Faster face detection using Convolutional Neural Networks & the Viola-Jones algorithm

Karina Enriquez *

B.S. Candidate, Department of Computer Science, California State University Stanislaus, 1 University Circle, Turlock, CA 95382

Received 18 April, 2018; accepted 15 May 2018

Abstract

If you have ever used social media, a digital camera, or a cell phone, chances are you have encountered face detection more than once. Popular social media applications ranging from Facebook to Snapchat use face detection for several of their popular features such as tagging friends and applying filters. Face detection is defined as computer technology that is used to detect human faces in digital images. There are various computer algorithms that are employed in the field of face detection, but this paper will focus on two of the most popular methods: Convolutional Neural Networks and the Viola-Jones algorithm. The motivation for this paper is a general curiosity about face detection in everyday life as well as a curiosity about how face detection algorithms work, in my classes so far I have talked about the structure of different types of computer systems but I have never delved into algorithms for these systems. This project offered the opportunity to not only learn about face detection but also delve into the world of computer algorithms.

Keywords: social media, privacy, face recognition,

Introduction

Face detection is something that is prevalent in our lives but also something many technology users have probably never thought about in depth. If you've ever used a digital camera, the camera on a phone, Facebook's tagging feature, or the social media app Snapchat, chances are you have encountered face detection before. Face detection is the application of computer technology to detect faces in digital images. All of the aforementioned apps and products use some form of facial detection, whether that be introducing tagging features, filters for pictures, or even just improving the quality of a photo. Given the prevalence of these products in day-to-day life, the digital industry is rapidly trying to improve these features in order to provide the best service. Because of this digital race, products such as the iPhone X have implemented new face detection and recognition software. Two of the most widely incorporated face detection methods at the moment are the Viola-Jones algorithm and Convolutional Neural Networks.

Background

The two methods that will be discussed are the Viola-Jones algorithm and Convolutional Neural Networks. In their 2001 paper *Robust Real-Time Object Detection* Paul Viola and Michael Jones proposed a visual object detection framework that was capable of fast image detection (1-16). At the time it was proposed this algorithm was game-changing given that it was the first object detection framework that provided "competitive object detection rates in real-time" (Ephraim, Hemmelman, & Siddiqi, 2009). Viola and Jones are largely credited with inspiring a new wave of face detection methods. One product of this face detection wave were Convolutional Neural Networks (CNNs). While the Viola-Jones algorithm offered high accuracy for frontally positioned faces, it proved to be less accurate if the same faces were slightly tilted or if a side profile was given. What CNNs offered was not only fast detection, but a more diverse capability when it came to facial positioning (Farfade, Aberian, & Li, 2015). While CNNs were able to improve upon this fault in the Viola-Jones algorithm, this does not mean the Viola-Jones algorithm is a completely inferior method.

The Viola-Jones Algorithm

The way that the Viola-Jones algorithm actually works is through the execution of four main steps: Haar-like Features, Integral Image, Adaboost Training, and the attentional cascade (Viola & Jones, 2001). The first step, Haar-like features, are a set of rectangular digital image features that break up the image into multiple sets of "two adjacent rectangles located at any scale and position within an image" (Ephraim,

* Corresponding author. Email: kenriquez1@csustan.edu

Himmelman, & Siddiqi, 2009). These rectangles are then applied to the image that has been opened within the program. In Figure 1 the Haar-like features are applied to the image. As demonstrated, there is first

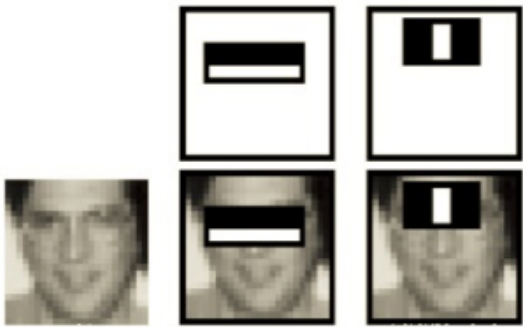

Figure 1: Haar-like features
This image shows Haar-like features being applied to the eye region and the space between the forehead and the eyes (Viola & Jones, 2001)

