

Face Detection & Recognition using Tensor Flow: A Review

Ishaan Chawala

Student, Department of Computer Science & Engineering, SBSSTC, Ferozepur

Ishaanchawla0001@gmail.com

ABSTRACT

Face recognition has become a popular topic of research recently due to increases in demand for security as well as the rapid development of mobile devices. There are many applications which face recognition can be applied to such as access control, identity verification, security systems, surveillance systems, and social media networks. Access control includes offices, computers, phones, ATMs, etc. Most of these forms currently do not use face recognition as the standard form of granting entry, but with advancing technologies in computers along with more refined algorithms, facial recognition is gaining some traction in replacing passwords and fingerprint scanners. Ever since the events of 9/11 there has been a more concerned emphasis on developing security systems to ensure the safety of innocent citizens. Namely in places such as airports and border crossings where identification verification is necessary, face recognition systems potentially have the ability to mitigate the risk and ultimately prevent future attacks from occurring. As for surveillance systems, the same point can be made if there are criminals on the loose. Surveillance cameras with face recognition abilities can aide in efforts of finding these individuals. Alternatively, these same surveillance systems can also help identify the whereabouts of missing persons, although this is dependent on robust facial recognition algorithms as well as a fully developed database of faces. And lastly, facial recognition has surfaced in social media applications on platforms such as Facebook which suggest users to tag friends who have been identified in pictures. It is clear that there are many applications the uses for facial recognition systems. In general, the steps to achieve this are the following: face detection, feature extraction, and lastly training a model.

Keywords: Face recognition, machine learning, tensorflow, Face detection.

INTRODUCTION TO FACE RECOGNITION

The word "biometrics" came from Greek words "bio" means life and "metrics" means to measure. Biometric is the process of identification of humans with the use of measurable biological characteristics or trait. In computer science, biometric is used for authentication and access control. User authentication can be done in three ways: "Something you know (password or pin)" "Something you have (key or ID card)" "Something you are i.e. biometrics (your face, voice, fingerprint or DNA) Advantages of biometrics over other authentication techniques are that they cannot be forgotten or lost. Also, biometrics characters are unique to individual humans, so they are more difficult to fake. There are many types of biometrics systems including face recognition, fingerprint recognition, iris recognition etc.

Nowadays, machine face recognition technologies still encounter a lot of challenges that it needs to overcome before it can be widely accepted and deployed. The aim of recognition technologies is to mimic or even exceed the ability of the human mind in recognizing faces and/or other objects [1]. In light of this, researchers have endeavored during the past years to produce and use various algorithms to address several problems that arose with the development and implementation of face recognition systems [2]. However, only a few of these systems really have the ability to recognize faces the way the human mind does. Another problem that surfaced with the implementation of face recognition systems is the fact that the use of facial databases invades the privacy of people and, if not properly secured, may also leak certain information about these individuals to unauthorized people [3]. Aside from these, researchers have also been dealing with the problem of varying illumination between trained and actual images. Certain studies show that the difference in illumination affects the efficacy of face recognition systems [4]. Also, the angle of rotation of the human face is also one factor that still requires further study since most face recognition systems limit the angle of rotation

of their subjects' faces to either just facing the camera directly or having only a slight difference from it. Face recognition is one of the important methods of biometric identification. Developing a face recognition model continues to be an extremely fascinating field for many researchers mainly because of its many real world applications like criminal identification, user authentication, security systems and surveillance systems. However, due to its complex and multidimensional structure, it is difficult to develop a face recognition model. The face recognition process basically involves four steps image acquisition, image pre-processing, feature extraction and classification as shown in figure 1.1. Image acquisition is the first step in face recognition which involves reading an input image from disk or camera and locating face in that image. In image pre-processing step, image enhancement techniques are applied on input image like histogram equalization, sharpening and smoothing. Generally, a combination of several image enhancement techniques is applied to get the best result. In feature extraction step, a feature vector unique to the image is generated by analysing the face and a template is created. In the final step, a test image is classified into either one of the known face or unknown face using classification techniques. The amount of computation and memory required in these face recognition steps is mainly affected by the approach used for face representation, that is, how to model a face. Depending upon the face representation approach, face recognition algorithms are classified into template based, feature based, and appearance based.

