Proceedings of the 5th Faculty of Industrial Technology
International Congress Bandung, Indonesia,
September 17-18, 2025
ISSN 2962 – 1798

Deep Learning based Approach for Face Mask Detection

Divya Rani [a], *Abhi* Kumar [a], *Khushi* Kumari [a], *Koushlendra Kumar* Singh [a,1], and *Danish Ali* Khan [a]

[a] Department of Computer Science and Engineering, Jamshedpur, Jharkhand, India, 831014

Abstract. The CRONA virus's malady (COVID-19) which is an enormous family of distinctive infections have gotten to be exceptionally common, spreadable, infectious and perilous to the whole world of human kind. It transmits for the most part through nose and mouth if a tainted individual sniffle or hack which clears out beads of the infection on distinctive surface which is at that point breathed in by other individual, he too catches the disease as well. So, it has ended up exceptionally pivotal to secure ourselves and the individuals around us from this circumstance. We require to take safety measures such as keeping up social separating, washing hands each two hours, utilizing sanitizer, and the most vitally wearing a cover. To prevent the spread of virus, face mask is important and with face mask it is difficult to recognize face of human being with machine. The proposed method developed an approach of face mask detection based on deep learning approaches. Proposed approach encompasses with pre-trained models VGG16 and VGG19. The proposed model demonstrated on a real-world information set and tried with live video gushing. Higher the precision value of the demonstrated dataset with diverse hyper parameters and different individuals at distinctive has been performed. The results have been reported in the present manuscript.

1 Introduction

The global pandemic of COVID-19 has led to widespread illness. Notably, during a COVID-19 outbreak aboard the USS Theodore Roosevelt, individuals who adhered to mask-wearing protocols demonstrated a 70% reduction in the risk of testing positive for SARS-CoV-2 infection [1]. Emphasizing the pivotal role of mitigating disease transmission, the importance of wearing masks has been underscored. In various public settings, including retail establishments and public transportation, the adoption of masks is strongly recommended to uphold the well-being of all individuals.

The exploration of face mask detection has emerged as a compelling avenue of research amid the COVID-19 pandemic. This pertains to the identification of individuals wearing masks, a critical consideration given the challenges associated with manually monitoring mask adherence in public spaces. Automated systems capable of real-time mask detection have gained prominence. Such technological solutions serve as an additional layer of protection against the transmission of the virus, contributing to heightened safety measures.

[1] Corresponding author: koushlendra.cse@nitjsr.ac.in

ISSN 2962 – 1798

This technology can be practically applied in various sectors, playing crucial roles in monitoring adherence at bustling locations like airports and train stations. Additionally, its integration into the healthcare industry, covering hospitals, clinics, and pharmacies, can bring substantial benefits by ensuring a heightened level of precautionary measures in the ongoing battle against the pandemic.

Figure 1. Face mask detection model implementation in public area

The utilization of machine learning and deep learning technology proves highly beneficial in discerning whether an individual is wearing a mask. Furthermore, the incorporation of technology for face detection in public spaces constitutes a pivotal aspect of this project. The investigation delved into the application of VGG16 and VGG19 models to ascertain the mask-wearing status of individuals. Meticulous training of the model was conducted using a dataset comprising approximately 7000 images to facilitate accurate classification. The proposed model achieved notable success, culminating in the development of a model with a training accuracy of 97.78% and a validation accuracy of 95.36%.

.

2 Problem Definition

The principal aim of this paper is to develop and create a system based on deep learning techniques that could identify whether or not a person is wearing a mask. Furthermore, the proposed work sought to integrate sophisticated facial recognition algorithms into pictures, with the goal of offering comprehensive annotations and forecasts for every individual recognized. The primary focus went beyond mask detection to include a comprehensive analysis of the facial features present in an image. The objective was to create a strong system that could not only identify whether a mask was present or not, but also produce detailed annotations for every person found in the picture. Deploying the developed model on real-time images is a crucial aspect of the project, with a focus on pragmatic and dynamic applications. This deployment aspect is designed to demonstrate the system's efficacy and adaptability in practical settings, highlighting its potential for immediate and pertinent application. With regard to mask classification and facial detection with

ISSN 2962 – 1798

annotations, the project aimed to provide a comprehensive response to the problems caused by the continuous requirement for safety precautions and public health measures.

3 Literature Review

An extensive review of earlier studies in the field of face mask detection is given in this section. Researchers have been very interested in real-time face mask detection, and over the last three years, many theories and algorithms have been proposed in response. The thorough analysis of the body of literature illuminates the developments as well as the recognized shortcomings in this field.

3.1 Face Mask Detection using Machine Learning

In image processing and computer vision, face detection has become a very intriguing problem. Its uses are numerous and include face motion capture and face recognition, both of which require extremely accurate face detection at first. With the development of convolutional networks, it is now feasible to classify images with high accuracy. Pixel-level data, which the majority of face detection techniques are unable to supply, is frequently needed after face detection. One of the most challenging aspects of semantic segmentation has been obtaining pixel-level details [2]. Researchers first concentrated on the edge of the face image and the gray value. One of the works used a pattern recognition model and had access to face model data beforehand [3].

