Modelling Epigenetic Mechanisms to Capture Dynamical Topological Morphology : Applications In Edge Detection

Alexander P. Turner *, Martin A. Trefzer * and Andy M. Tyrrell*

*Department of Electronics, University of York, Heslington, York, YO10 5DD, UK

***Abstract*—EpiNet is a novel computational model which is able to perform dynamic topological modification autonomously throughout execution. This approach is inspired by the functionality of eukaryotic gene regulation, specifically that of chromatin modification which is able to modify its structure dynamically, altering the structure of gene regulatory networks. In this work we utilise the dynamic properties of epiNet when applied to two different methods of edge detection, and analyse these networks and their dynamical properties via structural and dynamical systems analysis.**

I. Introduction

Over millions of years, biological systems have evolved solutions to complex problems such as locomotion, homoeostasis and information processing. Biological systems tend to have significant advantages when it comes to adaptability, robustness and evolvability [1], [2], [3], [4] when compared to their computational counterparts. Hence, it has been the goal of computer scientists and engineers to capture the underlying biological mechanisms *in silico* in an attempt to capture their specific behaviours, but also their advantageous underlying traits.

Since the beginnings of computer science, there have been many successful attempts at capturing biological models within a computation framework. This field, known as bio-inspired computation has given rise over recent years to algorithms which have been shown to perform objectively better than their biological counterparts. In certain instances this is even the case when performing complex tasks such as object recognition [5] against human beings. Indeed, many of the computational algorithms which are now state of the art are built upon these bio-inspired principles [6], [7], [8], [9].

Currently, modelling biological systems *in silico* in perfect detail is not possible due to both computational deficits and gaps in the understanding of biological processes at various levels of abstraction. Because of this, the design of bio-inspired algorithms fall on a spectrum. At each end of the spectrum there are generally two approaches present. The first approach is to capture as much detail from the biological system as possible to best promote the possibility of emergent properties. The second is to omit a vast amount of the biological realism to capure only what is necessary for the desired outcome.

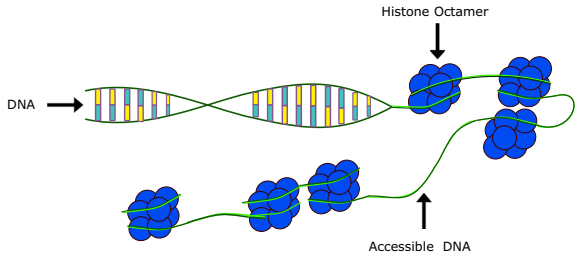

Fig. 1. DNA being wound around histone octamers over 1.67 turns into a chromatin fiber

Each of these perspectives has their positives and often the more simplistic models remain functional and are capable of capturing real-world complex dynamics [10], [11].The work in this paper falls towards the more detailed side of the spectrum.

In this work we present a novel model of artificial gene regulatory network, built upon previous models [12], [13], [14] which takes into account the dynamic nature of protein networks, their interactions and their relationships with dynamic epigenetic structures. This model is applied to the task of edge detection where we feel the dynamical behaviour of the networks can be well utilised.

II. Background

A. Epigenetics

Epigenetics refers to mechanisms which result in changes in gene expression without altering the underlying DNA [15], [16]. From both a logical and physical perspective, epigenetics can be considered to be acting on a different level of abstraction compared to genes. A gene can be considered to be a section of DNA which is typically used as an encoding for the primary structure of a protein [17], [16], [18]. Proteins are molecular machines which are responsible for a significant amount of the biochemical interactions within living organisms.

In Figure 1 a general eukaryotic arrangement of DNA is shown. Within the cell nucleus, DNA is wrapped around a histone octamer over 1.67 turns. This combination of DNA and histones is referred to as chromatin which forms one of the major epigenetic structures.