one main region that is being examined, the area from the forehead to the eyes. This region is selected given that the region of the eyes is darker than the region of the nose and cheeks so if the Haar-like features can be matched then the image can go to the next step with the Adaboost training. What Adaboost training does here is to better define the region from the eyes to the cheeks. The way it does this is by employing a learning algorithm that is used to "teach" the program to look over a set of possible areas and then choose the areas with the "best" features resulting in a reduced image with more defined regions (Viola & Jones, 2001). After the features are selected they are put through the Adaboost learning algorithm to narrow down the number of features that are looked at and then passed on to the cascading stage. In more broad terms, the Adaboost training makes sure that the region that is being examined is as precise as possible in order to help obtain the best accuracy. The last step is the attentional cascade. During this step, the program is trying, again, to maintain the smallest error percentage rate as possible. In order to do this, the attentional cascade's main focus is to eliminate as many false positives as possible. In practice, for instance, the initial steps identify an object, such as a bush, as a potential face then the attentional cascade takes the bush and discards it as a viable option. Decreasing the number of possible faces not only helps decrease false positives, it also helps to increase the number of correct detections. The Viola-Jones method, although it has a high accuracy rate, does come at the price of longer computation time. In contrast, CNNs have addressed computation time concerns and have successfully improved upon them.

Convolutional Neural Networks

Convolutional neural networks (CNNs) work in a similar way to the Viola-Jones method at least conceptually. So what is different? While both methods work in a series of steps, the steps in the Viola-Jones method are set whereas the steps in CNNs are much less structured. Although there is a general sense of what types of steps exist, the order is not uniform; additionally some steps are capable of running concurrently. This might seem counter intuitive given that the results of one step are passed on to the next step, but the fact that there are usually multiple pass-throughs of the various steps makes sure every step has the correct information for its execution. There are four basic steps used in CNNs: convolution, pooling, and ReLu ("How do" 2016).

Figure 2: CNN steps. This figure illustrates steps that may be implemented when evaluating an image ("How do", 2016)

Figure 2 illustrates a sample path a CNN might follow when evaluating an image. Convolution is the step that involves the most math, so when this step first is encountered the program does not know what it is looking for; in other words, it does not know where the face is. The convolution step is trained to recognize faces, so when a new image is inserted into the document the convolution step uses its previous knowledge to test where the faces might be. The math, is, in short, multiplying each pixel in the feature by the value in the corresponding pixel in the image. The answers are then divided by the total number of pixels in the image. Every matching pixel results in a value of 1 so anything else is -1. In Figure 3 a white cross is the item that is being looked at. For every pixel that matches the white cross, a value of 1 is given and the rest of the black pixels are given a value of -1. The CNN does this simple math over and over until it matches the features to a spot on the image. After the convolution step completes once, it must repeat itself multiple times until it has the desirable number of possible locations narrowed down. From here the pooling phase shrinks the image down enough to keep all possible features but still narrow the window of possible locations of the face.

The next step is the Rectified Linear Units or ReLU. In this step all negative features from the picture are changed to zeros which is mostly a step to

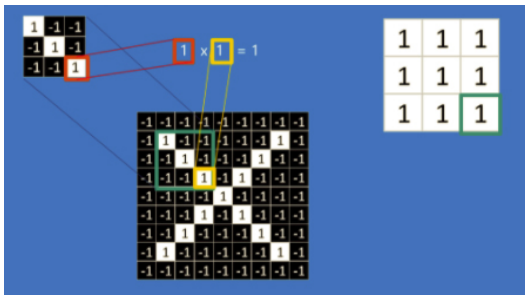

Figure 3: CNN Pooling math. This figure illustrates the white cross sample and how each pixel is assigned a value of 1 or -1 ("How do", 2016)

keep the CNN mathematically sane by keeping the CNN from approaching infinity ("How do", 2016).

The final layer is the fully connected layers; this final layer is the layer that gets the final say as to whether or not the final selection of features is correct. For every value that is output by the previous steps it gets a vote as to whether or not the value matches up with the image value.

For example, let's say there is a CNN that is trained to find eye positions. The first step would look through the image and find any possible location of the eyes. The program will scan all of the pixels in the image and once it has done this the pooling step eliminates areas that it does not believe are the eye region. If the picture being evaluated had been taken outdoors, the program removes the background and the bushes and keeps the entire facial region. The ReLU step will take only the values that were picked out from the photo that were negative and change them to zero. Conceptually there is no change to the image that is being processed, the area that is being examined has not changed and the entire face is still considered a possible location. After the initial pass-through the program will go through the steps again where the region that is being examined is shrinking with each pass-through until a small enough region has been identified and the program can proceed to the last step. The final step, the fully connected layers, will take a look at all of the values assigned to the selected part of the image and then will reference back to its training and try to closely match the values of what should be eyes to the values that it knows are eyes. If, for example, the eye region has pixel size of 0.11 in the training examples and the image that was scanned has a value of 0.14 the program will keep that region as a possibility, if it does not then the program will discard it. While this method provides for the analysis of a broader set of facial positions, the multiple pass-throughs CNNs must go through create a lot of overhead and thus require more space on the computer's memory. Both the Viola-Jones algorithm and CNNs have their strengths, whether that be with respect to accuracy or computation time. They do, however, both lack efficiency in some areas as well.

