Unsupervised Feature Descriptors Based Facial Tracking over Distributed Geospatial Subspaces

Shubham Dokania, Ayush Chopra(✉), Feroz Ahmad, S. Indu, and Santanu Chaudhury

Central Electronics Engineering Research Institute, Delhi Technological University, Pilani, India
shubham.k.dokania@gmail.com,
{ayushchopra_2k14,ferozahmad_2k14,}@dtu.ac.in, s.indu@dce.ac.in,
schaudhury@gmail.com

Abstract. Object Tracking has primarily been characterized as the study of object motion trajectory over constraint subspaces under attempts to mimic human efficiency. However, the trend of monotonically increasing applicability and integrated relevance over distributed commercial frontiers necessitates that scalability be addressed. The present work proposes a system for fast large scale facial tracking over distributed systems beyond individual human capabilities leveraging the computational prowess of large scale processing engines such as Apache Spark. The system is pivoted on an interval based approach for receiving the input feed streams, which is followed by a deep encoder-decoder network for generation of robust environment invariant feature encoding. The system performance is analyzed while functionally varying various pipeline components, to highlight the robustness of the vector representations and near real-time processing performance.

Keywords: Distributed facial tracking · Auto-encoders · Spark streaming

1 Introduction

Recently, visual representation and tracking has been subject to motivated research owing to increased relevance and interoperability with innumerable application domains such as criminal tracking, object tagging [3] etc. Efficient tracking requires learning of good feature representations that exhibit discriminative ability as well as robustness to data variance. Consequently, voluminous literatures have been produced and feature extraction methodologies have evolved significantly. These have largely been holistic or patch based [8]. Advancements in localized vectorization for generation of feature maps forms the basis of recent progress.

Auto-encoders [1] produce a non-linear representation which, unlike that of PCA or ICA, can be stacked to yield deeper levels of representation. More

B.U. Shankar et al. (Eds.): PReMI 2017, LNCS 10597, pp. 196–202, 2017.
https://doi.org/10.1007/978-3-319-69900-4_25

abstract features [5] can be perceived at deeper levels, enhancing the discriminative power of the feature descriptor. Facial features are subject to variance due to pose problems, background clutter, illumination variations. Using an implicit algorithm for capturing geometric information encoded into the descriptors, the issue of pose problem and misalignment can be tackled [2]. Simple elastic and partial metric proposed by Gang can also handle pose change and clutter backgrounds [4].

Object tracking has largely been characterised and defined as the problem of estimating the trajectory of a moving object [9] over constrained subspace. Several near real-time systems such as A Real-time face tracker [11], Pfinder [10], patch flow based [9] have been researched and reported with attempts to achieve human like accuracy in effortlessly tracking objects of interest. Eigenfaces, obtained by performing PCA on a set of faces are commonly used [11] to identify faces. However, increasing demands of real life applications such as vehicle navigation, traffic monitoring and surveillance, search and rescue operations to name a few imply that flexibility be exercised to include tracking that may require optimally fast and efficient search over large geographical subspaces that is beyond individual human capabilities involving the use of large scale distributed datasets.

2 Problem Formulation

In the present work, we propose a near real-time system employing an interval based programming approach for nodal tracking over distributed live streams or databases using a non-parametric supervised classification technique. The present work simulates the proposed approach with facial tracking over unconstrained geo-spatial subspaces owing to enhanced relative generality and reliability, while stating that similar work shall be extended to other vision based applications with relative ease. The system we propose seeks to leverage the particular computational prowess of large scale processing engines in applications involving reuse of working set across parallel operations [12] while assuring fault tolerance, consistency and seamless integration with batch processing, all which are critical considerables for scalable and reliable execution.

