A Comparison of Different Object Detection Strategies

Wyatt Kormick
kormi001@umn.edu

March 26, 2018

1 Introduction

For our final project we are likely going to be evaluating an AI created to play the Nintendo 64 game, Pokemon Snap. This game involves the main character travelling through a level on a set path with a set of predefined events occurring around him. These events are the same every play-through, and can be influenced by the main character. The goal is to take pictures of the Pokemon that appear throughout the level. At the end of the level the pictures are scored based on the picture's object's facing, pose, actions, size, and several other criteria. Though it doesn't directly solve the problem at hand, a useful thing for an artificial intelligence to be able to do in order to play this game would be to recognize dynamic objects, Pokemon in this case, in a dynamic environment. After doing some research on the subject, I have found three methods of object detection. Edge detection using a genetic algorithm[1], boundary detection using reinforcement learning on a classifier[2], and object detection in dynamic scenes using Bayesian modeling[3].

2 Genetic Algorithm

This method uses a genetic algorithm to find natural edges between objects in an image. The genetic algorithm takes in a gray-scale image and produces an image of the same size. The pixels of the output image have two values, 1 or 0 (white or black), corresponding to whether or not the pixel belongs to an edge. The fitness function used calculates probabilities of a pixels in the image belong to an edge based on dissimilarities between neighboring regions pixels and compares the output image to these values. The first generation were given an output image consisting of randomly selected values for pixels. Each generation used 512 images, and finished when there were 15 consecutive generations where the fittest image did not change. Reproduction involved trading regions of pixels between mates, and mutations involved flipping the values of pixels. The higher the fitness of an image, the higher its probability to reproduce.

3 Reinforcement Learning with a Classifier

This method uses a classifier, trained on a data set of 12,000 human-labelled images to detect boundaries between objects in an image. This classifier sampled circular regions of the input image, split them in half, checked the brightness, color, and texture in the regions, and if it found them to be sufficiently different, put a border of black pixels between the two regions in the output

image [2]. After the classifier put where it thought there would be boundaries in the image, it would check it against where the human place boundaries in the image. Depending on how far away the classifier's output image was from the human's the classifier would adjust the weights and biases of the its network through a process called back-propagation. The goal of this process is that after training itself on a large training set, it would be able to find the boundaries in images it had never seen before.

4 Bayesian Modeling

This method of object detection computes probability models of the foreground objects and the background and the two compete over ownership of the pixels. The models are each 5 dimensional (x, y, red, green, blue) and consist of every pixel in the image. Instead of modelling each pixel independently, it models the entire background as one image. This is to show the dependability of pixels on each other, as pixels nearby to a high probability background pixel are going to have a higher probability of being a background pixel, likewise for foreground pixels. For every frame the probability of a pixel belong to the background or the foreground is calculated for every pixel. A sliding window of frames is kept for both the foreground and the background, which corresponds to the learning rate of the system.

The background of an image should have a set structure. It is either stationary or it moves in some sort of a pattern, so pixel regions that follow this structure have a higher probability of being in the background. The foreground calculates its probabilities based on the idea of temporal persistence [3]. Basically what this means is that objects tend to stay in the same general area, and stay the same general color between frames. Changes have to occur over time and aren't instant. At first, the probability of any pixel belonging to an object is uniform across all pixels and frames. Then, pixel regions that vary from other dominant pixel regions have a higher probability of belonging to a foreground object. Once a pixel is found to have a high probability of belong to a foreground object, nearby pixels have a higher probability of also belonging to an object than those further away. At the next frame, there is a high probability of a pixel region to be in the foreground if, in the previous frame, there was a pixel region in the foreground in the same general area.

5 Comparison

None of the three strategies are guaranteed to come up with a correct answer to the object detection problem. This is because the first two are local searches, meaning that they will eventually find a local maximum in fitness, which may or may not be a global maximum, and the third is based on probabilities. Bayesian modeling as an advantage in that it attempts to solve the problem of dynamic object detection in a dynamic environment. The other two strategies, in their experimentation are used to find edges in a static picture. These edges may not even be between different objects in the scene.

In regards to accuracy, the Bayesian Modelling strategy has another advantage. In their experimentation, which involved filming for an hour in different environments and running their strategy on it, Bayesian Modeling had an object detection rate of 99.708% and a misdetection rate of 0.41%. In each scene nearly every human detected object was detected by the strategy and only one or two in each scene that were extraneously detected objects. This gives the the Bayesian Modeling

strategy a precision a steady percentage in the 90's. In comparison the classifier had a precision percentage starting in the 90's but with an exponential decrease as noise levels in the image increased.

References

[1] Suchendra M Bhandarkar, Yiqing Zhang, and Walter D Potter. "An edge detection technique using genetic algorithm-based optimization". In: *Pattern Recognition* 27.9 (1994), pp. 1159–1180.

[2] David R Martin, Charless C Fowlkes, and Jitendra Malik. "Learning to detect natural image boundaries using local brightness, color, and texture cues". In: *IEEE transactions on pattern analysis and machine intelligence* 26.5 (2004), pp. 530–549.

[3] Yaser Sheikh and Mubarak Shah. "Bayesian modeling of dynamic scenes for object detection". In: *IEEE transactions on pattern analysis and machine intelligence* 27.11 (2005), pp. 1778–1792.