Offloading the Computational Overhead of AI-application to the edge devices for face mask detection using Hybrid Computing Framework

MSc Research Project
Cloud Computing

Sumit Verma
Student ID: 20230851

School of Computing
National College of Ireland

Supervisor: Jitendra Kumar Sharma

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Sumit Verma
Student ID:	20230851
Programme:	Cloud Computing
Year:	2022
Module:	MSc Research Project
Supervisor:	Jitendra Kumar Sharma
Submission Due Date:	15/08/2022
Project Title:	Offloading the Computational Overhead of AI-application to the edge devices for face mask detection using Hybrid Computing Framework
Word Count:	6858
Page Count:	20

Signature:	Sumit Verma
Date:	15th August 2022

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Offloading the Computational Overhead of AI-application to the edge devices for face mask detection using Hybrid Computing Framework

Sumit Verma
20230851

Abstract

Cloud has become the important part of every business and organizational needs. Availability, scalability, Pay-per use model, easy connections with RESTful APIs are some of the popular features of cloud. Therefore, most of the business applications are highly depending on the cloud infrastructure to solve their business problems. On the other hand, Artificial intelligence technology is helping the businesses with automization of various applications and processes and these artificial intelligence-based applications utilizes the various deep learning and machine learning algorithms, which requires the large amount of storage and computing capabilities to train and run the applications. Due to the unavailability of large infrastructure, most of the AI-applications are being deployed to the cloud for model training and process the user requirements. The dependent application on cloud server requires high network bandwidth to establish the connection with cloud infrastructure, where latency is associated with each connection. The problem further arises when, the number of application users are much higher in number, which generates high amount of network traffic and computational overhead on the cloud server. In order to solve such challenges, this research is aimed to provide offloading-based hybrid computing architecture, where a small portion of code is installed on the edge device itself and dependency on the cloud is reduced. The study on the hybrid computing framework has been performed over the face mask detection application, which can detect the mask on the human face in the real time. This application has been deployed over both the cloud and hybrid computing based model and evaluated with different experiments and metrics.

1 Introduction

The modern era has introduced people to various connected technologies. Over this, technological advancement and adoption have been increasing more than ever. As the adoption rate of the connected technologies increased, the informational database also increased proportionally. Primarily, these data were stored in the local system which provided better security and privacy. Although, it couldn't handle the increasing database due to the limited storage, computational capacity, and ability. To combat the limitations, heavy investment had to be incurred to increase the server capacity. This again led to another limitation of higher input cost. Therefore, in the world of connected technologies, the cloud system could effectively combat the limitations with improved output. The cloud system is an over-the-air architecture which is distributed module of hardware over the internet and provided a reliable solution. This system provided on-demand storage, computational capacity and ability with reliability, convenience, accessibility, and feasibility. Several organizations and service providers in the technological domains have shifted their databases and system to the cloud computing architecture. From a wider perspective, the cloud system will benefit various businesses and technological users for an effective resolution towards increasing demand for computational capacity and ability. With this, it has several real-world applicationsAljanabi and Chalechale (2021).

On the other hand, in recent years, with the outbreak of lethal viruses and diseases, the healthcare system has been destabilized and was prone to collapse. The outbreak of viruses such as coronavirus has impacted various domains of human life considering mentally, physically, financially, and more. Due to coronavirus, several people have passed away through the transmission of this deadly virus. These occurrences have led to strict restrictions and curbs across the world both inter-border and intra-border. Moreover, to this, considering the novelty of the virus, there were no immediate remedies available for such virus. Therefore, several researchers, scientist, doctors, and healthcare practitioners around the globe joined their hands to bring an effective medical cure. Although research on the effective cure took a certain time. Meanwhile, the restrictions and curbs for a longer time could cause disruptions in multiple aspects such as economic, mental health and more. To overcome the disruptions, the certain preventive measure had to be taken. These measures should effectively allow human interaction by hindering the spread of the deadly virus. Some of the measure's sets were social distancing, proper sanitization, wearing of face masks and more. Although, certain people didn't consider the preventive measure which led to the spread of the deadly virus. Therefore, the authorities had to keep a check on the people in the public area to implement the preventive measure strictly**?**.

Considering the Face Mask measures, the authorities utilized both human interventions and technological interventions. Although the human interventions helped only to a certain extent which further lacked accuracy. To combat this challenge, technological interventions were enhanced and combined with artificial intelligence (AI). Through this, human interventions and effort could be eliminated thereby improvising accuracy and efficiency. The AI model is constructed utilizing the machine learning (ML) and deep learning (DL) algorithms and incorporated into the camera system. The video obtained through the cameras is fed into the model and interpretations or detections are obtained. This process is done in real-time settings. As this model generates a large amount of input information and heavy computational processes, it isn't suitable to operate in the local system. Therefore, the implementation of the cloud system is the most optimal solution here. It is due to its expandable computational and hardware capacity and ability. Although the implementation of the model in the cloud system has some fallacies. Firstly, it is due to lesser privacy than the local system. Secondly, its over-the-air or over-the-internet architecture may lead to a lag in data transfer, server downtime or latency overhead. Thereupon, it is observable that there are certain considerable features in both the local system and the cloud system which are supplements to each other. To overcome these challenges, the hybrid computing mechanism is being studied in this paper. In hybrid computing, both the local system and cloud system are utilized based on computational intensity. This type of computing work on the process called offloading. The concept of offloading is when the system transfers certain parts of models to the local devices which are less computationally intensive, and the cloud system processes the higher computationally intensive data. Through this mode, the latency overhead, lag and delay due to server downtime can be overcome. Furthermore, it provides a faster output with better efficacy and output. In addition to this, there are various other advantages of offloading in hybrid computing whether it be reduced financial inputs to improve task flowBhadani and Sinha (2020).

Hybrid computing with its advantages has a wide range of applications for AI. In our study, we have considered face mask detection. This detection model will be constructed on the different ML and DL algorithms which can effectively segregate the person with the mask and without the mask. In addition to this, the implementation of hybrid computing allows the offloading of the less computationally intensive task on the local system and heavy tasks on the cloud system. The process of obtaining the images and segregation is done by the local system whereas the pre-processing, feature extraction and implementation are done by the cloud system. The study is interpreted through its performance and accuracy which are evaluated through PRF score, accuracy score, runtime and more. In the next section, we will discuss the study related to offloading in the hybrid computing environmentWang et al. (2019).

1.1 Research Objectives

The purpose of this study to achieve the following research objectives.

- To implement the Cloud and Hybrid Computing Architecture.
- Implementation of Face Mask detection Application.
- Implement the concept of Offloading and analyse the performance difference between the Cloud and Hybrid computing architecture.
- Discuss the merits and demerits of both the approaches.

1.2 Research Question

RQ1: In what scenarios the hybrid computing architecture can be an efficient solution as compared to the cloud-based architectures?
RQ2: How Hybrid Computing architecture help to reduce the latency and computational overhead on the cloud server? How does it perform for AI-application such as Face mask detection?

2 Related Work

In this section, we have discussed the studies by various researchers on offloading in the cloud computing environment. The paper is further divided into subsections namely offloading in cloud computing based on computer intelligence, hybrid computing framework and its application and face mask detection using hybrid computation.

2.1 Offloading in Cloud Computing Based on the Computer Intelligence

In the paper by Wang and the Team (Wang et al., 2019), detailed research was done on the challenges, approaches and aspectual overview of the edge cloud offloading based on the artificial intelligence algorithms. The study suggested different types of algorithms based on the implementation and applications of servers. For the single cloud server, two types of offloading algorithms are MAUI and Clone Cloud. Although considering the multiple cloud servers, there were certain fallacies in these algorithms such as resource allocation and parallel computing, therefore to overcome the challenges, two other algorithms were also suggested which were Think Air and Cloudlet. Furthermore, different online modes of balancing and offloading were consideredBhattacharya et al. (2021).