3 Proposed System

The proposed system is used to achieve near real-time, efficient tracking of individuals over large geographical subspaces. The system constituents can be largely characterised as A Master Node, Worker Nodes, Camera Nodes and Request Tracking Node. A high level overview of the system architecture has been depicted in Fig. 1. The stages in the pipeline are

- Feature Generation
- Facial Identification

We utilize spark streaming, from the Apache Spark stack, as a core component for streaming computation tasks. The system defines multiple input streams obtained by receiving records feed directly from client or by interval defined loading from external data storage file systems, where it may be placed by a log collection system [13]. The feature generation stage is represented in Fig. 2. Facial Extraction and Component Definition is done using a region based Single Shot Detector [6]. The fast processing speed, 30–35 fps, and efficiency at various aspect scales enabled segregation of the components. The feature vectors are generated by passing the concatenated component vectors through the CAE. The feature vectors generated as a result of the computation associated with the previous phase are stored in the spark database distributed optimally over the worker nodes by the master. The database is characterised by 2 major tables T_1 and T_2. T_1 contains tuples of unique human faces identified or obtained from organizational records with attributes: *human id*, $feature\ vector$. T_2 characterises the occurrences of the human faces at different nodes as unique tuples, having attributes as: *auto id*, *human id*, *node id*, *timestamp*. This facilitates querying over the table using a nearest neighbor approach to identify the individual.

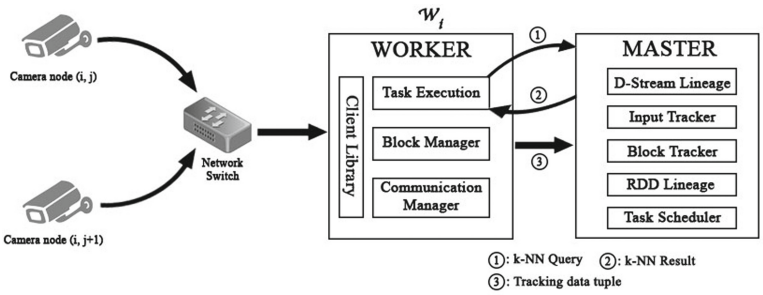

Fig. 1. Overall System Architecture - The j^{th} camera node with unique IP on the local network, of i^{th} block transmits the live feed to the block worker via network switch for the block. Apache Worker kept at control room of i^{th} block receives live feeds from all cameras are fed to Apache Spark Worker. For every face detected in the frame, the computation flow takes place as discussed in Fig. 2.

4 Experiments and Results

In the following section, the proposed system performance is analyzed as a function of various deterministic parameters, by simulating under approximate settings. For the simulation, we hand-picked images from standard facial benchmarks - Labeled Faces in the Wild-a (LFW-a) & IARPA Janus Benchmark A (IJB-A). The images were fed into the client library following an interval based

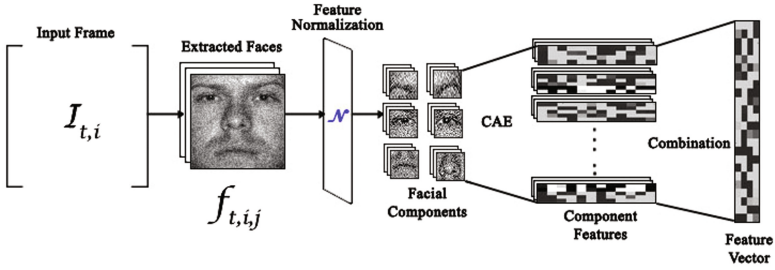

Fig. 2. Generation of Feature Vector - A representation of the generation of feature vectors for the j^{th} extracted face $f(t, i, j)$ from the i^{th} time variant input frame $I(t, i)$ at the time t. The output of normalization procedure $N(f(t, i, j))$ is operated upon to extract components for generation of robust descriptors using CAE.

approach to facilitate micro-batch generation for processing on the cluster workers. We used a multi-node Apache Spark cluster, with nodal configuration 2.6 GHz Intel i7 second generation processors and 8 GB RAM. Near real-time tracking was achieved employing k-Nearest Neighbours (k-NN) algorithm over the distributed database generated over the multi-node by integrating with MLlib [7]: the machine learning library supported by Apache Spark in simulated video feed environments while varying the parameters to obtain appropriate results as explained next.

