An Artificially Evolved Vision System for Segmenting Skin Lesion Images

Mark E. Roberts and Ela Claridge

School of Computer Science, University of Birmingham, B15 2TT, UK
{M.E.Roberts,E.Claridge}@cs.bham.ac.uk

Abstract. We present a novel technique where a medical image segmentation system is evolved using genetic programming. The evolved system was trained on just 8 images outlined by a clinical expert and generalised well, achieving high performance rates on over 90 unseen test images (average sensitivity 97% , average specificity 81%). This method learns by example and produces fully automatic algorithms needing no human interaction or parameter tuning, which although complex, runs in approximately 4 seconds.

1 Introduction

In many areas of medicine, images are used as a diagnostic aid. These could be "natural" such as direct views of the patient's skin, or produced by an instrument such as x-ray, ultrasound, CT or MRI. However, the images themselves only partially contribute and, crucially, input comes from interpretation of the image by an expert using the power of the human visual system. This human system works in real time, does not need carefully tuned parameters, and perhaps most importantly, is the able to learn by example to recognise general image features. It is these qualities which provide the inspiration for this work. We present here a method which learns by example, and produces fully automatic, parameter-free image processing algorithms to identify given features. This paper shows the application of the method to segmenting pigmented skin lesions as part of our group's work towards the early detection of malignant melanoma.

Malignant melanoma is a deadly cancer which most commonly arises in the skin. Its diagnosis at the primary care level is difficult because in its early stages it may look similar to totally innocent pigmented skin lesions – "moles". Moderate diagnostic rates achieved by dermatologists [1] confirm this difficulty and for this reason there is a growing body of work on applying image analysis methods to aid the diagnosis of melanoma [2]. Lesion classification is commonly carried out using automatically extracted features which characterise the lesion appearance. They are most frequently based on clinical features, for example the ABDCE list [3], which includes Asymmetry, Border, Diameter, Colour and Elevation. The first three of these features crucially depend on accurately segmenting the lesion body from the surrounding skin. This is the case also for other lesion features and the reason why the task of segmentation is such an important step prior

to further analysis. There has been much work on lesion segmentation and the interested reader is referred to one of the reviews, e.g. [4]. Segmentation is a difficult task, mainly due to the variability in lesion appearance. Some lesions are well delineated and make good contrast with the skin (see Fig. 3), whereas others are indistinct, variegated and difficult to see with an untrained eye (see Fig. 3). The published methods use a variety of approaches, including threshold-based methods [5], colour clustering and distance-based functions in a colour space [6,5], edge modeling [7,8] and various combinations of these methods [9].

What these methods have in common is the fact that they all have been developed by image analysis experts, in most cases informed by clinical practitioners. This paper describes a very different approach, in which the expertise in image analysis is not necessary to be able to create a well performing image processing system, in this case for lesion segmentation. This is achieved through the use of a Genetic Programming (GP) system, in which a lesion segmentation system is automatically evolved, purely on the basis of example segmentations provided by an expert clinician. The paper first outlines the concept of Genetic Programming. This is followed by the description of a study in lesion segmentation using GP, its results and discussion.

2 Materials and Methods

2.1 Pigmented Lesion Images

A set of 100 pigmented lesion images is used in this study. The images were acquired using a SIAscope [10], a device designed specifically for skin imaging, that takes a number of images of the same area of the skin at different wavelengths. In its normal mode of operation it uses an optical model of the skin to compute parametric maps showing the distribution and levels of individual histological components of the skin such as melanin, haemoglobin and collagen [10]. In this study images acquired in the blue band are used because of strong absorption by both melanin and blood makes the lesions stand out against the skin background. Image resolution is 40 microns per pixel and a circular area with radius of 280 pixels is used. A "ground truth" data set in the form of binary images, is created from outlines drawn by a clinical expert at Addenbrooke's Hospital, Cambridge, UK.

2.2 Genetic Programming

Genetic Programming (GP) is a method of creating computer programs to solve a specific problem by using methods inspired by natural evolution. GP emerged in the early 1990s as an offshoot of work on genetic algorithms (GAs). In a GA, populations of "chromosomes" normally represented as strings of binary numbers encode possible solutions to a problem.

GP is a powerful extension to this paradigm which actually evolves populations of computer programs rather than simple strings. These programs are represented as tree structures and are initially created randomly from sets of

Fig. 1. A demonstration of GP evaluation and crossover. The trees represent hierarchical programs. *I* represents the input image, *M* is a 50/50 image merge, *** multiplies images, *T* thresholds the image to the given value and *E* performs an edge detection. Crossover can be seen to generate children with some of each parent's properties.

functions and terminals. Each of the programs is run on the problem, and a fitness value is assigned based on how well it performs. These fitness values then serve as a means to implement “survival of the fittest” procedures which select, and then adapt, the fitter individuals using mutations (random changes to a single individual) and crossover (creating new offspring influenced by two parents). As the programs are represented as trees, mutation is easily implemented by replacing a randomly chosen branch of a tree with a randomly generated new sub-tree. Crossover simply selects random points from two trees and swaps over the subtrees beneath them to generate two new children. Over many generations, better and better solutions to the problem emerge. Effectively GP creates programs to solve a problem, without being told how to solve it, or knowing anything about its underlying nature. The programs produced are often quite novel, as the process is free from any human preconceptions about the problem or what constitutes a good solution.