978-1-4799-7560-0/15 $31.00 © 2015 IEEE
DOI 10.1109/SSCI.2015.176

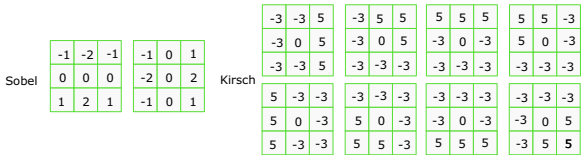

Fig. 2. The filters for both the Sobel and Kirsch operators which are responsible for determining gradient changes across 3*3 grids of pixels

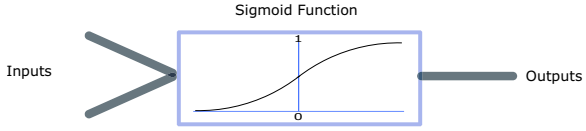

Fig. 3. The structure of the artificial gene which takes multiple inputs, processes these inputs using the sigmoid function and produces an output

Chromatin is principally responsible for controlling physical access to DNA. The dynamically varying structure of chromatin allows DNA to move relatively to it so that cellular machinery (e.g. transcription factors) can access it. Controlling this movement means controlling which genes are actively being transcribed at any given moment. This process underpins the philosophy of the model described in this paper, being that there exists a 'bank' of genes (the genome) which is inactive by default. Then, chromatin reorganises itself relative to the DNA in order to change which genes are accessible to the cellular machinery at any given time.

B. Edge Detection

Edge detection is a method of finding points within an image which contains sharp changes in brightness. Less formally, it is the process of mathematically creating borders within an image which relate to object differentiation [19]. There are many methods in which to do this, however some of the most popular are the methods which utilise discrete differentiation operators. Of these, two of the most common are the Sobel and Kirsch operators [20] (Figure 2).

This work uses both the Sobel and Kirsch operators to perform edge detection. The models are identical apart from the differences in the filters used which deduce the gradient difference.

For each pixel in the image, the 8 nearest neighbour pixels (3*3 grid) are then used along with the filters to approximate the derivatives or changes over those pixels. The Sobel operator uses 2 filters to achieve this, whereas the Kirsch operator uses 8. The additional Kirsch filters are more tuned to picking up gradient differences over multiple directions, where the Sobel filter generally operates in just the vertical and horizontal. Each pixel is analysed from left to right in the image.

III. The Model

The model developed in this paper (which from this point forward will be refereed to as epiNet), builds upon a range of previous work [21], [12], [13], [22], [14], [23]. Indeed, the most fundamental part of epiNet, the artificial gene is derived from ([24], and remains unchanged). The gene in this sense is a computational structure which takes inputs and processes them using a sigmoid function (Figure 3) producing a single output. Many of these genes are structured together to form an artificial gene regulatory network where certain genes map a task on to the network (input genes) certain genes process the information and certain genes provide an output for the network [24].

There have been several approaches to building epigenetic structures into, or on top of, existing artificial gene regulatory networks [21], [12], [13], [23]. In these models, genes are generally always active unless made inactive by an epigenetic analogue. Additionally, the epigenetic analogues are static and fixed in place and augment a small subsection of the network. With epiNet, we place an emphasis on separating genetic structure from their product; protein networks. Moreover, epiNet captures the dynamic properties of chromatin remodelling to allow for the dynamic control of transcription over *all* genes within the network and subsections thereof.

The main structure in this model is a genetic structure, consisting of a number of genes (30-100) which exist on a 1-dimensional linear scale (Figure 1) between [0,1]. These genes are static and are *not* directly executed. Execution of the genes occurs when genes are copied from the genetic structure to the protein network. The protein network functions in a similar way to the networks in [24], where it is the structure which directly interacts with an external environment (task) and is executable. However, *which* genes are copied from the genetic structure to the protein network is controlled by the epigenetic molecule.

The epigenetic molecule(s) within epiNet straddles the genome, existing in the same 1-dimensional space. Each epigenetic molecule has a position in this 1-dimensional space and a size (Figure 1). The size is fixed, but the position is a variable. At each time step, whichever genes exist within the space occupied (the molecules position $\pm$ its size) by the the epigenetic molecule are then copied from the genome to the protein network. At each time step, the epigenetic molecule takes selected inputs from the protein network, and processes them using a sigmoid function (identical to the genes (see Figure 3)) and this output then becomes the epigenetic molecules new position. Hence, the position of the epigenetic molecule on the genome is the product of the structure and state of the protein network. Only one epigenetic analogue is implemented upon initialisation, however multiple epigenetic molecules can be incorporated to the network throughout the optimisation process . After each execution the protein networks expression values are mapped back to the genome. This serves to give the protein network and genome a relative memory of its previous state. Following execution, all the proteins within the protein network are removed and will become repopulated according to the position of the epigenetic molecule on the next step. An external task can interface with epiNet by modifying protein expression within the protein

