Computer Vision for Human Detection

By Ben Francis, ECE '18

Introduction

The field of computer vision has one general goal: to make computers and other machines "see" as we do. If you stop and think about it, there is a lot of processing that goes on in our brains when we simply look around. We are constantly analyzing and gathering information about our surroundings. As we look around we see many different objects and attempt to categorize them and recognize them. For instance, we may see a shelf consisting of bags of chips. We must first identify that these are, indeed, bags of chips and furthermore what types of chips they are. If we are near people, we are constantly looking at peoples faces to see if there is anyone that we recognize. When we walk up to a crosswalk, we look at the oncoming traffic and track the cars to see if we can cross safely. While doing this, we are tracking the object as well as judging the speed of the car and how long it will be until the car is at the crosswalk. This allows us to make a decision as to whether or not to cross the road or not. These are all tasks that we take for granted in our everyday life. Computer vision consists of taking all of these tasks that utilize human vision and translating them into algorithms that computers and machines can utilize to perform similar tasks. When approaching a computer vision problem during a project, there are many different factors to consider depending on what your goal is. In the case of Team Chestnut, we aim to detect humans on a frame by frame basis in real time video. We hope to track the humans from frame to frame in combination with other sensor technology. With this in mind, we need to make sure that the algorithm that we choose to implement can be computed in real time with a reasonable amount of computing power. Additionally, we are concerned with primarily human detection, and not analyzing human motion itself. We aim to answer the question, "Are there humans in this video frame? If so, how many and where are they?" This paper will serve as a basic introduction to computer vision, specifically for the application of human detection.

CV Basics: Template Matching

Now, let's take a look at a basic computer vision algorithm: template matching. With template matching, the basic idea is that you have a source image and a template image. The goal is to find the template image somewhere within the source image. Let's take a look at an example. Say we have the following source image:

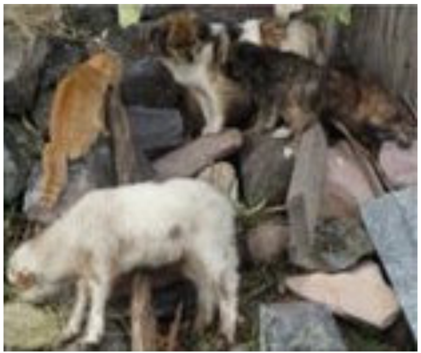

Figure 1. Source image for template matching

And the following template image (magnified), which we can see appears in the upper middle of the source image:

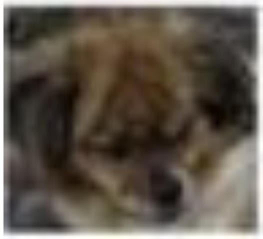

Figure 2. Template image for template matching

Now, we know that our objective is to find the template image within the source image. Simply put, this is done by sliding the template image over the source image, pixel by pixel, and determining which location is the "best match". Now, there are several different ways to quantify "best match". The simplest metric is the square difference. Given that the template image is at a certain location in the source image, the square difference at that point is the sum of all the differences squared between each pixel in the template image the current overlapping pixel in the source image. In a perfect match, this value will be zero [3].

Once the square difference has been calculated for every single candidate location in the source image, the pixel location that produced the minimum value is determined to be the "best match". We can represent the metric values visually with a grayscale heat map, with better matches appearing more towards the white end of the scale and poor matches appearing more towards the black end of the scale. The heat map produced by performing template matching on the above template and source images can be seen below:

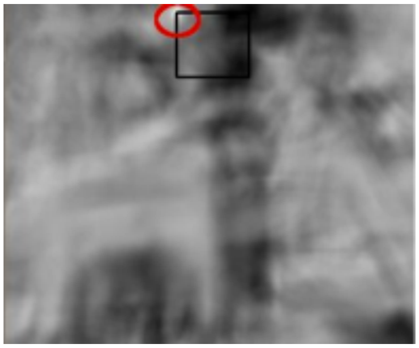

Figure 3. Template Matching Heat Map

The white area indicates the best match, and the black box represents the location of the template image corresponding to the best match within the source image. Looking back at the source image, it can be seen that this is the location that the template image appears.

