Facial Recognition in Collaborative Learning Videos[⋆]

Phuong Tran[1], Marios Pattichis[1], Sylvia Celedón-Pattichis[2], and Carlos LópezLeiva[2]

[1] Dept. of Electrical and Computer Engineering
University of New Mexico, Albuquerque NM 87106, USA
[2] Dept. of Language, Literacy, and Sociocultural Studies
University of New Mexico, United States.

Abstract. Face recognition in collaborative learning videos presents many challenges. In collaborative learning videos, students sit around a typical table at different positions to the recording camera, come and go, move around, get partially or fully occluded. Furthermore, the videos tend to be very long, requiring the development of fast and accurate methods.

We develop a dynamic system of recognizing participants in collaborative learning systems. We address occlusion and recognition failures by using past information about the face detection history. We address the need for detecting faces from different poses and the need for speed by associating each participant with a collection of prototype faces computed through sampling or K-means clustering. Our results show that the proposed system is proven to be very fast and accurate. We also compare our system against a baseline system that uses InsightFace [2] and the original training video segments. We achieved an average accuracy of 86.2% compared to 70.8% for the baseline system. On average, our recognition rate was 28.1 times faster than the baseline system.

Keywords: Human front-face detection and recognition · video analysis.

1 Introduction

We study the problem of face recognition in collaborative learning environments. Our goal is to develop fast and accurate methods that can be used to quantify student participation as measured by their presence in their learning groups.

Fig. 1 represents an example of a collaborative learning group. A collaborative learning group is represented by the group of students closest to the camera. Background groups are not considered part of the collaborative group that we are analyzing. There is a possibility that students or facilitators move between groups. Thus, we need to recognize the current members of our group from a larger group of students.

A fundamental challenge of our dataset is that face recognition needs to work at a large variety of poses. As long as a sufficiently small part of the face is visible, we need to identify the student. As an example, in Fig. 1, the student on the front right with a white hoodie has his face mostly blocked. Furthermore, students may disappear or
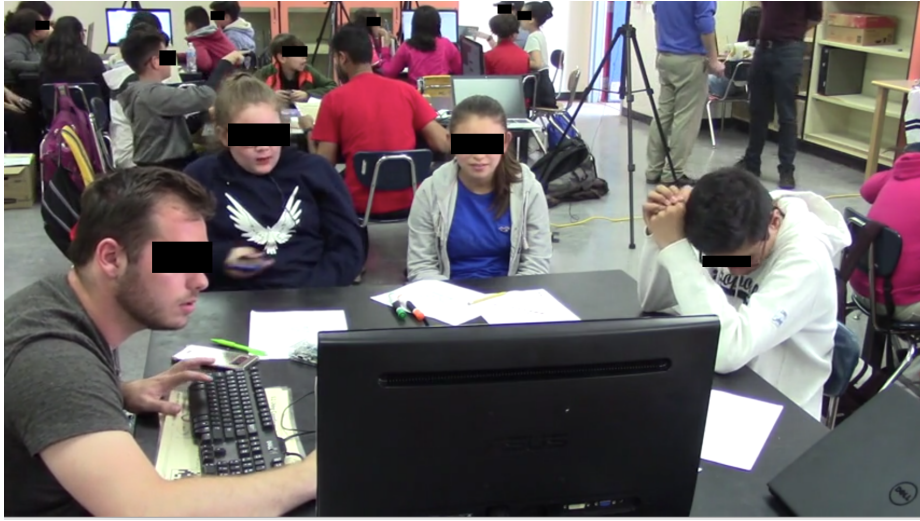

Fig. 1. Example of a collaborative learning group. We seek to recognize the students who are closer to the camera while ignoring students from all other groups.

reappear because the camera moves, or the students take a break, or because they have to leave the group. Hence, there are significant video properties to our problem that are not present in standard face recognition systems.

As part of a collaborative project between engineering and education, our goal is to assist educational researchers in analyzing how students who join the program learn and facilitate the learning of other students. The problem of identifying who is crucial for assessing student participation in the project. Furthermore, the developed methods need to be fast. Eventually, we will need to apply our methods to approximately one thousand hours of videos that we need to analyze.