The huge computational expense of running thousands of complex programs for many generations has, until recently, meant that GP was normally limited to dealing with numerical, logical, or simple symbolic manipulations. Now, however, we are at a stage where imaging problems can be tackled by using image processing operations in the function set, and input images in the terminal set. Figure 1 shows some very simple examples of GP image trees, the output they produce, and how a crossover operation would create two random children.

2.3 Experimental Setup

The data was divided into a training set containing 8 examples, and a test set containing 92 examples. The 8 images were chosen as they represented most of the variation found in the dataset.

The population of programs is then randomly created from the function and terminal sets. The function set contains imaging operations such as thresholds, morphological operations, logical operations, region intensity functions (mean, min, max), edge filtering, merging, quantisation etc. The terminal set consists of the input image, and numerical and coordinate values. All of the functions and terminals can be used any number of times within a tree. More detailed information about this type of system can be found in [11]

Every generation, each program in the population is run on each of the images in the training set. The fitness of each program is measured, and then used to influence the selection procedures deciding which ones are adapted and put into the next generation. In these experiments a population of 5000 programs is used, and the system was run for 75 generations.

2.4 Fitness Function

The fitness function is key to the success of the evolution. It should provide proportional feedback to the GP system in a way that correctly captures what it means for a solution to be better or worse than another. The fitness function used in this case is a modification of a function proposed by Poli [12] for similar problems, summed over all of the N images in the training set. It is based on the desire to maintain a good balance between sensitivity, the ability to correctly segment sections of the image containing the feature, and specificity, the ability to correctly segment the sections which do not contain the feature. These are formally defined using measures of true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN) which are used in this fitness function.

$$f = \sum_{i=1}^N \frac{FP}{FP + TN} + \frac{FN}{TP + FN} \exp \left(10 \left(\frac{FN}{TP + FN} - \alpha \right) \right) \quad (1)$$

where α (set to 0.4 in this case) allows the relative importance of sensitivity and specificity to be varied. It is important to consider these properties, as a simple pixel difference calculation could easily miss features; consider the case of a very small feature on a uniform background – a very high fitness could be achieved by just matching the background, but totally missing the feature. Also, as in Poli's work, a wrapper function thresholds the image before the fitness calculation. This is more effective as the system is actually trying to find the solution which, when thresholded, best matches the target output.

3 Results

The binary image that the program outputs is compared to the correct outline and each pixel is classified as a TP, FP, TN, or FN. From these classifications, measures of sensitivity and specificity are produced. These are shown in 10% divisions for each of the 92 test images on the 2-axis histogram shown in Fig. 2. To visually analyse performance, a four colour image is produced showing these classifications as light grey, black, white, and dark grey respectively. Some examples are shown in Fig. 3.

Fig. 2. Sensitivity-specificity histogram showing the percentage of results in each performance category for the 92 unseen test images

4 Discussion and Conclusions

4.1 Analysis of Performance

On the unseen test data the program performs very well on the majority of the examples, as can be seen from the histogram in Fig. 2. Most of the examples are clustered in the very high accuracy regions of the histogram. The examples on which the program performed less well were generally those which were highly irregular (such as Fig. 3(g) and (h)) which were not fully represented in the training set. Future training should use more of these irregular images.

Although the segmentations produced are not perfect, the algorithm would be a good first step to semi-automatic systems such as [7], which performs a more detailed analysis of the lesion borders but needs a human to mark the centre of the lesion. This system could quite easily produce this sort of input. Also, the success of the method shows great potential for other imaging problems.

4.2 Benefits of Method

The method presented here has several very important benefits over more traditional segmentation methods.

- The system can be used by non-experts as it only needs segmented examples.
- The system learns by example. This makes it applicable to many more problems than model based approaches which are very specific.

Fig. 3. Examples of performance on the unseen test set. Light-grey=TP, Black=FP, Dark-grey=FN, White=TN (a) an example of hair removal (b) shows the detection of another lesion which was not originally outlined. (c-f) good segmentations, in spite of irregularities and smooth borders (g-h) examples of bad segmentations.

- The programs produced are free from human preconceptions about the problem, and pick up on aspects of the problem that humans may miss.
- The programs produced are resolution independent, illustrating the fact that they have evolved to recognise general features of the images.
- This method generalises well from a very small training set of examples. In this case 8 images were used, and the resulting program performed well on over 90 unseen images.

