Automatic Detection of Defective Zebrafish Embryos via Shape Analysis

Haifeng Zhao¹ Jun Zhou^{2,3,4} Antonio Robles-Kelly^{2,3,4} Jianfeng Lu¹
Jing-Yu Yang¹

¹School of Computer Science and Technology,
Nanjing University of Science and Technology, Nanjing 210094, China

²Canberra Research Laboratory, NICTA*

Locked Bag 8001, Canberra ACT 2601, Australia

³College of Engineering and Computer Science,

The Australian National University, Canberra ACT 0200, Australia

⁴School of Engineering and Information Technology, UNSW@ADFA
Canberra, ACT 2600, Australia

Abstract

In this paper, we present a graph-based approach to automatically detect defective zebrafish embryos. Here, the zebrafish is segmented from the background using a texture descriptor and morphological operations. In this way, we can represent the embryo shape as a graph, for which we propose a vectorisation method to recover clique histogram vectors for classification. The clique histogram represents the distribution of one vertex with respect to its adjacent vertices. This treatment permits the use of a codebook approach to represent the graph in terms of a set of code-words that can be used for purposes of support vector machine classification. The experimental results show that the method is not only effective but also robust to occlusions and shape variations.

1 Introduction

Zebrafish is an interesting target of research in biology and genetics due to several reasons [9, 11]. Firstly, zebrafish grow quickly, with all major organs developed within 36 hours of fertilization. This is equivalent to 28 days of development of human embryo. Further, its fertilization process can be accurately controlled and can provide researchers with hundreds of embryos in short periods of time. These properties make zebrafish a good research supplement for higher-order vertebrates, such as mice. Secondly, the ge-

netic maps and development behavior of the zebrafish are well understood [19, 13], which makes it possible for researchers to capture and analyse the effects of gene mutation and disease in embryo development processes. Thirdly, the embryos of the zebrafish are transparent and easily observable and, hence, advanced physiology studies can be effected using zebrafish. As a result, in recent years, a lot of research based upon zebrafish has been reported in areas such as genetic defects analysis [1], disease modeling [5], and vision research [21].

In order to monitor the development process of zebrafish, high-throughput screen imaging devices have been used to capture microscopic imagery. The interpretation of these images can be used for multiple purposes, for example, for the analysis and statistics on the zebrafish phenotype. Traditionally, this process is done manually by visual inspection, which is a tedious and time consuming process. Moreover, a large amount of images can be generated when the development of zebrafish embryos is being scientifically documented. For example, a GE IN Cell Analyzer can generate 250 thousand images in a single experiment. This comprises a data set which can exceed a terabyte of data [8]. As a result, developing automatic or semi-automatic zebrafish image interpretation systems is an important need of the scientific community.

Surprisingly, the research on automatic zebrafish image interpretation is still limited. Tianming Liu et al. [14] have performed automatic detection of neuron loss and defective somites, as well as quantitative measurement of gene expression levels in zebrafish embryos. In [14], the authors used circular Hough transforms and intensity projections to detect circles from binary edge imagery. The defective somite detection was implemented through texture

*NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program.

Figure 1. Sample microscopic images of normal (left-hand panel) and defective (other panels) zebrafish embryos.

and edge direction analysis. The energy quantization of the Gabor convolution for the edge map was used to separate healthy from defective embryos. The gene expression level measurement was done by image segmentation using seed detection and region growing. The above method was integrated into ZFIQ, which is a zebrafish software package based on OpenCV library [15, 3]. An alternative method was presented in [22], where Vogt et al. implemented an interactive image interpretation tool to generate rule-based hierarchical image segmentation. These pre-processed images are then used for blood vessel quantification. Researchers have also used commercial software packages such as MetaMorph [20] and Cellomics ArrayScan II [22] to perform aided image interpretation. Despite effective, these tools do not permit automatic zebrafish image interpretation in volumes in par with the rate at which data is generated in large scale genetic experiments. This is mainly due to the complex shape detection and recognition required for processing zebrafish imagery in those intricate backgrounds commonly found in phenotype analysis.

