Masked Face Recognition using Deep Learning and MlOps

MSc Research Project
Cloud Computing

Akash Parapurath Govindarajan
Student ID: x20122101

School of Computing
National College of Ireland

Supervisor: Divya Elango

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Akash Parapurath Govindarajan
Student ID:	x20122101
Programme:	Cloud Computing
Year:	2021/2022
Module:	MSc Research Project
Supervisor:	Divya Elango
Submission Due Date:	16/12/2021
Project Title:	Masked Face Recognition using Deep Learning and MlOps
Word Count:	6345
Page Count:	21

Signature:	Akash Parapurath Govindarajan
Date:	31st January 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):	

Masked Face Recognition using Deep Learning and MlOps

Akash Parapurath Govindarajan
x20122101

Abstract

Facial recognition is a method of recognizing or verifying a person's identification by looking at their face. Identifying the person while wearing a mask is more important in this pandemic situation. In this paper, we proposed the use of a Deep Learning algorithm to extract the feature therefore it helps to identify the person behind the mask. This will improve security even in this pandemic situation. A Deep Learning Neural Network or DNN, a widely used image processing algorithm, which is implemented to recognise the masked person. Moreover, Azure Ml is a cloud service which is used to implement the MLOps technology in our end to end application. This project is developed end to end with Flask and the process is automated with the help of Azure Machine Learning. Through this we developed an automated lifecycle for our Masked face identification model. We achieved an accuracy of 91% in recognizing a person.

1 Introduction

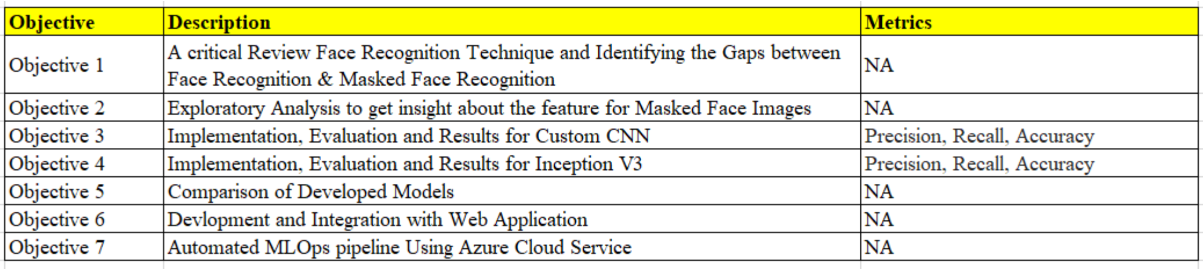
Artificial intelligence (AI) is a rapidly developing field with a diverse set of applications and research. The consequence of merging human intelligence with machine processing is artificial intelligence. Sub-domains such as machine learning and deep learning can be utilized to train a machine how to perform a task. We'll concentrate on Deep Learning, which can be used to train for image recognition. We utilize some algorithms to recognize faces because they are the most important aspect of picture recognition. The covid-19 is a pandemic situation of wide spreading of this coronavirus. Due to this wearing of mask is very important to reduce the widespread of this disease. Recently so many researches are done with coid-19 one example is shown in this paper Pathak et al. (2021). Facial recognition is a method of recognizing or verifying a person's identification by looking at their face. Biometric mass surveillance allows firms and governments to follow and identify us based on our person's unique data anytime, anywhere, and wherever we go. Face recognition and other biometric surveillance technology are increasingly being used in public places such as streets, train stations, protests, and sporting events to identify the person's true identity. This demonstrates that our ability to be anonymous in public spaces, our ability to simply be, is under assault. People can be identified in pictures, films, or in real time using facial recognition technology. Biometric security involves facial recognition. However, due to this pandemic, the whole concept of this face recognition has been collapsed by wearing a mask. So in this project we have overcome this major drawback by developing a model that can recognise the person while wearing a mask. This is implemented with the help of Deep Learning Neural Network or DNN which will train the model with very little features like Eyes, Eyes-brows and Ears with the idea of similar classification done in this paper (Nezamoddini and Gholami; 2019). We specifically used the Convolutional Neural Network (CNN). From the paper we can say Wang et al. (2020) this CNN is an efficient image processing deep learning type that is frequently used in computer vision. CNN employs a multilayer system that includes numerous convolutional layers, pooling layers, and fully connected layers, as well as an input layer, output layer, and hidden layer. On the validating test, we can see that accuracy is approaching 90%, indicating that a CNN model is performing well on accuracy measures. This model is developed completely with an end-to-end application that can be used to solve the business problem. We have used Flask which is a python web framework that is used in back-end integrated with deep learning model. And also this framework helps to build a responsive web application for our Masked Face Recognition model.

Moreover, Cloud computing is now widely used for business development. Cloud is being utilized for cloud architecture planning, organizing, and designing for cloud deliverable models. This study makes use of the cloud to achieve cloud hosting. DevOps (Development Operations) plays a vital part in the deployment of their application in every company. We can integrate software developer ops and IT ops with the help of DevOps Macarthy and Bass (2020). We will take DevOps to the next level of implementation in this study. Because our research project is completely developed with a deep learning algorithm to test and train the model, we have integrated Deep Learning and DevOps to provide a complete development solution. This implementation is called MLOPs. The MLOPs work concentrates on deploying and maintaining Machine Learning models in live environments. In order to make the entire ML cycle, MLOps takes many of the principles from DevOps to work more effectively and is easy to operate. MLOps' main purpose is to automate machine learning pipelines and measure timings and other metrics. The pipelines will be used for several iterations during the ML project's life-cycle. We'll use this to combine masked facial recognition with the MLOps technique to create an automated deployment pipeline for our study. This MLOps implementation is done with the Azure Cloud Service. Microsoft Azure Machine Learning is a set of services and technologies aimed at assisting developers in the development and deployment of machine learning models. These tools and services are available through Microsoft's Azure public cloud. Azure Machine Learning has so many advantages like Creating reusable software environments and reproducible machine learning pipelines. Models can be registered, packaged, and deployed from anywhere. Capturing governance data for a Machine Learning lifecycle, Notifications, and Alerts for events in the ML lifecycle. Azure Pipelines and Github may be used to build a self-contained and continuous integration system for training a model. The ML Lifecycle's majority of processes can be automated.

1.1 Research Question

How accurately can the face be recognized while wearing mask with the implementation of streamlined MLOps?

1.2 Research Objective



Objective	Description	Metrics
Objective 1	A critical Review Face Recognition Technique and Identifying the Gaps between Face Recognition & Masked Face Recognition	NA
Objective 2	Exploratory Analysis to get insight about the feature for Masked Face Images	NA
Objective 3	Implementation, Evaluation and Results for Custom CNN	Precision, Recall, Accuracy
Objective 4	Implementation, Evaluation and Results for Inception V3	Precision, Recall, Accuracy
Objective 5	Comparison of Developed Models	NA
Objective 6	Devlopment and Integration with Web Application	NA
Objective 7	Automated MLOps pipeline Using Azure Cloud Service	NA

1.3 Motivation

Face Recognition is one of the most discovered parts by many Scientists. Previously, Research on face recognition is a great topic where all the researchers build their innovation in this field to solve the business problem. This face recognition is being used in day to day life of people. For example, we use these face recognition techniques for security purposes, prevent retail crime, unlock phones, find missing persons, help the blind, etc. So this complete system is collapsed due to the wearing of masks in this pandemic situation. By this research, we can overcome this situation by directly identifying a person behind the mask. Through this, we can bring back face recognition into the world. In this paper, we use Deep Learning Neural Network to effectively identify the person behind the mask. And also this is developed as a complete business model by integrating a Machine Learning Operation(MLOps) with the help of the cloud service. The idea of this MLOps originates from the DevOps which is used in the development environment in Software development. Through the implementation of this cloud gives a developed model to perform as an end-to-end web application.

1.4 Contributions of the research

The rest of this document is written as follows. Firstly, The related works are analyzed and examined critically to identify the research gap in Section2. The methodology for the study is described in this Section3, and the steps that are carried out during the study. The design requirements details about the innovative methodologies applied in our project, the model architecture, and the algorithms used in the project are described in the following Section4. The next section goes over and specifics the implementation and the tools that were used while developing, Section5 evaluates and discusses the performance of a model that is implemented with CNN and Azure Machine Learning using appropriate tools. Finally, the conclusion and the future works are discussed in detail in Section7.