Convolutional Networks have been used in addressing image classification challenges. Common architectures such as AlexNet [4] and VGGNet [5] are characterized by a series of interconnected convolutional layers. AlexNet, boasting 5 convolutional layers and three fully connected layers, secured victory in the ImageNet LSVRC-2012 competition. VGGNet, an enhancement over AlexNet, distinguishes itself by employing 3x3 multiple kernels successively, deviating from larger kernels. These architectural nuances contribute to the effectiveness of Convolutional Networks in discerning patterns and features within images, showcasing their pivotal role in image classification tasks.

3.2 Deep Learning Models

Many studies carried out in previous years, using VGG16 architecture for image classification. The architecture of VGG16 comprises a total of 13 convolutional layers and 3 fully connected layers. The convolutional layers utilize 3 × 3 kernels, while the pooling layers employ 2×2 parameters. The organizational structure of VGG16 involves the segmentation of convolutional and pooling layers into distinct blocks labelled 1 through 5. Each of these blocks is characterized by the inclusion of multiple convolutional layers followed by a single pooling layer, contributing to a systematic and layered approach in the model's design. This hierarchical arrangement of blocks allows for a nuanced and compelling extraction of features across the network [6].

ISSN 2962 – 1798

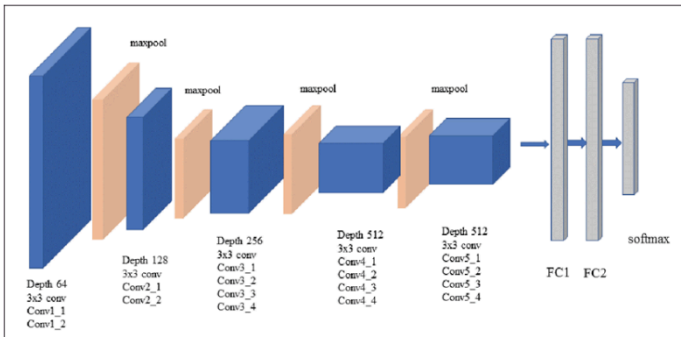

Figure 2. VGG16 Architecture

Meenpal et al. (2019) presented a study on facial mask detection using semantic segmentation [2]. The study proposed a methodology for obtaining segmentation masks directly from images containing one or more faces in different orientations.

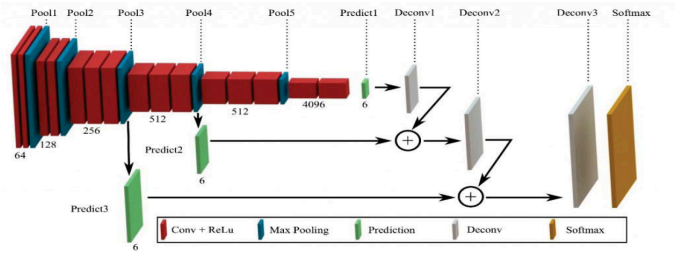

Figure 3. Architecture of FCN used generating segmentation masks

A similar study Lad et al. (2021) was carried out for Comparative Analysis of Convolutional Neural Network Architectures for Real Time COVID-19 Facial Mask Detection. Along with VGG16 CNN model, the study also worked on Sequential CNN model, and MobileNetV2 CNN [7]. Object detection models are generally being used so far for detecting objects in images. Same model can also be used for detecting multiple faces in images and later classifying as masked or unmasked. The study implemented Haar Cascade and HOG for detecting multiple faces in images. The feature engineering tasks was resolved using deep learning techniques like, overload was reduced and it do not require lots of manual feature engineering. Three CNN classifier was trained in this study.

- The Sequential CNN model consists of layers such as Conv2D, MaXPooling2D, Flatten, Dropout and Dense. It was trained with 20 epochs (iteration) which gives us training accuracy of 98.99% with loss of 0.0346 beyond which the accuracy changes due to increased training time and based upon parameter values.

ISSN 2962 – 1798

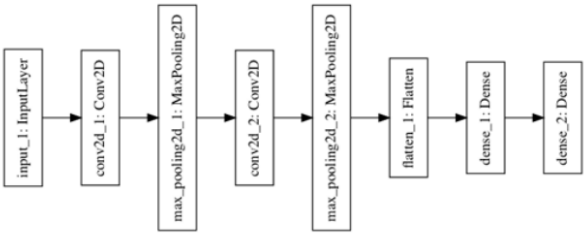

Figure 4. Sequential CNN model architecture

- VGG16 CNN is trained which has 17 convolution layers and 5 Max Pooling layers. The model is trained upon the training dataset and gives accuracy of 94.27% with loss of 0.1436.