In this paper, we aim at detecting defective zebrafish embryos with altered expressions of Alzheimer’s disease (AD)-linked genes. It has been discovered that injections of control, Psen1, Aph-1, or Pen2 antisense morpholino oligomers (MO) into fertilised zebrafish eggs can generate different effects on embryo somite development [14]. The purpose of the research is to automatically detect zebrafish embryos with defective somites caused by Psen1, Aph-1 or Pen-2 MO injection. These defects often manifest in shape alterations as those shown in Figure 1. More specifically, it can be observed, from Figure 1, that defective somites do not have clear V-shape, and the tails have different shape representation from that of the normal embryo. To this end, we propose a method to perform binary classification on embryo shapes. The method presented here

commences by building a graph from the contour of the zebrafish. It then computes and clusters distances over the graph vertices. Thus, each vertex in the graph can be represented by an attribute vector. We construct a histogram from these vertex attributes to summarise their distribution. In this way, we can quantise the graph representation of the target into a codebook, and thus, make the representation invariant to rotation and scale. This codebook is used to train a binary classifier so as to separate normal and defective image samples. In addition to being effective, the method provides a link between structural and statistical representation of graphs.

The remainder of the paper is organized as follows. In Section 2, we introduce the proposed method and elaborate upon the algorithm in detail. Section 3 gives a step-wise description on the implementation. Section 4 presents experimental results and conclusions are drawn upon in Section 5.

2 Method Overview

2.1 Graph Representation

Mathematically, object shape can be viewed as a set of points on a manifold. The shape of the object, as a finite subset of this set, can be obtained by sampling under some structural criterion [2]. In our case this criterion is given by the contour of the zebrafish embryos. Based on this rationale, we represent the zebrafish embryo shape as a graph, where the points on the manifold are the vertices and the relation between these is encoded in the graph edge-weights. Let $P = \{p_1, p_2, \dots, p_n\}, p_i \in \mathbb{R}^2$ be the set of n points under study. A graph can then be defined as $G = (V, E, W)$, where $V = \{v_1, v_2, \dots, v_n\}$ is the vertex set, $E = \{e_k = (v_i, v_j), k = 1, \dots, m\}$ is the edge set, and W is the edge-weight matrix. For complete graphs, i.e. an undirected graph without loops in which every pair of distinct vertices is connected by an edge, the entry of the weight matrix W corresponding to the edges between vertices v_i and v_j in G is given by

$$W(v_i, v_j) = \begin{cases} \mathcal{W}(v_i, v_j) & \text{if } v_i \text{ and } v_j \text{ are adjacent} \\ 0 & \text{if } i = j \end{cases} \quad (2.1)$$

where $\mathcal{W} : E \rightarrow [0, 1]$.

In this setting, a binary supervised graph classification problem can be formulated over the graph field $\mathcal{G} = \{G | G = (V, E, W)\}$ making use of the labels $\mathcal{L} = \{-1, 1\}$ and a training set $T = \{(G_i, L_i) \in \mathcal{G} \times \mathcal{L}, i = 1, 2, \dots, N\}$, where N is the number of training samples. The purpose is, as usual, to train a classifier so as to predict the labels of unknown graphs. This is a typical supervised learning problem except that the samples here are graphs

instead of feature vectors. Hence, any supervised learning algorithm can be used to solve this problem as long as we can convert inputs from graphical-forms into vectors.

Thus, in this paper, we focus on addressing the problem of graph vectorisation. We depart from the matrix W , which is a symmetric, positive-definite matrix which encodes all the structural information of the graph. Our goal is to convert W into a vector while preserving the structural information it contains. Note that the rows and columns of the weight matrix W are interrelated to one other. Therefore, simply reshaping the matrix into a vector would excise such relations. Moreover, different weight matrices can be generated from the same graph with different vertex orders. This is as the weight matrix W is not permutation invariant. This further complicates the task of vectorising a graph.

