Finding Tiny Faces & Viola-Jones Method of Rapid Object Detection – A Review

Sanidhya Singal
2015085, IIIT-Delhi
sanidhya15085@iiitd.ac.in

***Abstract*—Tremendous strides have been made in the field of Object Recognition, particularly, Face Detection, and many approaches to the task have been implemented over multiple decades. I summarize two prominent papers on the task of face detection – Viola-Jones method of rapid object detection[1] and Hu et al. method of finding tiny faces[2] – by means of this technical report. These papers discuss two fundamentally different problems and give insights on how to approach them. The performances of the models proposed have been evaluated on benchmark face detection data-sets and the results achieved are pretty promising. The technique provided in the former paper has been in use for some time now with practical applications, while the latter paper is a recent one.**

I. Introduction

The Viola-Jones object detection framework is the first object detection framework to provide competitive object detection rates in real-time[1]. Although it can be trained to detect a variety of object classes, it was motivated primarily by the problem of face detection[3]. Though human beings can detect faces easily, yet, a computer needs precise instructions and constraints. Viola-Jones paper describes a machine learning approach for visual object detection, which is capable of processing the images extremely rapidly and achieving high detection rates. It yields detection rates comparable to the then best systems[1].

The authors have constructed a frontal face detection system. The distinguishable feature of this system is its ability to detect faces extremely rapidly. None of the earlier published results could achieve this. This method is still widely used in face detection softwares and devices.

The authors propose three key concepts: *Integral Images*, AdaBoost and Cascade classifiers, as explained in the following lines. Integral Image is a new image representation that allows for very fast feature evaluation. This integral image can be computed from an image in *constant* time by using a few operations per pixel. The authors slightly modify the AdaBoost algorithm to construct a powerful classifier. The classifier uses AdaBoost to select a small number of important features in order to increase the speed of the process. The cascade classifier allows combining successively complex classifiers in a cascade structure. This dramatically increases the speed of the detector and makes it more accurate as well.

Hu et al.[2] cover another challenging problem in object recognition, which is detecting small objects. The paper primarily discusses the techniques for detecting tiny faces in an image. The authors talk about the importance of scale, context and resolution during object recognition. They demonstrate state-of-the-art results and claim to have reduced the error by a factor of 2.

The authors explore three aspects of the problem in the context of face detection: the role of scale invariance, image resolution, and contextual reasoning. I discuss them briefly as follows.

1) Hu et al. reason that *scale-invariance* will inherently make mistakes. The process for detecting an object 3px wide is much different than detecting one 300px wide. Our own human eye proves this assertion. We don't use the same technique when looking at something large or small. One has to squint one's eyes when looking at something small and trying to decipher the smaller object. Similarly, the authors train specific detectors for different scales in order to boost the detection/classification accuracy[4].
2) As a simple experiment, the authors ask some people to classify true and false positive faces. They observe that smaller faces are dramatically unrecognizable without their contexts, while this is not the case with large faces. They argue that due to the little signal on the object to exploit, one needs image evidence beyond the object extent. This is formulated as *context*. The authors show that context is mostly useful for finding low-resolution faces (and not large faces). Adding a fixed contextual window of 300px to the object of interest dramatically reduces the error on small faces by 20%[2].
3) Image *resolution* is significant during object recognition. Consider the ImageNet dataset where the average object size is between 40 to 140px. For such a dataset, generally, we build the templates (to detect the objects) at a fixed resolution. But this doesn't work with a dataset having a significant number of objects with varying sizes (other than the average size). To overcome this problem, the authors propose building templates at 2x and 0.5x resolutions to find small and large faces (or in general, objects), respectively. They show that by using this approach, the overall accuracy improves by more than 5%.

Both the papers are explained in detail in further sections.

II. Methods Used

In this section, we discuss the methodology and the algorithms used by the authors in their respective papers to approach the said problems.

To achieve the task of face detection, Viola-Jones propose new algorithms for an extremely rapid object detection. They begin their object detection procedure with selecting features from images and classifying the images based on the value of these features. The motivation behind not using the pixels instead, is that a feature-based system runs much faster. Viola-Jones make use of Haar features, particularly two-rectangle features, as shown below.

Haar Features

All human faces share some similar properties. These regularities may be matched using Haar Features[3].