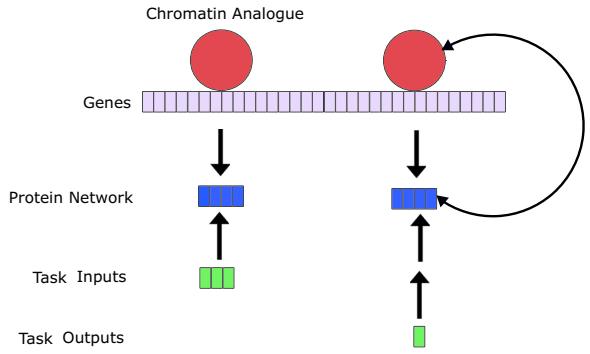

Fig. 4. A single iteration of epiNet. The genes which are directly below the genome are copied into the protein network. The protein network can then take inputs from the tasks and is executed to produce an output(s). The epigenetic analogue is connected to the protein network and post execution, it takes inputs from the protein network and processes them using a sigmoid function. The result then becomes the next position of the epigenetic analogue.

Variable	Type	Range
Gene Expression	Real	0;1
Positional Identifier	Real	0;1
Positional Proximity	Real	0;0.25
Weight	Real	-1;1
Sigmoid Offset	Real	-1;1
Sigmoid Slope	Int	0;20
Input Identifier	Real	0;1
Output Identifier	Real	0;1

TABLE I
RANGES OF THE VARIABLES WITHIN EACH GENE OR PROTIEN WITHIN THE NETWORK

network.

IV. Future Hardware Implementation

Within this body of work, there has been an interest in implementing the epiNet on a Field-programmable gate array (FPGA). FPGAs are a type of integrated circuit which can be configured using a hardware description language such as VHDL [25]. VHDL is used to describe the digital functions of an FPGA where one of the main strengths is parallelism. It is the strength of parallelism which is attractive to this body of work as it could significantly sped up edge detection performance. In addition, having epiNet on an embedded system would allow for greater versatility amongst applications. With the current epiNet existing only within an object orientated

Variable	Type	Range
Input Identifier	Real	0;1
Input Proximity	Real	0;0.25
Weight	Real	-1;1
Sigmoid Offset	Real	-1;1
Sigmoid Slope	Int	0;20
Transcriptional Proximity	Real	0;0.25

TABLE II
RANGES OF THE VARIABLES WITHIN EACH EPIGENETIC MOLECULE

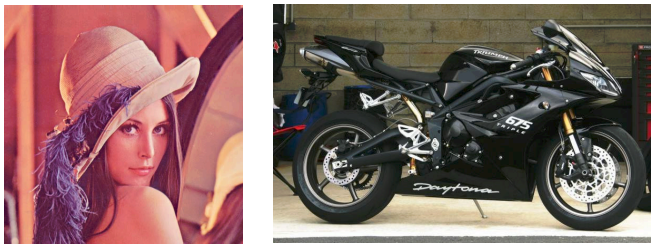

Fig. 5. The two images which will be used for training the network to perform edge detection. The picture of Lena contains 512*512 pixels and will be edge detected using the Kirsch operators and a threshold of 250. The picture of the bike contains 720 * 753 pixels and will be edge detected using the Sobel operators and a threshold of 100. The network will be trained on both of these images.

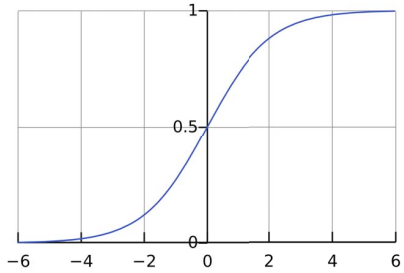

Fig. 6. The inputs (x-axis) against the outputs (y-axis) for the sigmoid function

language, there are many issues that need to be solved before spinet can function optimally on an FPGA.

A focus within this work is to remove high precision variables and functions from the networks which are notoriously computationally inefficient to work with in FPGAs. This includes the variables found within each gene and epigenetic analogue (tables I and II) and the sigmoid function (Equation 1). In addition, the sigmoid function is a differentiable real function which is executed for every single protein within the protein network and every epigenetic molecule at every time step, hence it is particularly important that this function can execute as fast as possible. In this instance, we are going to assume that a suitable reduction in precision will take floating point numbers and replace them with doubles fixed to 2 decimal places.