On the other hand, Cao (Cao et al., 2019), reviewed the intelligent offloading in multi-access edge computing. Due to the pacing usage of the connected and distributed technologies, the load on the servers has been exceeding. Therefore, to effectively offload the data, different machine learning and deep learning algorithms were considered in the paper. In addition to this, different advantages, disadvantages and applications were considered in the paper. Furthermore, the paper did a comparative analysis with different algorithms and parameters. In the conclusion, it was stated that the model was efficient enough for the wider adaptation, although these needed to be enhanced before the real-world application. Similarly, Miao and the team (Miao et al., 2020) proposed an intelligent task prediction and computation offloading based on mobile-edge computing. For this purpose, the proposed model considered the LSTM algorithm. Then the model was evaluated using certain factors. Furthermore, there were certain conditions to be met which were computation tasks to be implemented directly to the mobile devices and the computation must be offloaded to the cloud framework. In addition to this, the study also stated the future scope of work. The future scope of work included the three tasks which were joint optimization, MEC management and security and privacyCao et al. (2019).

A novel approach to mobile cloud offloading in heterogeneous wireless networks using the Lyapunov optimization-based trade-off policy has been proposed by (Li et al., 2022). Initially, the paper discussed the research paper by various other researchers and stated certain views. Then to overcome certain fallacies, the others considered the implementation of Lyaponouv policies. Various parameters of this study are also considered with the implementation. In addition to this, different other offloading optimization parameters are also considered which are the Lagriagian Optimization method and multistage scholastic programming method. The primary focus of this study is offloading optimization. A deep learning-based hybrid technique was used to investigate an autonomous compute offloading mechanism in Mobile Edge computing by Ali (Shakarami et al., 2021). This strategy was presented to address numerous issues during offloading, including long computing times and significant resource usage. To counter this, the author used different decision-making approaches for the study, including the hidden Markov model, hybrid model, multiple linear regression, and deep neural network. Other problems and experiments were mentioned in the reportChen et al. (2020). Furthermore, the study conducted extensive evaluations and comparative analyses on several approaches to arrive at the ultimate conclusion. Overall, the suggested technique could successfully deliver offloading outputs with different criteria such as latency, computational energy, performance ratio, and selection, according to the results of the assessment.

In addition, the authors specified the authors' future areas of study for real-world applications. Joeng (Jeong et al., 2018) suggested a compute offloading strategy for machine learning web applications in the edge server environment in a study. The authors looked at the snapshot-based offload architecture for DNN-based ML applications on the WebKit browser. The key goal here is to strike a compromise between computing time and accuracy while also protecting the user's privacy and security. The final model evaluation revealed that the suggested model may achieve improved performance, security, and reliability. Additionally, this technique allowed for less customization during the offloading operation, decreasing computational strain. Furthermore, the authors here outlined the future scope of the study, which will focus solely on the snapshot offloading assignmentDai et al. (2017).

Chen (Chen et al., 2020) investigated compute offloading and job scheduling for DNN-based applications for cloud-edge computing. The genetic algorithm and the greedy algorithm were presented in this work to address the domain's primary difficulties. The main issue here was the excessive latency, which might generate a backlog in the process. The author conducted various comparison evaluations and experiments to evaluate the module's performance. Each method suggested has distinct benefits and disadvantages in terms of performance, computing time, and so onDos Anjos et al. (2021). However, in the future, the author advised that learning algorithms be used for the study. In addition, Joeng (Jeong et al., 2020) presented PerDNN, a method for outsourcing deep neural network computations to ubiquitous edge servers. To deploy the DNN application effectively, the process must have lower latency and more computing capabilities. As a result of the widespread distribution of the job among the local system and the virtual system, the task might be processed with reduced latency and more computing efficacy. The model estimates the necessity and duration based on client movement prediction. Furthermore, by making such a forecast, the initial cost component is decreased since the cold start situation is eliminated. In the end, it was determined that the model's efficacy increased by 58 percent, lowering the cold start scenario by 90 percent**?**.

2.2 Hybrid Computing Framework and its Application

In contrast, Anjos (dos Anjos et al., 2021) provided a technique to reduce energy usage and elapsed time for IoT applications in a hybrid architecture. Most technical alternatives in mobile edge computing settings rely on batteries, which have a lengthier computational process. These analogues outsource the programmes to the over-the-air architecture to lessen the computational workload on the batteries. The author devised the TEMS algorithm to balance total expenses and computational processing time. This algorithm determines the most cost-effective and time-efficient processing path. During the evaluation, the model showed that it could achieve a promising outlook. Moreover, to this, the final outlook was conditioned as required**?**.

The utilization of cloud agents in the cloud computing environment has been performed with utilization of hybrid computing architecture (Pashinin and Bogdanova, 2021), The proposed system was providing the heterogeneous distribution of microservices. Various sophisticated programmes or methodologies can be decoded using dew computing in a hybrid environment. Even in the absence of internet access, the local system has modest processing capabilities. It also provides a stable mechanism for scaling the operation with increased system load. Furthermore, the proposed system performs rapid intelligent convergence, reducing the possibility of computational delay. The authors have recommended the inclusion of the intended user dew agent in this work. In the end, it was discovered that the dew agent automates the provision of numerous computing resources. [16] investigated NFV chaining in hybrid computing clouds. The purpose of this research was to improve the performance of the virtual network's operation. The author proposed that various networking functionalities be implemented for the hybrid computing architecture in this manner. In addition, various research theses on the current issue were reviewed in the study. In addition to this, numerous fundamentals such as Orchestors, AES NFV, and other algorithms were thoroughly addressed. For a quick comparison, these algorithms were built under various situations. This paper's experiments for VNF included an intrusion detection framework, a VNF deployment process management system, video stream processing, AES encryption, information compression (LZSS), and others. It was also determined that using the graphic processor can improve the efficacy of VNFs.

(Mao et al., 2021) investigated privacy preservation compute offloading for simultaneous deep neural network training in a study. The calculation workload for the DNN network consumes a lot of resources and energy. As a result, the air server is used to tackle these obstacles and effectively repeat the DNN models. Likewise, Shahid (Shahidinejad and Ghobaei-Arani, 2020) investigated a machine learning strategy for offloading computation and resource provisioning in an edge-cloud computing context. The author here contemplated using GPU to minimize calculation time for the suggested approach. [8] investigated the combination of edge computing and cloud computing with distributed deep learning for smart city internet of things in another article. A distributed deep learning-driven task offloading (DDTO) technique was proposed in this research. The authors of an article (Tang et al., 2021) investigated offload compute for low-end orbit satellite networks using hybrid cloud and edge computing. The most significant advantage of this study is that it is organized in a three-tier technical framework. Zhang and colleagues (Zhang et al., 2018) investigated computation offloading for smart home automation in mobile cloud computing. The authors used a heuristic technique based on particle swarm optimization to optimize the workload of a smart house. Silva (Silva et al., 2020) investigated adaptive computation offloading in a hybrid computing environment, whereas Liu (Liu et al., 2018) investigated hybrid-based mobile data offloading in mobile cloud computingJeong et al. (2018).