Comparison

While there exist similarities between the Viola-Jones algorithm and Convolutional Neural Networks, they do both differ in a few areas. For one, the Viola-Jones algorithm is not able to detect faces in varied positions. If the face is not presented in a front-facing position with proper lighting, the accuracy of the results drops dramatically (Farfade, Saberian, & Li, 2015). The Viola-Jones algorithm's Adaboost training is also not designed to get better results, but rather to have less error. Lawrence et al. state that a prevalent problem in face detection is the fact that "there are not enough training points…to allow accurate estimation of class probabilities throughout the input space" (1997). This means that some applications have a less extensive list of acceptable facial positions because a more extensive list would compromise even more computational memory. Some of these varying positions range from a slight head tilt to low lighting. The Viola-Jones algorithm encounters this problem because their Haar-like features do not map very well to varying positions. By contrast, CNNs have the ability to detect faces in different positions (including side views) and different lighting scenarios, and therefore CNNs are much more diverse in how they correctly handle their input. Figure 4 shows how CNNs are able to detect faces in varying positions and lighting situations. By contrast, the Viola-Jones algorithm can only detect well-lit and front facing faces. Another difference is the amount of space needed to run each program. Given that Convolutional Neural Networks require multiple pass-throughs and therefore store large amounts of information, CNNs require much more space locally than the Viola-Jones algorithm (Hanlon, n.d.). This becomes important when companies are looking to implement face

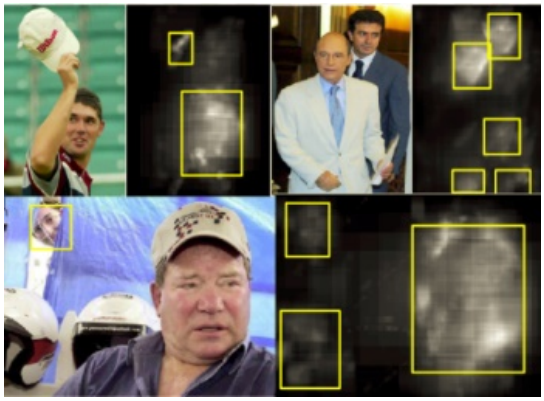

Figure 4: CNN Face detection. This image shows the potential faces found by the CNN after one pass through (Yang & Nevatia, 2016)

detection algorithms in mobile devices given that the amount of space provided by a phone chip is much more limited than that of a computer chip. Given this, the way that each of these methods is implemented is impacted significantly.

Discussion

As previously stated, the use of CNNs requires a larger storage capacity to run than does the Viola-Jones algorithm. Because of this, the places where CNNs can be implemented correctly become limited. Therefore, while CNNs are faster and much more reliable in terms of accuracy, the Viola-Jones algorithm is still widely used today. Two modern products using the Viola-Jones algorithm are various models of point-and-shoot digital cameras and the social media app Snapchat (Q, 2017). In the case of the Snapchat app, the user is meant to be facing the camera frontally in order to apply certain filters to the user's face and, given the fact that the app only runs on cell phones, the use of the Viola-Jones algorithm was a a clear choice. In contrast, CNNs are rapidly showing up in newer phones such as Apple's iPhoneX and Facebook's tagging protocol (Chamary, 2017). Apple's newest iPhone model is rumored to run its face recognition software using some form of a neural network while Facebook's CNN architecture is called DeepFace ("Top 8", 2015). These two uses are also obvious choices given that, in the case of the iPhone, one of their main selling points is that the facial recognition is applicable in all conditions, including low lighting, and that it should even work if the user is wearing sunglasses. If the iPhone had used the Viola-Jones algorithm then the promise of multiple angle recognition as well as low lighting recognition would be greatly compromised. The only way to guarantee these features actually work would be to use CNNs instead of the Viola-Jones algorithm. Facebook's algorithm would work in a similar way when tagging people in photos given that, when the feature debuted, cameras were not equipped to produce the high-quality images they do today, so low lighting was a given. The use of CNNs makes sense for both Facebook and the newest iPhone.

A problem that CNNs have encountered, however, is limits on both computational memory and physical memory, but as the accuracy of CNNs is proven in various products, more and more companies are equipping their products to support CNNs high memory occupancy. This proves that the direction companies are moving prioritizes versatile detection over accuracy. Given that technology is improving each day, it probably won't be long before CNNs come equipped with the same level of accuracy as the Viola-Jones algorithm offers.

