Artificial Intelligence Vaccination Strategy Through Kafka Using Human Face Detection

Umma Khatuna Jannat[1*], Dr.M.MohanKumar[2] and Syed Arif Islam[3]

[1*]Dept. Of Computer Science, Karpagam Academy of Higher Education, Coimbatore, India
ummakhatunajannat@gmail.com
[2]Dept. Of Computer Science, Karpagam Academy of Higher Education, Coimbatore, India
mohankumarcs@kahedu.edu.in
[3]Dept. Of Computer Science, Karpagam Academy of Higher Education, Coimbatore, India
syedarifislam@gmail.com

Abstract: Numerous issues have arisen in COVID-19 pandemic in many facets of human life. Numerous investigations have been carried out since the pandemic began more than a two year ago to identify different technical advancements and applications to combat the virus that has taken countless lives. Ongoing spread of the coronavirus infection can be stopped using vaccines. The use of face detection help to each record of the vaccination. In order to combat the pandemic formulation, this study article explores face detection and kafka technology research. Additionally, the value of face detection technology was examined, and the contributions of technology to five key areas were highlighted. These include having a unique ID with each person's face, saving each vaccination dose on a safe cloud platform, and certification. The analytical techniques of machine learning (PCA), Hadoop, Kafka, statistics, and mathematics show the accuracy and sustainability of this work, as well as its speed and time latency. People are the main data sources used to test this study effort. As a result, face detection technology will enable everyone to check his or her vaccination status. A scalable system for COVID-19 immunisation records with certificates is developed in this research study.

Keyword: Face Detection, Vaccination, Kafka, Hadoop, AI

Introduction

In the history of vaccines, the COVID-19 vaccinations have advanced at a rate that is unprecedented. Currently, 104 potential vaccines are being developed at the clinical stage, and 184 are in the preclinical stage. According to the latest research, there are currently 18 COVID-19 vaccines that have been approved and are being used globally. There were 621,797,133 confirmed cases of COVID-19 globally as of 5:49 PM CEST on October 17, 2022, with 6,545,561 fatalities. As of October 11, 2022, 12,782,955,639 doses of the vaccine had been administered. WHO set a target of 70% global immunisation coverage by the middle of 2022. By June 2022, just 58 of the WHO's 194 member states had reached the 70% target, and only 37% of healthcare workers worldwide had completed a full course of basic vaccines.

Even though, Many nations started requiring tourists to test negative for the new coronavirus before entering their borders in the fall of 2020. There soon developed a black market, as with anything valuable. Travelers might try to bypass the checkpoint by purchasing fraudulent negative COVID-19 test results illegally. At that time, there were already several cyberattacks against the COVID-19 vaccine. North Korean cyberterrorists allegedly targeted the UK-based vaccine manufacturer AstraZeneca in late 2020. Russian hackers were also discovered trying to steal vaccines throughout the summer. Therefore, cybersecurity risks to the COVID-19 vaccination are primarily.

In the context of automated inspection, process control, and robot guiding, machine vision is the use of imaging based automatic inspection and analysis in the industrial setting. Face recognition has become one of the hottest research areas in computer vision, image processing, and neural networks. In human-computer interaction, verification systems, and public safety, it is frequently used. Face detection has mostly progressed through three stages: on the basis of structural features, on the basis of statistical features, on the basis of massive data and complicated models. Technology for detecting faces has progressed from geometric features and template matching to artificial features and classifiers [1]. Due to the ongoing development of deep learning technology, this technology has shifted from conventional machine learning techniques to deep neural networks in recent years [2]. However, the calculation of neural networks frequently involves numerous matrix operations, which overload the hardware. The model computation time has been significantly decreased appreciations to the development of robust GPU hardware, which has for the broad adoption of facial detection technologies. After training large deep neural network models with tens of millions or even hundreds of millions of unknown parameters, researchers are continuously updating the top face detection accuracy record. Most facial detection systems, however, rely on cloud computing resources. Increasing numbers of connected devices are placing high demands on the network bandwidth and computational processing power of the physical data centers that represent the "cloud" in order to provide fast and accurate identification. The vulnerability of identification data and the open network environment also increase the risk of privacy leaks when using facial detection technologies in the real world. With the cloud-based technique, sensitive data is sent from many users to the cloud. If assaulted, user privacy is likely to be compromised. Cloud-based deployments, therefore, pose a degree of security risk to the personal data of customers. Researchers have become quite interested in edge computing as a complement to cloud computing in recent years. Edge computing offers end-to-end services, unlike cloud computing. The ability to analyse data immediately on the edge device rather than having to send it to the cloud demonstrates its exceptional performance in lowering bandwidth burden and communication delays.