The standard datasets for face recognition use a database of front-facing images. The Labeled Faces in the Wild (LFW) dataset [3] contains more than 13,000 face images with various poses, ages, and expressions. The Youtube Face (YTF) dataset [5] contains around 3,500 videos with an average range of 181 frames/video from 1,600 different people. The InsightFace system [1] developed the use of Additive Angular Margin Loss for Deep Face Recognition (ArcFace) on a large-scale image database with trillions of pairs and a large-scale video dataset, and tested on multiple datasets with different loss function models (ArcFace, Softmax, CosFace,..). InsightFace gave the best accuracies on LFW and YTF with 99.83% 98.02%. We adopt InsightFace as our baseline face recognition system because of their state-of-the-art performance.

We also provide a summary of video analysis methods that were developed specifically by our group for analyzing collaborative learning videos.

In [4], the authors introduced methods for detecting writing, typing, and talking activities using motion vectors and deep learning. In [6], the authors developed methods to detect where participants were looking at. In [7], the authors demonstrated that FM

images with low-complexity neural networks can provide face detection results that can only be achieved with much more complex deep learning systems.

We provide a summary of the contributions of the current paper. First, we introduce the use of a collection of face prototypes for recognizing faces from different angles. Second, we apply multi-objective optimization to study the inter-dependency between recognition rate, the number of face prototypes, and recognition accuracy. Along the Pareto front of optimal combinations, we select an optimal number of face prototypes that provides for a fast approach without sacrificing recognition accuracy. Third, we use the past recognition history to deal with occlusions and, hence, support consistent recognition throughout the video. Compared to InsightFace [2], the proposed system provides for significantly faster recognition rates and higher accuracy.

We summarize the rest of the paper into three additional sections. In section 2, we provide a summary of our methodology. We then provide results in section 3 and concluding remarks in section 4.

2   Methods

We present a block diagram of the overall system in Fig. 2. Our video recognition system requires a set of face prototypes associated with each participant. The video face recognition algorithm detects the faces in the input video and computes minimum distances to the face prototypes to identify each participant. To handle occlusion issues, the system uses past detection history.

2.1   Computation of Face Prototypes

We use two different methods to compute the face prototypes. First, we consider the use of K-means clustering to select a limited number of face prototypes from the training videos. Second, we consider a baseline approach where we use sparse sampling of the training videos to define the face prototypes. To achieve sparsity, we use one sample image per second of video. For our second approach, we used a video from Group D from urban cohort 2 level 1 (C2L1, Video 4) and Group E from C3L1 (Video 5), whereas the first approach tested on Group C from urban C1L1 (Video 1). Each sampled face is aligned and resized to 112×112.

We summarize the K-means approach in Fig. 2. We use K-means to cluster similar frames that appear when a student does not move very much. Hence, we expect that the centroids that result from the K-means algorithms would represent a small number of diverse face poses for each participant. To avoid unrealistic centroid images, we compute our face prototypes using the images closest to our cluster centroids. After finding a prototype image that is closest to the mean, we align and resize each prototype image to 112×112.

We use multi-objective optimization to determine the minimum number of face prototypes without sacrificing classification accuracy. We present our results for a representative video from group C, level 1, in Fig. 3. For the optimization, we use a log-based search by varying the number of face prototypes from 2^0 to 2^{11}. For K-means clustering, we see that the accuracy peaks at 79.8% for 1024 face prototypes, with a

---

Algorithm 2: Video Faces Recognition

Input:
   `video`: unlabeled video
   `facePrototypes`: list of images associated with each student pseudonym
Output:
   `vidResult`: store unique student identifiers and face locations along with face's
landmarks per frame
Local Variables:
   `ActiveSet` and `InactiveSet` store unique student identifiers, face locations,
`totalAppearances`,
     `totalFramesProcessed`, `continuousAppearances` & distance.

while frame `f` in initial part of video ▷Detect and Recognize all faces initial duration
   Detect faces in `f`
   Recognize faces in `f` using minimum distance to `facePrototypes`
   Update `vidResult`