For the sigmoid function (Equation 1) we can calculate that the bounds of the equation (where - sx - b returns either 0 or 1) are -5.29 , 5.29 (Figure 6). To 2 decimal places, all values for the sigmoid function can be calculated, approximated and stored within a table containing 1024 elements. As well as being more FPGA friendly, this is significantly faster.

$$f(n) = (1 + e^{-sx-b})^{-1} \quad (1)$$

Preliminary testing it was found that transforming the sigmoid function into a lookup table improved execution times of the network by around 40%. Hence, with the immediate benefits of reduced time to completion, the sigmoid function

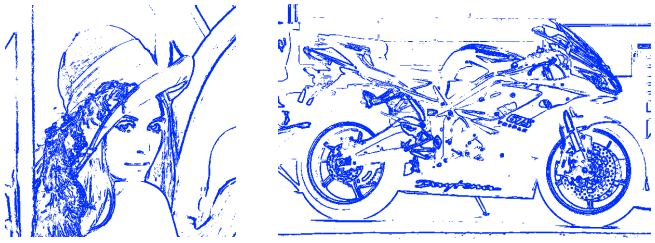

Fig. 7. The results from performing edge detection on Lena (Kirsch , threshold 250) and the bike (Sobel, threshold 100)

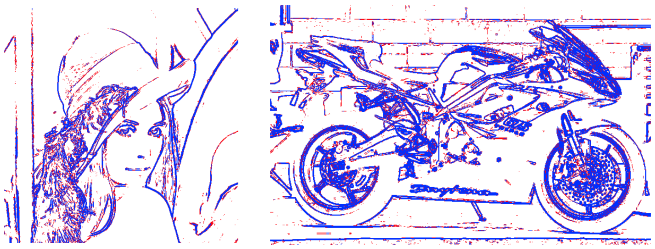

Fig. 8. The differences between the two images if the methods of edge detection are swapped (Lena using Sobel and the bike using Kirsch). The differences between the algorithms produce 15070 pixel differences for Lena and 24070 for the bike. The differences are denoted by red pixels and where the algorithms agree are denoted via blue pixels.

will be replaced with the lookup table throughout experimentation and analysis, with a view towards hardware friendly implementation.

V. Methodology

There are three points we wish to address with this investigation:

- What are the dynamic properties of epiNet when applied to edge detection and how does this affect network behaviour?
- How well can epiNet perform edge detection based on two mathematically different models within the same network?
- To understand the functionality of epiNet using a sigmoid lookup table rather than a high precision sigmoid function.

To address the first question we train the epiNet using a training set consisting of two images, Lena and the motorbike (Figure 5). The image of Lena uses the edge detection method utilising Kirsch filters with a threshold of 250. The image of the motorbike uses the edge detection method utilising Sobel filters with a threshold of 100 (Figure 7). The differences between these two methods can be seen in Figure 8. The purpose behind using these two images is because firstly, Lena is very commonly used in edge detection to assess functionality. This is because there are many varying features within the images, ranging from sharp edges to weak edges around the hair. The motorbike was chosen to provide a contrast to Lena, where generally hard edges are more common and pronounced.

Prior to execution of epiNet, two key parameters that have to be set. The number of genes in the network and the number of epigenetic molecules. In this instance the genome will consist of 30 genes, and only a single epigenetic molecule to begin with. Throughout evolution, the amount of genes within the network and the amount of epigenetic molecules can vary during to the optimisation process.

The networks were evolved using a multi-objective genetic algorithm NSGA-II. The population size was 200, with a crossover rate of 0.5 and a mutation rate of 0.05. NSGA-II will run for 50 generations. At each crossover operation 0.4% of each network is crossed over, not the whole network. This is done to minimise disruption. The mutation operator can add and remove genes as well as modifying variables within individual genes and epigenetic molecules. 40 individual runs