2 Literature Review

2.1 Different approaches for Masked Face Recognition and MLOps

2.1.1 Deep Learning approach:

Researchers are devising an improved solution to old problems on the planet using the potential mechanisms of machine learning and deep learning concepts. Neural networks are being used by researchers to solve problems in healthcare, technology, security apparatus, and space technology, among other fields. This shows the growth and usage of the technology in current world.

Many methods are used to detect faces; in this research, they employed R-CNN, which is a faster form of picture detection. With R-CNN, each region of the recommended network is likewise improved. To detect the multiple scale-faces, they trained and evaluated the data. VGG-16 was used to train the model in this case. This has a unique S-type classification that allows for two face detections. Even when the images are tiny, R-CNN can recognize them. The outcomes of R-CNN are compared using various models. Face detection accuracy is 89.6 percent when using multi-scale R-CNN. Anchor points are used to obtain feature maps from various CNN layers. The study Sun (2020) did not show how to accurately identify several faces in a single image.

The Covid-19 crisis has pushed all researchers to explore the next stage in facial recognition technology. Because face recognition is the only form of identification that can be verified through physical touch. Face recognition is mostly used for security purpose. A company that uses fingerprints for attendance must move to facial recognition. In this study Mundial et al. (2020), they present a system for identifying a person wearing a mask. They practice with datasets of people wearing masks and without masks. D-CNN is used to extract the feature, and SVM is used to categorize the facial feature. They employed supervised learning to execute the person's identification. We may learn suitable extraction techniques from these research Mundial et al. (2020); WU et al. (2019), which will aid us in detecting more features of the masked face in our model.

Face recognition in the scene of wearing a mask. Face recognition research on occlusion issues is less than that on gestures, lighting, age, and expression. For partial occlusion, they proposed a new effective face recognition system. They detect face and mask occlusion areas with the upgraded MTCNN. The decoding approach from the prior research helps us to extract all the features of the non-occlusion area after detecting the mask occlusion area, then feed the extracted features into the support vector machine to recognize the face. The non-occluded region of the face image is used to recognize the face. Any facial recognition system's performance is determined by the classifier that is employed to recognize the face. This research Qi and Yang (2020) offers a face identification technique based on upgraded convolutional neural networks to increase the recognition accuracy of current face recognition systems. CNN is specifically applicable to computer vision involving picture categorization and object recognition in deep learning research. CNN training is crucial, as it impacts the success of network training as well as the final recognition rate. The number of model layers, training algorithm, dropout value, and optimization procedure are all taken into consideration.

Gabor wavelet and deep transfer learning are used to tackle the problem of the face and mask recognition in this research. To build a more robust feature vector for optimized recognition, the Gabor wavelet feature is taken from the non-masked part of the face and combined with deep learning CNN features. This paper O et al. (2021) econsiders that Gabor wavelets seem to be the best starting point for separating primary highlights for face identification. From their research, we can assume that the Gabor wavelet's recurrence and directed representation properties are remarkably similar to those of the human visual framework. These were discovered to be ideal for surface depiction and separation. Deep CNN models, on the other hand, are seeing a significant push-back in masked face recognition. Moreover one more approach which show the same recognition of person in different scenario's. Masked Correlation Filters (MCFs) are designed to improve performance by leveraging prior knowledge of where partial occlusions would appear in test images, as well as the ZACF approach. For object detection under partial occlusion, They introduced Masked Correlation Filters (MCFs), a new technique of correlation filters. They have adjusted for the occlusion by using past knowledge of the obscured region, which improves recognition performance. MCFs were evaluated on 128x128 face pictures from the CMU-PIE, AR, and KACST databases, and Wang et al. (2020); WU et al. (2019) discovered that they can help correlation filters work better in the presence of massive, continuous occlusion.

2.1.2 Integration Frameworks:

Firstly, in this study Bura et al. (2018), They suggest using distributed cameras, edge computing, data analytics, and sophisticated deep learning algorithms to create a more adaptable and cheap smart parking system. They use cameras with zoom lenses and motorized heads to track vehicles as they enter and exit the parking lot and gather license plate information. They improve the technique and enable real-time deep learning inference in a device at the edge. They also completely implemented this project end-to-end with a help of the web framework Flask. This Flask is a web interface framework that helps to integrate back-end with Websites. By this, they can access and view all the running cameras and parking slots easily with this application. Secondly, from this paper Verma et al. (2021) we gathered that they create a dynamic online application that allows students, professors, and alumni to communicate on a common network. They created a web application that serves as a virtual platform for us to complete a variety of jobs from anywhere in the world. The purpose of this paper is to explain how a web application works, such as how to create a front-end part using React Js, a back-end part using Django framework, how to use a database in web apps to store data in the database, fetch and serve data in the user interface, how to deploy a website in the cloud, how to integrate all components, and how to use machine learning to create a Machine learning NLP Model for text analysis using Sci-kit learn package.By comparing both the paper Bura et al. (2018); Verma et al. (2021), we analyzed that both research implemented the end-to-end development with python. First paper implemented the development with Flask and the responsive of the UI is very good when compared with Second. Second paper developed their application with Django. From critically analysing we adopted the development implementation with Flask Framework.

2.1.3 Machine Learning Operation:

Machine Learning Operation or MLOps tools helps to collabrate and communicate to solve the buisness problems between data scientist and developers. The use of these methods improves the quality of Machine Learning and Deep Learning models, simplifies the management process, and automates their deployment in large-scale production contexts. Models can be more easily aligned with business demands and regulatory requirements. Let's look in details about which help to develop our model.

MLOps with DVC: Barrak et al. (2021) This article investigates the use of DVC in Github projects, 25 of which are examined in depth. More over half of the DVC files in a project are updated at least once every tenth of the project's lifespan, according to our findings. Modern machine learning (ML) applications necessitate complex data engineering, model construction, and deployment pipelines. Because machine learning models and workflows are based on data, it's critical to maintain track of the data used at each step and iteration of the workflow, i.e., capturing the history of the various ML stages, in order to assure the ML pipeline's reproducibility and track data provenance for their project. This paper also shows how the Data Versioning Control (DVC) ML pipeline's complexity changes over time. Through this paper we can analyze the ml pipeline that has been implemented in our project. The development of novel tools in the software engineering process, such as ML versioning tools to manage and recreate data, models, and pipelines through time, has resulted from the software engineering of machine learning applications. Our research on the DVC versioning tool reveals that ML versioning is becoming more common in open source repositories, and that it comes with a significant maintenance cost for developers and data scientists working on machine learning applications.

Azure Machine Learning: The paper Vadlapati et al. (2021) discusses more about the Azure ML and how Azure DevOps makes automation techniques like infrastructure as code available, as well as the seamless integration of verified frameworks like Machine Learning Operations (MLOps) with DevOps automated pipelines to provide and configure the infrastructure that apps require. With Azure ML's advanced functionality, MLops is simple. To see the test progress and examine the failed tests, we must select View test run. The error message, stack trace, console logs, and attachments are all included in the test results, which can be used to debug failed tests. Several strategies are used to detect model data drift and start the model development process automatically. There is a link to the Release that was used to conduct the tests, making it easy to locate the release that performed the tests if you need to go back and analyze the findings later.

Microsoft Azure ML is a cloud-based platform for building and managing machine learning solutions from start to finish. It's intended to assist machine learning engineers and data scientists in making the most of their existing abilities and model frameworks. It also allows businesses to scale, distribute, and deploy cloud-based workloads.The methodology that has been proposed in this paper is the mechanism we devised for detecting fraudulent online credit card transactions is demonstrated. A true positive (TP) transaction is projected to be fraudulent. Remember that the ratio of true positives to true positives plus false negatives is the ratio of true positives to true positives plus false negatives. TP / TP + False Negative = Recall (FN). Decision Jungle (DJ) outperforms Decision Forest (DF) because it has a higher Area Under ROC Curve (AUC) value and a higher Recall. The majority of popular open-source Python frameworks are supported by Azure ML. It effectively offers everything you'll need to build an end-to-end machine learning pipeline. Bot Frameworks, which are skeleton programs for building chatbots, are another aspect. From this paper Shivanna et al. (2020) we have analyzed the flow of this automation with the help of Azure ML. From the comfort of a basic web browser, users may do a range of operations such as importing the data-set, training, splitting, clustering, and a variety of other exploratory and ML tasks. Azure ML includes support for several popular open-source Python frameworks. It includes everything you'll need to set up a complete machine learning pipeline. By comparing both the paper's Shivanna and Agrawal (2020); Shivanna et al. (2020)they implemented ML pipeline using Azure ML which is used for different scenario's.