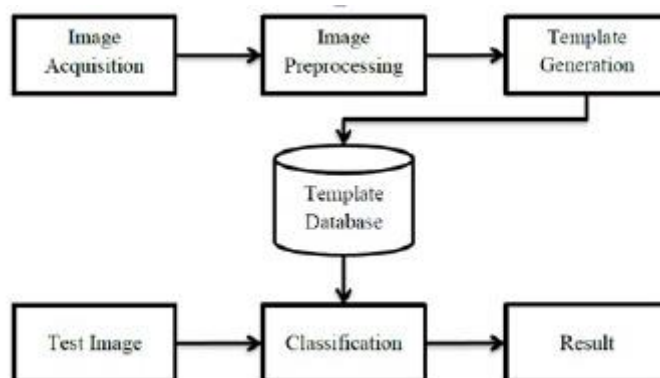


Figure 1. Steps involved in face recognition

Image acquisition is the first step in face recognition which involves reading an input image from disk or camera and locating face in that image. In image pre-processing step, image enhancement techniques are applied on input image like histogram equalization, sharpening and smoothing. Generally, a combination of several image enhancement techniques is applied to get the best result. In feature extraction step, a feature vector unique to the image is generated by analysing the face and a template is created. In the final step, a test image is classified into either one of the known face or unknown face using classification techniques. The amount of computation and memory required in these face recognition steps is mainly affected by the approach used for face representation, that is, how to model a face. Depending upon the face representation approach, face recognition algorithms are classified into template based, feature based and appearance based.

TEMPLATE BASED APPROACH

In a most simple template based approach for face representation, a single template, i.e. an array of intensities of face image, is used. Multiple templates of each face may be used in a complex method of template based approach where each template represents different viewpoint. Simplicity is the advantage of template based approach, but it has a major disadvantage of very huge requirements and inefficient matching.

FEATURE BASED APPROACH

Feature based approach uses facial features like relative position and/or size of eyes, nose and jaw to represent faces. To recognize a test face, same set of features are extracted from test image as earlier and are matched to precompiled features. The advantages of feature based approach are very little memory requirements and faster recognition speed. But in practical, it is very difficult to implement perfect feature extraction technique.

APPEARANCE BASED APPROACH

Appearance based approach is somewhat similar to template based approach. In this approach a face image is projected onto a linear subspace which has much lower dimensions compared to original image dimensions. This low dimensional subspace has to be precompiled by applying Principal Component Analysis (PCA) on training face image set. Appearance based approach requires very less memory as compared to template based approach and, also, it has fast recognition rate. The major drawback of appearance based approach is that it has high computation cost. As compared to template based and feature based approach, appearance based approach is simple and efficient except its high computation cost. If some way of lowering the computational cost is developed, then appearance based approach is a good practical approach for face recognition.

APPLICATIONS OF FACE RECOGNITION

Many applications for face recognition have been envisaged, and some of them have been hinted at above. Commercial applications have so far only scratched the surface of the potential. Installations so far are limited in their ability to handle pose, age and lighting variations, but as technologies to handle these effects are developed, huge opportunities for deployment exist in many domains.

1. Access Control: Face verification, matching a face against a single enrolled exemplar, is well within the capabilities of current Personal Computer hardware. Since PC cameras have become widespread, their use for face-based PC logon has become feasible, though take-up seems to be very limited. Increased ease-of-use over pass- word protection is hard to argue with today somewhat unreliable and unpredictable systems, and for few domains is there motivation to progress beyond the combinations of password and physical security that protect most enterprise computers. As biometric systems tend to be third party, software add-ons the systems do not yet have full access to the greater hardware security guarantees afforded by boot-time and hard disk passwords. Visionics face-based screen lock is one example, bundled with PC cameras. Naturally such PC-based verification systems can be extended to control authorization for single-sign-on to multiple networked services, for access to encrypted documents and transaction authorization, though again uptake of the technology has been slow. Physical access control is another domain where face recognition is attractive and here it can even be used in combination with other biometrics. Biold [23] is a system which combines face recognition with speaker identification and lip motion.