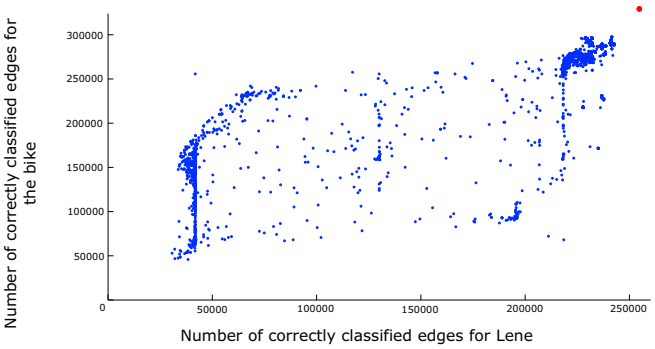

Fig. 9. The results of every population member at the 50th generation for each of the 40 runs with the optimum behaviour marked by the red dot. One of the most interesting things shown here is that the networks don't have a propensity towards being adept at one tasks over the other. This is especially the case for the fitter. It can be seen that there are generally two areas of dense solutions which correspond to poor performance and good performance. It is likely that these are local minima within the search space

of the experiment will be completed overall. The fitness of each network is calculated as the number of correctly classified pixels (according to each mathematical method) for each image. This will create a 2-dimensional fitness landscape.

VI. Results

The results show that epiNet was able to find solutions performing the two different types of edge detection and can be seen in Figure 9. The fitter solutions are able to achieve around 93% accuracy compared to Kirsch edge detection on Lena, and around 91% when compared to Sobel edge detection on the bike (Figure 10). From the results shown the epiNet is capable of capturing the inherent properties required to perform edge detection.

The results show that there is a certain amount of pooling within certain regions of in Figure 9. The fist one of these, where solutions are weak denotes the behaviour of a single non dynamic output from the network, for example always returning 0. This sets all of the pixels to a certain value, which returns a score around the range of the pooling. The other main area of pooling is the the fitter side of the map, where ever solution in this space is able to perform edge detection well and where NSGA-II is guiding the solutions.

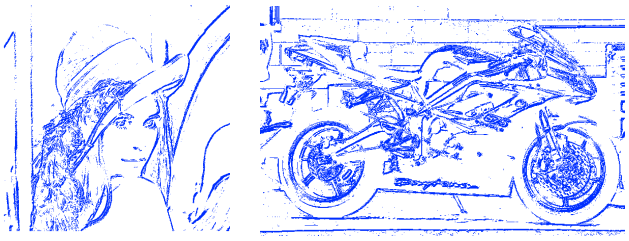

Fig. 10. The results from a single networks interpretation which correctly classified 242071 pixels of the Lena image (93%) and 294178 pixels of the bike image (91%)

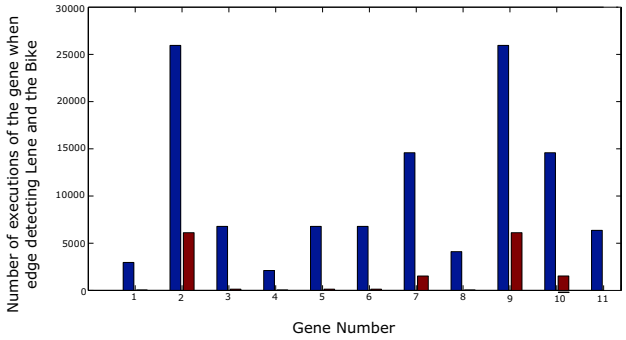

Fig. 11. A histogram plot of the non-permanently active genes within epiNet (The network analysed in this histogram is the same as the one used in Figure 10). The network contained 30 genes, 19 of which were either permanently active or jointly active. This permanently active or jointly active genes have been removed to better show the contrast between the non-permanently and non jointly active genes. This histogram depicts which of the non permanent and non jointly active genes are being used when applied to the different pictures (Lena = red, bike = blue)

A. *Dynamical Systems Analysis*

One of the goals of this work is to understand how the behaviour of the dynamics within epiNet when applied to tasks which require different properties to solve, and the dynamics required to achieve that. The first step taken to understand this is to discover the dynamical differences within epiNet during edge detection for the different pictures. The histogram in Figure 11 shows that depending on which picture is being sampled by the network, has a significant effect on what genes are being used. That is, epiNet is changing its topology and structure depending upon what task it currently faces.

