

PAPER • OPEN ACCESS

Fighting COVID-19 with Fever Screening, Face Recognition and Tracing

To cite this article: Weijun Tan *et al* 2020 *J. Phys.: Conf. Ser.* **1634** 012085

View the [article online](#) for updates and enhancements.

You may also like

- [ICRU Report 64: Dosimetry of High-Energy Photon Beams Based on Standards of Absorbed Dose to Water](#)
D W O Rogers
- [The activities and funding of IRPA: an overview](#)
Geoffrey Webb
- [An interview with Molly S Shoichet. Developing biomaterials and mobilizing resources for assaults on some of the most devastating medical problems](#)
Myron Spector



ECS
The
Electrochemical
Society
Advancing solid state &
electrochemical science & technology

DISCOVER
how sustainability
intersects with
electrochemistry & solid
state science research

Fighting COVID-19 with Fever Screening, Face Recognition and Tracing

Weijun Tan¹, Jingfeng Liu¹, Yue Zhuo¹, Qi Yao^{*1}, Xing Chen¹, Wei Wang¹,
Rushuai Liu¹, Yu Fu¹

¹ DeepCam, Shenzhen, Guangdong, 518110, China

* qi.yao@deepcam.com

Abstract. Since the outbreak of COVID-19 corona virus in late 2019, it has become a tremendous threat to the whole world. Driven by the mission to save lives, we develop a fever screening and tracing system which can detect patients with fever symptom, and identify the patients using face recognition. In addition, our big data AI platform enables the tracing of the patients possible. A real-time alert sent to the personnel on duty on a web or mobile app activates the action to trace the patient and the close contacts providing an effective means to control the spread of the virus.

1. Introduction

Since the outbreak of COVID-19 corona virus in late 2019, it has become a tremendous threat to the whole world. As of end of June 6, 2020, more than 6.6 millions of cases, and more than 392K deaths have been confirmed [1]. It becomes very critical to stop the spreading of this virus. Driven by the mission to save lives, we develop a fever screening system to detect patients with fever symptom. We understand that not every patient has fever symptom. But still, detection of patients with fever symptom can help the control of the virus spreading significantly. After a patient is confirmed, his identification can be recognized using face recognition. This can be used to notify the personnel on duty in real time and safety action such as quarantine can be taken to quarantine the person.

We also do face mask detection at the same time. Whoever does not wear a face mask will be detected and an alert message is sent to the personnel on duty. Our face recognition can recognize a person no matter he does or does not wear a mask. We go further to combine this fever screening and face recognition with a big data AI platform. With this platform and other face recognition cameras, we can easily find the time and location of this person's history, and therefore trace who has close contact with him. This makes the tracing of the virus spreading more effective.

2. Fever Screening

We use thermal imaging, also called Infrared thermography[2], to measure human body's skin temperature.

Since infrared radiation is emitted by all objects with a temperature above absolute zero according to the black body radiation law, thermography makes it possible to see one's environment with or without visible illumination. The amount of radiation emitted by an object increases with temperature; therefore, thermography allows one to see variations in temperature. When viewed through a thermal imaging camera, warm objects stand out well against cooler backgrounds; humans and other warm-



blooded animals become easily visible against the environment, day or night. As a result, thermography is widely used in the military and other users of surveillance cameras.

In our fever screening devices, we use dual cameras – one thermal camera to measure temperature, and the other visible-light RGB camera to do face detection, and recognition. After a face is detected, the thermal camera locates the forehead of the person, and measure the skin temperature. To improve the accuracy, a black body can be used as a reference for the temperature measurement. The black body is usually installed at the place where people pass by, which makes the installation not convenient. In our device using a black body, it is hidden in the camera module, making it very easy to use. Our device can be a tablet, a camera, or a box with HDMI interface to separate display screen. In Figure 1 a multiple person fever screening device is shown.