Silhouette Based Human Detection

Now that we are familiar with a basic computer vision algorithm, let's take a step further and apply what we've learned to problem of human detection. If we knew the exact template image of a human that we were looking for in a source image, this problem would be easy. The tough part is finding an arbitrary human in an image without knowing exactly what to look for, as is the case in template matching. One approach to this problem is detecting humans based on silhouettes [1]. This specific application specializes in detecting pedestrians. It utilizes a database of silhouettes and performs template matching on images and these silhouettes in order to detect pedestrians. An example silhouette template can be seen below [1]:

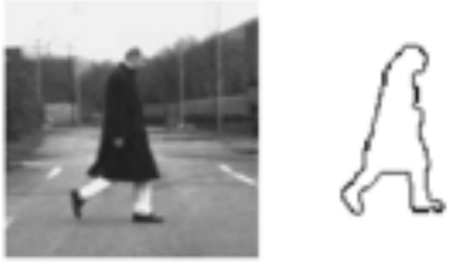

Figure 4. Silhouette Template

This approach to human detection is great for detecting humans in real time. In comparison to strict template matching, which is looking for all pixels in the template to match all pixels in the source, the silhouette based approach is looking for the general shape of an object in the source to match the shape of the template silhouette. The general shapes within the source image can be extracted by utilizing an edge detection algorithm. Without diving too much into the math, edge detection algorithms work by looking at the derivative, or change, across a set of pixels, among other things. Usually, this involves first utilizing a smoothing filter to blur the image. The purpose of this step is to make the edge detection less prone to error due to noise. Then, in the simplest case, the derivative across all dimensions of the image is examined. Finally, a threshold must be utilized to decide whether a given amount of change in pixel intensity constitutes an edge. This threshold is usually stated in terms of standard deviations from some portion of the image. The result of applying an edge detection algorithm could look like the following image [1]:

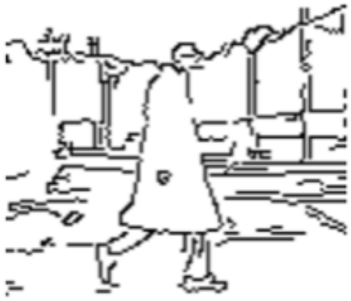

Figure 5. Application of Edge Detection Algorithm

After edge detection, there is one more transform that needs to be done before the actual template matching can be performed. An transform called the Distance Transform, DT, is performed. First, we will call the black pixels in the edge image "points of interest", as they represent the shapes within the image. The DT takes this image and produces a new image where each pixel represents the distance from that pixel to the closest point of interest. Black represents a distance of zero, and the whiter a pixel is the further it is from a point of interest. The distance metric can be as simple as Euclidean distance, rounded to an integer. The DT image produced from the previous edge image can be seen below [2]:

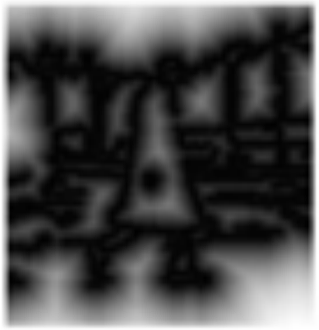

Figure 6. Application of the Distance Transform

Finally, the original silhouette image is overlaid onto the DT image. The sum of all pixels where the silhouette is black on the DT image is the metric of how good the match is. This can be done at different locations over the image, and if there exists a location that is above some user defined threshold, then it can be concluded that a human exists in the image at that location. This method relies on having a large database of many different human silhouettes, and assumes that there exists a silhouette in the database that closely resembles the shape of the human you are trying to detect.

Conclusion

In this paper, we have introduced the field of computer vision with specific focus on the task of human detection. We discussed the basic computer vision algorithm of template matching, and then extended this understanding to the problem of human detection through silhouette based human detection. Computer vision is an extremely large field and this paper has just skimmed the surface of it. This paper has only talked about one solution to one problem in the field. Hopefully, this paper has served as inspiration for further reading in the field of computer vision.

References

1. Dariu Gavrila. 2000. Pedestrian Detection from a Moving Vehicle. In Proceedings of the 6th European Conference on Computer Vision-Part II (ECCV '00), David Vernon (Ed.). Springer-Verlag, London, UK, UK, 37-49.

2. David A. Forsyth and Jean Ponce. 2002. Computer Vision: A Modern Approach. Prentice Hall Professional Technical Reference, 214-234.

3. David A. Forsyth and Jean Ponce. 2002. Computer Vision: A Modern Approach. Prentice Hall Professional Technical Reference, 749-750.