`ActiveSet` **= [];** `InactiveSet` **= [];** ▷Initialization
while `face` in all recognized faces
   if `totalAppearances(face)/totalFramesProcessed(face)` $>= 50\%$
     Add `face` to ActiveSet
   else:
     Add `face` to InactiveSet

for frame `f` in rest of `video`
   Detect faces in `f`
   if minor movement in detected face **then**   ▷Reuse face
     Reuse face from `ActiveSet`
     Update `ActiveSet`, `vidResult` with detected faces
   else if detected face found in `InactiveSet` **then**   ▷Update face
     Update `InactiveSet` with detected face
     if `totalAppearances(face)/totalFramesProcessed(face)` $>= 50\%$
       Move face to `ActiveSet`
       Remove face from `InactiveSet`
       Update `ActiveSet`, `vidResult` with detected faces
   else   ▷Possible new face
     Recognize face in f using minimum distance to `facePrototypes`
     Update `InactiveSet` with detected faces

  for face in `ActiveSet`
    if face not found in all detected faces **then**
      if `continuousAppearances` $>=$ `minAppearances` **then**
▷Occluded face
        Add face to `ActiveSet`, `vidResult`
       if `continuousAppearances` $<$ `minAppearances` **then**
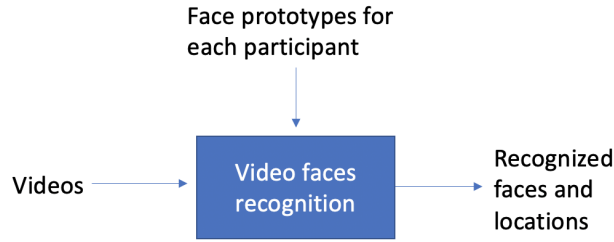▷Disappearing face
        Remove face from `ActiveSet`
        Add face to `InactiveSet`

  for all labels found in `f`   ▷Consistent assignment check
    if same label exists
     Set label with larger distance to Unknown

Face prototypes for
each participant

Videos → Video faces recognition → Recognized faces and locations

---

Algorithm 1: Computing Face prototypes using K-means

Input:
   video clips associated with each participant.
Output:
   facePrototypes associated with each participant.
for each participant
  Apply K-means clustering
  Select cluster means
  Find nearest images from cluster centroids
  Align faces to 112x112
end

Fig. 2. Block diagram for recognizing faces from videos. Each face is associated with a collection of face prototypes. The method computed face prototypes using video sampling or K-means clustering. K-means clustering is given here.

recognition rate of 4.8 seconds per frame. In this example, we only used one video to validate our approach. We expand the validation set to more videos as described in [8]. This process run on a personal Macbook Pro that ran Mac-OS with 2.3GHz, 4-core, Intel i5 processors.

2.2 Video Faces Recognizer

We present the algorithm for video face recognition in Algorithm 2. The input is an unlabeled video and the `facePrototypes` that provides a list of images associated with each participant. The detected faces for each video frame are returned in `vidResult`. To address occlusion, the algorithm maintains the face recognition history in `ActiveSet` and `InactiveSet`.

First, for the first two seconds of the videos, we detect all participants in each video frame using MTCNN [9]. For each detected face, MTCNN computes five landmark points: two for the eyes, one for the nose, and two for the mouth corners. The face detector uses a minimum area requirement to reject faces that belong to another group. Thus, we reject smaller face detections that are part of another group or other false positives because they appear smaller in the video. Second, we classify each detected face by selecting the participant that gives minimum distance to their associated prototypes stored in `facePrototypes`.

We use the initial face recognition results to initialize `ActiveSet` and `InactiveSet`. The faces that have been recognized in more than half of the frames are stored in the `ActiveSet`. The rest of them are stored in the `InactiveSet`. For each face detection, we use a dictionary to store: a pseudonym, location information, `totalAppearances` that stores the total number of frames where this face was detected, and `totalFramesProcessed` which represents the total number of frames processed since this face appeared for the first time. Hence, we use the `ActiveSet` to hold faces that appear consistently whereas `InactiveSet` contains faces that are still in doubt.

When a recognized face enters the `ActiveSet`, it gets a maximum value of 10 for its corresponding `continuousAppearances`. When a previously recognized face is missing, `continuousAppearances` gets decremented by 1. When a face re-appears, `continuousAppearances` is incremented until it reaches 10 again. We also set `minAppearances` to 5 as the minimum requirement on the number of prior continuous appearances for addressing occlusion issues. Thus, for each face in the `ActiveSet` that is not being detected in any frame, if `continuousAppearances` $\geq$ `minAppearances`, we declare the face as occluded, we mark it as present, and update `vidResult`. Else, if `continuousAppearances` < `minAppearances`, we declare the face as disappearing, and move it to the `InactiveSet`.