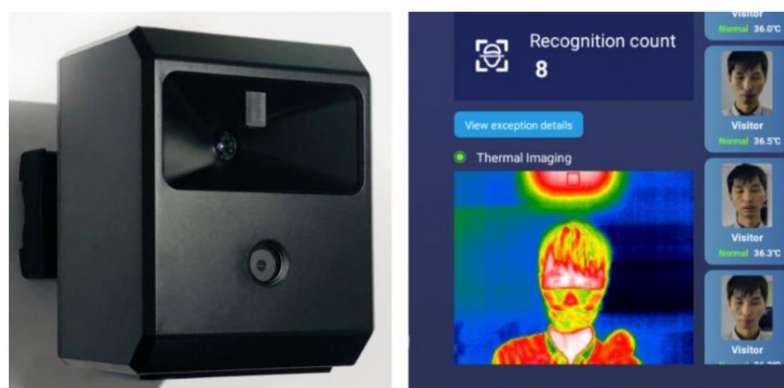


Figure 1: Demo of a multi-person fever screening device. The left hand is the camera module, which include a thermal camera, a black body, and a RGB camera. The right hand is a demo of the thermal image, and the face recognition result.

3. Face Recognition

There are two face recognition modules in the system, one in the edge device, another on the cloud. The face recognition module in the device can only recognize persons, called subjects, enrolled in the device; others will be reported as visitors. The enrolment can be managed locally on the device, or in the cloud platform. The latter is usually preferred. The second face recognition on the cloud is more powerfully. It uses powerful GPUs to train the model and do the inference to extract face feature vector.

3.1. Face Quality Network

Nowadays, most efforts on face recognition are to decrease in-class distance, while to increase inter-class distance, therefore to improve the discriminative capability. There are three approaches. First one is how to expand the data set and improve the quality. The introduction of datasets like VGGFace2 and MS1M [3] helps noticeably the performance of face recognition. Second one is to optimize the loss function. In same condition in terms of data set and network, new loss functions including SphereFace[4], CosFace[5], ArcFace[6], CircleLoss[7] have been proposed. Last one is to optimize the backbone network. In conditions where the computing power is limited, it is very important to design a backbone network which can achieve a good trade-off between performance and speed. For survey of the face recognition network, loss functions, and data set, please refer to [8-9].

We take all these approaches to improve our face recognition algorithm. In addition, we propose two unique algorithms[10] to improve the performance of face recognition. Firstly, we propose to extract a quality of face before the face recognition. While many researches spend so many efforts to improve the face recognition algorithms to do a better job to recognize low quality faces, including but

not limited to, blurry faces, back lighting faces, occluded faces, we choose to quantitatively define the quality of a face via a CNN. Then we filter out faces of quality lower than a quality and only do face recognition on faces not being filter out. This has been proved to be very good approach to improve the performance of face recognition in practical applications.

The face quality can be used not only to filter out low quality faces, but also in training of the face recognition network. One simple way is to use different weight of a face in the cost function. Good quality has larger weight and low quality face uses smaller weight. This has been proved to improve face recognition network when low quality faces are not filtered out.

3.2. Face Feature Evolving

Face pictures of a same person have some variations. Ideally, we can first get the centroid of the feature factors of these faces by averaging the feature vectors.

However, in reality, not all these face pictures are available all at a time. Similar to the behaviour of human, a person is recognized not by the first time he is seen, but by recent times he is seen. Reflecting to face recognition, the feature vector should be evolving as more pictures of this person become available. Therefore, we do not use the average of all feature vectors, but use a weighted average of the feature vectors. The choice of the weights reflects human's recognition behaviour. Simply speaking, the newer feature will have larger weight than older feature but the change of centroid feature vector should be gradual.

Table 1: Performance of our face recognition network

Method	LFW	CALFW	CPLFW	CFP_FP
Center Loss	98.75	85.48	77.48	-
Sphere Face	99.27	90.30	81.40	-
VGGFace2	99.43	90.57	84.00	-
ArcFace	99.82	95.45	92.08	-
Ours	99.85	96.03	98.27	99.60

3.3. Performance

We combine the above two algorithms, on top of the state of the art face recognition networks. Shown in Table 1 is the performance comparison of our face recognition networks with other best face recognition networks, on a few public data sets.