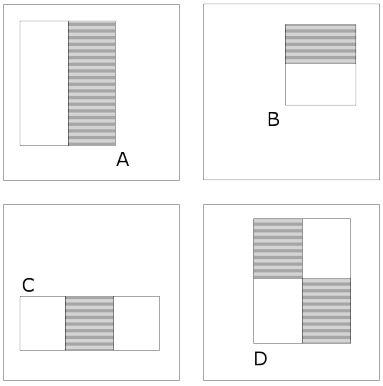
A few properties common to human faces:

1) The eye region is darker than the upper-cheeks.
2) The nose bridge region is brighter than the eyes.

Composition of properties forming matchable facial features:

1) Location and size: eyes, mouth, bridge of nose
2) Value: oriented gradients of pixel intensities

The four features matched by this algorithm are then sought in the image of a face.



Rectangle Features are shown. (Source: [3])

Value = Σ(pixels in black area) – Σ(pixels in white area) [3]

We can compute these rectangular features very rapidly by using the integral image. The integral image at location x, y contains the sum of the pixels above and to the left of x, y inclusive[1]:

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y')$$

Using the integral image, we can compute any rectangular sum in four array references. Clearly, we can compute the difference between two rectangular sums in eight references. Since the two-rectangle features defined above involve adjacent rectangular sums, we can compute them in six array references, eight in the case of the three-rectangle features, and nine for four-rectangle features[1]. Thus, the integral image evaluates rectangular features in *constant* time, which gives them a considerable speed advantage over more sophisticated alternative features.

Learning Algorithm

Viola-Jones algorithm uses a 24x24 window as the base window size to start evaluating these rectangular features in a given image. We place these features at every position with every size possible. Consequently, we have an exhaustive set of more than 160,000 features[5]. It would be very expensive to evaluate all of them while testing an image. Thus, the object detection framework employs a variant of the learning algorithm **AdaBoost** to both select the best features and to train classifiers that use them.

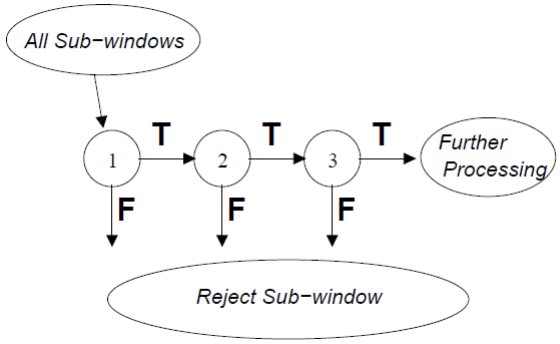
Adaboost is a machine learning algorithm which helps in finding only the best features among all these 160,000+ features. After these features are found, a weighted combination of all these features is used in evaluating and deciding if a given window has a face or not. These features are also called weak classifiers. This algorithm constructs a *strong* classifier as a linear combination of weighted simple *weak* classifiers.

$$F(x) = \alpha_1 f_1(x) + \alpha_2 f_2(x) + \cdots$$

Here, F is a *strong* classifier, while f_is are *weak* classifiers. $f_i(x) \in \{0, 1\}\ \forall i$. If a face is detected, it is 1, otherwise 0.

However, a single strong classifier formed out of a linear combination of all best features is not a good classifier because of the computation cost involved. Even if an image contains one or more faces, a large amount of it would still contain non-faces. Consequently, most of the f_is will be 0. This can waste a lot of time. Nevertheless and more importantly, we can minimize this time wastage.

We replace the linear classifier by a cascade classifier. The cascade classifier is composed of a number of stages, in the order of increasing complexity, each containing a *strong* classifier. This each stage evaluates a certain number of features. *For example*, one of the earlier stages might look for the presence of nose in the image. In case, no nose is detected, the image cannot contain a face. Hence, the image is discarded. In other words, we are able to discard a non-face image quite early in the process. This boosts up the speed of the classifier, and hence, the name *boosted* cascade classifier.