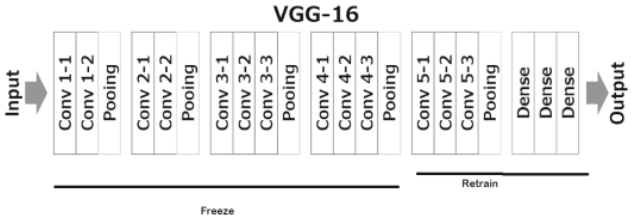

Figure 5. VGG16 CNN architecture

- MobileNetV2 architecture was next used. Model is trained for 20 epochs and gives accuracy of 99.22% with loss of 0.0282 for the training set.

Comparative analysis was further carried out for Sequential CNN, VGG-16 and MobileNetV2 architectures with performance measures. After 20 epochs, the MobileNetV2 CNN model exhibits the highest training accuracy at 99.2%, outperforming both the Sequential CNN (98.9%) and the VGG16 CNN (94.2%). MobileNetV2 also boasts the lowest training loss at 2.8%, suggesting efficient learning during training. In contrast, the Sequential CNN, while achieving a commendable training accuracy, demonstrates a higher loss (3.4%). The VGG16 CNN, although achieving a respectable training accuracy, presents a comparatively elevated training loss of 14.3%.

ISSN 2962 – 1798

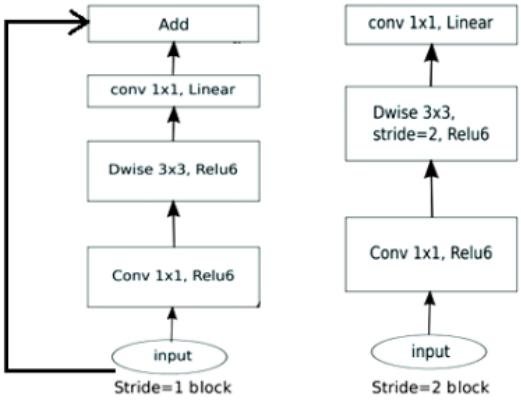

Figure 6. MobileNetV2 architecture

These metrics highlight MobileNetV2's potential superiority in face mask detection during the training phase, emphasizing the importance of considering both accuracy and loss in evaluating model performance. However, the ultimate choice should be guided by additional validation metrics and considerations related to the specific requirements of the face mask detection application.

In order to classify images into 1000 object categories, Simonyan and Zisserman (2014) [5] proposed the VGG19 convolutional neural network, which consists of 19 layers: 16 convolution layers and 3 fully connected layers. The ImageNet database, which has one million photos in 1000 categories, is used to train VGG19. Due to the fact that each convolutional layer uses multiple 3x3 filters, this method of image classification is highly popular [8]. The size of "VGG-19" network in terms of fully connected nodes is 574 MB. VGG-19 is so beneficial and it simply uses 3 × 3 convnet arranged as above to extend the depth [9]. Architecture CNN VGG19 in many cases of image classification always have good performance [10]. VGG19 stands out due to its focus on 3x3 filter convolution layers with a stride of 1, as opposed to having a large number of hyper-parameters. They also consistently used a max pool layer and comparable cushioning of 2x2 filter stride 2. The entire architecture consistently adheres to the convolution and max pool layer game plan. Two FC (fully connected layers) and a SoftMax for yield are present at the end [11]. The max-pool layer serves as a dividing line in the VGG-19 network, allowing the remaining sixteen layers to be divided into five blocks without having to calculate the fully connected layer [12].

Jian Xiao et al. 2020 [13], worked on utilizing a New and Enhanced VGG-19 Network to identify employees wearing masks. Initially, a certain quantity of mask pictures is gathered to serve as the model's training and test set. The enhanced network model's recall rate has increased by 8.39% and 11.4%, respectively, and its precision for determining whether or not to wear a mask has increased by 10.91% and 9.08%, respectively. As a result, there was a 11.4% and 8.39% decrease in mistake rates, respectively.

ISSN 2962 – 1798

To recognize the masked face, Md Abu, et al. 2022, employ CNN, the original VGG19, and the proposed extended VGG19 [14]. It also lessens the gradients' reliance on the parameter scale and the inter-covariant shift.

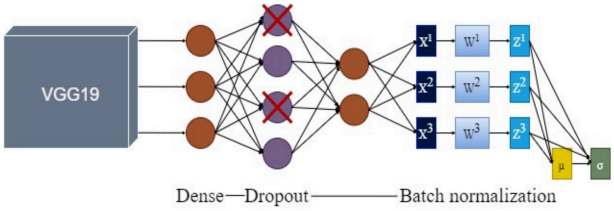

Figure 7. VGG19 Architecture

The activation functions, optimizer, dropout rate, and number of nodes in the dense layer were the hyperparameters used in this technique. The VGG19 model produced results with a 60% F1 score, 61% recall, 66% precision, and 61% accuracy. Our suggested extended VGG19 performs better than VGG19 and CNN using a supplementary dense layer with batch normalization and a 50% dropout layer.

