Supplementary Materials

Andreu Girbau, Tetsuro Kobayashi, Benjamin Renoust, Yusuke Matsui, and Shin'ichi Satoh

SI1. Robustness Check on Voting Threshold

In our system pipeline, the most commonly voted identity for the tracklet T should have at least 70% of the votes to be labeled as a specific individual. We checked the robustness of the performance of our system by varying this threshold. In the following robustness check, we varied the thresholds from 10% to 99% and examined the changes in Precision, Recall, and F1 scores. The results are presented in Figures S1 to S3.

Figure S1. Robustness check of Precision scores across voting thresholds

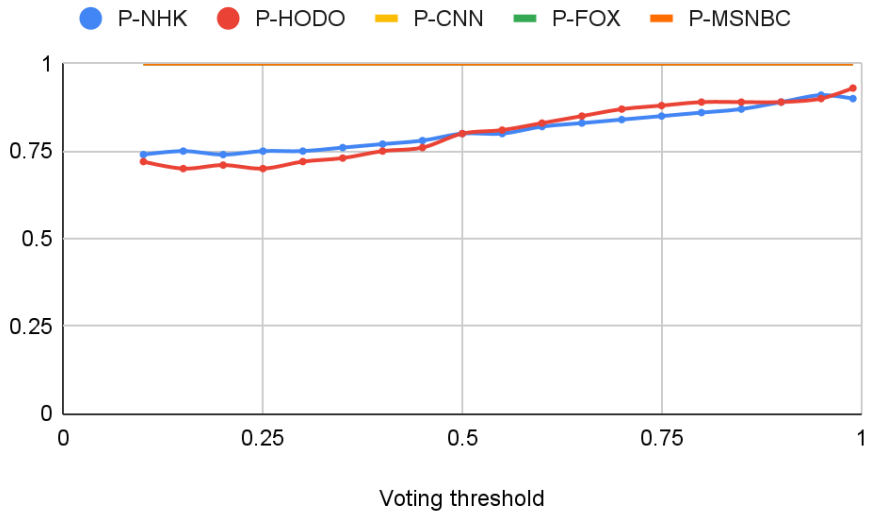


Figure S2. Robustness check of Recall scores across voting thresholds

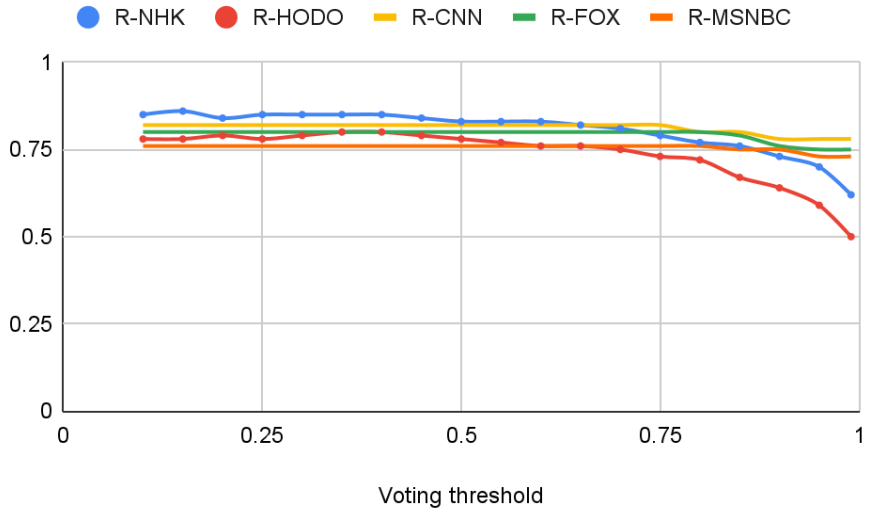
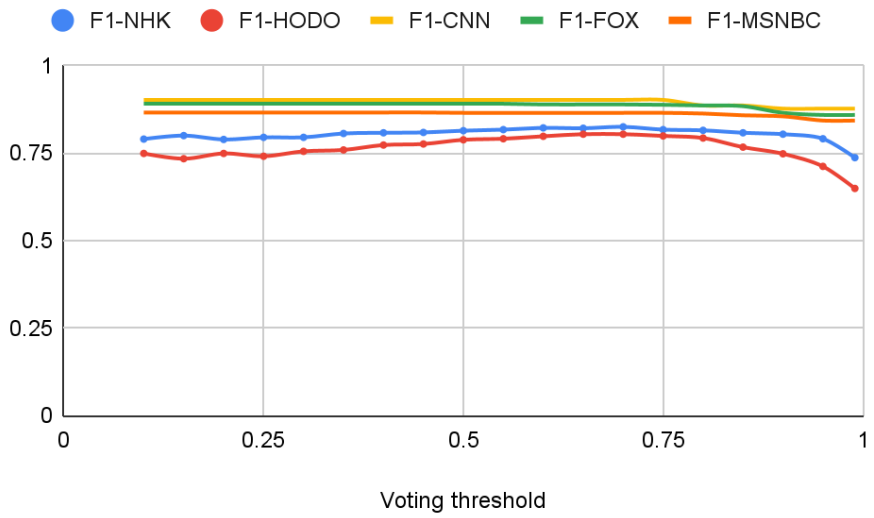