Although facial detectionn technology has made some significant strides, much more needs to be done before it can be applied in a practical setting [3]. For real time face detection and the qualities of Kafka are incredibly versatile and well-liked in current modern day. It is a distributed system made up of clients and servers that communicate with each other using the fast TCP network protocol. It is free software that functions as a distributed transaction log and is used to process real-time data streams. Kafka is a communications system and a platform for streaming events, to put it simply. It is excellent for maintaining uptime, growing quickly and easily, and handling high data volumes.

Hadoop is an open-source framework that will be used for a significant portion of this work because of its rapid processing and economical storing. Hadoop is capable of processing datasets of any size, from gigabytes to petabytes, with high efficiency [4]. Hadoop can analyse a large number of datasets in parallel because it uses a sizable cluster of parallel processing units to store and process data. Generally speaking, Hadoop Distributed File System (HDFS) works with inexpensive or standard hardware. In comparison to conventional file systems, HDFS offers faster data throughput.

Real-Time Facial Detection is Facilitated with Apache Kafka

Given the aforementioned difficulties, Apache Kafka presents as the best option for enabling businesses to detect fraud in real time [5][6]. When ingesting enormous volumes of data, Apache Kafka is helpful. Because it is highly scalable, any volume of data may be managed for real-time face detection. The foundation of Kafka's delivery architecture is a producer and consumer based messaging system. Being a distributed system, it operates in clusters of several nodes, known as Kafka Brokers. Kafka can read large amounts of data from a variety of sources, including user interactions, financial transactions, database records, analytics, and more.

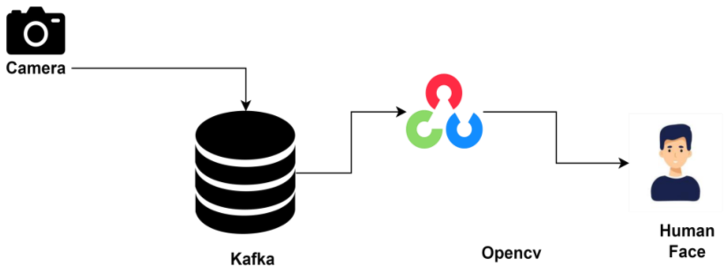

Fig. 1 Process

The fantastic piece of technology known as Apache Kafka makes it easier to pull data from various sources. It converges data pipelines from several sources, assisting organisations in extracting insights from massive data sets. Additionally, such a technology must be able to take in large amounts of data quickly. Based on the data it gathers and the analysis it performs, the tool should be able to generate predictions about face detection in real time. Camera images are sent to Kafka through a machine vision technology that is native to the platform. Metadata is added during preprocessing, and data from other backend systems is correlated with it. One or more programmes then consume the message.

Although facial recognition technology has made some significant strides, much more needs to be done before it can be applied in a face authentication system. With the use of the unique data set, a model is created that can identify and name any persons in a live image of human. Information about the immunisation is received and the recognition response made on a vaccination certificate issued.

Methodology

The COVID-19 epidemic was resulted in a massive increase in data [7] that makes it difficult to stay current on development in the field of Kafka technologies. As of this work seeks to close gap by examining the kafka research for COVID-19 to determine the state of the field [8]. Studies focusing on current deep learning techniques and their potential applications [9]. Furthermore, highlighted how Hadoop and AI can be used [10]. A wider view of kafka technologies is presented in our research work in comparison with earlier studies, encompassing its use in a variety of contexts, analytical approaches, and data management capabilities. Furthermore, this field has not been thoroughly investigated.

The Literature Review Process

Relevant and academic investigations have shown how AI technology can be used to combat the COVID-19 epidemic and comprehend the study's current situation. With search phrases published between January 2018 and January 2022, publications from the reputable journal citation databases IEEE, Science Direct, Springer, and Google Scholar were chosen for the analysis.

Additionally, the inclusion of journals and proceeding articles in the English language was a requirement for the selection process. The second criterion was subjected to qualitative examination based on the abstract, with articles that had weak ties to the context being disregarded. In order to categorise the findings into research and review articles, full-text reading was done. After that, articles relating to the domain that were less firm and non-empirical were removed. All academic studies were eventually screened. Papers describing data analytics work in academic journals, as well as actual investigations that utilised Covid 19 vaccine , human face detection and kafka. Out of the entire number of publications, 78 were categorised as research and the remaining articles as reviews. The 78 research articles that make up this study's focus. The methodical technique of literature review is provided in Fig. 2.