3.3 Face detection models

In computer vision, face detection is the most widely pursued research topic. Face recognition produces the detected picture topic after receiving an image from a camera or from a video. Regions of the face, changes in the anatomy of the face, face cuts, and formatted and styled angles can all be considered facial features. A face's extraction process involves obtaining its features from the camera. Various models are popular in this topic. Some of them includes, YOLOv5, Haar Cascade, camshift algorithm.

A series of "square-shaped" functions that collectively create a base or a family of wavelets is known as a haar cascade. It also focuses on "Haar Wavelets," which divide image pixels into squares using a Haar wavelet technique that was proposed in Paul Viola and Michael Jones' 2001 work "Rapid Object Detection Using an Enhanced Cascade of Simple Features" (Viola et al., 2001[14]). Kumbhar, P. Y., et al., implement a real-time Face detection and tracking the head poses position from high-definition video using Haar Classifier through Raspberry Pi BCM2835 CPU processor which is a combination of SoC with GPU based Architecture [16]. Using OpenCV's Haar-Like Feature function, the prototype was constructed. A search window that slides through an image to determine whether or not a certain area of the image resembles a face was created using the Haar classifier face detection. To determine if an image is a face or not, a single feature is used by a vast set of very weak classifiers and Haar-like features. Every feature has a description that includes its coordinate in relation to the search window, which serves as the feature's origin for size. Once learned, the classifier can be applied to an input image's region of interest (of the same size as used during training). The output of the classifier was "0" if the region is not likely to display the face, and "1" otherwise. Processing uses the OpenCV library to analyze the video after receiving video input from the webcam. The OpenCV library will provide the Processing sketch with the face's coordinates if a face is found in the footage. The processing sketch will identify the face's location within the frame in relation to the frame's

ISSN 2962 – 1798

center and transmit this information to the Raspberry Pi over a serial connection. The servos linked to the Servo setup will be moved by the Raspberry Pi using the data from the Processing sketch.

Figure 8. Displaying result of haar Cascade

Figure 8 displays the face detection result. These are the frames that were taken from the streaming HD video. Even though there is only one face in the frame, the face detection system occasionally returns many results. In this instance, the OpenCV and SimpleCV Haar Classifier packages are utilized in post-image processing to extract the precise facial coordinates.

YOLO series algorithm is another famous algorithm now a days, for object detection. The YOLO series of algorithms directly performs object identification, location, and classification using a single-stage neural network. YOLOv5's input end makes use of the same mosaic data augmentation technique as YOLOv4, which has improved small target detection performance. The function of adaptive anchor frame calculation is added in YOLOv5. The value of the ideal anchor frame in various training sets is adaptively determined throughout each training session [17].

The YoloV5 model developed by Ieamsaard et al. (2021) was composed of two parts, namely the training model and the face mask detection model [18]. 682 photos from the face mask dataset were utilized in the training model. The YoloV5 face mask detection model processed a prediction score of three classes: "With_Mask", "Without_Mask", and "Incorrect_Mask". It then produced output photos with their predicted classes and detection scores. The original images from the face mask dataset were the model's input. With five distinct epoch counts, the YoloV5 trains the model. The deep learning model for face mask identification with 300 epochs performs best when compared to the 86 examined photos, showing 96.5% accuracy in relation to the highest precision and recall. In further studies, a computer vision system might be used to determine the right or erroneous mask wearing in real time.

4 Proposed Methods

Amidst the global turmoil caused by the COVID-19 pandemic, governmental authorities worldwide sought to curb the spread of the deadly virus by implementing a range of preventative measures. Chief among these measures was the universal recommendation to

ISSN 2962 – 1798

wear masks in public spaces. While a significant portion of the populace embraced and adhered to these guidelines, achieving universal compliance remained a formidable challenge. In response to this challenge, researchers and technologists undertook the task of devising innovative solutions to enhance the enforcement of COVID-19 safety protocols. One such pioneering idea that emerged from the research community involved the development of face mask detection technology. This groundbreaking approach aimed to enable real-time identification of individuals who were not wearing masks, providing a tool for authorities to ensure strict adherence to the prescribed guidelines. By employing advanced computer vision and deep learning algorithms [10] [7] [12], these systems could analyse real time live videos or images to determine whether individuals in public spaces were complying with the mask-wearing rules. This technological intervention proved instrumental in supplementing traditional enforcement methods, allowing governments to more effectively monitor and enforce adherence to COVID-19 regulations. The implementation of face mask detection technology not only facilitated a more nuanced approach to public health management but also served as a testament to the adaptability and ingenuity of the scientific community during times of crisis.