2. Identification Systems: Two US States (Massachusetts and Connecticut [3]) are testing face recognition for the policing of Welfare benefits. This is an identification task, where any new applicant being enrolled must be compared against the entire database of previously enrolled claimants, to ensure that they are not claiming under more than one identity. Unfortunately face recognition is not currently able to reliably identify one person among the millions enrolled in a single states database, so demographics (zip code, age, name etc.) are used to narrow the search (thus limiting its effectiveness), and human intervention is required to review the false alarms that such a system will produce. Here a more accurate system such as fingerprint or iris-based person recognition is more technologically appropriate, but face recognition is chosen because it is more

acceptable and less intrusive. In Connecticut, face recognition is the secondary biometric added to an existing fingerprint identification system. Several US States, including Illinois, have also instituted face recognition for ensuring that people do not obtain multiple driving licenses.

3. Surveillance: The application domain where most interest in face recognition is being shown is probably surveillance. Video is the medium of choice for surveillance because of the richness and type of information that it contains and naturally, for applications that require identification, face recognition is the best biometric for video data. Though gait or lip motion recognition have some potential. Face recognition can be applied without the subject's active participation, and indeed without the subject's knowledge. Automated face recognition can be applied live to search for a watch-list of interesting people, or after the fact using surveillance footage of a crime to search through a database of suspects. The deployment of face-recognition surveillance systems has already begun though the technology is not accurate enough yet [14]. The US government is investing in improving this technology [10] and while useful levels of recognition accuracy may take some time to achieve, technologies such as multiple steerable zoom cameras, non-visible wavelengths and advanced signal processing are likely to bring about super-human perception in the data gathering side of surveillance systems.

4. Pervasive Computing: Another domain where face recognition is expected to become very important, although it is not yet commercially feasible, is in the area of pervasive or ubiquitous computing. Many people are envisaging the pervasive deployment of information devices. Computing devices, many already equipped with sensors, are already found throughout our cars and in many appliances in our homes, though they will become ever more widespread. All of these devices are just now beginning to be networked together. We can envisage a future where many everyday objects have some computational power, allowing them to adapt their behaviour to time, user, user control and a host of other factors. The communications infrastructures permitting such devices to communicate to one another are being defined and developed (e.g. Bluetooth, IEEE 802.11). So, while it is easy to see that the devices will be able to have a well-understood picture of the virtual world with information being shared among many devices, it is less clear what kind of information these devices will have about the real physical world. There are many examples of pervasive face recognition tasks: Some devices such as Personal Digital Assistants (PDAs) may already contain cameras for other purposes, and in good illumination conditions will be able to identify their users. A domestic message centre may have user personalization that depends on identification driven by a built-in camera.

MACHINE LEARNING

In recent years, machine learning has driven advances in many different fields. We attribute this success to the invention of more sophisticated machine learning models, the availability of large datasets for tackling problems in these fields [9, 64], and the development of software platforms that enable the easy use of large amounts of computational resources for training such models on these large datasets. TensorFlow supports both large-scale training and inference: it efficiently uses hundreds of powerful (GPU-enabled) servers for fast training, and it runs trained models for inference in production on various platforms, ranging from large distributed clusters in a datacenter, down to running locally on mobile devices. At the same time, it is flexible enough to support experimentation and research into new machine learning models and system-level optimizations. TensorFlow uses a unified dataflow graph to represent both the computation in an algorithm and the state on which the algorithm operates. We draw inspiration from the high-level programming models of dataflow systems and the low-level efficiency of parameter servers. Unlike traditional dataflow systems, in which graph vertices represent functional computation on immutable data, TensorFlow allows vertices to represent computations that own or update mutable state. Edges carry tensors (multi-dimensional arrays) between nodes, and TensorFlow transparently inserts the appropriate communication between distributed sub computations. By unifying the computation and state management in a single programming model, TensorFlow allows programmers to experiment with different parallelization schemes that, for example, offload computation onto the servers that hold the shared state to reduce the amount of network traffic. We have also

built various coordination protocols, and achieved encouraging results with synchronous replication, echoing recent results that contradict the commonly held belief that asynchronous replication is required for scalable learning.