From the table, we see that our face recognition networks achieve similar or better performance than the public available networks. Please note that, in these results, low quality faces not worth recognizing are filed out. Also note that, since our algorithms are trained on proprietary data sets, which are mostly Asian, while the public data sets are mostly white, so the improvement of our algorithms have not been fully demonstrated.

4. Big Data AI Cloud Platform

We develop a big data AI cloud platform to manage multiple types of devices including fever screen tablets, face recognition cameras, license plate recognition cameras, gun detection cameras and so on. This platform is a modular, open interface platform. So it is easy to add other smart devices and applications.

We use the popular Spring Cloud[6] big data framework in our platform design for its micro-service architecture. It provides tools for developers to quickly build some of the common patterns in distributed systems (e.g. configuration management, service discovery, circuit breakers, intelligent routing, micro-proxy, control bus, one-time tokens, global locks, leadership election, distributed sessions, cluster state). For more information, readers can refer to Spring Cloud website [6].

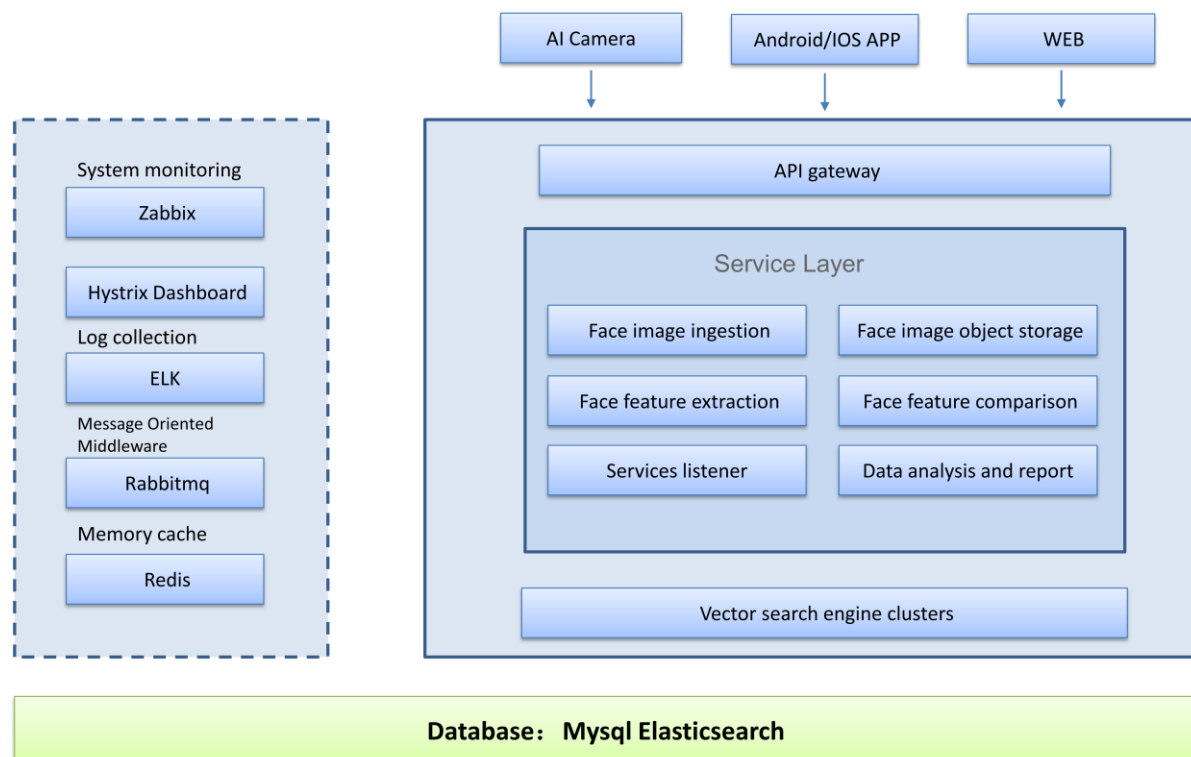


Figure 2: System diagram of our big-data AI platform

4.1. System Diagram

The top-level system diagram of this system is shown in Figure 2. Please note that we use face recognition as an example to demonstrate the block diagram. It actually supports other type of object detection, recognition, and reporting.