4.1 Architecture of the Study

This section will mainly deal about the follow of methodology followed in this research to do real time mask detection. In this section, the research will focus on the architecture of the study and the methods follow to achieve the result.

4.2 Problem description:

This section mainly deals with the problem description and the basic understanding about the research conducted. The model choice and the basic knowledge about dataset is discussed in this section.

4.3 Data loading and basic EDA

The dataset is loaded from the Kaggle. The folder is stored in zip files and now extracted from the zip file. The research also extracts the classes present in the dataset. The dataset basically contains two classes. One class contains only images with face mask present and other folder contains images with only without face mask.

4.4 Data preprocessing

The dataset only contained 7553 images. Thus, the process of data augmentation becomes necessary. Thus, this research focuses on applying data augmentation to increase the size of the dataset. The final total number of images in the dataset is 7553 images with 6043 images in the training set and 1053 images in the validation set. The process also involves rescaling the image pixels between 0 and 255 to between 0 and 1

4.5 Training the model

The research is centered on the training of two pre-existing models, namely VGG16 and VGG19, for the task of face mask detection. In the initial phase, the VGG16 model undergoes a comprehensive training and validation process using the designated dataset. Subsequently, the model is put to the test by making predictions on the validation set, assessing its performance and accuracy in face mask detection. In parallel, the research extends to the VGG19 model, which undergoes a similar training and validation regimen on

ISSN 2962 – 1798

the provided dataset. Notably, the VGG19 model boasts a more intricate architecture with 19 trainable layers, distinguishing it from its VGG16 counterpart, which comprises 16 trainable layers. This disparity in layer complexity implies that the VGG19 model is more resource-intensive, contributing to its potentially heightened capacity for feature extraction and representation. To complement the training efforts, the research leverages the pre-trained model Haar Cascades, integrating its capabilities into the training process. This strategic utilization enhances the models' ability to discern facial features and facilitates a more nuanced approach to face mask detection. In essence, the research unfolds as a dual exploration into the capabilities of VGG16 and VGG19 models, showcasing their distinct architectural variances and the impact of their pre-trained features. The integration of the Haar Cascades model further enriches the training process, promising a more robust and sophisticated approach to the crucial task of face mask detection.

4.6 Result and Evaluation

The assessment of a model extends to both the test set and the validation set, pivotal components in gauging its performance. Beyond these structured datasets, evaluations are conducted on randomly acquired images, introducing an element of unpredictability to assess the model's adaptability to diverse scenarios. Furthermore, a forward-thinking aspect of the evaluation process involves the incorporation of real-time video testing. This innovative feature allows researchers to scrutinize the model's efficacy in dynamically changing environments, mirroring the fluidity of real-world applications. By subjecting the model to this dynamic assessment, its responsiveness to unfolding events and its ability to process information in real-time are scrutinized, offering a comprehensive understanding of its practical utility beyond static datasets. In essence, the model's evaluation protocol encompasses a multi-faceted approach, ranging from structured datasets to spontaneously collected images and, notably, real-time video scenarios.

4.7 Discussion and related works

The concluding segment of the research delves into the broader landscape of related endeavors within this domain, emphasizing the evolutionary strides made through this study. It meticulously outlines the advancements achieved and the hurdles encountered during the research journey. This study stands as a benchmark in face mask detection, elucidating its significance by comparing and contrasting with prior works employing either the same dataset or similar models. This comparative analysis serves to underscore the relevance and impact of this research in augmenting the field's understanding and capabilities in face mask detection methodologies.

Moreover, the comprehensive evaluation against existing studies not only highlights the strengths and uniqueness of this research but also elucidates the challenges overcome, contributing to the broader discourse on this critical subject. The comparative insights foster a deeper appreciation for the advancements made and the innovative approaches adopted within the study. Closing on a forward-looking note, the report delineates the future prospects and the uncharted avenues awaiting exploration. It offers a roadmap for potential research trajectories, suggesting directions where further investigations could be conducted. This futuristic outlook not only consolidates the study's contributions but also beckons forth continued advancements, inspiring forthcoming researchers to build upon this foundation and pioneer novel solutions in the realm of face mask detection.

ISSN 2962 – 1798

4.8 Dataset

The Dataset used for this study was downloaded from Kaggle. The dataset contains the masks and non-masks images and consists of 7553 images. The link to the dataset is https://www.kaggle.com/datasets/omkargurav/face-mask-dataset.

4.9 Model choice

Various models have been employed in this study for the purpose of face mask detection, with particular emphasis on the utilization of the well-known VGG16 pre-trained model. The selection of this pre-trained model was significantly influenced by considerations related to storage limitations and GPU constraints. VGG16, characterized by a total of 21 layers, including 16 trainable layers and 5 pooling layers, served as a pragmatic choice within the resource constraints imposed by the study. Notably, modifications were made to the final layers of the VGG16 model to adapt it for binary classification, specifically for the task of face mask detection.