In the following subsections, we present a method to vectorise a graph. The idea presented here is a simple one which aims at extracting statistical information from the structure of the graph. This is reminiscent of the work in [4]. The proposed method entails two steps. The first of these involves computing a representation of the edge-weight distribution for every graph-clique so as to characterise the pairwise relationships between vertices. Secondly, we generate a permutation invariant vector from the vertex-weight distribution computed in the first step of the method.

2.2 The Clique Histogram

As mentioned above, the first step of the graph vectorisation method proposed here is to characterise the distribution of pairwise relations between vertices making use of a clique histogram. Given a graph $G = (V, E, W)$ with n vertices as defined in the above section, a clique histogram H_{v_i} of a vertex v_i is a histogram over the distribution of weights between v_i and its adjacent vertices.

A straightforward way of constructing the clique histogram is to generate a frequency histogram with s bins, where each of these is a partition of the entries in W . This generates clique histograms comprised by the same number of bins. Its important to note that clique histogram also captures the cardinality of a vertex v_i which is given by

$$n = \sum_{k=1}^s H_{v_i}(k) \quad (2.2)$$

The clique histogram is also an approximation of the edge-weight probability distribution. Let the peak of the histogram be

$$H^* = \max_{i,k} H_{v_i}(k) \quad (2.3)$$

The probability distribution of the edge-weights for the clique is given by normalising the following clique histogram

$$H_{v_i}(k) = \frac{H_{v_i}(k)}{H^*} \quad (2.4)$$

In this way, we can use the clique histogram to represent, in a vector form, the structural information about v_i . Making use of the clique histogram, we can represent the graph G making use of the matrix

$$\mathbf{H} = [H_1, H_2, \dots, H_n]^T \quad (2.5)$$

where n is the number of vertices in the graph G . Please note that, due to the normalisation, the clique histogram becomes scale invariant.

2.3 The Graph Vector

Although the clique histogram for a vertex is invariant to scale and permutation, the matrix $\mathbf{H}$ is not permutation invariant. To solve this drawback, instead of using the graph Laplacian and its eigen decomposition, we adopt an approach akin to that presented in [23]. Here, we treat the graph as a collection of vertices, and extract the statistical information on the distribution of clique histograms so that the permutation of vertex sequences does not affect the graph representation.

To proceed, we view a graph as a collection of clique histograms. The graph codebook is, hence, generated by clustering all the clique histograms in the training set $T = \{G_i, i = 1, 2, \dots, N\}$ over N graphs. Each clique histogram in the matrix $\mathbf{H}$ is assigned to a cluster based upon the distance between the histogram and the cluster centers. Let the number of clusters be K , this procedure generates a histogram of codewords for each graph, which is a K -dimensional graph vector.

The above process is similar to the “bag-of-words” model which has been widely used in image classification and retrieval [12]. The difference lies in that instead of acquiring pixel or region-wise image feature statistics, we recover statistics on structural information of graphs. This allows us to summarise a graph set into a codebook. This graph-to-codebook conversion is important. Firstly, it makes possible to use statistical classifiers to perform graph classification tasks. It also combines the advantages of structural and statistical methods by making use of information from both areas. In this manner, it provides a link between graph statistics and their structure. Secondly, the vector representation of the graph is permutation invariant and independent to the number of vertices and edges in the graph while, in its un-normalised form, still reflecting the graph cardinality. Thirdly, the method presented here is very general in nature. The codeword representation can be used for both classification and retrieval.