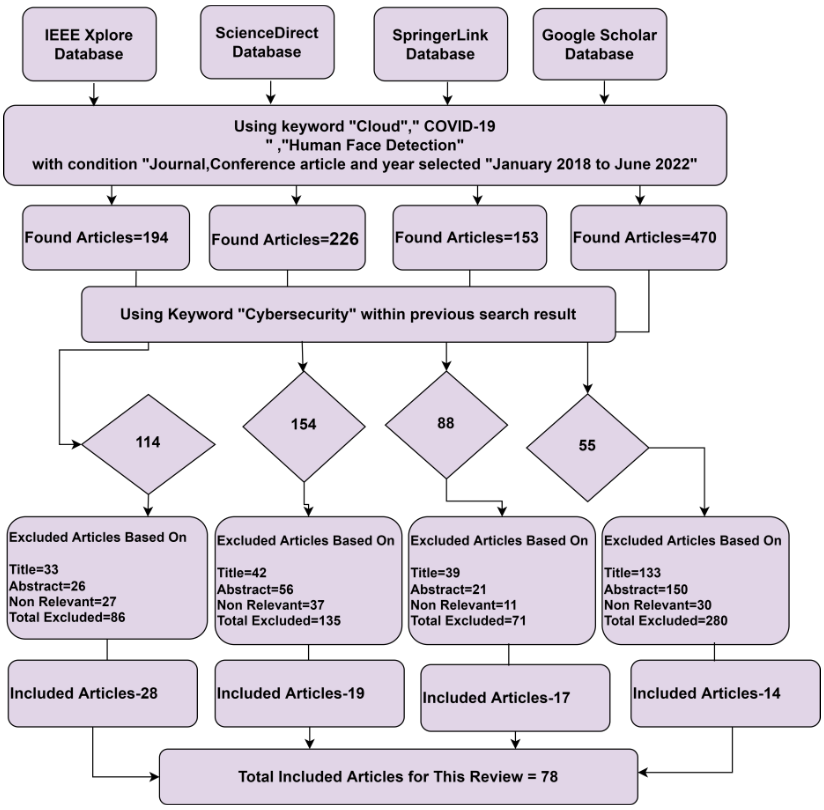

Fig .2 Methodical Technique of Literature Review

Area of Research and Contribution

The articles are put into practise based on real-time goals categorised into eHealth systems for the public and private sectors. The areas of contribution can be covered by this system in many areas. It also reduced the time complexity and give the exact accuracy. The descriptions of the contributor regions are shown in Table 1.

Table 1. The areas of contribution

Contribution	Covered Areas
eHealth	The eHealth sector has benefited from advancements made elsewhere employing this smart technology appreciations to research and development.

Government	This enables medical experts to remotely monitor public health.
Time	As predictors are utilised to cut down on the time and danger, the population-based tactics that follow this approach will be beneficial.
Usability	Comparatively, this software is easier to use.

Implementation

Kafka Streaming

In order to avoid being continuously accommodated in memory, real-time stream data is continuously generated. The memory capacity restriction prevents such a vast volume of stream data from being processed in parallel in a single node. Frequent data exchange between the main processor and the coprocessor also contributes to communication overhead [11]. The communication cost caused by just using distributed processing on numerous nodes results from extensive parameter exchanges between nodes in the cluster. A novel architecture and various approaches for apache kafka-based large-scale face recognition processing in order to address this issue [12]. Three components make up the concept for large-scale image processing in real-time. The first is where the frame from the stream channel is detected. The stream channel could be enabled remote channel that is physically connected to the camera. This component, which is made up of numerous nodes, is in charge of producing messages. One node is capable of detecting data coming from one or more channels. How many channels may be detected in one node depends on the resource capacity of each node. The image frame is sent to the Kafka broker during the channel detection section. A Kafka broker is in the second section. A significant amount of the data that was communicated in the previous section is saved in a buffered queue in this section. The processing rate is typically slower than the generation rate. This makes it necessary to temporarily store the identified data in order to handle a huge volume of data without loss. We employed Kafka for a variety of purposes. We chose Kafka for large-scale image processing since the file system ensures the data's long-term viability by storing the human face data [13]. This benefit makes it possible to keep data indefinitely even when none of the nodes can process it. At the moment the user specifies, the data is removed from the file system. Additionally, the real-time produced file's enormous size allows it to overcome the drawback of being challenging to load in main memory. Each processing node accepts a message from the data and processes it rather than disseminating it to the scattered nodes in the master node. Even though each node's performance varies in this scenario, high-speed processing can still be done without a stream of data bottleneck since the node that is ready to handle the data.

The final component is that each node includes a face detection processing programme that uses stream data from the Kafka broker to process faces. Each node's detection processing programme is designed to carry out the same task. One frame can be thought of as a group of linked data as each node only possesses one frame at a time. Data is communicated or processed in units of tuples in the text processing system. The face detection processing unit is viewed as a frame in the suggested system. The processing of face detection includes numerous iterations of matrix or vector operations [14]. By using a GPU accelerator and treating a frame as the basic processing unit, it may be processed quickly. Regardless of the sequence in which the images are generated, we use the asynchronous way of data processing in this section.