On the left side of Figure 2 are the Spring Cloud components we use for system monitoring, log collection, message and event queuing, and memory cache management. On the right side of Figure 2 is the main pipeline of our system. On the top are the AI cameras which ingest data to the API gateway. Also interfaced to the gateway are the Android/IOS mobile apps, and web. The app and web acts as the user account manager, device manager, data analysis and display terminal. When an alert event is detected in the service layer, an alert message is sent to them in the form of push notification, email, text message etc.

In the middle are the main service layer, including the face image ingestion, face image object storage, face feature extraction, face feature comparison, service listener, and data analysis and report. More blocks can be added as they are needed. For example, if there are separate face attribute (like age, gender) extraction block, it will be put here.

In the bottom is the vector search engine (VSE). It consists of clusters of servers, and is very easy to scale up by adding more hardware. This VSE is a very critical part of the system to provide big data support, real-time processing, and subject retrieval.

On the very bottom of the whole diagram is the mySQL elastic search. This advanced search technology manages the large amount of data, so called big data, in the system.

4.2. VSE

The VSE is most critical block in the diagram to provide accurate, fast (perhaps real-time depending on applications) search and data retrieval in big data face images in the scale of tens of millions to billions.

Depending on different applications, we support both a global search engine, and a smaller VIP search engine. In the global search engine, all data coming to our system forms a big data base. The search engine can find nearest neighbours in the data base for a particular subject, or form clusters of faces for a variety of applications. The VIP search engine searches among given set of data, for example, for faces in a customer's one store, or a number of stores, or all stores. The VIP examples include important customers of a bank or retail store, or shoplifter suspects in a pharmacy store.

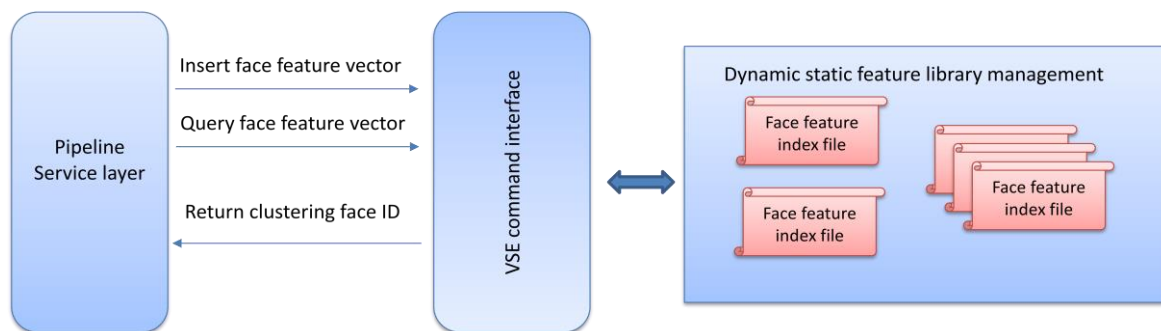


Figure 3: VSE clusters

The key algorithm in the VSE is a k nearest neighbour (KNN). In practice, to improve the efficiency by a few orders, while only losing marginal performance, an approximate nearest neighbour (ANN) [12] is usually used. There are many different algorithms for ANN, including graph-based, tree-based, local sensitivity hashing based and so. Our ANN algorithm is one of the latest state of the art. It supports retrieval of subjects among hundreds of millions data in milliseconds.

Shown in Figure 3 is our VSE diagram. The VSE command interface received face feature vector from the pipeline service layer. It also receives query face feature vector. When no valid record is found in the face feature vector data base, a new record is inserted into the data base. When valid records are found in the data base, the cosine similarity is compared with a given threshold. Records whose similarity is higher than the threshold are returned with their face ID.

There is one dynamic search index and multiple static search indices. When the data amount in the dynamic search index reaches a certain amount, the dynamic index is converted to a static index file, and new dynamic index is created. The reason for using a dynamic index is to include the most recent new records in the search, which is required in many applications.