(Aljanabi and Chalechale, 2021) investigated the hybrid fog-cloud offloading strategy for improving IoT services The resource has been significantly boosted due to the continuous high demand for these services. The work is offloaded from the IoT nodes to the cloud node or fog node to balance the demand level and resource-heavy processing. The most significant benefit of this study is that it considered the two primary characteristics for analysis for offloading, which are load balancing and delay minimization. Based on the sequential choice, the offloading node for each job is evaluated. Additionally, the Q-learning model has been examined. The model effectively provides a definitive forecast for the proper offloading details using the proposed technique called Markov Decision Process (MDP). Similarly, using hybrid quantum-behaved particle swarm optimization, Dai (Dai et al., 2018) investigated hybrid computation offloading for the Internet of Things. Mobile edge computing (MEC) is a much-needed and well-adapted form in today's world. Although the IoT's massive database has resulted in technological improvement. As a result, to improve the overall efficiency, the paper's author developed a unique technique for computing resource allocation based on hybrid quantum-behaved particle swarm optimization. The study examined numerous features of the model and conducted a thorough comparative analysis. In addition, the research demonstrated a simulated examination of the suggested technique to produce the final attitude. Finally, it was determined that iterations were considerably decreased and computing capability was much increasedZhang et al. (2018).

2.3 Face Mask Detection Using Hybrid Computation

Amrit proposed a face mask detection technique using the machine learning algorithm and image processing approach (Bhadani and Sinha, 2020). The proposed model utilized two algorithms which are Mobile Net V2 and the ADAM optimizer. Here, ADAM is a novel algorithm which is an adaptive scholastic optimization algorithm. Furthermore, it was evaluated with other DNN algorithms which are ANN, RCNN, PCA, CNN and FNN. For the dataset, the paper sourced the image from the real-set environment. In the primary flow, the data visualization and pre-processing take place. Once done, the process of feature extraction takes place in the process. Certain required features are extracted here. Then the model is implemented and pre-trained. After this model is implemented and applied in real-time. During the evaluation, it showed a promising result and could be considered for real-world application. Similarly, Loey and the team proposed a hybrid approach of deep transfer learning with machine learning for face mask detection (Loey et al., 2021). The paper considered three datasets which are Real-World Masked Dataset (RWMD), Labelled face in the Wild (LFW) and the Stimulated Masked Dataset (SMD). Primarily, the dataset was pre-processed and feature engineered utilizing ResNet-50 and some of the machine learning algorithms. The ML algorithms utilized were SVM, Decision Tree and Ensemble algorithms. After the implementation, the algorithms were evaluated for performance using the evaluation metrics such as PRF score and accuracy score. With a thorough analysis, the models had moderate output and could be improved.

In a paper by Khatri (Khatri et al., 2021), a hybrid learning approach was proposed for the face detection model. For the implementation, the model was sourced from online repositories such as Kaggle. Furthermore, the proposed model utilized two modules which are the Mobile Net framework and the Caffe model. In the final interpretation, the model achieved an average accuracy of 96.5 percent. For the real-world scenario implementation, the model could be enhanced further. On the other hand, Mrinmoy (sadhukhan and Indrajit, 2021) proposed a novel hybrid approach for face mask detection. Initially, the dataset is sourced, and the further process of pre-processing and feature extraction is done by the DNN algorithms such as the CNN algorithm. Then for the classification, the random forest classifier is utilized. It is then compared with the algorithms such as gaussian naïve Bayes, XGBoost and SVM. Although the model had a poor accuracy of 62 percent. A novel hybrid learning face detection model named FMNet was proposed by Ravikumar and the team (Ravikumar et al., 2022). This proposed model was based on CNN and VGG framework. This model could detect whether the person wearing or not wearing the mask. During the evaluation of the model, it achieved an average accuracy of 99.6 percent . Furthermore, the study stated the future scope of the study for the next study.

3 Methodology

Today, Artificial intelligence has become an important part of modern life. With the increasing demand of data, the computation requirement to process such large amount of data is also increasing. In such cases, the cloud computing-based solutions are being adapted by most of the organizations. Cloud provides hassle free data storage, computation, globally accessible and scalable solutions. However, to utilize the cloud services high bandwidth network connection is required, where latency is one of major concern specially for the real-time applications. Other than the latency, computation overhead is another area of focus for this research, where on increasing the user base of application the memory/CPU utilization increases to a significant level. Considering such certain limitation on cloud-based architecture, this research is mainly focused to adapt the hybrid computing architecture for real-time AI based applications. In this research a real-time AI-based application has been developed for face mask detection, which has been deployed over the 2 different architectures those are cloud-computing based architecture and Hybrid Computing based architecture. Both the architecture utilizes some common components, the detailed description about these components is shown in further subsectionsJeong et al. (2020).

3.1 System Components

• Edge Devices : Edge device can be any Equipment, which has the availability of the network system, storage and computing capabilities. Edge devices also can have the multiple sensors through which data is collected and sent to the cloud computing environment. In order to run the face mask detection application, the edge devices should have webcam, networking, browser, storage and some computing capabilities. For this research, the edge devices can be used such as Smartphone, Laptop, tablet etc. With these edge devices the real-time video data will be transmitted to cloud server for analysing the face with the mask or not.

• WebRTC : For maintaining the real-time communications, the WebRTC helps to establish the peer-to-peer connections. Almost all the applications, providing the video call facility are utilizing the WebRTC services. The applications such as Facebook Messenger, google hangout, Discord, Snapchat, Google Duo utilizes the webrtc at a large scale for both personal and business use. In this project, face mask detection application will also utilize the WebRTC service through which it can be used any device user for real-time face mask detection.

• Cloud Server : Cloud server is the main component in our system, it is a place where the application resides, model is trained and computation task such as inference is performed. Cloud also provides the application access globally and usually have higher storage and computational capabilities as compared to the local servers, systems or devices. In order to run the multiple applications, cloud server contains the multiple microservices, each microservices hosts the different set of applications where each service run on its own processors and provided computational capacity, it is also responsible for communication with the help of Restful APIs.

3.2 Cloud-Server Architecture

The main component used for Cloud-server architecture are explained in Section 3.1, in this architecture the main functionality of the application is dependent on the cloud server. Where the cloud server stores the vast amount of data to train the machine learning/deep learning models. In order to train the AI-based models, a high computational capability resources such as CPU, RAM, GPU are required. In this architecture, the edge devices transmit the real-time video streaming data (video frames) from the browser to the cloud server using the WebRTC. The WebRTC connects with the associated microservices of the cloud and provides the videos frames to the microservices in order to perform the inference, generated by already trained deep learning model. The deep learning model mainly identifies the mask from the human face, add their prediction outcomes (labels) and send back to the edge devices with the help of WebRTC. The edge device displays the predictions on the video frames provided by deep learning model in cloud server over the canvas in the browser. In this process of rendering and inference, a high network bandwidth along with a high computational memory is being utilized by the microservices to server the customers and this memory consumption increases further, when a greater number of customers are using the application. As the application is highly depending on the cloud server, the latency in the performance can be observed due to the dependency on the internet, which can delay the prediction on device screen. The Cloud-server based architecture for Face mask detection application is shown in Figure 1.

