore:

$$S_{control} = S_{mutant}$$

$$N(A_c + 1) = (12 - N)(A_m + 1)$$

Ap.3 Cyclical learning rate

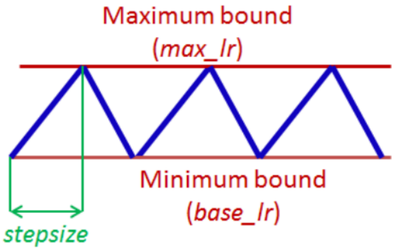


Figure 9: Triangular policy of cyclical learning rate method [19]

Ap.4 Zebrafish Heart Anatomy

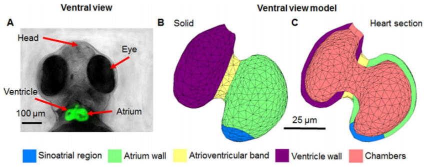
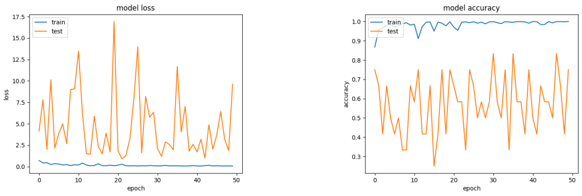


Figure 10: Morphology of the zebrafish heart [23]

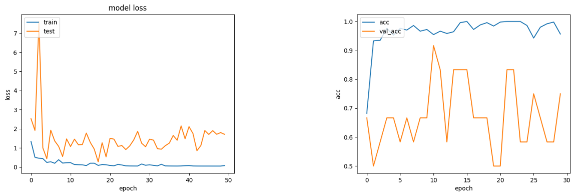
Ap.5 Loss graphs with different parameter values



(a) Validation and Training loss

(b) Validation and Training accuracy

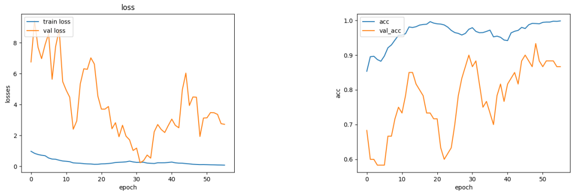
Figure 11: Training results with learning rate 0.05 with no regularisation used



(a) Validation and Training loss

(b) Validation and Training accuracy

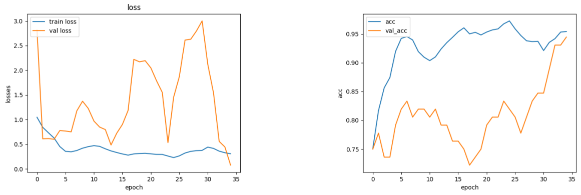
Figure 12: Training results with learning rate 0.01 with no regularisation used



(a) Validation and Training loss

(b) Validation and Training accuracy

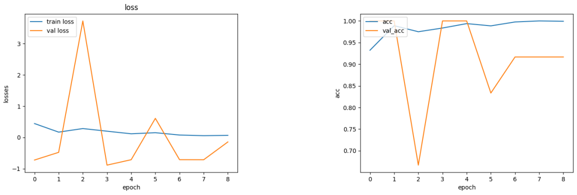
Figure 13: Training results with learning rate 0.008 with point translations as augmentation



(a) Validation and Training loss

(b) Validation and Training accuracy

Figure 14: Training results with learning rate 0.008 with Dropout Layers, point jittering as augmentation



(a) Validation and Training loss

(b) Validation and Training accuracy

Figure 15: Training results with learning rate 0.005 with Dropout Layers and early stopping, point jittering as augmentation, augmentation number of 500