In addition to VGG16, the evaluation phase incorporated the VGG19 model, a natural extension of the VGG16 architecture and widely recognized among researchers for its extensive usage. The VGG19 model encompasses 19 trainable layers accompanied by 5 pooling layers, contributing to its enhanced capacity for feature extraction and representation. Furthermore, the study integrated the Haar cascade model from OpenCV into its methodology to assess its performance in face mask detection. This model, commonly employed by researchers in the field, was deemed essential for comparative analysis and evaluation within the study. The selection and integration of these models were underpinned by a comprehensive consideration of their merits, coupled with practical considerations such as storage capacity and GPU resources. This approach ensured a judicious balance between model effectiveness and the inherent constraints of the study environment.

5 Implementation

This section will discuss the implementation and coding section of the study. The flow of implementation would be followed as discussed in the methodology section.

5.1 Model choice importing required libraries

A fundamental component of the setup involves the installation of TensorFlow, a versatile library instrumental in deep learning tasks, encompassing domains such as image classification and natural language processing. TensorFlow's compatibility with Keras, a high-level neural networks API, is also established through installation, providing convenient access to pre-trained model weights. In addition to TensorFlow and Keras, the installation of the Matplotlib library is facilitated to enable robust data visualization, an indispensable aspect for interpreting and presenting results effectively. The inclusion of Matplotlib empowers the project with diverse plotting capabilities, enhancing the clarity of visual representations. Furthermore, the installation of NumPy and Pandas, two powerhouse libraries in the realm of numerical computing and data manipulation, augments the project's computational capabilities. NumPy's array-based operations and Pandas' versatile data structures collectively contribute to efficient data handling, a crucial facet in the journey from raw data to meaningful insights.

ISSN 2962 – 1798

In essence, this section serves as the foundational step, fortifying the project environment with the essential tools required for advanced data analysis, model training, and visualization. The diverse capabilities of TensorFlow, Keras, Matplotlib, NumPy, and Pandas collectively empower the project to navigate the intricacies of deep learning and data manipulation seamlessly.

5.2 Data loading and basic EDA

Given the substantial size of the dataset, a streamlined approach was adopted for seamless acquisition by directly downloading it from Kaggle to Google Colab. Following the download, the dataset arrived in a compressed zip format, necessitating a subsequent step to unzip and organize the content within Google Colab. Upon extraction, the resultant folder structure comprises two distinct directories: one housing images depicting individuals wearing masks, and the other featuring images without masks. This meticulous organization facilitates a categorical division, crucial for the subsequent stages of the study. To operationalize the dataset efficiently within the Google Colab environment, generators were employed for data loading. This approach not only optimizes memory usage but also ensures a dynamic flow of data during training. The dataset was further partitioned into training and validation subsets, a pivotal step in facilitating robust model training and evaluation.

5.3 Data Pre-Processing

Within the data pre-processing phase, a crucial set of transformations is applied to ensure optimal model training and performance. One integral step involves rescaling the pixel values of images, transitioning them from the original range of (0,255) to a normalized scale of (0,1). Furthermore, as part of the pre-processing pipeline in this study, the dimensions of the images are specified to a standardized size of 256×256 pixels. This uniformity in image size establishes consistency in the input data, streamlining the subsequent stages of model training and validation. To enhance the dataset's variability and robustness, a data augmentation strategy is implemented. This involves the incorporation of transformations such as zoom, horizontal flip, and shear. By introducing these augmentations, the dataset undergoes variations that emulate real-world scenarios, thereby fortifying the model against potential overfitting and enhancing its generalization capabilities.

In essence, the data pre-processing pipeline not only standardizes the input data through rescaling and sizing specifications but also strategically augments the dataset to imbue the model with resilience and adaptability. These meticulous steps collectively contribute to the preparation of a well-optimized and diversified dataset for effective training and evaluation.

5.4 Training the Model

The VGG16 pre-trained model serves as the cornerstone for the training and validation procedures employed in the context of face mask detection. The weights associated with the VGG16 model are acquired directly from the Keras library, specifically pre-trained on the ImageNet dataset. To expedite the training process and conserve computational resources, the initial layers of the VGG16 model are frozen, mitigating the need for retraining these layers. In contrast, the final dense layers are strategically appended to the architecture to enable fine-tuning tailored to the nuances of face mask detection. Following the integration of the VGG16 model, a meticulous process of hyperparameter tuning ensues, exploring various configurations of layers and neurons. After careful experimentation, a configuration

ISSN 2962 – 1798

featuring a layer with 124 neurons, followed by an output layer housing a single neuron, emerges as the optimal configuration, delivering superior results in face mask detection. To visually represent the architecture of the refined model, the Keras plot model functionality is employed. This graphical representation offers a succinct and insightful depiction of the model's structure, encapsulating the nuanced layers and configurations optimized for the task at hand. In summary, the utilization of the VGG16 pre-trained model, coupled with strategic adjustments and hyperparameter tuning, culminates in an intricately designed architecture conducive to achieving exemplary results in the realm of face mask detection. The graphical representation provided by the Keras plot model serves as a visual testament to the sophistication and efficacy of the refined model.