Technique

We are employing methods to handle real-time human face detection. Following are a few of the technique we present.

- The resolution of the human face image needs to be synchronised since the same process is carried out at the application level. A facial image gets shuffled when it is distributed across several distributed resources. Therefore, the application may demand an excessive effort, which may affect performance, when face images of varied resolutions are transferred to a distributed environment. Because of this, the application level in the component detecting the frame must receive the size information of the facial image.
- The frame of an image of a human face is the unit that is conveyed at once. One frame delivered from a camera channel typically has several channels. In order to send one frame to the Kafka broker, it is serialised into a byte array. The original human face image is re-encoded based on factors like the resolution and number of channels when the byte array is sent from the Kafka broker.
- Synchronously send human face image data to Kafka Broker. This method accelerates the processing of face images in a distributed resource system with numerous nodes. This allows for the face photos to be processed more quickly and in any order. It can be used to identify an object in an image that is generated in a vast camera channel and

permanently store the value by capturing only the information of the detected object. In order to retrieve and parallelize data from subjects, use a number of distributed nodes. Distributed nodes can retrieve and analyse data from each topic using a frame from a single channel that is broadcast to a number of topics. As an alternative, numerous nodes can access and process the same data. The node grows larger and utilises its memory capacity more as the number of channels increases since more memory is required to process frames.

Frame Transmission and Image Distribution Gateway

To send the frame to the Kafka broker, the data must be serialised. Gateways are represented as a matrix in the data flowing from the gateway. This is transmitted by converting a matrix made up of two RBG gateways into a byte array. The height and width of the frame, which indicate how many gateways a single frame forms, are lost during this operation. Once the gateway is identified, this information should only be delivered once at the time of connection with the Kafka broker because it is required to re-encode the human face picture at the application level. The system offered by Kafka's brokers will divide the queues for each gateway that comes in. The distribution of each gateway among various systems is depicted in Fig. 3 below. Before taking data straight from the Kafka system, many distributed applications handle data based on the resources they have. As depicted in the picture, the message can be spread by permitting access from different nodes, as well as retrieved by accessing numerous systems from a single node.

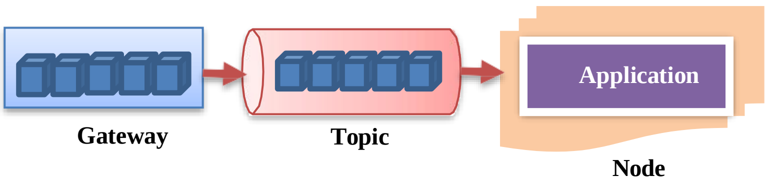

Fig .3 Distribution Gateway Frame

Streaming Data Pipeline

The method must be implemented on a specific platform. The overall method entails the input source's frames, processing the algorithm, and deploying it on a target platform that runs the programme with a CPU instance and responds to client requests via an API. Employ the Apache Kafka platform's open source client library to accomplish this. Using Kafka's native characteristics for data parallelism, distributed coordination, fault tolerance, and operational efficiency. Kafka streams streamlines the application infrastructure for executing the algorithm using apache Kafka, as shown in Fig. 4.

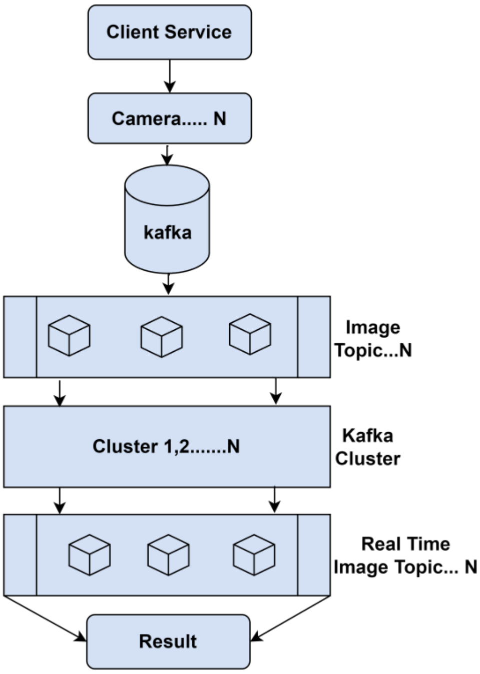

Fig .4 Executing the Algorithm using Apache Kafka

derived from the bullet camera's input frames. The message layer, which consists of producers subscribed by system cam1…camN, receives the bytes after the message layer. The messages are then delivered to the Kafka cluster, which is represented by the nodes Nodes1. NodesN and corresponds to the processing utilities or algorithms available to handle the byte value messages received from the producer. Each unique node is a subscriber to the producer's system. Additionally, the nodes process the byte messages received as input and deliver the processed streams to various groups of the systems. The consumer or group of consumers to which these systems are subscribed. The consumer system retrieves the message stream that has been processed and sends it to the intended application which assures data parallelism, in order to process streams more quickly. Our face detection algorithm, which is located on the nodes, receives input frames from the camera through the topics subscribed to the producer, processes them, and then sends them to the system subscribed by consumer. For our face detection software, we use a camera as our input source. In order to display results in real time.