2.1.4 Development Operation(DevOps):

In this article Vuppalapati et al. (2020) they proposed a DevOps framework for building intelligent Tiny ML dairy agriculture sensors as part of the paper, as well as the benefits that DevOps provides in terms of developing high-quality products in the most cost-effective manner and serving small-scale farmers at the bottom of the economic pyramid. They constructed a future world that is more proactive in ensuring continuous food stability, i.e., ideas drawn from data. TinyML sensors built with DevOps ensure that small-scale farmers can benefit from Artificial Intelligence and Data Science. High-productivity frameworks, such as Azure DevOps, provide product cost savings that can be passed on to small-scale farmers who are struggling and need the tools to improve our planet.

When we need a demand dependable model so we have to adopt different methods and tools for such management operations . Changes must start at the software engineering level when designing cloud applications, mandating the usage of cloud and non-cloud DevOps automation solutions to automate our DevOps operations.This paper's goal Agrawal and Rawat (2019) is to move DevOps to the cloud and make software more accessible. Simultaneously, consider how to enlarge those Integrate DevOps procedures and automation into both public and private environments. Clouds were the primary method used in this study. They show how DevOps and Cloud work together. Collaborate to solve difficulties in the corporate world They give a high-level overview of the DevOps lifecycle. To make the process go faster, specific tools are used for automation testing and deployment. Following the DevOps methodology allows for rapid implementation, security patches, and etc.They can improve their job running performance by practicing this.

2.2 Summary and research gap

The main idea of developing our model arises from a technology that has been used by all the people in the world. Face recognition technology is the main idea that is discussed in our project. This is the topic where most of the researchers did their research and discovered and also implemented their idea to improve the technology. The heading deep learning gives details about the research and ideas carried out with neural networks. Here we observe that how a neural network works like the human brain with a set of algorithms that seek to discover hidden patterns in a batch of data. These papers show a clear understanding of the performance and use that we get with the help of a deep learning algorithm. From the research we analyzed that face recognition technology can be implemented with Neural Networks and also with Image processing tools, but the accuracy of identifying a person is the only difference. Because With image processing tools it only compares with an image but when we perform with Neural Networks it only extracts all the features and identifies by sharing information. The paper Mundial et al. (2020) clear show how to train and fit a model in our case. In our scenario, we use this deep learning algorithm to identify the person with the mask. With the help of these ideas, we developed a deep neural network model that identifies a person with a mask. From these paper's Bura et al. (2018); Verma et al. (2021) , we understand the development style of a business model. Both the development research did their model with different frameworks like Flask and Django. These frameworks are written in python language. Through this, we are able to implement a model with an end-to-end application. The paper shows the integration of web frameworks with the Deep Learning model. With the knowledge of this paper Bura et al. (2018), we adopt a flask framework to implement our project.

Following the end-to-end development of the deep neural network model, we also performed a DevOps implementation in deep learning with the help of Azure Machine Learning. The process of DevOps with Machine Learning is called Machine Learning Operation(MLOps) The paper gives a clear flow of performing automation processes with the help of the cloud. This automation of the Machine Learning lifecycle is demonstrated in this paper Shivanna et al. (2020). In our case, We duplicate the process of implementing the Machine Learning Operation for a Deep Learning Model. This helps us to develop a deep learning automation lifecycle for our model.

3 Methodology

This research is carried out with the knowledge of Deep Learning Neural Network And process flow of the complete development is explained in short in this figure 1. Through this flow diagram we can have a quick idea about the implementation of our model. The highlighted portion of the flow diagram illustrates the significant improvements implemented and also the research uniqueness.

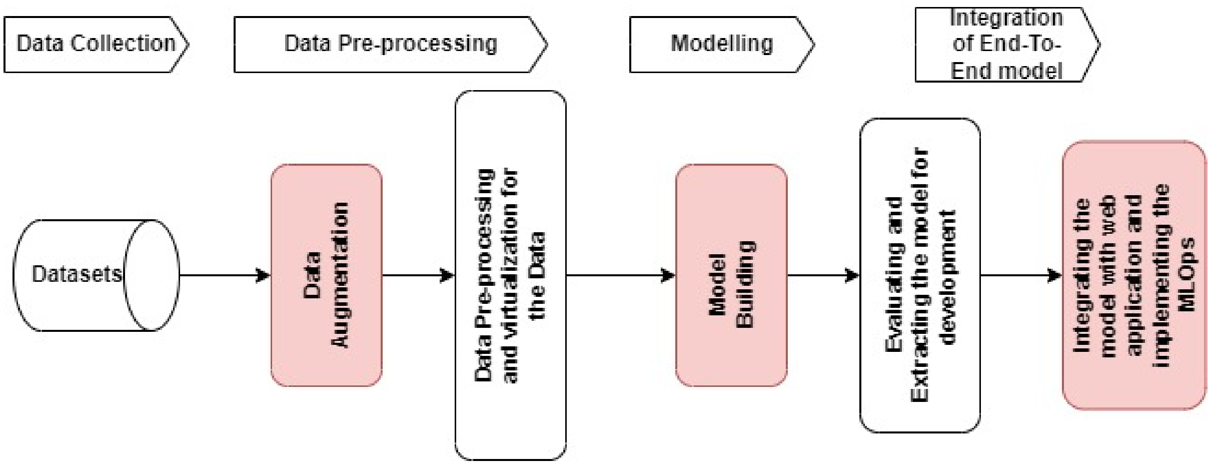

Data Collection
Data Pre-processing
Modelling
Integration of End-To-End model
Datasets
Data Augmentation
Data Pre-processing and virtualization for the Data
Model Building
Evaluating and Extracting the model for development
Integrating the model with web application and implementing the MLOps

Figure 1: Process flow of the methodology followed

3.1 Data Collection

The procedure of collecting, measuring, and evaluating correct insights for research using established approved procedures is referred to as data collection. Based on the information gathered, a researcher might evaluate their hypothesis. Data collection is the important and initial stage of developing a Machine learning or deep learning model. In our scenario, we are developing a deep learning model to identify a person behind the mask.

In this project, we have generated the data by taking the images of a person wearing a mask and labeling them to identify the person. We manually clicked five hundred images for a single person. From the previous study by Vadlapati et al. (2021) we are able to adapt the data collection in our implementation. This data collection is done with the help of OpenCV. OpenCV is a programming library geared mostly at real-time computer vision. So with this, we are able to access the camera and automatically collect the data of a person for training and testing a model. The collected images are around 1000 for two-person and classify them to identify the person and to authenticate them. Through this collected data we have developed a model that can identify a person behind the mask. Let's deep dive into the implementation of steps with the collected data.

3.2 Data Augmentation

Deep learning neural networks' performance is often boosted by the amount of data provided. Only the given datasets are subjected to this data augmentation. Data augmentation is a technique for generating new training data from old data. To obtain new data, we simply made a few minor adjustments to our old dataset. Minor modifications include flips, translations, and rotations. Our neural network will identify these as distinct images in all scenarios. Our planned application might come in a variety of forms, including different orientations, positions, scales, and picture quality. We train our neural network utilizing the modified data to solve the problem of building a proper model. We have increased the photos by rotating them on each side, which is also considered data for training, using this data augmentation technique. We have used this technique and increased our dataset from three hundred images to six hundred images per person. By this data augmentation we able to generate

3.3 Data Pre-processing

The actions taken to format images before they are utilized in model training and inference are known as image preprocessing. This involves some of the techniques such as resizing, orienting, and color corrections, among other things. So the preprocessing is mainly carried out to make the data or images in the same format because if the format of the image varies this can make the model perform very badly. So to avoid this we have done the preprocessing. By comparing both papers Vadlapati et al. (2021) and Baik et al. (2019) we have implemented the pre-processing technique we followed the pre-processing techniques which are followed in this paper Baik et al. (2019). With this preprocessing, we are able to input proper and perfectly pre-processed images to the training of the model. Through this, we can able to develop a perfect model for training and testing.