Generally amongst all the evolved solutions there was an approximate trend in the structure of the networks. This trend consisted of two epigenetic analogues within the network, where one was dynamic and the other was generally static, but with a slight deviation at a few points during the edge detection process. To understand the interactions within the network and its behaviour we analyse a network (the same from Figure 10) and remove each of the epigenetic analogues one at a time and assess the difference. The results show that one of the molecules is vital to function and removing it perturbs the network unto an unrecoverable static state. However, upon removal of the other molecule, the network

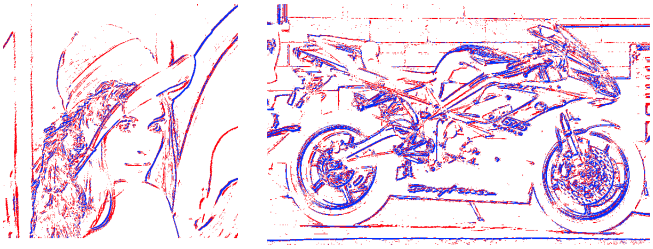

Fig. 12. A large sum of the networks solved the edge detection task by using 2 epigenetic analogues. In the network used in Figure 10. One of these was vital for performance, and the other although not vital (as in, the network could produce meaningful results without it) was responsible for correctly classifying around 10% of the pixels. These images show the difference between the non-vital epigenetic molecule being included vs knocked out. Blue pixels show the performance when the network had the epigenetic knock-out, red the full functioning network

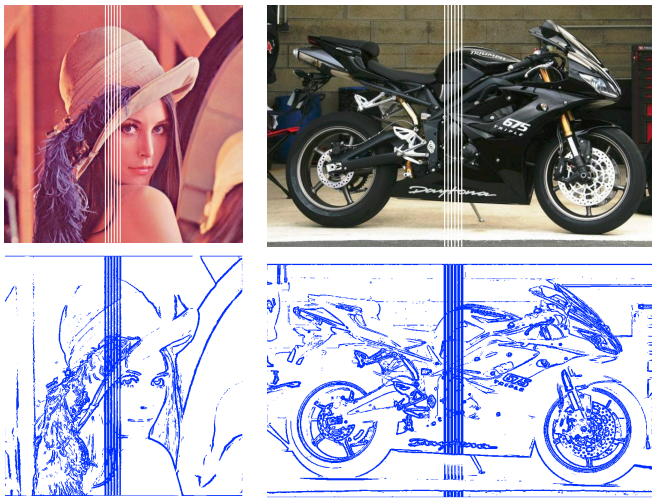

Fig. 13. An illustration of how a pertubation can affect the dynamical state of the epiNet. This is done by using noise in the form of 6 vertical lines of 2 pixels in width spaced over 4 pixel gaps. The results of the perturbations can be more closely seen in Figure 14

still functions, yet mis-classifies approximately 10% of the pixels over each images. Interestingly, the epigenetic analogue which is responsible for the majority of the classification is the dynamic one suggesting that a wide range of genes may be needed to optimally perform edge detection.

From these results we can make some general statements about the function of the network. The network has learnt to partition itself where specific genes can become active according to specific behaviours which are required to solve certain aspects the task. More broadly, it means that certain genes and certain behaviours can be autonomously selected depending on the current task in hand.

To generate a level of understanding about the dynamical properties of the network when it comes to non-structure based perturbations, perturbations were introduced to the training images and the networks recovery is analysed (Figures 13 and 14). What can be seen is that epiNet does not immediately recover for these perturbations and takes around 1 - 3 iterations

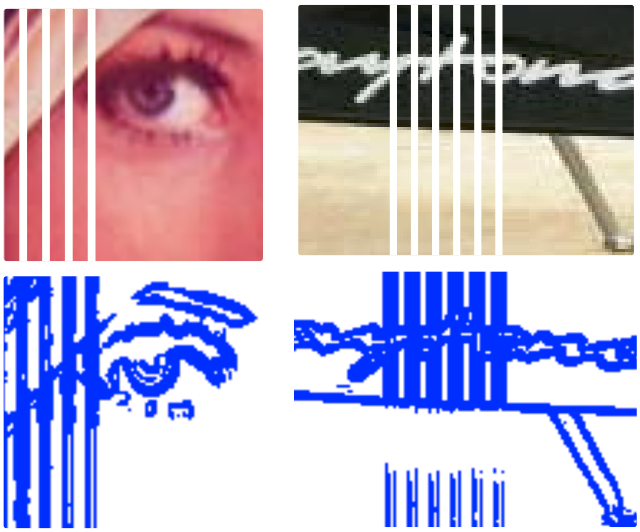

Fig. 14. A close up of the images from Figure 13. It can be seen that the noise has a knock on effect for between 1 and 3 steps (scanning left to right). This provides evidence to suggest that they dynamical behaviour of epiNet is altered by the perturbations and takes time to recover. During this recovery it can be seen to incorrectly classify certain pixels.

of the network to do so. However, it is to be noted that throughout all the image-based perturbation analysis epiNet was always able to autonomously recover to a functional state.