Cloud on Real Time Face Detection for eHealth System

Webcams gather data from people and transmit it to the cloud for immediate analysis. Cloud-based, real-time electronic health record systems offer distributed cloud architecture contributes to the reliability of eHealth software. To provide real-time communication service for eHealth software, communication is made directly with edge devices. Large nodes can be connected in the cloud, which is beneficial in this system, and data processing in the cloud is done extremely close to the source of information to take prompt remedial action. The cloud offers less latency than is necessary for eHealth system real-time analysis. In comparison to cloud computing, it offers a variety of protocol standards and greater security.

Real Time Vaccine Data Using Kafka and Hadoop

A real-time many-to-one or many-to-many eHealth system can exchange messages at predetermined intervals. Kafka is a pipeline architecture that is utilised to store massive volumes of data in real time in a database. For real-time healthcare systems, it can be set as a many to many or many to one mode. On the server, Kafka is configured and put into operation. The pipeline is built using the Kafka topic. When the Kafka topic is active, more actuator nodes can read data from the Kafka broker and more sensor nodes can transmit data to the Kafka topic. Data can be buffered for a while by Kafka, and the data can then be processed in the pipeline itself in real time. In the system design, sensors in a cluster communicate with one another via sending messages to the cluster. Data from sensor nodes is gathered using a gateway. Data is written to a Kafka topic.

Research Experiments and Results

Experimental Environment

We deployed the Kafka broker with 12 X 1 TB discs and separate OS discs from Kafka storage on physical nodes with Intel® coreTM Dual 12 core socket CPUs and 64GB RM memory. In order to make it simpler to reshape the number of nodes, nodes at the application level have been run in a virtual environment.

In this experimental research, preprocessing and model training are done in the cloud while also designing the cloud architecture. We use openCV, which keeps track of the openCV ID, the quantity of images that need to be trained, and the time the request was submitted. In testing, the method used in the study is contrasted with conventional methods and characteristics that extract depth. Fig. 5, displays the typical process of the detection with certificate and Fig. 6, displays the real time face database detection.

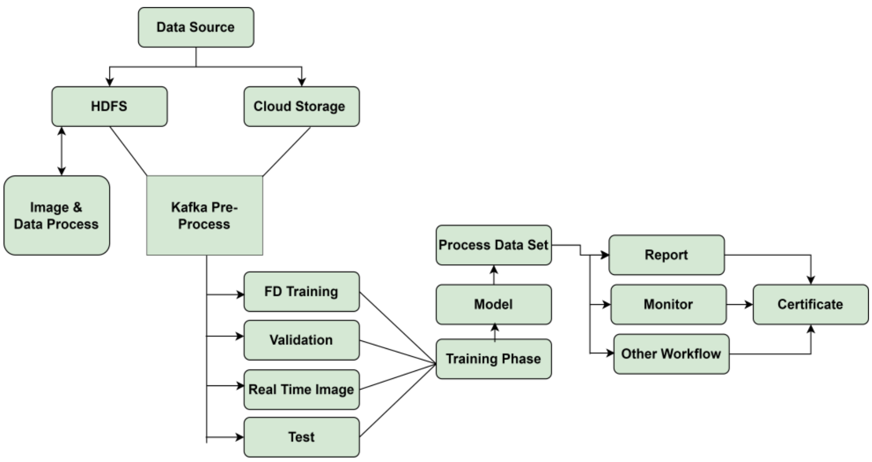

Fig .5 Typical Process of the Detection with Certificate

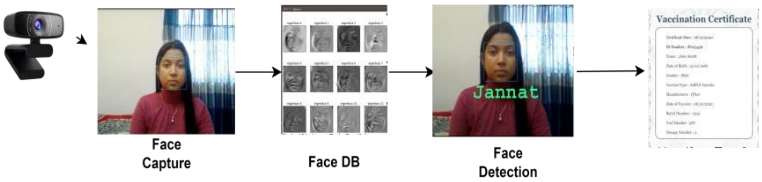

Fig .6 Displays the Real-Time Face Detection

Sign-in module for software using unique ID and registration for the human validation, and open a web cam to capture a live, detected human face image. human face image is first converted into a grey scale from various perspectives and train of the same person's faces images. A classifier will be trained using those photos store in Face DB. Each time, human have to the show face as a secret key to download documents. All the data record into a cloud server.