The recognition of a human face is one of the most basic activities that humans perform with ease on a daily basis. However, when this problem is tried to be solved using algorithms, it proves to be an extremely challenging one. The idea of a machine capable of knowing who is the person in front of them has existed for a long time, the first attempts happening on the 70s [Kelly, 1971]. The researchers have ranged from computer engineers to neural scientists [Chellappa, Wilson, and Sirohey, 1995]. However, during many years no quality solutions were obtained. It has not been until the late 2000s and beginning of the 2010s that functional systems have started to appear. The uses for an automatic face recognition system are many. Typical ones are biometric identification usually combined with other verification methods, automatic border control, or crowd surveillance. One of its main advantages is its non-intrusively. Most identification methods require some action from people, either putting the fingerprint in a machine, introducing a password, etc. On the contrary, face recognition can work by simply having a camera recording. Among other uses, some of its most well known uses belong to the social network field. As of 2016, there are already system being used that rely on face recognition, a brief sample of which are introduced here. This sample is by no means exhaustive, but it tries to show the variety of applications. It comes as no surprise that one of the most uses that draws most attention is to track criminals. As forensic TV series have shown, having a system automatically scanning city cameras to try to catch an escapee would be of great help. In fact, United States is already using this technology. Although far from the quality level depicted in fiction, they are already using it although there is some scepticism regarding whether it works to identify people from afar. Although the large criticism there is involving this kind of methods, there is little doubt that in the future they will become widely used. A not so well known use of face recognition is to authorize payments. As a part of a pilot test, some users are, under some circumstance, asked to take a picture of themselves before the payment is accepted. This kind of applications have a double goal: to facilitate the process to users being easier than remembering a password, and to discourage credit card thefts.

TENSOR FLOW

The Google Brain project started in 2011 to explore the use of very-large-scale deep neural networks, both for research and for use in Google's products. As part of the early work in this project, we built DistBelief, our first-generation scalable distributed training and inference system [14], and this system has served us well. We and others at Google have performed a wide variety of research using DistBelief including work on unsupervised learning [11], language representation [5, 2], models for image classification and object detection [8], video classification [27], speech recognition, sequence prediction [7], move selection for Go [4], pedestrian detection [2], reinforcement learning [8], and other areas [17, 5]. In addition, often in close collaboration with the Google Brain team, more than 50 teams at Google and other Alphabet companies have deployed deep neural networks using DistBelief in a wide variety of products, including Google Search [11], our advertising products, our speech recognition systems [6], Google Photos [43], Google Maps and StreetView [19], Google Translate [18], YouTube, and many others. Based on our experience with DistBelief and a more complete understanding of the desirable system properties and requirements for training and using neural networks, we have built TensorFlow, our second-generation system for the implementation and deployment of large scale machine learning models. TensorFlow takes computations described using a dataflow-like model and maps them onto a wide variety of different hardware platforms, ranging from running inference on mobile device platforms such as Android and iOS to modestsized training and inference systems using single machines containing one or many GPU cards to large-scale training systems running on hundreds of specialized machines with thousands of GPUs. Having a single system that can span such a broad range of platforms significantly simplifies the real-world use of machine learning system, as we have found that having separate systems for large-scale training and small-scale deployment leads to significant maintenance burdens and leaky abstractions. TensorFlow computations are expressed as stateful dataflow graphs and we have focused on

making the system both flexible enough for quickly experimenting with new models for research purposes and sufficiently high performance and robust for production training and deployment of machine learning models. For scaling neural network training to larger deployments, TensorFlow allows clients to easily express various kinds of parallelism through replication and parallel execution of a core model dataflow