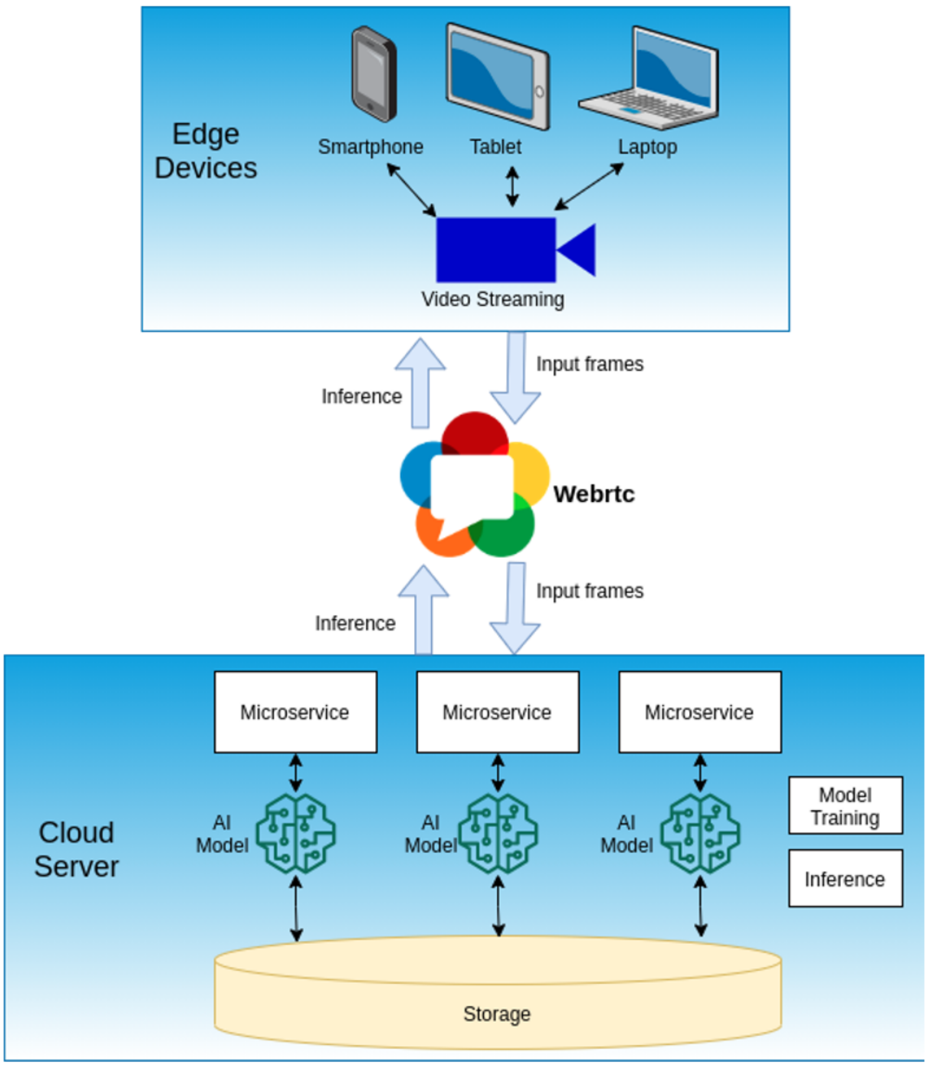


Figure 1: Cloud Server Based Architecture for Face Mask Detection Application

3.3 Hybrid Computing Architecture

The concept of Hybrid computing architecture is different than the cloud-server based architecture. However, both the architecture utilizes the same set of components as explained in Section 3.1. The concept of the hybrid computing based architecture is the combination of both the cloud and local system-based computation. In the process of Hybrid computation, the edge devices establish the one-time connection with the microservices on the cloud server, during this connection, edge device request for the already trained models available over the cloud server microservice. Cloud server transfer this model to the requested edge device, once the model is downloaded on the edge device, the inference task such as frame extraction, label prediction can be performed on edge-device itself. However, the task which requires heavy computational resource, large storage system such as model training, evaluation is still performed on the cloud-servers. On the other hand, the ligh-weight operations such as inference, prediction can be performed on the edge-device as most of the devices are highly equipped with CPU processors, RAM and even GPUKhatri et al. (2021). Hybrid-computing architecture is scalable solution, as computation task are being performed on the local device itself, which does not create any computation overhead on the cloud server, even if the number of application users increases significantly. The flow diagram of Hybrid computing architecture for Face mask detection application is shown in Figure 2.

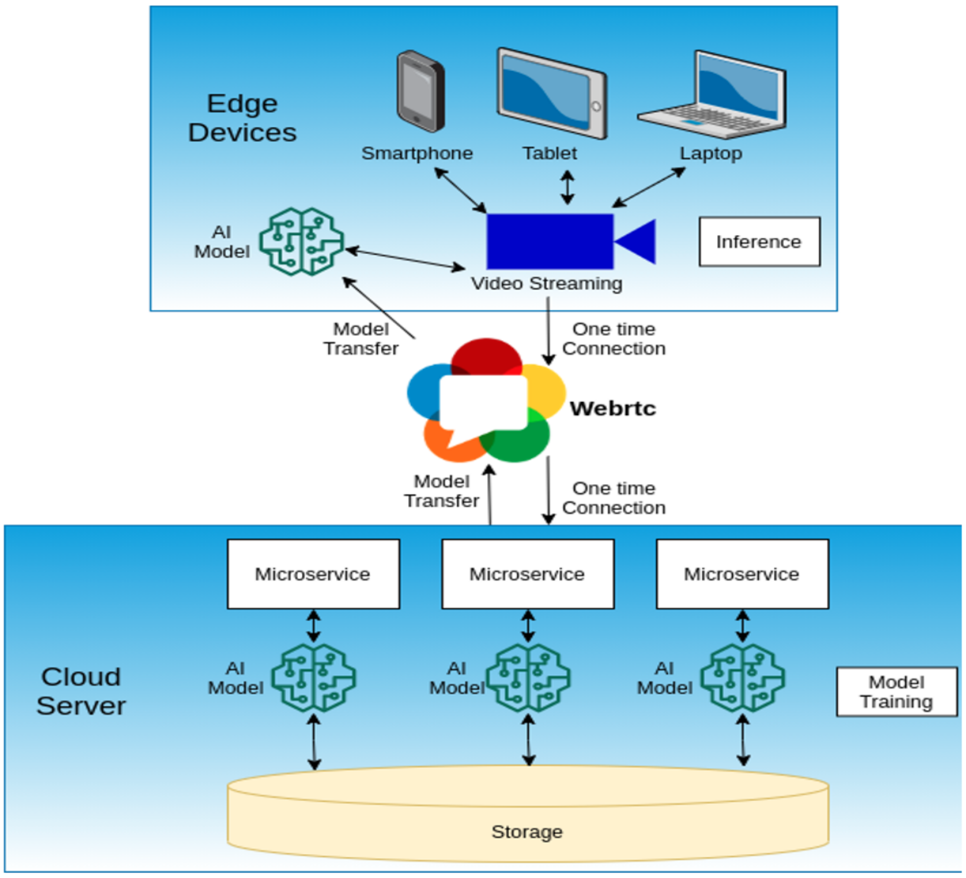


Figure 2: Hybrid Architecture for Face Mask Detection Application

4 Design Specification

In the Chapter 3, a high-level overview of Hybrid and cloud-server architecture has been provided. However, the detailed explanation about the working mechanism of application still needs to be discussed. Therefore, in this section we will discuss about the functionality, features and working mechanisms of face-mask detection application. As the advance deep learning algorithms are utilized for development of this application, several sets of process have been followed, which will be discussed in further subsections. The detailed flow diagram of face mask detection application is shown in Figure 3.