Therefore, high throughput, dependable delivery, and horizontal scalability Kafka will be employed as a messaging service. When messages are saved, Kafka categorises them according to subject. The message sender now acts as the producer, and the message recipient now acts as the consumer. Additionally, each Kafka instance in the cluster transforms into a broker, making up the Kafka cluster.

The training of the data delivered via Kafka is the responsibility of the cloud training host. We can set up a high-performance training host or employ a cluster-based organisation. For priority training of data that requires prompt training, we can fulfil the demands of real-time identification and finish the training. After that, send the model to the associated openCV and allow Hadoop to store it so that calculations can be performed on large amounts of data using MapReduce.

It is clear that the cloud as a whole functions as a whole. It begins by preprocessing the data received via openCV, which includes image preprocessing, image categorization, and basic data storage. After that, Kafka receives the preprocessed data. Following a second round of classification by work, Kafka sends the data for training and detection before sending some model data to Hadoop for storage.

The training data will get more and bigger as time goes by. On the other hand, we can upgrade the hardware setup and speed up the computing process in order to preserve the effectiveness of system training. However, we can decrease the training quantity by judiciously minimising the training data. We can decide on a limit. When the number of training images with a particular degree of accuracy exceeds this threshold, the excess photos are randomly destroyed to maintain the same number of training samples.

In terms of the training period for cloud data, there are two different scenarios: the first is when the system must first be able to recognise a pattern, in which case training begins as soon as the data is received; the second is when training is not necessary right away, as in the case of training accuracy that is located. We schedule training at off-hours.

Table 2. Face Detection Accuracy

Face Detection	Yes	90%
	No	10%
Time Effective	Yes	100%
Integration with Kafka and Hadoop	Yes	90%
	No	10%

Face Detection Accuracy, Sustainability, Time Latency

The first series of studies is to assess the accuracy analyzer's performance. As a binary classifier, the following metrics for face detection are of relevance to us:

$$A = \frac{AP + AN}{AP + AN + A\bar{P} + A\bar{N}}$$

Where A=Accuracy, AP is the number of true positives, AN is the number of true negatives, $A\bar{P}$ is the number of false positives, and $A\bar{N}$ is the number of false negatives.

Our initial test was designed to gauge the accuracy. To do this, we added 200 additional images to the original faces dataset and used those 200 images to perform face detection. In addition, the graph demonstrates that growing accuracy values are correlated with an increase in the number of photos per person. The accuracy, in figure 7 shows A = 90%. and Figure 8 and 8.1 displays the two scenarios' sustainable input rates. Results reveal that for both scenarios, Kafka consistently beats the other two frameworks. We think this behaviour is caused by the fact that the simplest version of Kafka is operating well, even though its results are fairly good and sustainably for a cloud environment. When comparing the performance of centralised solutions with those built on HDFS platforms, the former achieves a higher throughput than the latter in the case of a single machine since the "centralised" situation lacks the overhead of the underlying platform. The use of HDFS, on the other hand, assists in achieving sustainable input for attainable in the centralised situation as the number of machines rises. We demonstrate the acceleration in sustainable input in Fig. 9. Every framework speeds up when the number of nodes is constrained. When a system expands, its scalability starts to deviate from linearity; in both instances, kafka is the system that first displays this behaviour. Last but not least, Fig. 10 shows the latency. The average latency of the photographs contained for Kafka is the term "delay." The very lowest latency is regularly attained with Kafka.