The Merger

In order to allow for less memory usage and computational run-time, one approach would be to use aspects of the Viola-Jones algorithm in conjunction with Convolutional Neural Networks. We have already established the repetitiveness of the convolution step of CNNs largely fills the memory after the first step. The Viola-Jones algorithm's use of Haar-like features relies on a similar idea as convolution does but does not come with the same memory usage. As in CNNs, the Haar-like features and integral image go through each section of the image in order to correctly identify where possible known features are. If it were possible to use Haar-like features as a replacement for some of the convolution, not only would CNNs be as accurate as the Viola-Jones algorithm, it would also have the potential to decrease how much memory is required to apply CNNs. Replacing some of the CNN architecture would help improve accuracy of front face detection and keeping some of the CNN architecture would help reduce the length of time the program occupies the computer's RAM. There have already been similar approaches in taking aspects from the Viola-Jones algorithm and applying it to CNNs. Li, Lin, Shen, Brandt, & Hua (2015) used a cascade architecture in order to better reject false positives and eliminate background noise in the photos. Yang & Nevatia (2016) used a convolutional cascade in order to better zoom into the faces and used it to train the network in making these distinctions. While many face detection methods have used the cascade CNN combination, these combinations do not address the problem of the large memory requirements for CNNs.

In their paper *Compact Convolutional Neural Network Cascade for Face Detection*, Kalinovskii and Spitsyn (2016) propose an idea to help reduce the amount of computational complexity that is needed when applying CNNs for face detection (p.1). Kalinovskii and Spitsyn's idea is to limit the number of parameters by removing the fully connected layers so that the CNN does not take up too much of the computational memory for too long of a time. They explain that, due to the natural parallelism of CNNs, if you were to reduce the amount of time a process uses computational memory the accuracy shouldn't be compromised if done well. This means their approach keeps accuracy while improving performance time.

Another approach to reducing the amount of computational memory was proposed by Bong, Choi, Kim & Yoo and addresses the problem of computational memory specifically in cell phones (2017, p.30). Bong et al. propose that the hardware be changed first, in order to implement an always-on face detection using Viola-Jones (32). This approach reduces the restrictions on the amount of computational memory the software can use at one

time. While this idea is potentially revolutionary for cell phones, the question of whether it would be applicable to other forms of technology remains.

If a company is not producing a cell phone product, a final approach that is considered comes from Sarwar, Panda, & Roy (2017) from Purdue University. In their paper *Gabor Filter Assisted Energy Efficient Fast Learning Convolutional Neural Networks.* They propose reducing the memory required by CNNs by literally reducing the amount of work a CNN can do at a time (p.1). They explain that, while it is common to try and reduce the amount of testing a CNN does, their approach is to instead reduce the amount of training a CNN needs (p. 1). Sarwar, Panda, & Roy propose an idea of incorporating Gabor Filters into the testing phase rather than the CNN go through multiple training algorithms. A Gabor Filter is an image texture analysis filter that basically tells the user if what they're looking for is potentially there. Similarly to how Viola-Jones algorithm looks for specific facial features, the Gabor Filter looks for specific textures in an image (Murthy, 2015).

After analyzing all 3 proposals, one different proposal can be constructed. Similar to how Kalinovskii and Spitsyn reduce the amount of time by removing the fully connected layer, I propose we reduce another step in the CNN process, the training or convolutional step. Like Sarwar et al., I propose using a filter to reduce the amount of computational memory needed for CNN training but, instead of using a Gabor Filter, I would use Viola-Jones' Haar like features. In doing so, the CNN would run for less time and would also have the trusted accuracy that comes with using the Viola-Jones algorithm. A potential problem with using Haar-like features would be the real possibility that by, incorporating the Haar-like features into CNNs, the speed of detection would decrease. Because CNNs do so much in so little time, the drawback of possibly slowing down the process by introducing a new concept is real. However, the inherent rapidness of CNNs could prevent this effect from actually being impactful and, by minimizing the parameters of the verification in the Fully Connected Layers, speed should not be largely impacted. No matter the effects, tackling the problem of memory usage by CNNs is a problem worth looking into if the face detection industry is to help improve everyday products.