We thus process the rest of the `video` based on the following four cases:
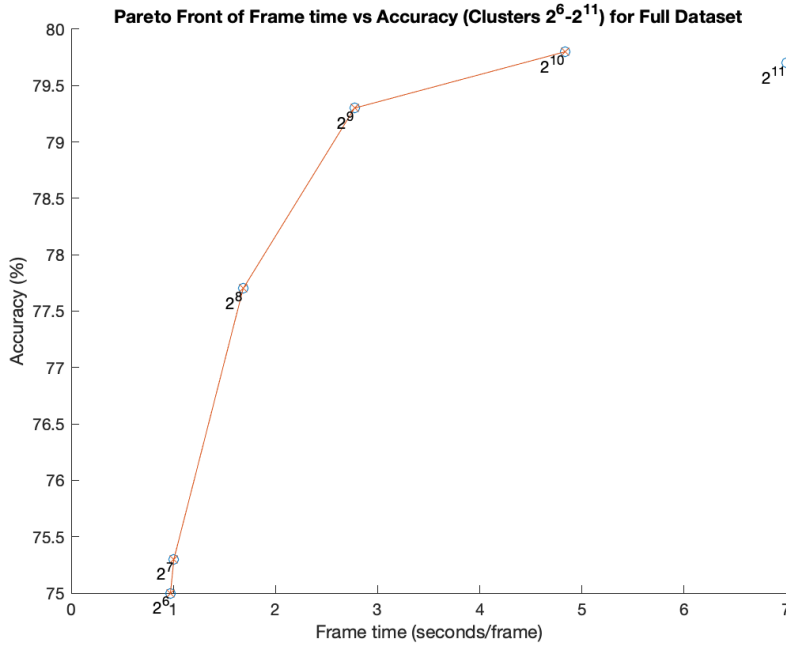

Fig. 3. Pareto Front on the K-mean Clustering Results

(i) If a newly detected face corresponds to a minor movement of a prior detection, we keep it in the `ActiveSet`. This approach leads to a significant speedup in our face recognition speed.

(ii) If a newly detected face is in the `InactiveSet`, we update the `InactiveSet` with the new detection, and look at the ratio of `totalAppearances`/`totalFramesProcessed` to determine if it needs to move to the `ActiveSet`. Otherwise, the face stays in `InactiveSet`.

(iii) If a newly detected face does not belong to either set, then recognize it and move it to the `ActiveSet`.

(iv) If a face that belongs to the `ActiveSet` no longer appears, we consider the case for occlusion and that the participant has left the frame. As described earlier, we check `continuousAppearances` to determine whether to declare the face occluded or not.

Lastly, we do not allow the assignment of the same label to two different faces in the same frame. In this case, the face that gives the minimum distance is declared the recognized face while the other(s) are declared Unknown.

3 Results

We sampled twenty-four participants from our video dataset (11 boys and 13 girls). For training, we used 80% of the data for fitting and 20% for validation using different video sessions. From our video sessions, we randomly select short video clips of 5 to 10 seconds for training our video face recognizer. Overall, we use more than 200,000 video frames from 21 different video sessions. We have an average of about 10,000 image examples per participant. Furthermore, as the program lasted three years, the testing dataset used later on videos (e.g., later cohorts and different levels). For the testing dataset, we used seven video clips with a duration of 10 to 60 seconds. However, we do assume that we have trained for all of the students within the collaborative group. For reporting execution times, we use a Linux system with an Intel(R) Xeon(R) Gold 5118 CPU running @ 2.30GHz with 16GB Memory and Nvidia Quadro RTX 5000 GPU with 3072 Cuda cores.

We present face recognition accuracy results in Table 1 using said system. For InsightFace, we use all of the training video frames as face prototypes. From the results, it is clear that the proposed method is far more accurate than the baseline method. The difference in accuracy ranged from as low as 2% to as large as 25%. Out of 19 participants in these five videos, our method achieved higher or same accuracy in 17 cases. Overall, our method achieved 86.2% compared to 70.8% for the baseline method.