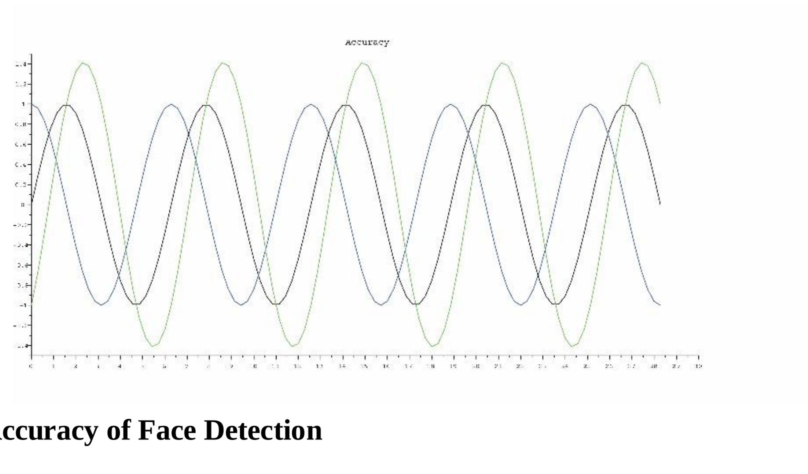

Fig .7 Accuracy of Face Detection

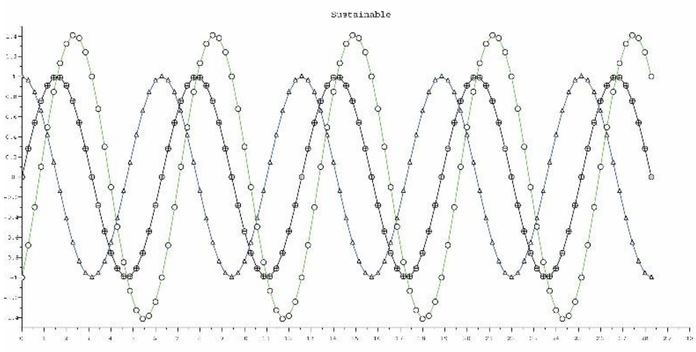

Fig .8 Sustainable

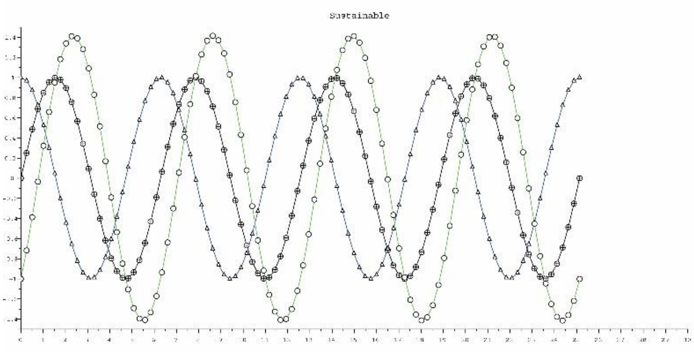

Fig .8.1 Sustainable

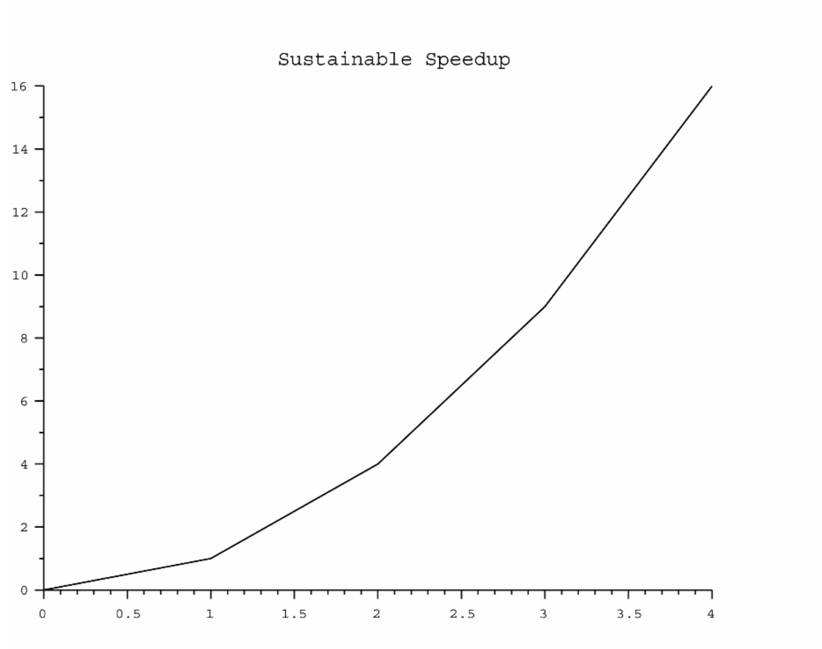

Fig .9 Sustainable Speedup

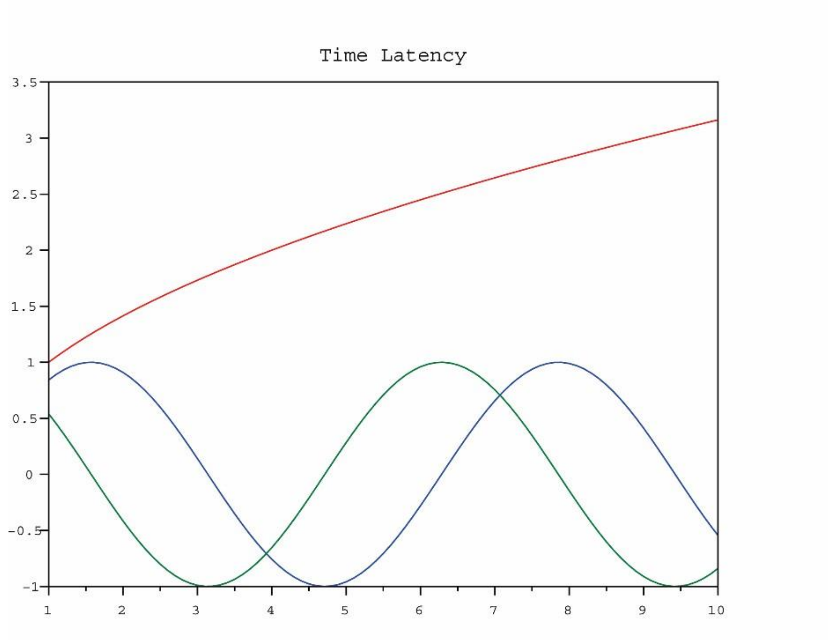

Fig .10 Time Latency for the Face Detection in Cloud

Conclusion

In this research article, we strive to present the exceptional viable records for a face detection tool this is used for figuring out people and without difficulty discover what number of doses were completed with vaccine name. Since it's totally based on cloud and makes use of Kafka and hadoop on this information framework, this software program is absolutely secure with the certification. The cameras are skilled to become aware of human faces and identities with the useful resource of kafka. The face detection device is absolutely reliable, easy to use, and time and price effective. It could be properly applicable to be implemented in each sector.

References

[1] Ahmed, S. B., Ali, S. F., Ahmad, J., Adnan, M., & Fraz, M. M. (2020). On the frontiers of pose invariant face recognition: a review. Artificial Intelligence Review, 53(4), 2571-2634.

[2] Alber, M., Lapuschkin, S., Seegerer, P., Hägele, M., Schütt, K. T., Montavon, G., ... & Kindermans, P. J. (2019). iNNvestigate neural networks!. J. Mach. Learn. Res., 20(93), 1-8.

[3] Anand, R., Shanthi, T., Nithish, M. S., & Lakshman, S. (2020). Face recognition and classification using GoogleNET architecture. In Soft computing for problem solving (pp. 261-269). Springer, Singapore.

[4] Alkathiri, M., Jhummarwala, A., & Potdar, M. B. (2019). Multi-dimensional geospatial data mining in a distributed environment using MapReduce. Journal of Big Data, 6(1), 1-34.

[5] ARMEL, A., & ZAIDOUNI, D. (2019, April). Fraud detection using apache spark. In 2019 5th International Conference on Optimization and Applications (ICOA) (pp. 1-6). IEEE.