The provided code snippet initiates the download of the pre-trained VGG16 model from the Keras library. Upon retrieval, the VGG16 model, encapsulated within the 'conv_base' variable, is structured to accommodate input images with dimensions of 256×256×3, conforming to the required input shape. The pre-trained weights of the VGG16 model, gleaned from the ImageNet competition's extensive dataset comprising over 10,000 images, serve as a robust foundation for feature extraction and representation. Expanding upon this foundation, additional layers are appended to the 'conv_base' layer. Initially, a flattening operation is applied to facilitate the transition from convolutional layers to densely connected layers. Subsequently, a dense layer comprising 124 neurons, activated by the Rectified Linear Unit (ReLU) function, is incorporated to enhance the model's ability to capture intricate features within the data. Finally, the output layer culminates the model architecture, employing the softmax activation function to facilitate classification among the two classes inherent in the face mask detection task. This outputs layer configuration, featuring two neurons, aligns with the binary nature of the classification problem. To provide a comprehensive overview of the model's architecture, the 'model.summary()' function is invoked, revealing the intricate configuration of layers, their respective shapes, and the total number of trainable parameters within the model. In essence, this code snippet orchestrates the construction of a tailored VGG16-based neural network, adeptly fine-tuned for face mask detection, leveraging the powerful pre-trained weights from the ImageNet competition while customizing the model architecture to suit the specific requirements of the classification task.

The VGG19 model is also used in training and validation in face mask detection. The VGG19 model is a little heavier as compared to VGG16 model. The same approach is used in VGG19 model as for VGG16 model. The weights are downloaded and taken from imagnet dataset using keras library. The code snippet orchestrates the acquisition of the pre-trained VGG19 model from the Keras library, storing it in the 'conv_base' variable. Weights for the VGG19 model are retrieved from the ImageNet competition, where the model's prowess has been fine-tuned on an extensive dataset encompassing more than 10,000 images. The specified input shape is tailored to accommodate images of dimensions 256×256×3. Distinguished by its increased complexity, the VGG19 model boasts 19 trainable layers and 5 max pooling layers, making it a more intricate counterpart to the VGG16 model. While the convolutional layers are directly imported, the dense layer from VGG19 is intentionally omitted during download and will be incorporated separately into the model. Extensive hyperparameter tuning endeavors revealed that configuring a hidden layer with 124 neurons, activated by the Rectified Linear Unit (ReLU) function, alongside an output layer housing 2 neurons (reflecting the binary nature of the task) with the softmax activation function, yielded superior performance. This meticulous fine-tuning of hyperparameters contributes to optimizing the model's capacity to discern features and

ISSN 2962 – 1798

make accurate predictions. In essence, the code snippet establishes a VGG19-based neural network, capitalizing on the pre-trained weights from ImageNet while strategically incorporating a separately added dense layer. The culmination of hyperparameter tuning results in an architecture characterized by 124 neurons in the hidden layer and 2 neurons in the output layer, encapsulating the optimal configuration for effective face mask detection.

The Adam optimizer, renowned for its effectiveness in optimizing deep neural networks, is selected. Given the binary nature of the classification task, categorical crossentropy is chosen as the loss function, adept at handling such scenarios. The fitting process, executed through the 'fit' function, encompasses the model training phase. The training is conducted over a span of 6 epochs, with the number of steps per epoch aligned with the length of the training data. This strategic choice ensures comprehensive coverage of the training dataset during each epoch, contributing to robust learning and parameter adjustments. Crucially, concurrent with the training process, the model undergoes validation on the designated validation set. This entails assessing the model's performance on data it has not encountered during training, offering insights into its generalization capabilities and ability to perform effectively on unseen instances. In essence, the compilation, fitting, and validation stages collectively constitute a coherent and dynamic process, equipping the model to iteratively learn from the training data, fine-tune its parameters, and validate its performance against previously unseen data. Now, whenever we need to test the model on images or videos, the model need not be trained again and can be loaded using the load_model function of the keras. For predicting the face detection on random set of images, a pipeline is created. The pipeline would consist of taking an input image, converting the image to numpy array and normalizing it so that every pixel value can come in between 0 and 1. Now, the result is predicted using predict function to predict the output.