Copyright Law Compliance

In a world where technology is constantly changing and so much information is shared through various mediums including the internet, one must be aware of copyright protection laws when creating software. According to the U.S. Copyright Office, while copyright law protects authored works, it does not protect "ideas, concepts, systems, or methods of doing something" ("What does", n.d.). With this in mind program designers must still be wary of how they take inspiration from previous programs. According to David Carnes of Legalzoom.com (2017), copyright law cannot prevent you from creating your own software for a similar product "as long as you don't copy the algorithm used in that product". Since the Viola-Jones algorithm is regularly incorporated into face detection software, the process of acquiring permission to incorporate it into this potential hardware shouldn't be a difficult process. From there, the programmer would need to figure out the best way to condense the parameters of the CNN in a unique way. As long as the programmer is aware of the limitations on sharing ideas and takes the correct preventative measure to comply with copyright law, copyright infringement should not be a problem.

Conclusion

When looking at the Viola-Jones algorithm and Convolutional Neural Networks for face detection, it is difficult to say which one is best overall. Both methods have strengths and weaknesses when it comes to certain areas of face detection. While CNNs are much faster, they do require more memory space and are therefore are more expensive to implement. The Viola-Jones algorithm is older but given its minimal memory requirement it is implemented much more easily. The next best thing would be to implement some features from the Viola-Jones algorithm along with the Convolutional Neural Networks. While there has been some talk of implementing CNNs using Viola-Jones' cascading feature, the issue of CNNs accessing too much RAM is a constant problem. This paper proposes a way to combine the two methods, however, proper experimentation and development of this idea is beyond the scope of this project. The hope for the future is that this paper inspires more conversation and experimentation into decreasing the amount of space required to implement CNNs without compromising speed and accuracy.

References

Bong, K., Choi, S., Kim, C., & Yoo, H. (2017). Low-Power Convolutional Neural Network Processor for a Face-Recognition System. *IEEE Micro, 37*(6), 30-38. doi:10.1109/mm.2017.4241350

Carnes, D. (2017, November 21). Software Copyright Issues. Retrieved from http://info.legalzoom.com/software-copyright-issues-23662.html

Chamary, J. (2017, September 18). How Face ID Works On iPhone X. Retrieved from https://www.forbes.com/sites/jvchamary/2017/09/16/how-face-id-works-apple-iphone-x/2/#66d9568138f4

Ephraim, T., Himmelman, T., & Siddiqi, K. (2009, 05). Real-Time Viola-Jones Face Detection in a Web Browser. *2009 Canadian Conference on Computer and Robot Vision.* doi:10.1109/crv.2009.48

Farfade, S. S., Saberian, M. J., & Li, L. (2015). Multi-view Face Detection Using Deep Convolutional Neural Networks. *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval - ICMR '15*. doi:10.1145/2671188.2749408

Hanlon, J. (n.d.). Why is so much memory needed for deep neural networks? Retrieved from https://www.graphcore.ai/posts/why-is-so-much-memory-needed-for-deep-neural-networks

How do Convolutional Neural Networks work? (2016, August 18). Retrieved from https://brohrer.github.io/how_convolutional_neural_networks_work.html

Kalinovskii, I., & Spitsyn, V. (2015). Compact Convolutional Neural Network Cascade for Face Detection.

Lawrence, S., Giles, C., Tsoi, A. C., & Back, A. (1997). Face recognition: A convolutional neural-network approach. *IEEE Transactions on Neural Networks, 8*(1), 98-113. doi:10.1109/72.554195

Li, H., Lin, Z., Shen, X., Brandt, J., & Hua, G. (2015, 06). A convolutional neural network cascade for face detection. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. doi:10.1109/cvpr.2015.7299170

Murthy, K. (2015, September 12). Gabor Filters: A Practical overview. Retrieved from https://cvtuts.wordpress.com/2014/04/27/gabor-filters-a-practical-overview/

Q. (2017, March 17). The Inner Workings of Snapchat's Faceswap Technology. Retrieved from https://www.forbes.com/sites/quora/2017/03/17/the-inner-workings-of-snapchats-faceswap-technology/#3327327d64c7

Sarwar, S. S., Panda, P., & Roy, K. (2017). Gabor filter assisted energy efficient fast learning Convolutional Neural Networks. *2017 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)*. doi:10.1109/islped.2017.8009202

Top 8 Ways Facial Recognition Software is Being Used Today. (2015, July 14). Retrieved from http://www.techguruit.com/top-8-ways-facial-recognition-software-used-today/

Viola, P., & Jones, M. (2001). Robust real-time face detection. *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*. doi:10.1109/iccv.2001.937709

What Does Copyright Protect? (n.d.). Retrieved from https://www.copyright.gov/help/faq/faq-protect.html

Yang, Z., & Nevatia, R. (2016, 12). A multi-scale cascade fully convolutional network face detector. *2016 23rd International Conference on Pattern Recognition (ICPR)*. doi:10.1109/icpr.2016.7899705