TENSORFLOW WORKING

TensorFlow is the successor to DistBelief, which is the distributed system for training neural networks that Google has used since 2011 [10]. DistBelief uses the parameter server architecture, and here we criticize its limitations, but other systems based on this architecture have addressed these limitations in other ways [11, 14, 49]; In the parameter server architecture, a job comprises two disjoint sets of processes: stateless worker processes that perform the bulk of the computation when training a model, and stateful parameter server processes that maintain the current version of the model parameters. DistBelief programming model is similar to Caffe's [18]: the user defines a neural network as a directed acyclic graph of layers that terminates with a loss function. A layer is a composition of mathematical operators: for example, a fully connected layer multiplies its input by a weight matrix, adds a bias vector, and applies a nonlinear function (such as a sigmoid) to the result. A loss function is a scalar function that quantifies the difference between the predicted value (for a given input data point) and the ground truth. In a fully connected layer, the weight matrix and bias vector are parameters, which a learning algorithm will update in order to minimize the value of the loss function. DistBelief uses the DAG structure and knowledge of the layers' semantics to compute gradients for each of the model parameters, via backpropagation [13]. Because the parameter updates in many algorithms are commutative and have weak consistency requirements [11], the worker processes can compute updates independently and write back "delta" updates to each parameter server, which combines the updates with its current state. Although DistBelief has enabled many Google products to use deep neural networks and formed the basis of many machine learning research projects, we soon began to feel its limitations. Its Python-based scripting interface for composing pre-defined layers was adequate for users with simple requirements, but our more advanced users sought three further kinds of flexibility:

Defining new layers For efficiency, we implemented DistBelief layers as C++ classes. Using a separate, less familiar programming language for implementing layers is a barrier for machine learning researchers who seek to experiment with new layer architectures, such as sampled softmax classifiers [37] and attention modules [53].

Refining the training algorithms many neural networks are trained using stochastic gradient descent (SGD), which iteratively refines the parameters of the network by moving them in the direction that maximally decreases the value of the loss function. Several refinements to SGD accelerate convergence by changing the update rule [23, 66]. Researchers often want to experiment with new optimization methods but doing that in DistBelief involves modifying the parameter server implementation. Moreover, the `get ()` and `put ()` interface for the parameter server is not ideal for all optimization methods: sometimes a set of related parameters must be updated atomically, and in many cases, it would be more efficient to offload computation onto the parameter server, and thereby reduce the amount of network traffic.

Defining new training algorithms DistBelief workers follow a fixed execution pattern: read a batch of input data and the current parameter values, compute the loss function (a forward pass through the network), compute gradients for each of the parameter (a backward pass), and write the gradients back to the parameter server. This pattern works for training simple feed-forward neural networks, but fails for more advanced models, such as recurrent neural networks, which contain loops [19]; adversarial networks, in which two related networks are trained alternately [16]; and reinforcement learning models, where the loss function is computed by some agent in a separate system, such as a video game emulator [14]. Moreover, there are many other machine learning algorithms—such as expectation maximization, decision forest training, and latent Dirichlet

allocation—that do not fit the same mold as neural network training, but could also benefit from a common, well-optimized distributed runtime.

CONCLUSION

Face recognition has become a popular topic of research recently due to increases in demand for security as well as the rapid development of mobile devices. There are many applications which face recognition can be applied to such as access control, identity verification, security systems, surveillance systems, and social media networks. In this paper, we discussed a facial recognition system, its architecture and its applications.