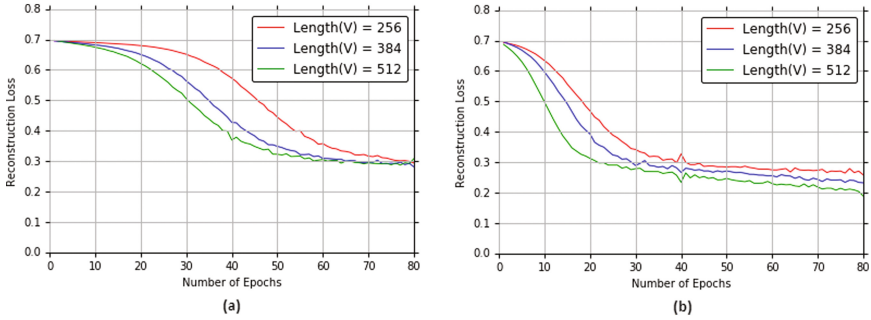

Fig. 3. The reconstruction loss is plotted against the number of training epochs for varying lengths of feature vectors for the auto-encoder trained with (a) adadelta (b) adam optimizer.

4.1 Reconstruction Loss Analyzed on the Variation of Feature Length

From Fig. 3(a) and (b) we see that applying Adam optimizer produces better results in the present problem setting. Adam, if compared to Adadelta, is seen to

perform better because, in addition to saving a functional average of past squared gradients, it also stores the functional average of past gradients. Decreasing reconstruction loss depicts the increasing descriptive efficiency and invariance of the feature descriptors by a contractive auto-encoder.

4.2 Query Processing Time Analyzed as a Measure of the Facial Components

As evident from Fig. 4, an increase in number of facial components extracted is associated with a marked increase in the processing time for 300 queries in case of a single worker node scenario. Graphical representation for 2 and 3 node scenario exhibit a slightly leaner growth rate of the processing time as a functional measure of the number of components, but these may have significant considerations in practical settings.

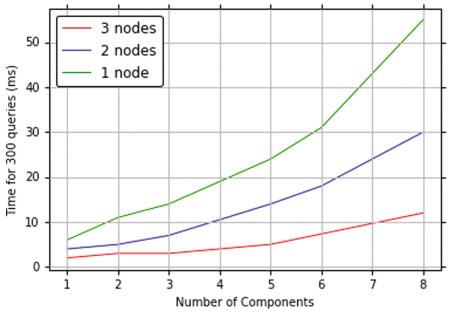

Fig. 4. Processing time for 300 queries, measured in ms, is plotted against differing number of components for 1, 2 and 3 worker nodes.

4.3 Query Processing Time with Varying Worker Nodes

Increasing range to greater geographical subspaces implies the need for enhanced computational processing power, which can be achieved by more worker nodes. The master ensures locality of computation on worker nodes, however in specific scenarios it may shift records between the worker nodes to ensure more equitable load balancing. Keeping the total record penetration of the database constant, a decrease in query processing time is observed on increasing the number of slave worker nodes in Fig. 5. Further this rate of growth is witnessed to depict flattening tendencies as number of worker nodes are further added to the cluster.

4.4 Performance of Proposed System

The number of facial components impacts the dimensional complexity of the feature descriptors and correspondingly their descriptive power. The performance

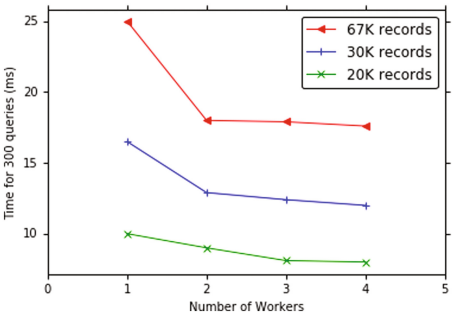

Fig. 5. The processing time for 300 queries on the database generated by the system is plotted against the number of workers for varying number of records. The system exhibits near real-time performance taking time less than 20 ms for 300 queries.

of the system in terms of True Acceptance Rate at False Acceptance Rate = 0.01 and Recognition Rate at Rank-10 is presented in Table 1. Query processing is analyzed for different number of facial components (hence, varying feature length) under consideration.

