

Marcellus Hunt

Biometric Computing

Assignment 1

1/23/2020

Robust Real-Time Face Detection

I will describe the method of robust real-time face detection by Paul Viola and Michael J. Jones; including how this type of detector is implemented and how it works intuitively, mathematically, and schematically. This face detection process uses features instead of pixels because a system that uses features instead of pixels works much faster. The three kinds of features used are rectangular based: two-rectangle, three-rectangle, and four-rectangle. Depending on the type of rectangle feature, it calculates the difference between the sum of the pixels in the regions. These show a representation model for the image, which is referred to as the integral image. The location is based on the positional points ('x', 'y') and the sum of the pixels in relation to that point (to the left and above it). So the formula would be:
$$integral\ image(x, y) = \sum image(x^2, y^2), x^2\ and\ y^2\ represent\ the\ x\ and\ y\ points\ for\ the\ original\ image\ (not\ the\ points\ squared).$$
Next, I will go over learning classification functions.

With a feature set, machine learning approaches could be utilized to learn a type of classification function. In this system, a version of AdaBoost, which is a learning algorithm that is used to enhance the classification performance of a training procedure, is used to select certain features and teach the classifier. Based on experiments in the paper, the system gave a detection rate of 95% with a false positive rate around 1 in 14084. This is good but stated that the rate should be lower in order to guarantee great security and results. These features could be for example, around the eyes, mouth or even more specific features such as around the nose or forehead. This is a good way for facial detection but in order to improve it, there needs to be more features that are added to the classifier. But this also increases computation time. In order to decrease computation time and to accomplish this, the system uses a cascade of classifiers.

A cascade of classifiers increases detection time and performance while decreasing computation time. The key to this part of the system are smaller classifiers that are also more efficient and will reject negative-sub windows in most positive instances. The simpler classifiers that are used to achieve low false positive rates before the smaller and more complex classifiers. This works like a tree: a result from the first classifier class for the second classifier to be evaluated which had been altered to get high detection rates. So, the results from the first classifier trickles down to the next. The cascade tries to get rid of as many negatives it can at the earliest time. Each stage gets more difficult than the previous stage as it tries to reject those negative-sub windows to get better detection rate. The rate for the false positive of the cascaded classifier is defined by the product of each of the false product rates and up to the number of

classifiers. Formula: $(F = C \prod f_i)$ where F is the false positive rate, C is number of classifiers, f_i is each of the false positive rates. The rate of detection is defined by the formula: $(DR = C \prod d_i)$ where DR is the detection rate of the cascaded classifier, C is the number of classifiers, and d_i is the detection rate of the classifiers that through what is being processed. The important attribute of all the classifiers is the positive rate which depending on the proportion of windows, contains a possible face. This plays a key in detecting the face and the computation speed. Also, the system needs to figure out the number of features that will be needed to be evaluated. This is done by the algorithm: $[NF = f_0 + C \sum_{i=1}^n (f_i \prod_{j>i} p_j)]$ where NF is number of features, C is the number of classifiers, p_i represents the positive rate, and f_i represents the number of features. Also, to note, because faces are rare and unique, this positive rate is equal to the false positive rate. The overall process requires some type of tradeoff though. The more features, the higher the detection rates and lower false positive rates. Also, there is a need for more computation time. The amount of classifier stages, features, and max amount of at each stage are variables that will require some type of tradeoff between each other. So, with that, the speed of the detector is based on the total amount of features for the window/area scanned.

Next let's move on to image processing. This involves using the integral image mentioned earlier. The number of squared pixels is constructed by an integral image of the original image squared. Also, to note from this section, that multiplying the values from the features can achieve the effect of image normalization instead of trying to work with the pixels. Next, the detector is scanned. It is scanned across the image at different areas and scales. In order to scan at different locations, the sub-window needs to be moved a certain number of pixels. This can affect the computation speed and the accuracy of the detector.

This paper showed an approach to construct a facial detection system that attempts to achieve a high detection accuracy with as little computation time as possible. It introduced the usage of integral image which reduces the image processing for detection. It does this by using features instead of pixels which is faster. Because of the use of features, the system needs a classifier from the features using AdaBoost to choose them. The usage of classifiers using AdaBoost to select features and teach the classifiers. This approach needed more features to be added to increase accuracy, but this also increases computation time. So cascaded classifiers are introduced to minimize computation time as much as possible. This works like a tree structure as it trickles down and adjusts the next set of classifiers for a higher detection rate. In summary to "Robust Real-Time Face Detection" by Paul Viola and Michael J. Jones, it demonstrated a very effective way for facial detection that can processes these images very quickly.