4.3 Programs Produced

One obvious question arises from this work; what do the programs do that makes them so good? This is difficult to answer because of the programs' complexity and the often unconventional steps they use. The evolutionary computation literature is littered with examples of evolutionary processes exploiting subtle aspects of the problem, bugs in programs, or even properties of matter (in hardware evolution), to produce solutions which would never have been thought of by humans.

However, the innovative nature of the GP process also has some disadvantages. When humans design programs, they are always thinking about efficiency, ease of implementation, and maintainability. GP totally ignores all of these factors. As a result, it is difficult to fully understand what the program is doing, and they also contain inefficient and unnecessary steps which slow down the program execution. To overcome this the program needs to be post-processed to remove operations that do not contribute to the final output, but because of reasons mentioned previously, care must be taken not remove something which looks to a human like it doesn't do anything, but is actually exploiting a subtle feature. The optimisation and analysis of the programs is an ongoing task.

Training Time. The best program produced after 75 generations contained 330 nodes and executes in about 4 seconds. Pruning and optimisation of the program tree could easily reduce this to real time. As previously mentioned, the training time of a image based GP run can be enormous. The run that produce the results shown took approximately 20 hours to complete, running on a cluster of about 40 computers. This amount of processor time may at first make this approach seem prohibitive, but all of this training time was spent in search of just one single, and actually relatively simple (compared to what it could have produced), program which takes only a few seconds to run. This time spent training only needs to occur once to produce this program, and thereafter it can be used without the need for large computational resources.

Another issue is the repeatability of these experiments. GP is a highly random process and generally evolutionary computation experiments are run many times to assess the amount of computation needed to achieve a good probability of success. The long training time hampers efforts to generate these sorts of statistics in this case.

4.4 Summary and Future Work

We have presented preliminary results of a system which uses genetic programming to evolve a program to segment pigmented skin lesions. The evolved programs generalise from a small set of examples, and perform very well on a large set of unseen examples.

There is enormous scope for future work in this area. The method could be applied to almost any binary segmentation problem and a few modifications to the paradigm could make the system applicable to non-binary problems. All that is required is for an expert to provide manual segmentations for training. Specific future work includes using outlines drawn by multiple experts in order to reduce the intra and inter-expert ambiguities which confuse the learning process.

References

1. Morton, C., MacKie, R.: Clinical accuracy of the diagnosis of cutaneous malignant melanoma. *British Journal of Dermatology* **138** (1998) 283–287
2. Day, G., Barbour, R.: Automated melanoma diagnosis: where are we at? *Skin Research and Technology* **6** (2000) 1–5
3. Fitzpatrick, T., Rhodes, A., Sober, A., Mihm, M.: Primary malignant melanoma of the skin: the call for action to identify persons at risk, to discover precursor lesions, to detect early melanoma. *Pigment Cell* **9** (1988) 110–117
4. Computerised Medical Imaging & Graphics: Special issue on skin imaging **16** (1992)
5. Xu, L., Jackowski, M., et. al.: Segmentation of skin cancer images. *Image and Vision Computing* **17** (1999) 65–74
6. Schmid, P.: Segmentation of digitized dermatoscopic images by two-dimensional color clustering. *IEEE Transactions on Medical Imaging* **18** (1999) 164–171
7. Claridge, E., Orun, A.: Modelling of edge profiles in pigmented skin lesions. In Houston, A., Zwiggelaar, R., eds.: *Proceedings of MIUA 2002*. (2002) 53–56
8. Gao, J., Zhang, J., Fleming, M., Pollak, I., Cognetta, A.: Segmentation of dermatoscopic images by stabilised inverse diffusion equations. In: *Proceedings of the International Conference on Image Processing, Chicago* (1998) 823–827
9. Ganster, H., Pinz, A., Rohrer, R., Wildling, E., Binder, M., Kittler, H.: Automated melanoma recognition. *IEEE Transactions on Medical Imaging* **20** (2001) 233–239
10. Claridge, E., Cotton, S., Hall, P., Moncrieff, M.: From colour to tissue histology: Physics based interpretation of images of pigmented skin lesions. In Dohi, T., Kikinis, R., eds.: *Proceedings of MICCAI'2002*, Springer (2002) 730–638
11. Roberts, M.: The effectiveness of cost based subtree caching mechanisms in typed genetic programming for image segmentation. In Cagnoni, S. et.al., eds.: *Proceedings of EvoIASP 2003*. Volume 2611 of LNCS., Essex, UK, Springer-Verlag (2003)
12. Poli, R.: Genetic programming for feature detection and image segmentation. In Fogarty, T., ed.: *Proceedings of the AISB'96 Workshop on Evolutionary Computation*. Volume 1143 of *Lecture Notes in Computer Science*., Springer (1996) 110–125