Schematic representation of a detection cascade. (Source: [1])
Thus, the Viola-Jones algorithm has four stages:

1) Haar Feature Selection
2) Creating an Integral Image
3) Adaboost Training
4) Cascading Classifiers

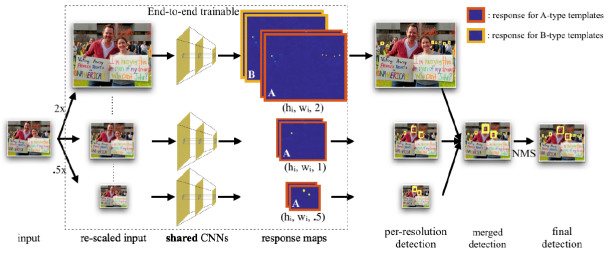
Hu et al. present a machine learning based method, involving Neural Networks, for detecting tiny faces. This method improves upon existing detectors and incorporates a multi-task training model.

Multi-task Learning

Multi-task learning involves learning how to do separate actions while sharing information that may be useful in ac-

complishing all of these tasks[4]. In this case, the separate actions are really the different scale detections, while sharing information across scales between each of the specific detectors. The authors propose to train a bunch of templates of faces at different resolutions and then, select the ones that do the best across all scales. An interesting thing is that you only need 3 regimes. For large images (>140px), use 2x smaller resolution; for smaller images (<40px) use 2x larger resolution; and just use the same resolution for anything in between, where most objects fall, as discussed earlier as well.

The detection pipeline for finding faces is split into a 3-level image pyramid, which includes both a 2x upsampling space (for small faces) and a 2x downsampling space (for large faces). This scaled input is fed into three separate CNNs (Convolutional Neural Networks) to predict template responses at every resolution. The CNNs extract the hypercolumnal features (important for context) and then, predict response maps of the corresponding templates, generated from the multi-task learning above. Given those, we extract bounding boxes and then merge them back into a single image. For predicting the template response, they use both an A-type template (tuned for normal faces, 40-140px) and B-type template (tuned for ¡40px). The A-type is run on all 3 levels of the pyramid, while the B-type is only run on the 2x upsampled image, for detecting smaller faces[2][4].



Detection Pipeline (Source: [2])

III. Results and Observations

Viola-Jones worked on MIT+CMU frontal face test set. This set consists of 130 images with 507-labelled frontal faces. On a 700 MHz Pentium III processor, the face detector could process a 384x288 pixel image in about 0.067 seconds [1]. The cascaded classifier had 38 layers and the base resolution was 24x24. The high speed of the classifier was largely because the first or the second layer in the cascade rejected majority of images. The classifier could run approximately 15 times faster than any known detector at that time.

Hu et al. worked on WIDER FACE and FDDB datasets. The model proposed by them achieves state-of-the-art performance on WIDER FACE dataset. More significantly, it reduces the error on *hard* set (set containing all faces taller than 10px) by 2x. The results are similar with FDDB dataset. The difference lies in the fact that WIDER FACE uses bounding boxes while FDDB dataset uses bounding ellipses. The authors transform bounding box predictions to ellipses using a post-hoc linear regressor. With this, the detector again achieves a state-of-the-art performance. The *run-time* of the detector is independent of the number of faces in an image [2].

IV. Conclusions

Viola-Jones have presented an approach for object detection, which minimizes computation time while achieving high detection accuracy. They use the approach to construct a face detection system, which is approximately 15 times faster than any previous approach. This paper brings together new algorithms, representations and insights that are quite generic and have broader application in computer vision and image processing. An extremely fast face detector has many practical applications. These include user interfaces, image databases, and teleconferencing. It can be implemented on a wide range of small low power devices, including hand-helds and embedded processors[1].

Hu et al. propose a simple yet effective framework for finding small objects, demonstrating that both large context and scale-variant representations are crucial. They also explore the encoding of scale in existing pre-trained deep networks, suggesting a simple way to extrapolate networks tuned for limited scales to more extreme scenarios in a scale-variant fashion. Finally, they use their detailed analysis of scale, resolution and context to develop a state-of-the-art face detector that significantly outperforms prior work on standard benchmarks[2].

References

[1] P. Viola and M. Jones, *Rapid object detection using a boosted cascade of simple features*, 2001

[2] P. Hu and D. Ramanan, *Finding tiny faces*, CoRR, 2017

[3] https://en.wikipedia.org/wiki/Viola-Jones_object_detection_framework

[4] http://mohsaad.com/2017/09/19/Finding-Tiny-Faces/

[5] https://stackoverflow.com/questions/1707620/viola-jones-face-detection-claims-180k-features