3 System Implementation

In this section, we present the implementation of the method in Section 2 as used for classifying the zebrafish

Figure 2. Sample results for each step of the graph generation process. (a) Original image. (b) Edge response from the Canny edge detector [6]. (c) Texture classification results. (d) Segmentation results after the application of the morphological operations. (e) Subsampling of the contour points used for the graph generation.

embryos. In order to construct graphs from images, we need to extract zebrafish contours. The system is composed of the following steps

1. Extract the texture descriptors from images.
2. Perform foreground/background segmentation using texture features.
3. Use morphological operations to remove the background noise and recover the zebrafish contours.
4. Calculate the clique histograms using the method in section 2.2.
5. Generate the codewords via graph vectorisation using the method in section 2.3.
6. Train an SVM classifier making use of the recovered codeword in order to predict the new zebrafish embryos images.

In the remainder of the section, we provide further details on the steps above.

3.1 Image Segmentation

Given an input image, the purpose of this step is to segment the zebrafish from the background. The segmentation will generate a fish contour. Many segmentation algorithms have been based on pixel intensity, edge maps, or region growing [17, 10]. As can be observed from Figure 2(a), many pixels within the zebrafish embryo have similar intensity as that exhibited by the background. Further, noise and occlusions may be present in the scene. These suggest that pure intensity or gradient based methods cannot be used

Figure 3. Top: Gaussian and Laplacian filter. Bottom-left: 6-edge filter at three different scales. Bottom-right: 6-bar filter at three different scales. The sizes of the edge and bar filters are, from top to bottom, 1×3 , 2×6 and 4×12 .

in isolation to solve the segmentation problem. For example, in Figure 2(b), many false edges have been generated by Canny edge detector [6].

On the other hand, it can be observed that the region within the embryo contains richer texture than the background. This suggests we can use texture to segment the objects from the background. This is, again, a binary classification process. Thus, we make use of a texture descriptor for each pixel and train an SVM classifier. To extract the texture descriptor, we use the MR-filters presented in [18]. The MR-filters are a bank of filters that contains Gaussian responses, Laplacian masks, the gray-scale values of the image, edge-filters and bar-filters at three different scales. The MR-filter bank is shown in Figure 3. The filtering step gen-

Figure 4. Sample images from the zebrafish embryo dataset.

erates an 8-dimensional texture descriptor for each pixel.

We treat the texture descriptor as a feature vector which we can employ to train a binary classifier so as to separate the object from the background. To do so, we randomly select foreground and background pixels in manually segmented training images and use the sampled texture descriptors as the input to an SVM classifier. Then, we apply this classifier to all the images in the training and testing set. One example of foreground/background classification is presented in Figure 2(c).

The above texture classification step can generate false positives. This is due to the background regions that have similar texture to those in the body of the zebrafish embryo. Within the body or along the boundary of the embryo, there are also false negative responses. We remove these artifacts by performing interleaved opening and closing morphological operations in order to obtain a binary image with the object as foreground. An example of the contour obtained after the morphological operations have been applied is shown in Figure 2(d). In practice, we only make use of the external contour of the zebrafish, such as that shown in Figure 2(e).

Using the contour, the object can be represented making use of a weighted graph by treating points on the boundary of the zebrafish as vertices. Here we assume that each vertex is connected to all the other vertices and that the edge-weights can be computed as the Euclidean distance between vertices. These weights are normalized between 0 and 1. This makes the graph invariant to the size of the embryo on the image. With the graph at hand, we use the method in Section 2 to extract the clique histogram for each vertex and vectorise the graph. The final step is the image classification via an SVM classifier trained from the graphs in the training set.

4 Experiments

Here, we present experimental results on a dataset of zebrafish embryo images [8]. The dataset has been acquired using a high-throughput screen imaging system [15]. It contains 98 images among which there are 81 normal samples and 17 defective samples. We separate the whole data into a training set with 10 images and a testing set of 88 images. In Figure 4, we show some example embryos images in the dataset.