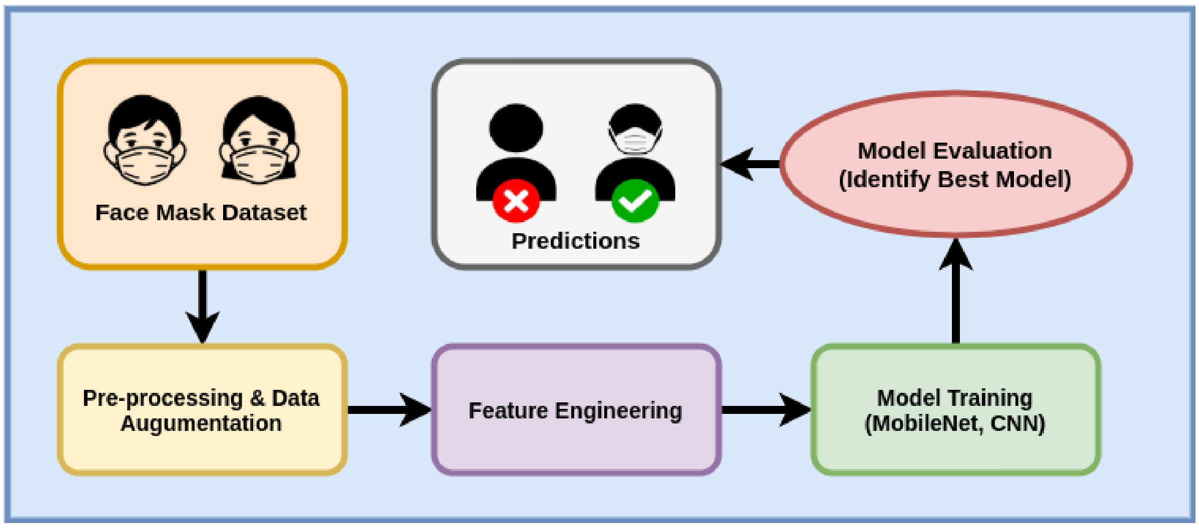

Face Mask Dataset
Predictions
Model Evaluation (Identify Best Model)
Pre-processing & Data Augumentation
Feature Engineering
Model Training (MobileNet, CNN)

Figure 3: Flow diagram of Face Mask Detection Application

4.1 Dataset Description

To apply the deep learning models, the first step is to collect the dataset, the dataset should contain the images of human with face mask and without mask, which has been collected from Kaggle (Face Mask Detection Dataset, 2022)Gurav (2022). The dataset contains the 7500 RGB images where the images are classified as with mask and without mask. The dataset is balanced where approx. 3700 images of faces with mask are available and remaining 3800 images of faces without mask. The sample dataset containing the labels with mask and without mask is shown in Figure 4.

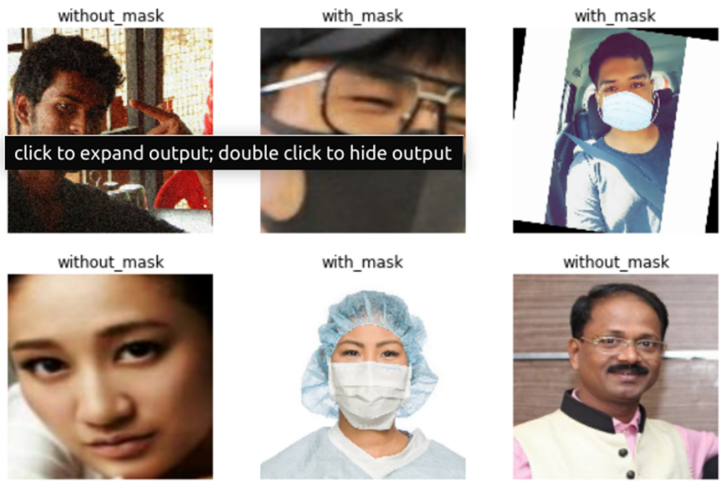


Figure 4: Sample Dataset (with labels)

4.2 Data Pre-processing Image Augmentation

In order to enhance the model performances and outcomes, the facemask data needs to pre-processed, where height and width of images are set to (160X160) size. Images augmentation has been performed where the existing images are stored with different camera angles, different angle of rotation and flip operations are performed. TensorFlow provides the inbuilt support of data augmentation.

4.3 Model Training

There are 2 different models, which has been utilized for accurate classification between the mask and without mask images. The models used for this research are Convolutional Neural Network (CNN) and MobileNet. Among which the mobilenet model is a pre-trained model. TensorFlow along with keras library has been used for model training, where the number of epochs is set to 10.

4.4 Model Evaluation Prediction

Model Evaluation is the most crucial step, where the performance of each deep learning model is monitored and different performance metrics are calculated to get the most optimal and accurate model for face mask detection. There are 2 different metrics have been used in this research that are accuracy and loss. Model with highest accuracy and minimum loss will be identified as most optimal model for face mask detection. The Validation accuracy and validation obtained from each algorithm after training will be discussed in the further subsections?.

4.4.1 Accuracy and Loss of Convolution Neural Network (CNN)

CNN model has been trained over the 10 number of epochs after following all the required steps such as data augmentation, pre-processing and feature extraction etc. After these steps CNN model has been developed from scratch and trained over the face mask dataset.

The performance of model has been examined over the test set, where it has been observed over every epoch the accuracy of model is increasing and loss is decreasing. The highest validation accuracy obtained using CNN model is 97.29 percent and minimum validation loss obtained using CNN model is 0.121. The training accuracy is found to be continuously increasing, which represents no overfitting in the modelLi et al. (2019). The validation accuracy and validation loss over every epoch is shown in Figure 5.

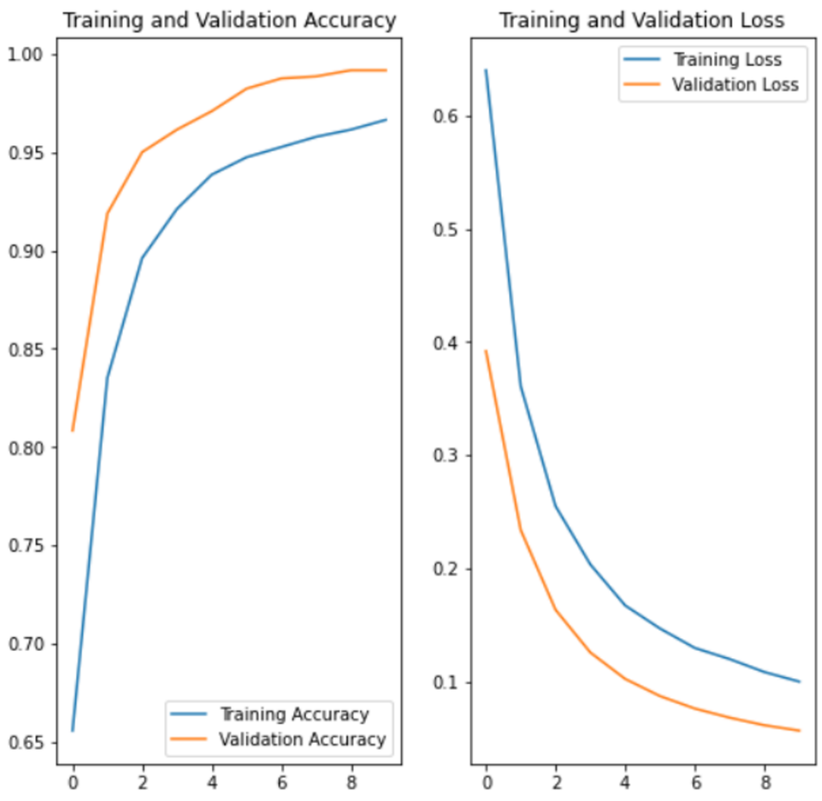


Figure 5: Validation Accuracy and Validation Loss using MobileNet-V2

4.4.2 Accuracy and Loss of MobileNet-V2 (Pre-Trained Model)

Another effective pre-trained model MobileNet-V2 has been utilized for this research, which has been trained over the face mask dataset. This model is also has been trained over the 10 number of epochs with face mask dataset. The obtained training and validation accuracy along with loss is shown in Figure 6. After analysing the line graph, it has been observed that MobileNet-V2 has achieved the highest accuracy of 99.17 percent over the test data. On the other hand, minimum loss obtained by MobileNet-V2 model is 0.056. The obtained outcomes from MobileNet-V2 is quite optimal as compared to the CNN modelLi et al. (2019).