3.4 Building a Model

In this model building heading, we will see the approaches we have developed to identify the Masked Face Recognition of a person. Here we use Deep Neural Network which acts like the human brain and shares the features with each neuron to perform well. The upcoming heading demonstrate principles on how to utilize TensorFlow to generate custom code and to gain a knowledge of the background operations in a deep neural network. We have implemented three approaches such as Custom CNN, Inception V3 that will effectively perform with image recognition. These Deep Neural Networks help to extract the features from the image to build a perfect model for recognition. Let's deep dive into the model building for each approches.

3.4.1 Custom CNN Approach

Convolutional Neural Networks (CNN) is a type of deep neural network that is frequently used to analyze image data. We implemented the custom CNN model with a Deep learning framework such as TensorFlow and Keras. Both are used together to build a neural network model. Masked Face Recognition dataset is a set of colored images with 1,000 images of each person for training and testing. Here we use 80% of the images for the train set and 20% of the images for the test set. Then we use the labeling that are categorical values, they must be encoded twice. We use keras to encode the values which is in the format of 0 and 1. We used a data iterator to demonstrate a datatypes graph and process. We return a batch size, number of images, and labels with each time of running the model and repeat for an indefinite number of epochs to get a trained model. The epoch will do the forward and backward cycle technique to train a model. Each epoch will train the model with the previously obtained epochs output. This will help to develop a perfectly trained model. By increasing the epochs we can also increase the accuracy. The max-pooling technique is used in this model, this max-pooling will select the maximum element and give a feature map this feature map will give the extracted feature from the image. Activation functions are involved to improve the model accuracy. The figure 2 shows the architecture flow diagram which is used in developing our model.

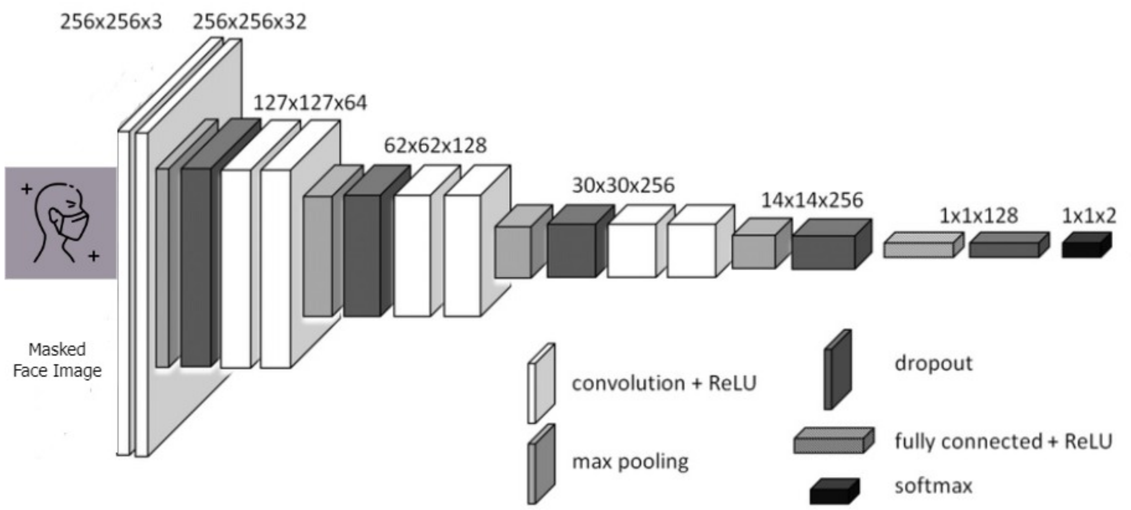


Figure 2: Architecture Diagram For Custom CNN

3.4.2 Inception V3

Inception v3 is a convolutional neural network that is developed from the Googlenet module to assist in image processing and object detection. The Inception v3model is widely used in developing image and face recognition model that has shown a major improvement in model accuracy calculation. The model's symmetric and asymmetric building components include average pooling, convolution, dropouts, and completely linked layers. Batchnorm is utilized extensively throughout the model by appling to activation inputs. Softmax is used to calculate the loss Image preprocessing is an important aspect of the system that can have a big impact on the model's maximum accuracy during training. Model fitting is done by encrypting and resize of the image. In inception V3 special pre-processing technique is done to achieve more accuracy. Image preprocessing is an important aspect of the system that can have a big impact on the model's maximum accuracy during training. For the preprocessing stage, Inception v3 Nabil et al. (2021) provides a variety of solutions, ranging from simple and computationally inexpensive to complex and extremely intensive. By analyzing we build a model with Inception V3 which also gives a good accuracy of identifying a masked person. This model is performed effectively by the use of average pooling, concat, dropout, fully connected networks and softmax. The figure 3shows the architecture diagram of the Inception V3.

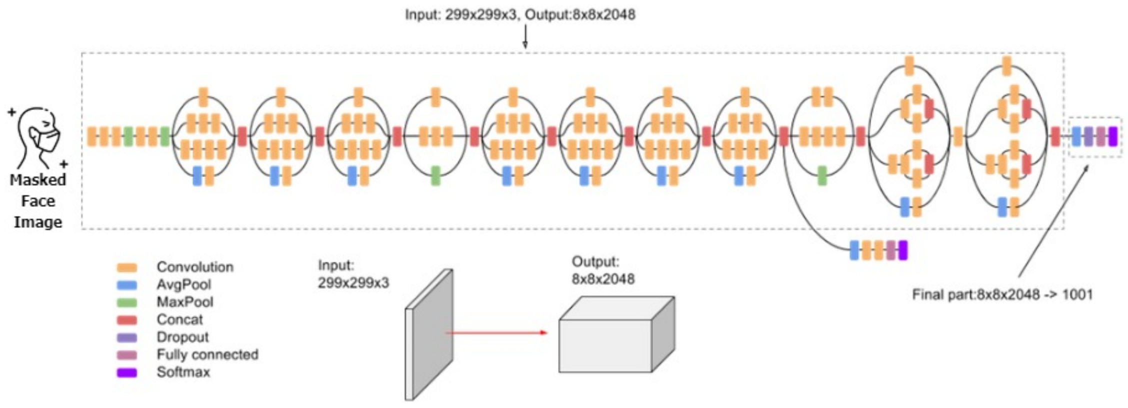


Figure 3: Architecture Diagram For Inception V3

3.5 End-to-End Integration With Framework And Azure Cloud

In our project we developed a end-to end model by integrating a the extracted model with the framework. We use a framework written in python. Flask is an excellent framework for modern programming. On most frameworks, you'll see two ideologies. You have basic micro-frameworks or libraries that give an ecosystem of components that can be mixed and matched as needed, with a focus on only adding what you require. we adopted the model integration from this paper Bura et al. (2018). In this development we use flask framework for the back-end and Html, Css, JavaScript for the Front-end. The framework has so many advantages apart from other languages like java, JavaScript, etc because this is the only framework that supports the integration of deep learning and machine learning model to build a complete business use project. Through excellent support of python language, we have to build an application that authenticates a person by identifying the person with the mask. The integration involves a deep learning model with flask and that flask communicates with front-end UI, this establishes the real-world use model for the user. This frame also supports cloud integration by accessing the package of the cloud service. Here in our implementation, we use Azure Cloud Service. The Azure Cloud Service gives a service called Azure ML, this service provides us a default pipeline creation feature to implement our Masked Face Recognition model. This complete model training is done in Cloud with the help of this service. And also allows us to implement the MLOps technique Shivanna et al. (2020) in our project. The trained model is accessed by the API that is generated by the Azure ML. This API is integrated with a flask to run the model in a development environment. The architecture explains the flow diagram of the Framework and MLops concept in our project.