To verify the effectiveness of our graph vectorisation algorithm, we first test the method on the dataset by manually segmenting the zebrafish embryo contours. These manually segmented contours will be used as a baseline, later in the paper, for purposes of comparison with the texture based background-foreground segmentation. With these manually segmented contours at hand, we abstract each embryo silhouette into a graph, whose vertices correspond to the contour points. In order to reduce the computational complexity, we sample the contour points in uniform intervals across its arc-length. In our experiments, we have set the interval step to 3. Our complete graph edge-weights are the Euclidean distances between the contour points on the image plane. The graph vectorisation process follows the method introduced in Section 2. For the recovery of the cluster centers used in the histogram and codebook computation we use the K-means clustering algorithm. The SVM classifiers used for graph classification has been implemented using the LIBSVM [7]. In all our experiments, we perform ten-fold cross validation (CV) and recover the mean and variance of the correct classification rate for the optimum SVM parameters.

Here, we have performed two experiments so as to analyse how the parameters in the graph vectorisation method affect the final performance. In the first set of these, we evaluate the effect of the number of clusters used to generate the clique histograms, which determine the dimen-

Figure 5. Performance as a function of the number of clique histogram bins

Figure 6. Performance as a function of codebook size

sionality of codeword. The mean and standard error of the correct classification rate are shown in Figure 5. The best performance is achieved when the number of clusters is set to 3. Note that, as the number of clique histogram bins increases, the performance decreases. The reason for this may be that, as the number of clusters increases, the graph vectorisation becomes more sensitive to noise corruption. The experiments show that the above method can achieve up to 96.5% correct classification rates. This compares favorably against the 92.0% classification rate reported in [8], which uses skeleton extraction and curve fitting.

In the second set of experiments, we study the effect of the codebook dimensionality on the recognition results. The results are shown in Figure 6. From the figure, we can see that the larger the size of the codebook, the better the performance. This observation is reasonable because too few codewords can not effectively characterise the distribution of graph attributes. The observed effects of the code-

word and codebook sizes are interesting. They are an intuitive confirmation of two important properties of the proposed method. Firstly, the clique histogram characterises the structural information within the graph. Secondly, the graph vector characterizes global feature distributions at a graph level.

Finally, we perform experiments on the unsegmented dataset. We do this following the steps in Section 3 and the same settings employed for the manually segmented imagery. Our aim is to study the effect of noise corruption on the automatic segmentation scheme presented previously as compared to the accuracy when manual segmentation is used. The results in Figure 7 show that our automatic segmentation method delivers a level of accuracy which is comparable to that yielded using manually segmented imagery. There is a slight decrease in performance which becomes more obvious when the dimensionality of codebook is large. However, when the dimensionality of the codebook

Figure 7. Performance comparison between automatic and manual segmentation procedures

is small, the performance does not decrease greatly. This suggests that the proposed graph vectorisation algorithm is robust to noise corruption and contour perturbation due to imperfect zebrafish background segmentation.

5 Conclusion

In this paper, we have proposed a graph-based method for automatic detection of defective zebrafish embryos. We depart from a structural setting, where we abstract the zebrafish contour into a graph. To extract the contour, we use texture descriptor to segment the image. The graph then can be generated from the contour points. Due to the normalisation, the graph representation is invariant to the object size and pixel resolution. We have shown how these graphs may be vectorised for purposes of classification. Thus, our method provides a link between structural and statistical pattern recognition techniques. Our experiments have shown that the method is both effective and robust to noise corruption and object contour variations. In the future, we will try more methods, for example, the level set method, to improve the segmentation step [16].

6 Acknowledgement

This work was done while Haifeng Zhao was a visiting scholar at the Canberra Laboratory of NICTA. It is supported by the National Natural Science Foundation of China (NSFC) under No.60775015. The authors would like to thank Dr. Tianming Liu, Dr. Weiming Xia and Dr. Scott Holley for providing the zebrafish dataset used in this paper.