[6] Aziz, K., Zaidouni, D., & Bellafkih, M. (2018, April). Real-time data analysis using Spark and Hadoop. In 2018 4th international conference on optimization and applications (ICOA) (pp. 1-6). IEEE.

[7] Awotunde, J. B., Jimoh, R. G., Oladipo, I. D., Abdulraheem, M., Jimoh, T. B., & Ajamu, G. J. (2021). Big data and data analytics for an enhanced COVID-19 epidemic management. In Artificial Intelligence for COVID-19 (pp. 11-29). Springer, Cham.

[8] Talmele, G., & Shrawankar, U. (2022). Real-Time Cyber-Physical System for Healthcare Monitoring in COVID-19. International Journal of

Web-Based Learning and Teaching Technologies (IJWLTT), 17(5), 1-10.
[9] Tuli, S., Basumatary, N., Gill, S. S., Kahani, M., Arya, R. C., Wander, G. S., & Buyya, R. (2020). HealthFog: An ensemble deep learning based Smart Healthcare System for Automatic Diagnosis of Heart Diseases in integrated IoT and fog computing environments. Future Generation Computer Systems, 104, 187-200.
[10] Vione, E., & Darmawan, J. B. (2019, April). Performance of K-means in Hadoop Using MapReduce Programming Model. In Proceedings of the 1st International Conference on Science and Technology for an Internet of Things, 20 October 2018, Yogyakarta, Indonesia.
[11] Van Dongen, G., & Van den Poel, D. (2020). Evaluation of stream processing frameworks. IEEE Transactions on Parallel and Distributed Systems, 31(8), 1845-1858.
[12] Wäldchen, J., & Mäder, P. (2018). Machine learning for image based species identification. Methods in Ecology and Evolution, 9(11), 2216-2225.
[13] Yang, J., Lu, Z., & Wu, J. (2018). Smart-toy-edge-computing-oriented data exchange based on blockchain. Journal of Systems Architecture, 87, 36-48.
[14] Zhi, H., & Liu, S. (2019). Face recognition based on genetic algorithm. Journal of Visual Communication and Image Representation, 58, 495-502.