4 Design Specification

In this Design specification, we will see the exact use and in-depth tweaking of the model. Here we discuss the algorithms which we have used to develop our Masked face recognition project. The tweaking of parameters helps us to analyze and implement different neural network approaches. Here we explain about different layers which perform their role in model building. These layers are the heart of a model because we specify all the required layers and the process which has to be followed in these layers. In our design specification, we have used two approaches such as Custom CNN and Inception V3. Both of them have their own way of building a model, but we give different parameter layers here to achieve the desired output for our model.

4.1 Transfer Leraning Model

Transfer learning is a machine learning technique in which a model created for one job is utilized as the basis for a model on a different task. Given the large computation and time resources necessary to develop neural network models for these business problems, using pre-trained models as the starting point for computer vision. Here we use the inception V3 algorithm. The algorithm iterates all the images and identify the features from the model. Let's discus the use of these layers of which is implemented in training a model.

4.1.1 Inception V3

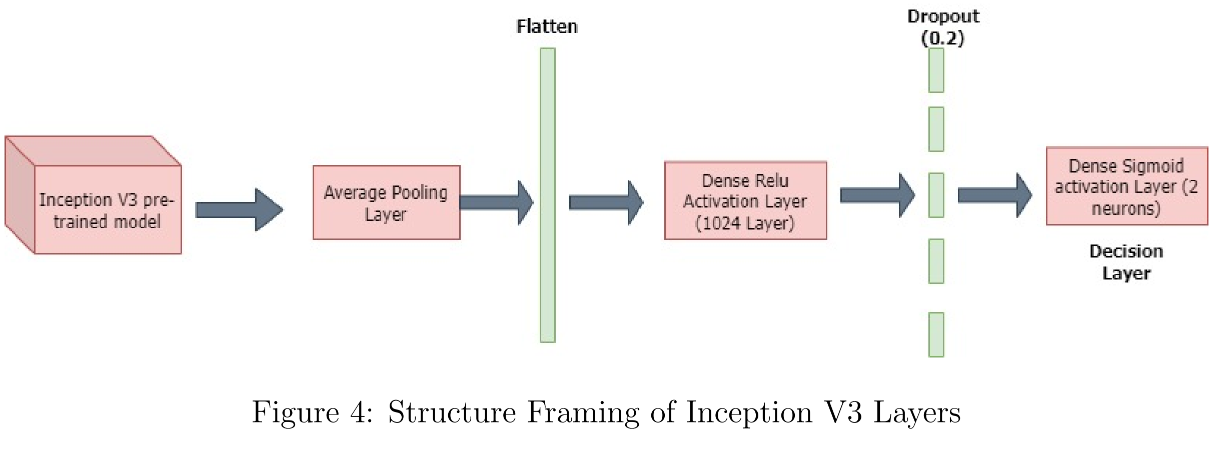

Inception V3 pre-trained model
Average Pooling Layer
Flatten
Dense Relu Activation Layer (1024 Layer)
Dropout (0.2)
Dense Sigmoid activation Layer (2 neurons)
Decision Layer

Figure 4: Structure Framing of Inception V3 Layers

We use inception V3 to train our model. This model is mainly used for image classification. So we implemented this transfer learning inception v3 to classify our masked face images with some of the optionally loaded pre-trained weights. These weights are initialized to list the layers of the model. The trainable weights are also given to update the training loss while training. *"Batch Normalization"* is where we perform these trainable weights. Then we perform *"Average pooling"* which calculates the average score for each pixel, which reduces the size of the image and improves the process of training to train fast. From the input of these pooled images, we do the flattern option to fine-tune the image for processing the model effectively. Next, the flattern images are passed to the activation layer, *"Relu Sigmoid"* activation functions are used to create a linear graph and classify the images by cross comparing. Then *Drop-out* is initialized to randomly pick the images for training, rather than appending all the images in the training model. This complete customization of these layers will finely perform well in the identification of our masked face recognition. The framed structure is clearly shown in Figure 4.

4.2 Custom Convolution Neural Network

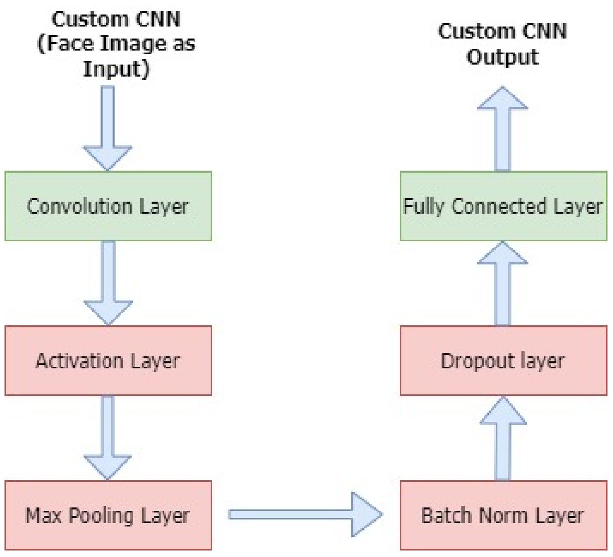


Figure 5: Structure Framing of Custom CNN Layers

Even though we implemented the model with Inception V3, we also carried out model training with the Custom CNN model. This algorithm is also introduced for image recognition. The only advantage is we can customize all the layers in the CNN model building. The building of a neural network has many sequential layers that are organized like a stack. We follow the process from the top by assigning the layer specification that should be followed on each layer. All the layers are interconnected with each other. This is followed to reduce the dimension and identify the patterns for the given image. The first stage of this convolution layer involves *"Conv2D"*. This applies a 3x3 pixel filter for the images and returns a certain feature for the input images that are called Feature Map. Next followed by the *"MaxPooling2D layer"* . This reduces the image dimension by 2x2 pixel size to one dimension, which provides the maximum activation value in the compared grid. This complete process will reduce the size of the image by 50%. The implemented model has 128 layers that are fine-tuned to extract all the features for the Feature Map. Then we perform Batch Normalization which verifies all the activation values given are the same for each batch process. Now after the process we use this "Relu" activation to consider the difference we get from each class. Again we initialize activation for better classification of the image. Instead of using softmax, we'll utilize a "sigmoid" activation function. For this Custom CNN, we finally add the Dense layer that classifies our masked face person by different classes. Finally to improve the learning rate of the model we use this "ADAM" optimizer to perfect the trained output for our model. After stacking all the layers we finally get an output of a perfectly predicted image of the masked face person. The framed structure is clearly shown in Figure 5.