Every static index holds a certain amount of data records, typically a few millions. We do not use a search index with more than 10 millions of data records, because when the data records grow too big, the training of the search index, fast retrieval, dynamic-static conversion will become problematic.

When a query request is received by the VSE command interface, it is sent to all index files. Returned results from all index files are combined, sorted, and filtered against the threshold.

The static and dynamic index files can be hosted on same or different VSE servers. All these servers work as a cluster with automatic assignment, load balancing, fault protection etc powered by Spring Cloud.

4.3. Fever Screening Reporting and Tracing

As the outbreak of the COVID-19 pandemic, we are motive to modify our big data AI platform to support the fever screening reporting, and patient tracing using face recognition.

Firstly, we added reporting of fever screening into our system. Our previous subject and data type did not include temperature. In order to do the fever screening, we added the fever screening devices to our supported hardware. We also added temperature as one entry of our data type, and added data analysis, alert type, message notification etc. This demonstrates the easiness of our system's capability to integrate new device, new data, and new event processing.



Figure 4: Demo of the fever screening and face mask detection.

Secondly, we added detection of face mask in the edge device. Face data sent to the cloud also included face mask information.

Thirdly, and most importantly, combining with the regular face detection and recognition, the fever screening data is made possible for patient tracing. When a patient is identified in our fever screening device, his appearance in other fever screening devices, or other regular face detection devices (mostly cameras), then by retrieving this patient's history and clustering results in the global VSE search engine, close contacts of him can also be identified. Follow up actions can be taken according to stop the wide spread of the virus.

Shown in Figure 4 is a demo of the fever screening and face mask detection. In each of the small pictures, the one on the right is the detected one from edge device, and the left one is the recognized person. When the left picture is empty, this detected subject is not recognized, mostly because the person is not enrolled in the account. The number on the left corner is the measured temperature, green means normal temperature, red means a fever. Please note that the fever temperature is as high as 54.5 degree because the person holds a hold water cup to mimic a fever.

Also please note that under each picture, there is information like store name, device ID, date and time, and group number. This information can be used to trace the history and contacts of an identified patient by a SQL query. In addition to a face picture, a snap shot of the video frame is also saved. All persons showing up on same video frame are the close contacts.

5. Conclusion

We develop a fever screening and face recognition system to help detect COVID-19 patients who has fever symptom.

On top of that, our big data AI platform makes the tracing of the patients based on face recognition possible. When a suspect patient is detected, a real-time alert is sent to personnel on duty on a web or

mobile app. With this alert, safety actions can be taken to trace and quarantine the patient and the close contacts. Therefore, our system provides an effective means to control the spread of the virus.

References

- [1] WHO report, available at <https://www.who.int/emergencies/diseases/novel-coronavirus-2019>
- [2] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, A. Zisserman, “VGGFace2: A dataset for recognising face across pose and age”, International Conference on Automatic Face and Gesture Recognition, 2018.
- [3] W. Liu et. al., “SphereFace: Deep Hypersphere Embedding for Face Recognition”, CVPR 2017.
- [4] H. Wang et. al., “CosFace: Large Margin Cosine Loss for Deep Face Recognition”, CVPR 2018.
- [5] J. Deng et. al., “ArcFace: Additive Angular Margin Loss for Deep Face Recognition,” CVPR 2019.
- [6] Y. Sun et. al., “Circle Loss: A Unified Perspective of Pair Similarity Optimization” , <https://arxiv.org/abs/2002.10857>.
- [7] I. Masi, Y. Wu, T. Hassner, and P. Natarajan, “Deep Face Recognition: A Survey”, IEEE SIBGRAP Conf, Oct-Nov 2018.
- [8] Y. Kortli, M. Jridi, and M. Atri, “Face Recognition Systems: A Survey”, Sensors, 20(2): 342, January 2020.
- [9] R. Liu et. al, “Face Quality for Recognition”, to be submitted to ACCV 2020.
- [10] Spring Cloud, available at <https://spring.io/projects/spring-cloud>
- [11] Near neighbour search, available at https://en.wikipedia.org/wiki/Nearest_neighbor_search.