After analysing the results of both the models it can be concluded that MobileNet-V2 model is correctly able to classify among the images with face mask and without facemask. Thus, the mobilenet model will be saved in the cloud server and will be utilized for real-time classification of with mask and without mask images.

5 Implementation

After selecting the most optimal model for face mask classification, the application will be developed and will be deployed over both the architectures (Cloud-server and Hybrid Computing). The implementation of these architecture is performed with the help of various tools and technologies.

5.1 Implementation of Cloud-Based Architecture

For the Cloud based architecture the video stream is sent to the Cloud server frame by frame from the Edge device browser, this communication between the edge device browser and the cloud server is handled using the web-rtc. We are using the python implementation of webrtc (aiortc) on top of aiohttp apiMao et al. (2020). To ensure the users privacy and security our proposed architecture does not allow the video streams to be sent across un-encrypted(http) channels.In order to achieve that ngrok client is used to quickly create a secure https tunnel. On the server side the image is preprocessed and passed through the model for inference, as this transfer and processing happens for every frame the load increases for every new application user added into the system. Due to such certain limitation and high dependency on network bandwidth, the solution is not scalable at all , especially in case of real time apps(face-mask detection). There are several sets of other libraries which has been used for developing the application such as TensorFlow, keras, numpy and matplotlib. OpenCV library has been used for pre-processing the image during the training and prediction phase. As the dataset contains mainly the images, the dataset can be trained over the GPU in a faster manner. On running the cloud based architecture the delay in the video frames has been observed, when the number of users were increasing more than the server capacity, the denial of service also has been observed.

5.2 Implementation of Hybrid-Computing Based Architecture

Most of the libraries utilized for Cloud-based and Hybrid computing based framework are similar. Although the working mechanism of both the architectures are different from one another. As described in the Chapter 3.3, for hybrid computing architecture the trained model over the cloud server is saved and stored in the Json Format using tensorflow js. On the client end, the edge device downloads the AI assets (model and logic) and stores it into the browser's local storageLiu et al. (2018). Once the model is downloaded, then no transfer of data takes place to the cloud and when the system needs to run the model, it does on the client side(locally). The edge devices utilize, tensorflow js with webgl as backend for gpu hardware acceleration. The web-rtc utilizes the media devices such as web-cam from client machine and establishes a secure https-based tunnel. There are certain advantages of using Hybrid computing architecture as the network latency involved in transfer of video frames from client side to server side and vice-versa is removed. The computation overhead on the server associated with every user is negated, which makes the system highly scalable. The detection of face mask with our developed application is shown in Figure 7.

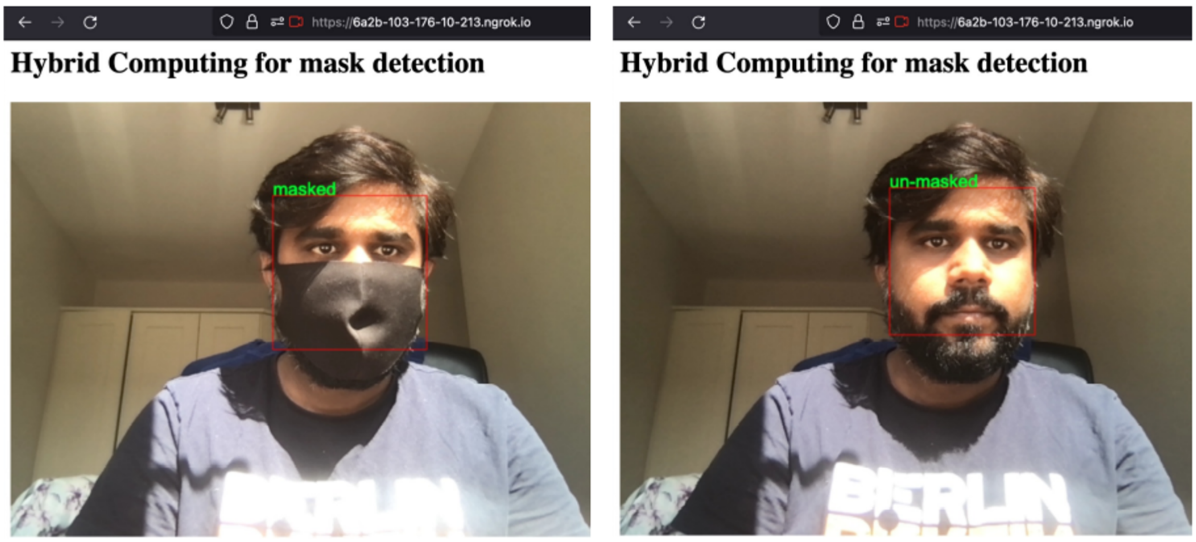
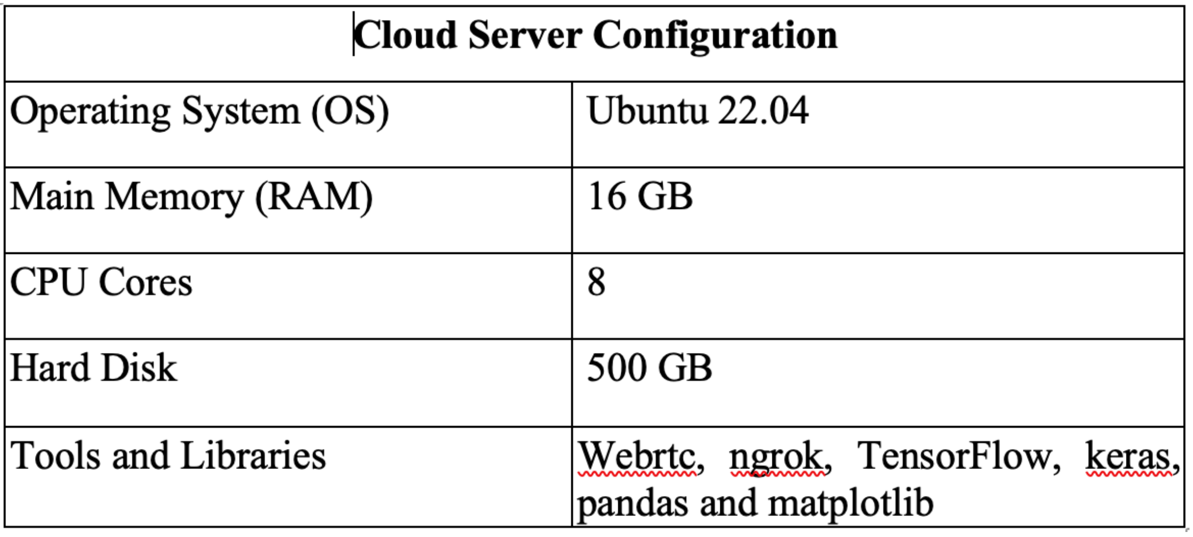

https://6a2b-103-176-10-213.ngrok.io
Hybrid Computing for mask detection
masked
https://6a2b-103-176-10-213.ngrok.io
Hybrid Computing for mask detection
un-masked

Figure 6: Identification of Face Mask using Laptop (Hybrid Computing)

Our developed application does not need to be installed on any system, it is purely browser based, all the computation will be also consumed even with browser with the help of Ngrok and webrtc. As our developed application is browser-specific it can run on any device having the camera and browser support. However, the response of application dependes on the computing capability of device, as inference and extraction of video frames is performed locally on device. Our cloud server consists of following specificationsLoey et al. (2021).