5 Implementation

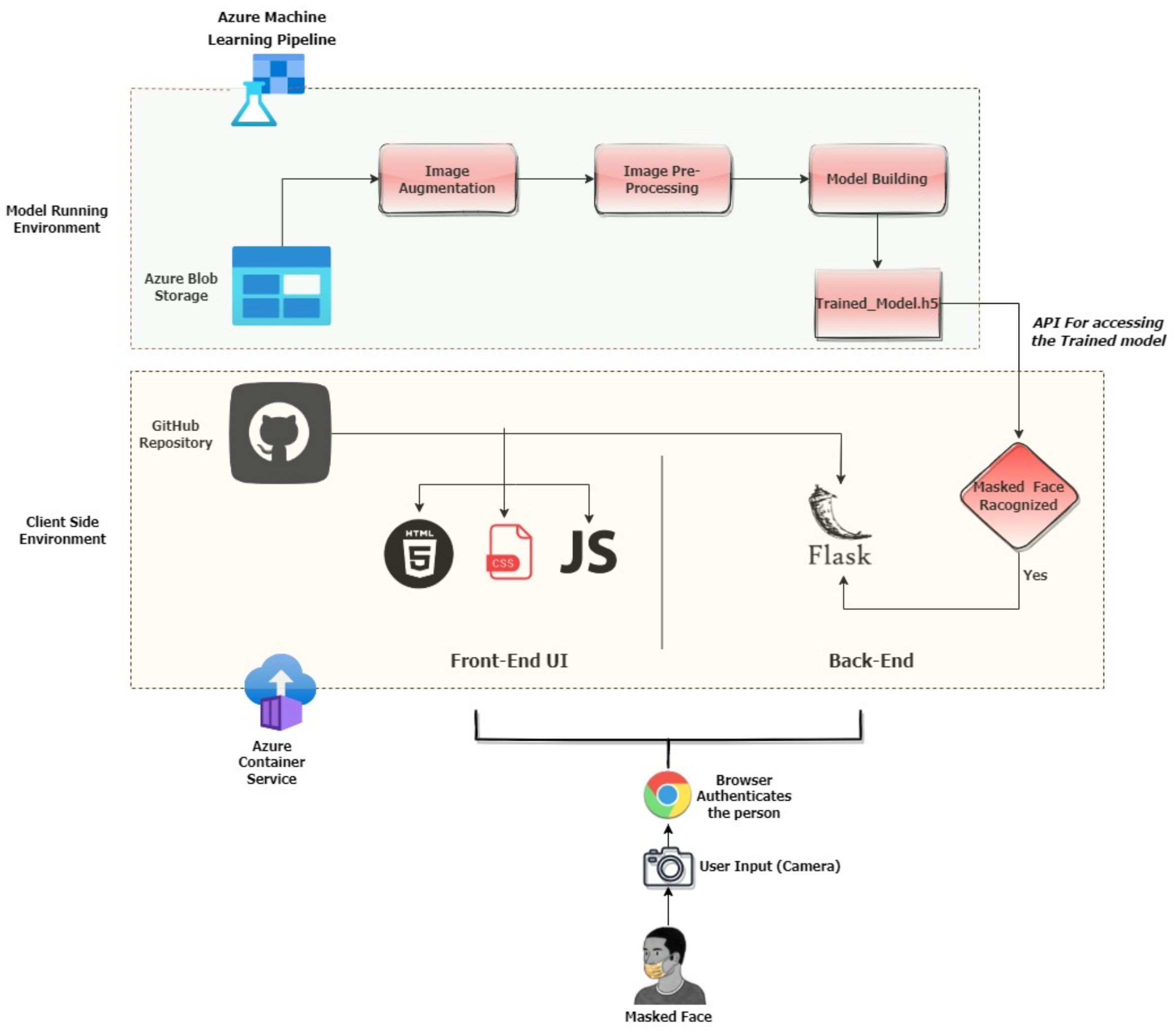


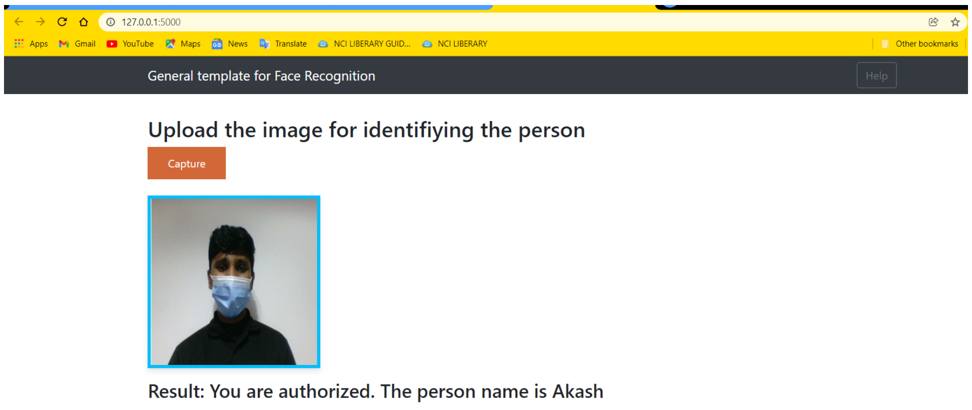
Figure 6: Complete Flow of Masked Face Recognition Application

The implementation of this research is carried out with Python language, Which has the support to develop Deep Learning and Machine Learning projects. Here we use this Python language to handle the complete development of the model and integration with the cloud in the back-end. The model development is done with Keras which supports the TensorFlow which is used to access GPU for running our model. We did this model development with the help of Azure Cloud Service. Using the Jupyter Notebook of Azure ML we are able to develop our Masked Face Identification model. And this model is completely integrated into the web application. We used Flask Framework, which back-end runtime environment to handle the web application with Python. Through this Framework, we are able to integrate our Masked Face Identification model with the web application. The Front End is handled with Html, CSS, and JavaScript to design the UI design for testing our model. This complete application is hosted in an Azure Cloud.

5.1 Preparation of Model

The complete model development is coded with Jupyter Notebook which is an IDE that supports writing deep learning codes. Previously we discussed the models which we developed using Deep Neural Network. For the implementation, we chose the Custom CNN algorithm for the Masked Face Recognition. This model building of Custom CNN is done in the Azure Machine Learning workspace with TensorFlow and Keras. The Figure 6 clearly shows the environment setup made in the cloud for the model development. The dataset of our project is stored in the Azure Blob storage. We do the data augmentation by accessing the data from the blob storage then we perform data preprocessing and finally the model building of Custom CNN. This Custom CNN accuracy is achieved by giving the correct amount of epochs for our masked face identification model. After completing the model building we are extracting the Trained model by h5 file format. This h5 file is a Hierarchical Data Format. And this h5 file is stored in the blob storage. This h5 file is later accessed through the API from flask to test and identify the person through the web interface. The complete model development gives an accurate identification of a person by easily checking through the web application.

5.2 Integration of Model with Web Application



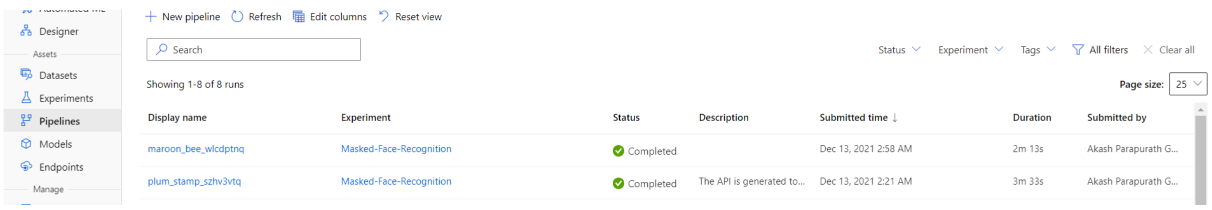
Upload the image for identifiying the person

Result: You are authorized. The person name is Akash

Figure 7: Identification of Masked Face using Web Application

The Development of our application is done with VS Code IDE. It is IDE which has gives a programmer-friendly experience to run the code. Here we use Flask a back-end framework and HTML, CSS, and JavaScript to develop the Front-end. The flask Web framework enables us to construct numerous routes based on a URL, allowing us to provide various features. This allows us to expose several endpoints. The framework also includes useful features such as integrated unit testing, a built-in server, a debugger, and HTML templating. The HTTP protocol takes care of this. What action the server and client must perform in response to various commands and request kinds, as well as how the recognized face information is formatted and delivered. The client sends a Masked face identification request, which the server processes using algorithms and then returns to the client with a response. When a user wishes to identify a person's masked face, all he has to do is send a request to that exact endpoint and wait for a response. Here we use the endpoint from the Azure Machine Learning to access the model. The complete web application is Hosted in a Azure Container. The figure shows the integrated web Application of Masked Face Recognized model.