Figure S3. Robustness check of F1 scores across voting thresholds



As Figure S3 shows, the F1 score is fairly robust to variance in voting thresholds; only when the threshold is set very high, close to 1, is performance slightly degraded for the Japanese data. This trend is also true for the value of Recall. For Precision, the value for the Japanese data increases as the voting threshold increases, while the US data always show a value of 1. As noted in Section 4.2 of the paper, this is because the US data are more lenient in terms of the annotation and evaluation criteria, specifically in evaluations of false positives.

Overall, Figures S1 to S3 indicate that the proposed voting approach has a tradeoff between Precision and Recall, with the maximum F1 scores found in the voting thresholds of 0.7 for NHK, and 0.65 for HODO Station. Therefore, it is reasonable to set the voting threshold at 70%. With regards to the US data, adjusting the threshold only affects the Recall metric because the evaluation criterion for the US dataset is very lenient toward false positives, with maximum F1 scores from 0.1–0.7.

Table S1. Performance with different voting thresholds of US TV data

		CNN			FOX			MSNBC	
Threshold	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
0.1	1	0.82	**0.902**	1	0.8	**0.891**	1	0.76	**0.866**
0.15	1	0.82	**0.902**	1	0.8	**0.891**	1	0.76	**0.866**
0.2	1	0.82	**0.902**	1	0.8	**0.891**	1	0.76	**0.866**
0.25	1	0.82	**0.902**	1	0.8	**0.891**	1	0.76	**0.866**
0.3	1	0.82	**0.902**	1	0.8	**0.891**	1	0.76	**0.866**
0.35	1	0.82	**0.902**	1	0.8	**0.891**	1	0.76	**0.866**
0.4	1	0.82	**0.902**	1	0.8	**0.891**	1	0.76	**0.866**

0.45		1	0.82	**0.902**		1	0.8	**0.891**		1	0.76	**0.866**
0.5		1	0.82	**0.902**		1	0.8	**0.891**		1	0.76	0.865
0.55		1	0.82	**0.902**		1	0.8	**0.891**		1	0.76	0.865
0.6		1	0.82	**0.902**		1	0.8	0.889		1	0.76	0.865
0.65		1	0.82	**0.902**		1	0.8	0.889		1	0.76	0.865
0.7		1	0.82	**0.902**		1	0.8	0.889		1	0.76	0.865
0.75		1	0.82	**0.902**		1	0.8	0.888		1	0.76	0.865
0.8		1	0.8	0.886		1	0.8	0.886		1	0.76	0.863
0.85		1	0.8	0.886		1	0.79	0.884		1	0.75	0.858
0.9		1	0.78	0.877		1	0.76	0.865		1	0.75	0.855
0.95		1	0.78	0.877		1	0.75	0.859		1	0.73	0.843
0.99		1	0.78	0.877		1	0.75	0.859		1	0.73	0.843

Table S2. Performance with different voting thresholds for Japanese TV data

			NHK				**HODO**	
Threshold		**P**	**R**	**F1**		**P**	**R**	**F1**
0.1		0.74	0.85	0.79		0.72	0.78	0.749
0.15		0.75	0.86	0.8		0.7	0.78	0.734
0.2		0.74	0.84	0.789		0.71	0.79	0.749
0.25		0.75	0.85	0.795		0.7	0.78	0.741
0.3		0.75	0.85	0.795		0.72	0.79	0.755
0.35		0.76	0.85	0.806		0.73	0.8	0.759
0.4		0.77	0.85	0.808		0.75	0.8	0.773
0.45		0.78	0.84	0.809		0.76	0.79	0.776
0.5		0.8	0.83	0.814		0.8	0.78	0.788
0.55		0.8	0.83	0.817		0.81	0.77	0.791
0.6		0.82	0.83	0.822		0.83	0.76	0.798
0.65		0.83	0.82	0.821		0.85	0.76	**0.804**
0.7		0.84	0.81	**0.825**		0.87	0.75	**0.804**
0.75		0.85	0.79	0.817		0.88	0.73	0.799
0.8		0.86	0.77	0.815		0.89	0.72	0.793
0.85		0.87	0.76	0.808		0.89	0.67	0.767
0.9		0.89	0.73	0.804		0.89	0.64	0.748
0.95		0.91	0.7	0.791		0.9	0.59	0.712
0.99		0.9	0.62	0.737		0.93	0.5	0.649

SI2. Examples of Misclassification

No system pipeline is perfect, and misclassifications occur. The following are examples of misclassification. The images on the right are the key political figures the system aims to detect, and the images on the left are misclassified faces.