REFERENCES

1. Bernd Heisele, Purdy Ho, Jane Wu, Tomaso Poggio Face recognition: component-based versus global approach *Computer Vision and Image Understanding*. February 2003.
2. Guodong Guo, Stan Z. Li, Kap Luk Chan Support vector machines for face recognition. *Image and Vision Computing*. January 2001.
3. Harihara Santosh Dadi, Gopala Krishna Mohan Pillutla Improved Face Recognition Rate Using HOG Features and SVM Classifier. *IOSR Journal of Electronics and Communications Engineering*, Vol. 11, Issue 4, pp 34-44, July 2016.
4. Navneet Dalal and Bill Triggs Histograms of Oriented Gradients for Human Detection. *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 2005.
5. O. Deniz, G. Bueno, J. Salido, F. De La Torre Face recognition using Histogram of Oriented Gradients *Pattern Recognition Letters*. 2011.
6. P. Jonathon Phillips Support Vector Machines Applied to Face Recognition. *National Institute of Standards and Technology*
7. Qiang Zhu, Shai Avidan, Mei-Chen Yeh, Kwang-Ting Cheng Fast Human Detection Using a Cascade of Histograms of Oriented Gradients. *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 2006.
8. Y. Atoum, Y. Liu, A. Jourabloo, and X. Liu. Face anti-spoofing using patch and depth-based cnns. In *IJCB*, pages 319–328. *IEEE*, 2017.
9. A. Bansal, C. Castillo, R. Ranjan, and R. Chellappa. The dos and don'ts for cnn-based face verification. *arXiv preprint arXiv:1705.07426*, 5, 2017.
10. A. Bansal, A. Nanduri, C. Castillo, R. Ranjan, and R. Chellappa. Umdfaces: An annotated face dataset for training deep networks. *arXiv preprint arXiv:1611.01484*, 2016.
11. J. Bao, D. Chen, F. Wen, H. Li, and G. Hua. Cvae-gan: fine grained image generation through asymmetric training. *arXiv preprint arXiv:1703.10155*, 2017.
12. J. Bao, D. Chen, F. Wen, H. Li, and G. Hua. Towards open-set identity preserving face synthesis. In *CVPR*, pages 6713–6722, 2018.
13. P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE trans. Pattern Anal. Mach. Intel.*, 19(7):711–720, 1997.

14. J. R. Beveridge, P. J. Phillips, D. S. Bolme, B. A. Draper, G. H. Givens, Y. M. Lui, M. N. Teli, H. Zhang, W. T. Scruggs, K. W. Bowyer, et al. The challenge of face recognition from digital point-and-shoot cameras. In BTAS, pages 1–8. IEEE, 2013.
15. S. Bianco. Large age-gap face verification by feature injection in deep networks. *Pattern Recognition Letters*, 90:36–42, 2017.
16. V. Blanz and T. Vetter. Face recognition based on fitting a 3d morphable model. *IEEE Transactions on pattern analysis and machine intelligence*, 25(9):1063–1074, 2003.
17. N. Bodla, J. Zheng, H. Xu, J.-C. Chen, C. Castillo, and R. Chellappa. Deep heterogeneous feature fusion for template-based face recognition. In WACV, pages 586–595. IEEE, 2017.
18. K. W. Bowyer, K. Chang, and P. Flynn. A survey of approaches and challenges in 3d and multi-modal 3d+ 2d face recognition. *Computer vision and image understanding*, 101(1):1–15, 2006.
19. K. Cao, Y. Rong, C. Li, X. Tang, and C. C. Loy. Pose-robust face recognition via deep residual equivariant mapping. arXiv preprint arXiv:1803.00839, 2018.
20. Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman. Vggface2: A dataset for recognising faces across pose and age. arXiv preprint arXiv:1710.08092, 2017.
21. Z. Cao, Q. Yin, X. Tang, and J. Sun. Face recognition with learning based descriptor. In CVPR, pages 2707–2714. IEEE, 2010.
22. T.-H. Chan, K. Jia, S. Gao, J. Lu, Z. Zeng, and Y. Ma. Pcanet: A simple deep learning baseline for image classification *IEEE Transactions on Image Processing*, 24(12):5017–5032, 2015.
23. B. Chen, W. Deng, and J. Du. Noisy softmax: improving the generalization ability of dcnn via postponing the early softmax saturation arXiv preprint arXiv: 1708.03769, 2017.
24. B.-C. Chen, C.-S. Chen, and W. H. Hsu. Cross-age reference coding for age-invariant face recognition and retrieval. In ECCV, pages 768–783. Springer, 2014.
25. D. Chen, X. Cao, L. Wang, F. Wen, and J. Sun. Bayesian face revisited: A joint formulation. In ECCV, pages 566–579. Springer, 2012.
26. D. Chen, X. Cao, F. Wen, and J. Sun. Blessing of dimensionality: High Dimensional feature and its efficient compression for face verification. In CVPR, pages 3025–3032, 2013.