5.3 Implementation of MLOps with Azure Cloud Service

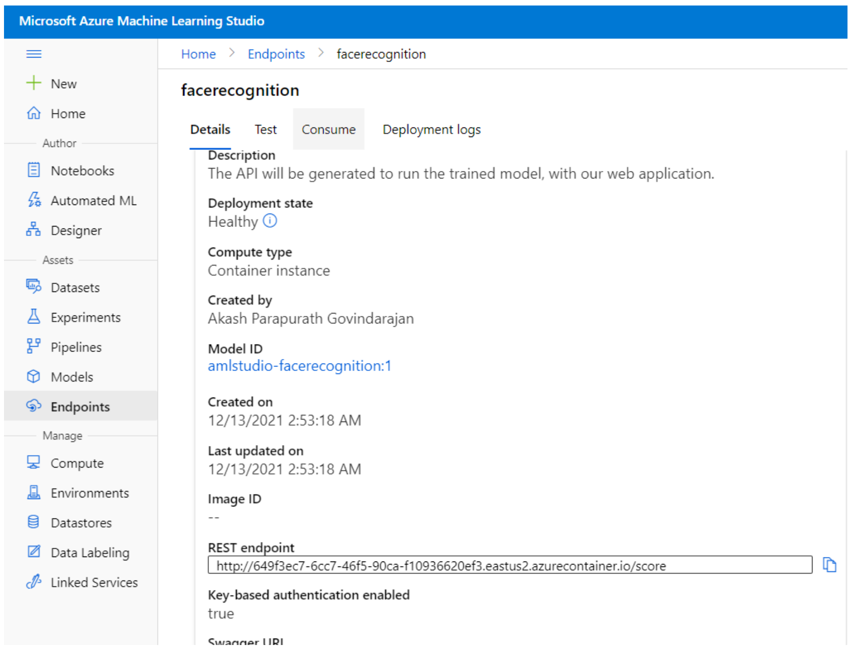


Showing 1-8 of 8 runs

Page size: 25

Display name	Experiment	Status	Description	Submitted time ↓	Duration	Submitted by
maroon_bee_wlcdptnq	Masked-Face-Recognition	✓ Completed		Dec 13, 2021 2:58 AM	2m 13s	Akash Parapurath G...
plum_stamp_szhv3vtq	Masked-Face-Recognition	✓ Completed	The API is generated to...	Dec 13, 2021 2:21 AM	3m 33s	Akash Parapurath G...

Figure 8: Running MLOps Pipeline



facerecognition

Description
The API will be generated to run the trained model, with our web application.

Deployment state
Healthy

Compute type
Container instance

Created by
Akash Parapurath Govindarajan

Model ID
amlstudio-facerecognition:1

Created on
12/13/2021 2:53:18 AM

Last updated on
12/13/2021 2:53:18 AM

Image ID
--

REST endpoint
http://649f3ec7-6cc7-46f5-90ca-f10936620ef3.eastus2.azurecontainer.io/score

Key-based authentication enabled
true

Swagger URI

Figure 9: Azure End Point API for Identifying the Model

Machine Learning Operations (MLOps) is based on DevOps principles and practices that increase the efficiency of workflows.MlOps implies continuous integration, Continuous-delivery, and continuous deployment. We apply these principles in our masked face detection deep learning model. We used the Azure Machine Learning workspace to develop the MLOps Pipeline. The figure shows the running Machine Learning Operation Pipeline for our model. If you made any changes in the code the model is re-trained by collecting the data from the blob and producing the trained model into the production. This is a continuous process this creates a business development environment for a deep learning project. This Azure Machine Learning Service provides proper details with the help of the experiment option. Through this, you can view all the logs of the trained model. The completely trained model is deployed in development by generating an end-point API which is accessed by the flask to communicate our model with the web application. The figure shows the end-point access of our trained model. This server is hosted in the Azure Container instance.

6 Evaluation

The evaluation gives a judgment of all the possible experiments which we have carried out during the model building. After strongly analyzing we have written this complete analysis. So through this, we will understand the different approaches performance and also cross comparing trained model's prediction. We collected images of a person and appended them into a separate folder. This folder name is the label of a person. We tried out certain cases and calculated the accuracy for identifying the masked person. Let's see the complete experiment scenarios while training and testing two models like Custom CNN and Inception V3 in the upcoming headings.

6.1 Experiment 1: Model Training With Custom CNN Model

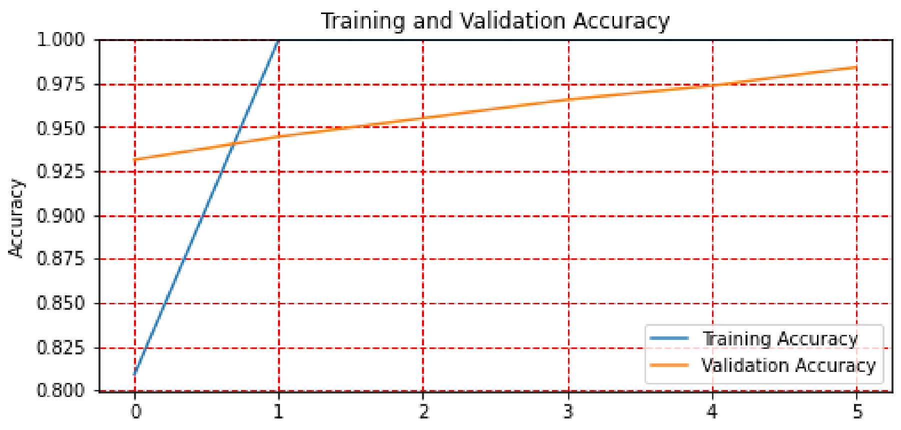


Figure 10: Training Accuracy for Custom CNN model

In Experiment 1, we discussed the experiment results while running a *Custom CNN* model. The Figure 11 shows the experiment results in identifying the model. We have given different epochs to train the model. The *epochs* are the main deciding part while training the model. Let's compare each epochs result's like *loss, accuracy, validation loss, and validation accuracy.* Loss is the bad prediction of training the model. The graph shows the increase of accuracy in each training epochs. Accuracy is a good prediction result. If we get good accuracy, it is a well-trained model. We did ten epochs to completely build a perfectly trained model. This model is extracted for testing the model and integrated with the web application. The Figure 10 shows the comparison Line graph of training accuracy and validation Accuracy.