B. Generality

The training of epiNet consisted of using two different images, each using a different method of edge detection. It is therefore difficult to objectively specify how well the network is performing on new, unseen images as there are technically two correct ways of determining fitness (Sobel and Kirsch). In this instance we attempt to show that epiNets are capable of performing edge detection of unseen images without negative impacts on the dynamics of the network. The images which show this can be seen in Figures 15 and 16.

VII. Conclusion

In this work we have shown that the epiNet is capable of dynamically performing edge detection based around two different edge detection methods. EpiNet utilised its internal dynamics to autonomously select specific set of genes according to its current state and inputs. This dynamical behaviour partitioned the genome into different types of behaviours required to perform edge detection.

In addition, the sigmoid function, which has been at the core of many bio-inspired networks preceding this was replaces by a lower precision lookup table which served to both increase speed without noticeably limiting performance.

In future work the focus should be upon increasing the efficiency of all the networks so that a larger number of generations can be used when optimising the networks. In addition, due to the subjective nature of edge detection (in that the objective is very specific and individual for a given task),

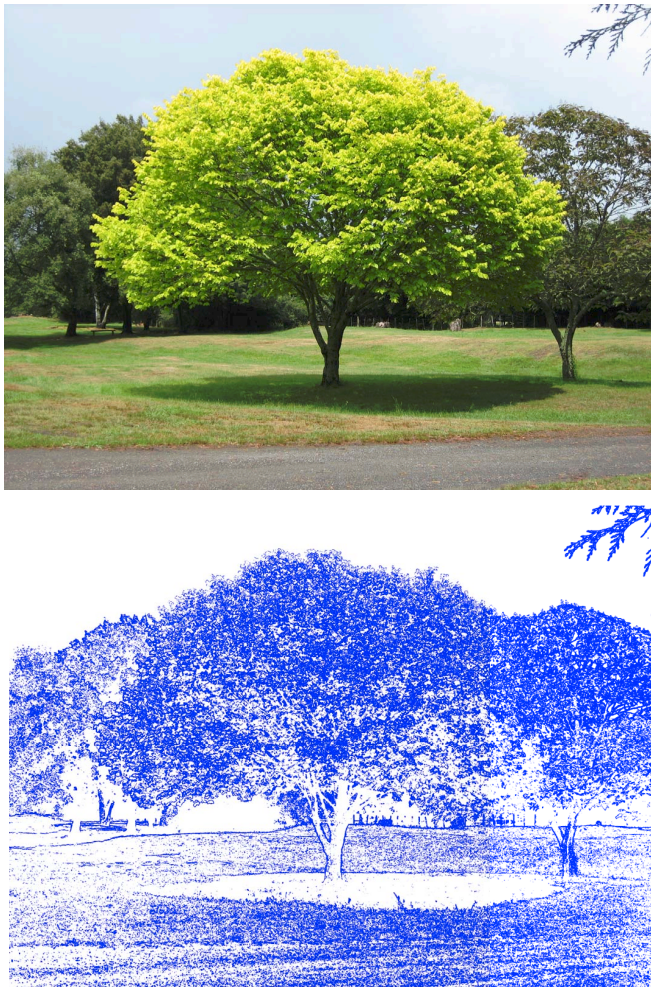

Fig. 15. The networks output when applied to an unseen image of a tree.

a better method may be to utilise edge detected images from visual effects studios to ascertain if epiNets can learn user defined edge detection, and how this can help reduce work load. Additionally, if the networks are successfully ported to FPGAs it would be good to understand their performance when applied to video steams rather than individual images.

Acknowledgments

Acknowledgements The authors would like to thank the EPSRC for their support of this work through the Platform Grant EP/K040820/1. Data created during this research is available at the following DOI: INSERT

References

[1] J. Masel and M. V. Trotter, "Robustness and evolvability," *Trends in Genetics*, vol. 26, no. 9, pp. 406–414, Sep. 2010.
[2] S. Huang, "Reprogramming cell fates: reconciling rarity with robustness," *Bioessays*, vol. 31, no. 5, pp. 546–560, 2009.
[3] H. Kitano, "Biological robustness," *Nature Reviews Genetics*, vol. 5, no. 11, pp. 826–837, 2004.