SI3. Additional Performance Comparison with Hong et al. (2021)

We conducted the following comparisons to determine why our system outperformed that of Hong et al. (2021). Because the most critical issue is the effect of the clustering used in this system on performance, we controlled for differences in ways other than comparing classifiers and clustering as much as possible. Specifically, we used the same FaceNet features as Hong et al. (2021) and compared performance with and without tracking using tracklets.

When we and Hong et al. (2021) refer to FaceNet features, we refer to the same concept (face classifiers trained with triplet loss). This was introduced by Schroff et al. (2015), using a more advanced network backbone of FaceNet, inception-resnet-v2, in contrast to the inception-resnet-v1 used by Hong et al (2021). To examine the effects of clustering vs. classification, we extract the same base FaceNet features. Moreover, as Hong et al. (2021) did not use tracklets, we examined performance without them. Note that Hong et al. (2021) did not use the extracted FaceNet features directly, but trained a classifier on top of them and used it together with the unspecified Amazon Rekognition Celebrity Recognition API face classifier (see Section 2.2 of Hong et al.'s (2021) supplementary material). Therefore, comparing our system with the older backbone of FaceNet used by Hong et al. (2021) without tracking would produce the strictest comparison between classifiers Hong et al. (2021) and clustering in our system.

In Table S3, MTCNN-FaceNetv1 indicates the backbone of FaceNet used by Hong et al. (2021), while MTCNN-facenet-v2 indicates the (more advanced) backbone used in our system. The comparison between Systems 2 and 3 demonstrates the detrimental effect of clustering on overall performance compared with the classifier (Overall F1 scores were 0.660 for clustering and 0.768 for the classifier). Therefore, we conclude that clustering provides flexibility at the expense of classification performance. The comparisons between Systems 1 vs. 4 and 2 vs. 5 show that using MTCNN-facenet-v2 improves performance compared with MTCNN-facenet-v1. On the other hand, the comparisons between Systems 1 vs. 2 and 4 vs. 5 demonstrate that tracking with tracklets improves performance. In summary, while clustering makes the system flexible and renders classification and retraining unnecessary, which is the advantage of our system, it also undermines overall performance compared with classifiers. However, this performance loss is compensated for by using more advanced backbones (with their corresponding training data), and tracking with tracklets to assign IDs, resulting in a flexible system (with no need to retrain the models) with great performance. The best F1 scores are shown in bold in the following table, indicating that our proposed system consistently performs best across the three media outlets.

Table S3. Performance of our system using the same FaceNet features as Hong et al. (2021)
Note: Note that while using the same base FaceNet features, Hong et al. (2021) trained a classifier on top of them.

System	Classifier vs. Clustering	Detector-feature	Tracking	Overall			CNN			FOX			MSNBC		
				P	R	F1	P	R	F1	P	R	F1	P	R	F1
1	Clustering	MTCNN-FaceNet-v1	Yes	1.00	0.66	**0.797**	1	0.7	**0.826**	1	0.67	**0.802**	1	0.62	**0.762**
2	Clustering	MTCNN-FaceNet-v1	No	1.00	0.50	0.665	1	0.54	0.698	1	0.49	0.655	1	0.47	0.642
3 Hong et al. (2021)	Classifier	MTCNN-FaceNet-v1		0.96	0.64	0.768	-	-	-	-	-	-	-	-	-
4	Clustering	MTCNN-FaceNet-v2	Yes	1.00	0.74	**0.847**	1	0.77	**0.869**	1	0.75	**0.858**	1	0.69	**0.814**
5	Clustering	MTCNN-FaceNet-v2	No	1.00	0.63	0.775	1	0.65	0.787	1	0.66	0.793	1	0.59	0.744

References

Hong, J., Crichton, W., Zhang, H., Fu, D.Y., Ritchie, J., Barenholtz, J., Hannel, B., Yao, X., Murray, M., Moriba, G. and Agrawala, M., 2021, August. Analysis of faces in a decade of us cable tv news. In KDD'21: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining.

Schroff, F., Kalenichenko, D. and Philbin, J., 2015. Facenet: A unified embedding for face recognition and clustering. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 815-823).