References

- [1] P. Bang, P. Yelick, J. Malicki, and W. Sewell. High-throughput behavioral screening method for detecting auditory response defects in zebrafish. *Journal of Neuroscience Methods*, 118(2):177–187, 2002.
- [2] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(24):509–522, 2002.
- [3] G. Bradski and V. Pisarevsky. Intel’s computer vision library: Applications in calibration, stereo, segmentation, tracking, gesture, face and object recognition. In *Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 796–797, 2000.
- [4] H. Bunke and K. Riesen. Graph classification based on dissimilarity space embedding. In *Proceedings of the 2008 Joint IAPR International Workshop on Structural, Syntactic, and Statistical Pattern Recognition*, pages 996–1007, Orlando, FL, 2008.
- [5] W. Campbell, H. Yang, H. Zetterberg, S. Baulac, J. Sears, T. Liu, S. Wong, T. Zhong, and W. Xia. Zebrafish lacking alzheimer presenilin enhancer 2 (pen-2) demonstrate excessive p53-dependent apoptosis and neuronal loss. *Journal of Neurochemistry*, 96(5):1423–1440, 2006.
- [6] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986.
- [7] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- [8] L. Cheng. Morphometry analysis of zebrafish. Master’s thesis, Nanjing University of Science and Technology, Nanjing, China, 2009.
- [9] H. Detrich, M. Westerfield, and L. Zon. Overview of the zebrafish system. *Methods in Cell Biology*, 59:3–10, 1999.
- [10] C. X. Dzung L. Pham and J. L. Prince. Current methods in medical image segmentation. *Annual Review of Biomedical Engineering*, 2:315–337, 2000.

- [11] J. Eisen. Zebrafish make a big splash. *Cell*, 87(6):969–977, 1996.
- [12] L. Fei-Fei and P. Perona. A bayesian hierarchical model for learning natural scene categories. In *Proc. of IEEE Computer Vision and Pattern Recognition*, pages 524–531, 2005.
- [13] C. Kimmel, W. Ballard, S. Kimmel, B. Ullmann, and T. Schilling. Stages of embryonic development of the zebrafish. *Developmental Dynamics*, 203(3):253–310, 1995.
- [14] T. Liu, J. Lu, Y. Wang, W. Campbell, L. Huang, J. Zhu, W. Xia, and S. Wong. Computerized image analysis for quantitative neuronal phenotyping in zebrafish. *Journal of Neuroscience Methods*, 153(2):190–202, 2006.
- [15] T. Liu, J. Nie, G. Li, L. Guo, and S. Wong. ZFIQ: Zebrafish image quantitator. In *IEEE/NIH Life Science Systems and Applications Workshop*, pages 59–62, Bethesda, MD, 2007.
- [16] J. Sethian. *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*. Cambridge University Press, 1999.
- [17] L. G. Shapiro and G. C. Stockman. *Computer Vision*. Prentice-Hall, New Jersey, 2001.
- [18] J. Sivic, B. Russell, A. Efros, A. Zisserman, and W. Freeman. Discovering objects and their location in images. In *Proceedings of the International Conference on Computer Vision*, pages 370–377, 2005.
- [19] J. Sprague, E. Doerry, S. Douglas, and M. Westerfield. The zebrafish information network (ZFIN): a resource for genetic, genomic and developmental research. *Nucleic Acid Research*, 29(1):87–90, 2001.
- [20] T. Tran, B. Sneed, J. Haider, D. Blavo, A. White, T. Aiyejorun, T. Baranowski, A. Rubinstein, T. Doan, R. Dingle-dine, and E. Sandberg. Automated, quantitative screening assay for antiangiogenic compounds using transgenic zebrafish. *Cancer Research*, 67(23):11386–11392, 2007.
- [21] M. Tsujikawa and J. Malicki. Analysis of photoreceptor development and function in zebrafish retina. *Journal of Developmental Biology*, 48(8-9):925–934, 2004.
- [22] A. Vogt, A. Cholewinski, X. Shen, S. Nelson, J. Lazo, M. Tsang, and N. Hukriede. Automated image-based phenotypic analysis in zebrafish embryos. *Developmental dynamics*, 238(3):656–663, 2009.
- [23] R. Wilson, E. Hancock, and B. Luo. Pattern vectors from algebraic graph theory. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(7):1112–1124, 2005.