Fig. 16. The networks output when applied to an unseen image, that of York minster.

[4] J. Kim, M. Choi, J. Kim, H. Jin, V. Kim, and K. Cho, "The co-regulation mechanism of transcription factors in the human gene regulatory network," *Nucleic Acids Research*, 2012.

[5] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," *CoRR*, vol. abs/1502.01852, 2015. [Online]. Available: http://arxiv.org/abs/1502.01852

[6] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Learning and transferring mid-level image representations using convolutional neural networks," in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*. IEEE, 2014, pp. 1717–1724.

[7] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *Signal Processing Magazine, IEEE*, vol. 29, no. 6, pp. 82–97, 2012.

[8] X.-S. Yang, Z. Cui, R. Xiao, A. H. Gandomi, and M. Karamanoglu, *Swarm intelligence and bio-inspired computation: theory and applications*. Newnes, 2013.

[9] A. Zhou, B.-Y. Qu, H. Li, S.-Z. Zhao, P. N. Suganthan, and Q. Zhang, "Multiobjective evolutionary algorithms: A survey of the state of the art," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 32–49, 2011.

[10] L. Bull, "Consideration of mobile dna: new forms of artificial genetic regulatory networks," *Natural Computing*, vol. 12, no. 4, pp. 443–452, 2013.

[11] R.-S. Wang, A. Saadatpour, and R. Albert, "Boolean modeling in systems biology: an overview of methodology and applications," *Physical biology*, vol. 9, no. 5, p. 055001, 2012.

[12] A. P. Turner, M. A. Lones, L. A. Fuente, S. Stepney, L. S. D. Caves, and A. M. Tyrrell, "The artificial epigenetic network," in *10th Internation Conference on Evolvable Systems, IEEE press, Singapore*. IEEE Press, April 2013, pp. 66–72.

[13] A. P. Turner, M. A. Lones, L. A. Fuente, A. M. Tyrrell, S. Stepney, and L. S. D. Caves, "Controlling complex tasks using artificial epigenetic regulatory networks," *BioSystems*, vol. 112, no. 2, pp. 56–62, 2013.

[14] A. P. Turner, "The artificial epigenetic network," Ph.D. dissertation, University of York, 2013.

[15] A. Bird, "Perceptions of epigenetics," *Nature*, vol. 447, no. 7143, pp. 396–398, 2007.

[16] B. Turner, *Chromatin and gene regulation: mechanisms in epigenetics*. Blackwell Science, 2001.

[17] L. A. Moran, H. R. Horton, G. Scrimgeour, and M. Perry, *Principles of biochemistry*. Pearson Boston, 2012.

[18] R. Dawkins, *The extended phenotype: the long reach of the gene*, ser. Popular Science. Oxford University Press, 1999. [Online]. Available: http://books.google.co.uk/books?id=j7aPVHze9i8C

[19] K. K. Jena, "Application of com-sobel operator for edge detection of images," 2015.

[20] K. Plataniotis and A. Venetsanopoulos, *Color Image Processing and Applications*, ser. Addison-Wesley series in electrical and computer engineering: Digital signal processing. Springer, 2000. [Online]. Available: https://books.google.co.uk/books?id=vUM9-HDespYC

[21] M. A. Lones, L. A. Fuente, A. P. Turner, L. S. D. Caves, S. Stepney, S. L. Smith, and A. M. Tyrrell, "Artificial biochemical networks: Evolving dynamical systems to control dynamical systems," *Evolutionary Computation, IEEE Transactions on*, 2013, in press.

[22] T. Reil, "Dynamics of gene expression in an artificial genome - implications for biological and artificial ontogeny," in *Proceedings of the 5th European Conference on Artificial Life (ECAL'99), number 1674 in Lecture Notes in Artificial Intelligence*, 1999, pp. 457–466.

[23] A. P. Turner, M. Trefzer, and A. M. Tyrrell, "Evolving efficient solutions to complex problems using the artifcial epigenetic network," in *10th International Conference on Information Processing in Cells and Tissues, San Diego*. IEEE Press, September 2015.

[24] M. A. Lones, A. M. Tyrrell, S. Stepney, and L. S. Caves, "Controlling complex dynamics with artificial biochemical networks," in *Genetic Programming*. Springer, 2010, pp. 159–170.

[25] S. Harris and D. Harris, *Digital Design and Computer Architecture: ARM Edition*. Elsevier Science, 2015.