We present face recognition examples in Fig. 4. Figs. 4(a) and 4(b) show results from the same video frame of Video 2. The baseline method recognized Javier67P (lower right in (a)) and Kenneth1P (white shirt with glasses in (a)) as Unknown, whereas our approach correctly identified all four participants. A second example for video 5 is presented in Figs. 4(b) and (e). The baseline method identified all three people correctly. However, the baseline method also detected and incorrectly claimed recognition of background participants that we did not train. Our proposed method used projection and small-area elimination to reject this false-positive recognition. A third example for

Table 1. Accuracy for Facial Recognition. Each video represents a different group session segment.

Video	Duration	Person Label	Ours	Insightface
1 (Face prototypes using K-means)	10 seconds	*Antone39W* *Jaime41W* *Larry40W* *Ernesto38W*	**36.5%** **86.7%** **99.3%** **96.5%**	**36.5%** 84.2% 98.3% 95.3%
		Average	**79.8%**	78.6%
2 (Face prototypes using Sampling)	10 seconds	*Chaitanya* *Kenneth1P* *Jesus69P* *Javier67P*	**95.3%** **91%** **100%** **100%**	80.3% 83.1% **100%** 69.1%
		Average	**96.5%**	83.1%
3 (Face prototypes using Sampling)	60 seconds	*Chaitanya* *Kenneth1P* *Jesus69P* *Javier67P*	**80.0%** **98.3%** **99.3%** **80.6%**	56.1% 61.5% **96.3%** 39.0%
		Average	**89.5%**	63.2%
5 (Face prototypes using Sampling)	10 seconds	*Melly77W* *Marisol112W* *Cristie123W* *Phuong*	**96.0%** **84.0%** 8.67% **77.4%**	59.7% 60.5% **27.3%** 21.4%
		Average	**66.5%**	42.2%
5 (Face prototypes using Sampling)	60 seconds	*Alvaro70P* *Donna112P* *Sophia111P*	**96.4%** **100%** 99.5%	60.8% 99.8% **99.9%**
		Average	**98.6%**	86.8%
		Overall Average	**86.2%**	70.8%

video 4 is shown in Figs. 4(c) and (f). The baseline method only succeeded in recognizing Melly77W (pink sweater) and wrongly recognized Cristie123W (lower right) as Phuong, who is actually on the far left wearing glasses. Our method used history information to address the partial occlusion issue and correctly recognized Phuong who is in the far left of Fig. 4(f). Furthermore, our method rejected the wrong assignment of Phuong because it does not allow the assignment of the same identifier to two different faces. Instead, the wrong assignment was re-assigned to Unknown. A fourth example of our method is shown in Fig. 4(g). In Fig. 4(g), we can see that our method works in occlusion cases. Herminio10P (dark blue polo, right) and Guillermo72P (blue T-shirt) were correctly recognized even though their faces were partial. We also present challenges in Figs. 4(h) and (i). In Fig. 4(h), Antone39W did not get recognized because

Table 2. Recognition time for facial recognition. Each video represents a different group session segment.

Video	Duration	GT Faces	Insightface (seconds/frame)	Ours (seconds/frame)	Speedup factor
1	10	4	9.91	2.8	3.5x
2	10	4	9.96	0.8	12.5x
3	60	4	15.8	0.3	52.7x
4	10	4	10.1	0.9	11.2x
5	60	3	18.2	0.3	60.7x
		Average	12.8	**1.1**	**28.1x**

he had his back facing the camera. In Fig. 4(i), Kirk28P was not recognized due to significant changes in appearance through time.

We present speed performance comparisons in Table 2. The baseline method required 9.9 to 18.2 seconds/frame whereas our proposed method required 0.3 to 2.8 seconds/frame. On average, the proposed method was $28.1\times$ faster. Our speedups can be attributed to our use of a reduced number of face prototypes and the fact that we do not rerun the minimum distance classifier if there is little movement in the detected faces. For example, for 3 and 4, InsightFace took a very long time (more than ten seconds/frame) because it compared each participant against (almost) ten thousand images. For 5, in addition to comparisons to about ten thousand images for the main group, InsightFace also had to compare against faces from the background groups. In comparison, our approach rejected the need to recognize background groups by applying a minimum face size constraint.