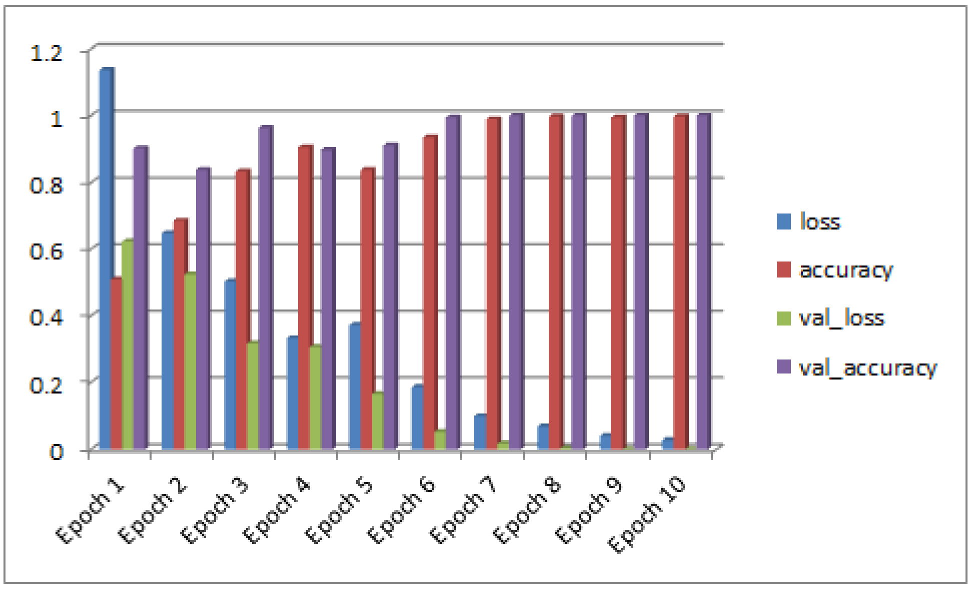


Figure 11: Experiment 1:Accuracy, Loss for each Custom CNN model

6.2 Experiment 2: Model Training With Inception V3

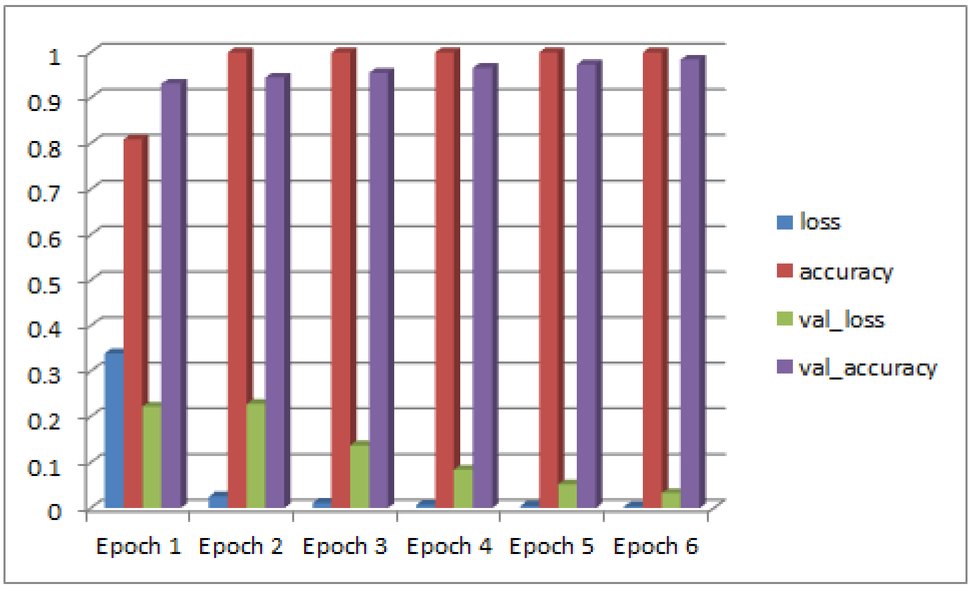


Figure 12: Experiment 2:Accuracy, Loss for each Inception V3 model

The next experiment is the model building of Inception V3. This evaluation and analysis of the model performance are completely reviewed and discussed in this heading. This model experiment results from different epochs are shown in Figure. This model's performance is achieved with fewer epochs. The Accuracy is achieved in the third iteration of the epochs. But the difficulty which I face here is, even though it has good accuracy but the masked face identification is not accurate when it comes to the end-to-end implementation. This is completely free trained so we cannot customize the layer in building the model. This is because the dataset is not sufficient for training the model.

Finally, this model experiment is trained and tested.

6.3 Discussion

Model	Accuracy(%)
Custom CNN	98.50
Inception V3	95.61

Table 1: Obtained Accuracy for each Model

Overall Comparison of this research is clearly described in this subheading. Here for our Masked Facer Recognition model, we developed two models Custom CNN and Inception V3. This Custom CNN is a Customized model by us and this Inception V3 is a pre-trained model with different weights. While coming to comparison The customized Custom CNN gives better performance and accuracy when identifying a masked person when compared with InceptionV3. Inception V3 accuracy is achieved with Fewer epochs but it doesn't perform well when it is integrated with a web application. The Identification accuracy and speed of identifying a person is good with the Custom CNN model when is integrated with the web application. The MLOps feature provided by the Azure ML that are used in our implementation of our model. With this, we create a pipeline that handles and monitor our model. All the models running can be viewed with the UI in the form of a graph. With this Azure ML, we create a model and do the Continuous Integration in our model, and Continuous development, deployment is done by running an Azure Container Instance. This model is deployed in instance and accessed by Rest API generated by the Flask to integrate with the web application.

7 Conclusion and Future Work

The complete research focused on building a deep neural network that Recognizes a masked face of a person. This complete implementation is developed and tested with two models, such as Custom CNN and Inception V3. Here in this research, we have used so many metrics which helped to improve the recognition of identifying a masked face person. Finally, we choose this Custom CNN model for integrating with the web application. In this paper, we overcome all the masked face identification limitations that were faced by the real world during this pandemic. In our case, we chose Custom CNN to perform the masked face prediction with the web application, because it better performing than the Inception V3 in end-to-end implementation. This is also automated by a pipeline using Azure Machine Learning Pipeline. Through this, we perform all the Machine Learning Operation (MLOps) process is carried out in Azure Cloud Service. The MLOps process gives a good business development environment for a run and deploys in the real world.

In the future, the developed Masked Face identification model can be implemented in the real world such as used for security purposes in mobile phones, company authentication, website login authentication, etc. The MLOps process also can be taken to the next level by using the Azure DevOps service which provides more options for using different resources for the application. The complete research is implemented and executed with deep learning to Recognize a masked Face person and authentication is performed through a web application.

References

Agrawal, P. and Rawat, N. (2019). Devops, a new approach to cloud development amp; testing, *2019 International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT)*, Vol. 1, pp. 1–4.

Baik, C., Jagadish, H. V. and Li, Y. (2019). Towards facial recognition problem in covid-19 pandemic, *2019 IEEE 35th International Conference on Data Engineering (ICDE)*, IEEE, pp. 374–385.

Barrak, A., Eghan, E. E. and Adams, B. (2021). On the co-evolution of ml pipelines and source code - empirical study of dvc projects, *2021 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER)*, pp. 422–433.

Bura, H., Lin, N., Kumar, N., Malekar, S., Nagaraj, S. and Liu, K. (2018). An edge based smart parking solution using camera networks and deep learning, *2018 IEEE International Conference on Cognitive Computing (ICCC)*, pp. 17–24.

Macarthy, R. W. and Bass, J. M. (2020). An empirical taxonomy of devops in practice, *2020 46th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, pp. 221–228.

Mundial, I. Q., Ul Hassan, M. S., Tiwana, M. I., Qureshi, W. S. and Alanazi, E. (2020). Towards facial recognition problem in covid-19 pandemic, *2020 4rd International Conference on Electrical, Telecommunication and Computer Engineering (ELTICOM)*, pp. 210–214.

Nabil, M., Rady, M., Moussa, K., Wessam, M., Hossam, M., Yousri, R. and Darweesh, M. S. (2021). A preprocessing approach to improve the performance of inception v3-based face shape classification, *2021 3rd Novel Intelligent and Leading Emerging Sciences Conference (NILES)*, pp. 205–209.

Nezamoddini, N. and Gholami, A. (2019). Integrated genetic algorithm and artificial neural network, *2019 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC)*, pp. 260–262.

O, M. A. K., C, M. Y., C, M. H., P, D., G, D., D, A., S, F. A. and P, S. (2021). Gabor-deep cnn based masked face recognition for fraud prevention, *2021 5th International Conference on Computing Methodologies and Communication (ICCMC)*, pp. 990–995.

Pathak, Y., Shukla, P. K. and Arya, K. V. (2021). Deep bidirectional classification model for covid-19 disease infected patients, *IEEE/ACM Transactions on Computational Biology and Bioinformatics* **18**(4): 1234–1241.

Qi, C. and Yang, L. (2020). Face recognition in the scene of wearing a mask, *2020 International Conference on Advance in Ambient Computing and Intelligence (ICAACI)*, pp. 77–80.

Shivanna, A. and Agrawal, D. P. (2020). Prediction of defaulters using machine learning on azure ml, *2020 11th IEEE Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, pp. 0320–0325.

Shivanna, A., Ray, S., Alshouiliy, K. and Agrawal, D. P. (2020). Detection of fraudulence in credit card transactions using machine learning on azure ml, *2020 11th IEEE Annual Ubiquitous Computing, Electronics Mobile Communication Conference (UEMCON)*, pp. 0268–0273.

Sun, X. (2020). Research on face detection method based on deep learning, *2020 International Conference on Big Data Artificial Intelligence Software Engineering (ICBASE)*, pp. 200–203.

Vadlapati, J., Senthil Velan, S. and Varghese, E. (2021). Facial recognition using the opencv libraries of python for the pictures of human faces wearing face masks during the covid-19 pandemic, *2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, pp. 1–5.

Verma, A., Kapoor, C., Sharma, A. and Mishra, B. (2021). Web application implementation with machine learning, *2021 2nd International Conference on Intelligent Engineering and Management (ICIEM)*, pp. 423–428.

Vuppalapati, C., Ilapakurti, A., Chillara, K., Kedari, S. and Mamidi, V. (2020). Automating tiny ml intelligent sensors devops using microsoft azure, *2020 IEEE International Conference on Big Data (Big Data)*, pp. 2375–2384.

Wang, D., Yu, H., Wang, D. and Li, G. (2020). Face recognition system based on cnn, *2020 International Conference on Computer Information and Big Data Applications (CIBDA)*, pp. 470–473.

WU, S., WANG, K. and OUYANG, Y. (2019). Study on small samples sar image recognition detection method based on transfer cnn, *2019 3rd International Conference on Electronic Information Technology and Computer Engineering (EITCE)*, pp. 718–722.