Cloud Server Configuration	
Operating System (OS)	Ubuntu 22.04
Main Memory (RAM)	16 GB
CPU Cores	8
Hard Disk	500 GB
Tools and Libraries	Webrtc, ngrok, TensorFlow, keras, pandas and matplotlib

Figure 7:

6 Evaluation

The developed application can be hosted over any cloud computing platform such as AWS, Azure or Google Cloud Platform (GCP). The main objective of developing any cloud-based application is to provide the services to multiple user at a single point of time. The main difference between the two architectures can be observed in terms of CPU and Network Bandwidth utilization. Therefore, with our experimental set those reading will be taken, which will help us to identify the optimal architecture for real-time face mask applicationTang et al. (2021).

6.1 Experiment-1 / Cloud-Server based Architecture

Our application has been executed over the cloud-based architecture, where CPU utilization has been measured with the increasing number of users. In this architecture, the edge devices continuously send the video frames in the real-time to the cloud server for labelling with mask and without mask and sends back those videos frames back to the edge device. This architecture is based on the two-way communication. In this architecture for any type of processing, the complete dependency is on the cloud server. Therefore, when the number of users has been increasing, a significant rise in the CPU utilization has been observedSilva et al. (2020). Where the number of users served depends on the computing capability of cloud, our system configuration was able to serve only 5 users at a same time for face mask detection application. The CPU utilization with respect to user for face mask detection application is shown in Figure 8.

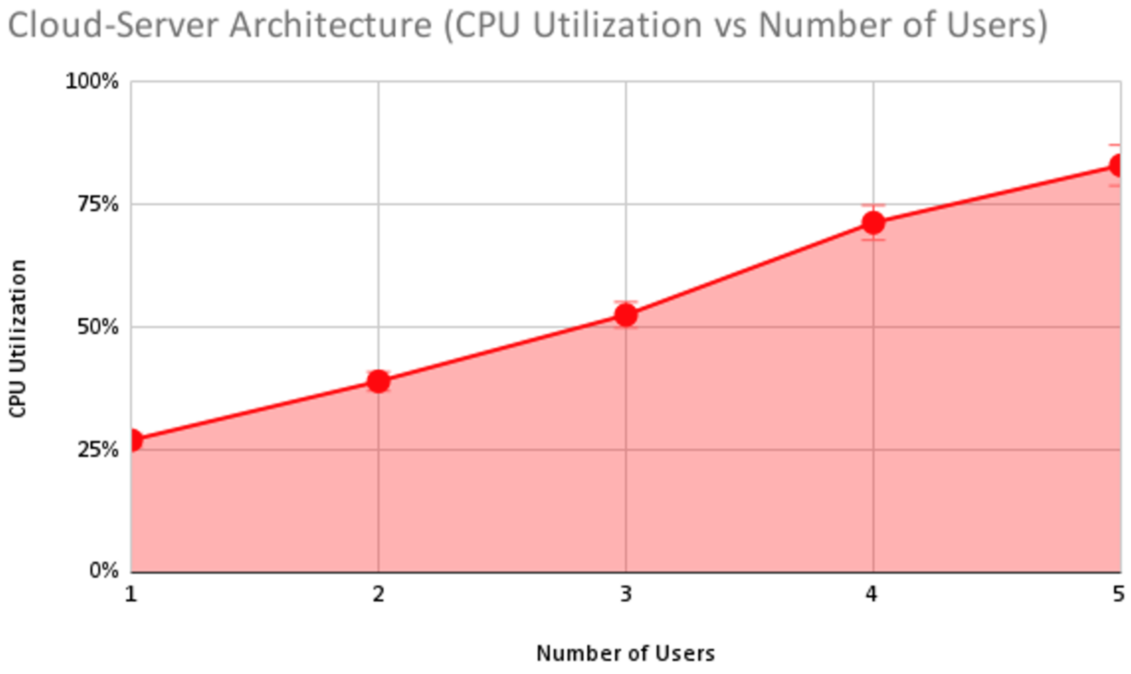


Figure 8: CPU Utilization vs No. of Users in Cloud-Server Architecture

After analysing the graph as shown in Figure8, it has been observed that when 1st user has been running the application, CPU utilization was approximately 25 percent, which has been increased further on adding the 2nd user where the CPU utilization raised to 38 percent approx. Similarly, on increasing the number of users the CPU utilization was found to be increasing. With the 8 Cores of CPU, the cloud server was able to serve only 5 users at a same time.

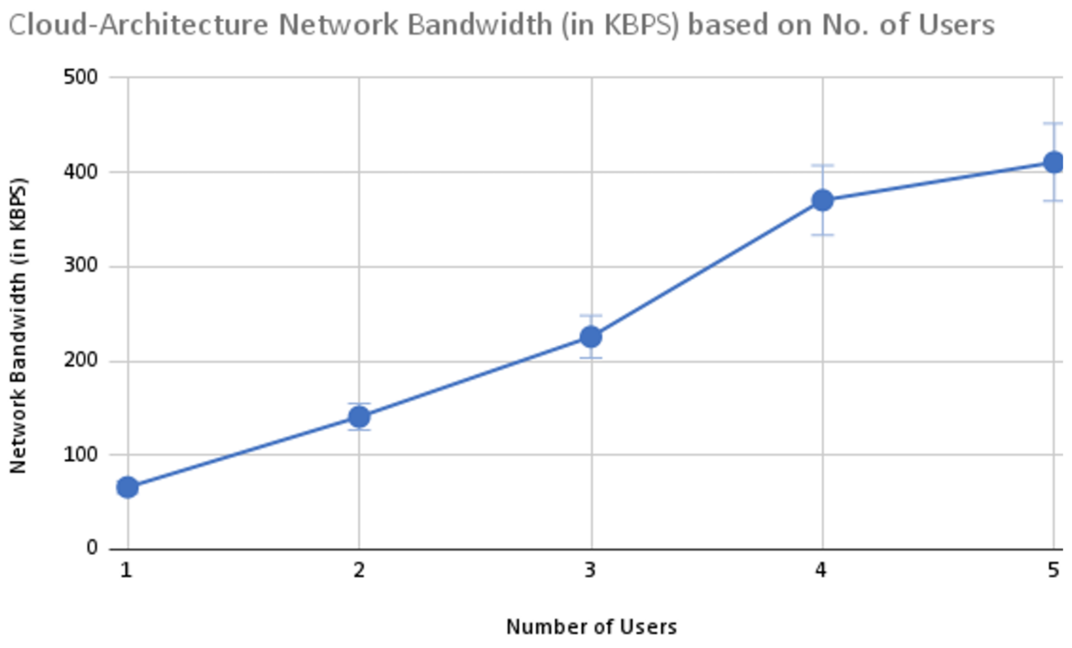


Figure 9: Network Bandwidth vs Number of Users in Cloud-Server Architecture

Another observation on the cloud-server architecture has been performed based on the Network bandwidth utilization, as the application is purely depending on the cloud server a high amount of bandwidth is utilized for receiving the videos frames and send them back. Therefore, with respect to every user the network bandwidth has been calculated. The line graph indicates the network utilization of cloud server with respect to multiple number of users shown in Figure 9. An significant increase in network utilization on cloud server has been observed, with increasing number of users. While running the application, a delay in the video frames has been observed due to the latency issues with network connectionPashinin and Bogdanova (2020).