4   Conclusion

The paper presented a method for video face recognition that is shown to be significantly faster and more accurate than the baseline method. The method introduced: (i) clustering methods to identify image clusters for recognizing faces from different poses, (ii) robust tracking with multi-frame processing for occlusions, and (iii) multi-objective optimization to reduce recognition time.

Compared to the baseline method, the final optimized method resulted in speedy recognition times with significant improvements in face recognition accuracy. Using face prototype with sampling, the proposed method achieved an accuracy of 86.2% compared to 70.8% for the baseline system, while running 28.1 times faster than the baseline. In future work, we want to extend our approach to 150 participants in about 1,000 hours of videos.
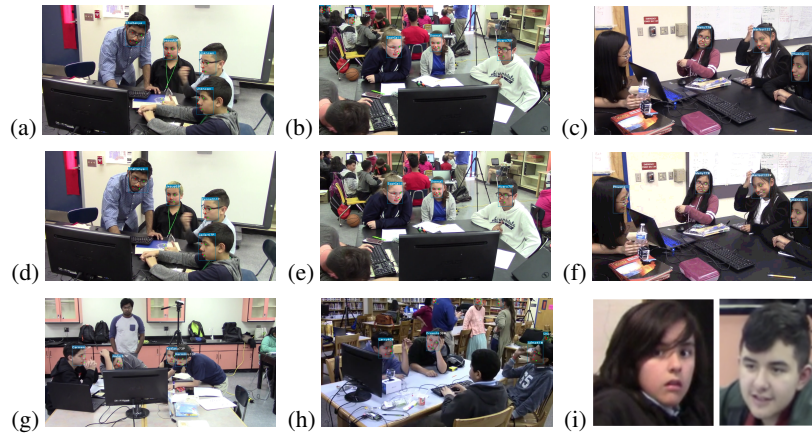
Fig. 4. Video face recognition results for three collaborative groups. The first row shows results from the use of InsightFace (baseline). The second row shows our results using the sampling method. In (g), we show successful detections despite occlusions. Results from the use of K-means clustering are shown in (h). Then, we show dramatic changes in appearance in (i).

References

1. Deng, J., Guo, J., Xue, N., Zafeiriou, S.: Arcface: Additive angular margin loss for deep face recognition (2019)
2. Guo, J., Deng, J., An, X., Yu, J.: Deepinsight/insightface: State-of-the-art 2d and 3d face analysis project (Jul 2021), https://github.com/deepinsight/insightface
3. Huang, G.B., Ramesh, M., Berg, T., Learned-Miller, E.: Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Tech. Rep. 07-49, University of Massachusetts, Amherst (October 2007)
4. Jacoby, A.R., Pattichis, M.S., Celedón-Pattichis, S., LópezLeiva, C.: Context-sensitive human activity classification in collaborative learning environments. In: 2018 IEEE Southwest Symposium on Image Analysis and Interpretation (SSIAI). pp. 1–4 (2018). https://doi.org/10.1109/SSIAI.2018.8470331
5. Learned-Miller, G.B.H.E.: Labeled faces in the wild: Updates and new reporting procedures. Tech. Rep. UM-CS-2014-003, University of Massachusetts, Amherst (May 2014)
6. Shi, W., Pattichis, M.S., Celedón-Pattichis, S., LópezLeiva, C.: Robust head detection in collaborative learning environments using am-fm representations. In: 2018 IEEE Southwest Symposium on Image Analysis and Interpretation (SSIAI). pp. 1–4 (2018). https://doi.org/10.1109/SSIAI.2018.8470355
7. Tapia, L.S., Pattichis, M.S., Celedón-Pattichis, S., Leiva, C.L.: The importance of the instantaneous phase for face detection using simple convolutional neural networks. In: IEEE Southwest Symposium on Image Analysis and Interpretation, SSIAI 2020, Albuquerque, NM, USA, March 29-31, 2020. pp. 1–4 (2020). https://doi.org/10.1109/SSIAI49293.2020.9094589
8. Tran, P.: Fast Video-based Face Recognition in Collaborative Learning Environments. Master's thesis, University of New Mexico (12 2021)
9. Zhang, K., Zhang, Z., Li, Z., Qiao, Y.: Joint face detection and alignment using multi-task cascaded convolutional networks. CoRR **abs/1604.02878** (2016), http://arxiv.org/abs/1604.02878