Table 1. Performance of the proposed system. TAR is reported at FAR = 0.01 for verification, Recognition Rate at Rank-10 is reported for identification.

Feature length	TAR	Rank-10
256 bits (4 components)	0.587	0.608
384 bits (6 components)	0.651	0.694
512 bits (8 components)	0.718	0.732

5 Conclusions

Recent attempts in tracking have sought to implement paradigm like human vision for analysing motion trajectory. No work to our knowledge has performed tracking at similar scale in real-time, rather major works have focussed on subspaces small as frames of single camera feed. The present work tries to provide a solution into the particularly untapped and critical task involving large subspaces that is beyond individual human capabilities. The work aimed to provide a baseline to propel further penetration in the domain. The proposed system is a multilevel hierarchical model based on Spark streaming to feed input streams that uses a deep contractive encoder-decoder model to generate robust vector encoding and a penultimate classifier for searching over the distributed database prior to final path determination. The processing and path retrieval depicted near real-time performance that provides encouragement for applicability into varied commercial settings.

References

1. Bengio, Y., Lamblin, P., Popovici, D., Larochelle, H., et al.: Greedy layer-wise training of deep networks. Adv. Neural Inf. Process. Syst. **19**, 153 (2007)
2. Cao, Z., Yin, Q., Tang, X., Sun, J.: Face recognition with learning-based descriptor. In: 2010 IEEE Conference on CVPR, pp. 2707–2714. IEEE (2010)
3. Cui, J., Wen, F., Xiao, R., Tian, Y., Tang, X.: EasyAlbum: an interactive photo annotation system based on face clustering and re-ranking. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 367–376. ACM (2007)
4. Hua, G., Akbarzadeh, A.: A robust elastic and partial matching metric for face recognition. In: 2009 IEEE 12th ICCV, pp. 2082–2089. IEEE (2009)
5. Lee, H., Grosse, R., Ranganath, R., Ng, A.Y.: Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In: Proceedings of the 26th Annual ICML, pp. 609–616. ACM (2009)
6. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., Berg, A.C.: SSD: single shot MultiBox detector. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9905, pp. 21–37. Springer, Cham (2016). doi:10.1007/978-3-319-46448-0_2
7. Meng, X., Bradley, J.K., Yavuz, B., Sparks, E.R., Venkataraman, S., Liu, D., Freeman, J., Tsai, D.B., Amde, M., Owen, S., Xin, D., Xin, R., Franklin, M.J., Zadeh, R., Zaharia, M., Talwalkar, A.: Mllib: machine learning in apache spark. CoRR abs/1505.06807 (2015). http://arxiv.org/abs/1505.06807
8. Mishra, R., Kumar, P., Chaudhury, S., Indu, S.: Monitoring a large surveillance space through distributed face matching. In: 2013 Fourth National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG), pp. 1–5. IEEE (2013)
9. Prabhu, N., Ramakanth, S.A., Babu, R.V.: Patch flow based visual object tracking. In: Proceedings of the 2014 ICVGIP, p. 86. ACM (2014)
10. Wren, C.R., Azarbayejani, A.J., Darrell, T.J., Pentland, A.P.: Pfinder: real-time tracking of the human body. In: Photonics East 1995, pp. 89–98. International Society for Optics and Photonics (1996)
11. Yang, J., Waibel, A.: A real-time face tracker. In: 1996 Proceedings of 3rd IEEE Workshop on Applications of Computer Vision, WACV 1996, pp. 142–147. IEEE (1996)
12. Zaharia, M., Chowdhury, M., Franklin, M.J., Shenker, S., Stoica, I.: Spark: cluster computing with working sets. In: Proceedings of the 2nd USENIX Conference on Hot Topics in Cloud Computing, HotCloud 2010, pp. 10–10. USENIX Association, Berkeley (2010). http://dl.acm.org/citation.cfm?id=1863103.1863113
13. Zaharia, M., Das, T., Li, H., Hunter, T., Shenker, S., Stoica, I.: Discretized streams: fault-tolerant streaming computation at scale. In: Proceedings of the Twenty-Fourth ACM Symposium on Operating Systems Principles, pp. 423–438. ACM (2013)