6.2 Experiment-2 / Hybrid Computing based Architecture

An overall working mechanism of Hybrid-computing based architecture is discussed in the Chapter 3.3. As hybrid computing based architecture establishes the one-time connection with the cloud server to download the AI assets such as (Model, configuration and Logic). Only network bandwidth is utilized when the Model is being transferred to the edge device of user (mainly during the offloading). After successful deployment of model to the edge device, the dependency on the cloud is no more required, as the computation required for Face mask application is performed by local device itself which reduces the computation overhead to a great extent and provides scalability with respect to the number of users. With our experimentation and observation, more than 10 application users were able to utilize the face mask detection application at the same point of time without creating the computation overhead on cloud and also no delay has been observed in terms of computation and network bandwidth utilizationShahidinejad and Ghobaei-Arani (2020).

6.3 Discussion

In this research, an offloading mechanism for hybrid computing with the real-time AI application has been explored. The Hybrid computing architecture was not only able to reduce the computational and latency overhead but it is also scalable solution to deliver the services to maximum number of users. On the other hand, with the traditional cloud-server based architecture high latency, delay in video processing and computational overhead has been observed. With the cloud-server based architecture, only limited number of user applications can be served that depends only the computing capability of cloud server such as RAM, CPU Cores and GPUShakarami et al. (2021).

7 Conclusion and Future Work

In this research, the main goal was to analyse different architecture for deploying AI-based application using the advancement of modern-day browsers such as webgl, webassembly etc. Nowadays almost every website or portal on web has some sort of AI/ML analytics build into it. These AI/ML analytics engines are a wide range of models varying in size and models generally having higher computation requirements then any of the conventional BI (Business Intelligence) algorithms also serving these cloud-based application to multiple application users generates an enormous amount of computational overhead on the cloud server. After certain set of experiments and analysis this can be concluded that offloading based mechanism such as hybrid computing is useful to solve the latency and computation overhead challenges by distributing the prediction model to local device itself. Hybrid computing model is only functional, when computation requirement of this code is smaller than the actual capability of device. If the computational requirement of deployed application model is higher, it may affect the other processes and service of the edge devices and application may not function properly. However, hybrid computing based solution is very useful for tiny applications such as identifying the face mask from human face. There are certain tasks and application, where complete dependency on the cloud server is necessary and required for exchanging the data with synchronization. In such cases, cloud server-based application has their own advantages. Training the model and correctly identifying the mask from human face was another challenging task for this research, after evaluating the performance of MobileNet-V2 and CNN model, it has been found that mobilenet algorithm was correctly able to recognize the mask from human faces with high accuracy and minimum loss. Highest accuracy score of 99.17

References

Aljanabi, S. and Chalechale, A. (2021). Improving iot services using a hybrid fog-cloud offloading, *IEEE Access* **9**: 13775–13788.

Bhadani, A. K. and Sinha, A. (2020). A facemask detector using machine learning and image processing techniques, *Eng. Sci. Technol. Int. J* pp. 1–8.

Bhattacharya, I. et al. (2021). Hybridfacemasknet: A novel face-mask detection framework using hybrid approach.

Cao, B., Zhang, L., Li, Y., Feng, D. and Cao, W. (2019). Intelligent offloading in multi-access edge computing: A state-of-the-art review and framework, *IEEE Communications Magazine* **57**(3): 56–62.

Chen, Z., Hu, J., Chen, X., Hu, J., Zheng, X. and Min, G. (2020). Computation offloading and task scheduling for dnn-based applications in cloud-edge computing, *IEEE Access* **8**: 115537–115547.

Dai, S., Liwang, M., Liu, Y., Gao, Z., Huang, L. and Du, X. (2017). Hybrid quantum-behaved particle swarm optimization for mobile-edge computation offloading in internet of things, *International Conference on Mobile Ad-Hoc and Sensor Networks*, Springer, pp. 350–364.

Dos Anjos, J. C., Gross, J. L., Matteussi, K. J., González, G. V., Leithardt, V. R. and Geyer, C. F. (2021). An algorithm to minimize energy consumption and elapsed time for iot workloads in a hybrid architecture, *Sensors* **21**(9): 2914.

Gurav, O. (2022). Face mask detection dataset, *Disponıvel em: https://www. kaggle. com/omkargurav/face-mask-dataset. Acesso em* pp. 04–02.

Jeong, H.-J., Jeong, I., Lee, H.-J. and Moon, S.-M. (2018). Computation offloading for machine learning web apps in the edge server environment, *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*, IEEE, pp. 1492–1499.

Jeong, H.-J., Lee, H.-J., Shin, K. Y., Yoo, Y. H. and Moon, S.-M. (2020). Perdnn: offloading deep neural network computations to pervasive edge servers, *2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS)*, IEEE, pp. 1055–1066.

Khatri, N., Jayswal, A. K. and Tyagi, R. (2021). Face mask detection using hybrid approach, *2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO)*, IEEE, pp. 1–4.

Li, Y., Xia, S., Cao, B., Liu, Q. et al. (2019). Lyapunov optimization based trade-off policy for mobile cloud offloading in heterogeneous wireless networks, *IEEE Transactions on Cloud Computing* .

Liu, D., Khoukhi, L. and Hafid, A. (2018). Prediction-based mobile data offloading in mobile cloud computing, *IEEE Transactions on Wireless Communications* **17**(7): 4660–4673.

Loey, M., Manogaran, G., Taha, M. H. N. and Khalifa, N. E. M. (2021). A hybrid deep transfer learning model with machine learning methods for face mask detection in the era of the covid-19 pandemic, *Measurement* **167**: 108288.

Mao, Y., Hong, W., Wang, H., Li, Q. and Zhong, S. (2020). Privacy-preserving computation offloading for parallel deep neural networks training, *IEEE Transactions on Parallel and Distributed Systems* **32**(7): 1777–1788.

Pashinin, A. A. and Bogdanova, V. (2020). Application of user dew agent in hybrid-computing environments., *AICTS*, pp. 135–145.

Shahidinejad, A. and Ghobaei-Arani, M. (2020). Joint computation offloading and resource provisioning for e dge-cloud computing environment: A machine learning-based approach, *Software: Practice and Experience* **50**(12): 2212–2230.

Shakarami, A., Shahidinejad, A. and Ghobaei-Arani, M. (2021). An autonomous computation offloading strategy in mobile edge computing: A deep learning-based hybrid approach, *Journal of Network and Computer Applications* **178**: 102974.

Silva, J., Marques, E. R., Lopes, L. M. and Silva, F. (2020). Jay: adaptive computation offloading for hybrid cloud environments, *2020 Fifth International Conference on Fog and Mobile Edge Computing (FMEC)*, IEEE, pp. 54–61.

Tang, Q., Fei, Z., Li, B. and Han, Z. (2021). Computation offloading in leo satellite networks with hybrid cloud and edge computing, *IEEE Internet of Things Journal* **8**(11): 9164–9176.

Wang, J., Pan, J., Esposito, F., Calyam, P., Yang, Z. and Mohapatra, P. (2019). Edge cloud offloading algorithms: Issues, methods, and perspectives, *ACM Computing Surveys (CSUR)* **52**(1): 1–23.

Zhang, J., Zhou, Z., Li, S., Gan, L., Zhang, X., Qi, L., Xu, X. and Dou, W. (2018). Hybrid computation offloading for smart home automation in mobile cloud computing, *Personal and Ubiquitous Computing* **22**(1): 121–134.