6 Result and Evaluation

The performance evaluation of the VGG16 model on the training set reveals an impressive accuracy of 99.16%, coupled with a minimal loss of 0.0236. This underscores the model's exceptional capability to classify instances within the training dataset correctly. The validation set results are also commendable, with an accuracy of 96.75% and a corresponding loss of 0.1038. These figures attest to the model's robust generalization to unseen data, albeit with a slight increase in loss compared to the training set. Similarly, the VGG19 model demonstrates noteworthy metrics on the training set, achieving an accuracy of 97.78% and a relatively low loss of 0.0632. This underscores the model's proficiency in learning the intricacies of the training data. The validation set accuracy, although slightly lower at 95.36%, is indicative of the model's adeptness in generalizing to new instances, despite a moderate increase in loss to 0.1473. In essence, both the VGG16 and VGG19 models exhibit commendable performance metrics, showcasing their efficacy in face mask detection. The nuanced differences in accuracy and loss between the training and validation sets provide valuable insights into the models' generalization capabilities and highlight their competence in handling unseen data. As the model is tested on random set of images of researchers taken for evaluation. The prediction from the model is depicted along with image in Figure 9.

ISSN 2962 – 1798

The bounding box predicts the confidence level of the model of mask or no mask prediction. The model is also evaluated on random test images from the internet to find out how the model performs in real life. The model is again tested on images taken during research to evaluate the robustness of the model. Furthermore, a real time video detecting is created to detect mask and non-mask images in real life. The face is detected from the video using Haarcascade and then sent to the model saved to predict whether the face consist of masks or not. The final output is shown in the video by using putText function of opencv. The model is saved on the test.

Figure 9. Testing on random images

directory and predicted the output following the machine learning pipeline implemented in this study. The Cascade model from opencv helps in determining the face in the video and then the bounding box is created around that face. The images are sent to the saved model and predicted on whether the image of the person has masks or not. The argmax function is used to find the maximum probability among the two. If the maximum probability predicts that the image contains a masks, the bounding box is shown in green color or else the bounding box is shown in red color. The confidence level just near the bounding box determines with how much accuracy the model can predict whether the image contains the mask or not. Following this, the putText function puts the text in the image and using

ISSN 2962 – 1798

imshow function it shows the output. The results with the pre-trained models is given in the table 1.

Table1. Results with pre-trained models

Parameters	**VGG19**	**VGG16**
Training Loss	0.0632	0.0236
Training Accuracy	97.78%	99.16%
Validation Loss	0.1473	0.1038
Validation Accuracy	95.36%	96.75%

- Comparing the results:

  Training Accuracy: VGG16 slightly outperforms VGG19 with a higher training accuracy of 99.16% compared to VGG19's 97.78%.

  Validation Accuracy: VGG16 maintains a higher validation accuracy of 96.75%, surpassing VGG19's still impressive 95.36%.

Both models demonstrate excellent performance, VGG16 stands out with slightly higher accuracy on both training and validation sets. The choice between VGG16 and VGG19 may depend on factors such as computational resources, model size, and the specific requirements of your application. Suppose the increased depth of VGG19 does not yield significantly improved results for your use case. In that case, the more lightweight VGG16 may be a favorable choice, offering similar accuracy with potentially reduced computational demands.

7 Conclusion

The objective of the proposed work is to create a binary classification model capable of distinguishing between individuals wearing face masks and those without mask. Leveraging the power of transfer learning, we employed pre-trained weights from the ImageNet dataset. These weights, learned during training on a vast and diverse dataset, enabled our model to capitalize on the knowledge acquired, thereby enhancing its performance in image classification tasks even when trained with limited face mask dataset. The achieved accuracy of the model underscores the effectiveness of the proposed model. Continuous optimization is a key aspect of present work, involving the fine-tuning of hyperparameters to build a highly accurate solution. Notably, this model stands out as a compelling use case for edge analytics, showcasing its potential applicability in scenarios where on device processing is crucial. Moreover, our method demonstrated state of the art results when evaluated on a public face mask dataset. The proposed face mask detection models are doing really well, achieving accuracy rates of 96-97%. For instance, VGG19 and VGG16, two popular models, showed excellent performance. VGG19 had an accuracy of 97.78%, and VGG16 had an impressive 99.16% accuracy.

References

[1] D.C. Payne, S.E. Smith-Jeffcoat, G. Nowak, U. Chukwuma, J.R. Geibe, R.J. Hawkins, J.A. Johnson, N.J. Thornburg, J. Schiffer, Z. Weiner, B. Bankamp, SARS-CoV-2 infections and serologic responses from a sample of US Navy service members—USS Theodore Roosevelt, April 2020, Morb. Mortal. Wkly. Rep., 69(23), 714 (2020)
[2] T. Meenpal, A. Balakrishnan, A. Verma, Facial mask detection using semantic segmentation, in 2019 4th Int. Conf. Comput., Commun. Secur. (ICCCS), 1–5 (IEEE, 2019)
[3] T. Ojala, M. Pietikainen, T. Maenpaa, Multiresolution gray-scale and rotation invariant texture classification with local binary patterns, IEEE Trans. Pattern Anal. Mach. Intell., 24(7), 971–987 (2002)
[4] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, Adv. Neural Inf. Process. Syst., 25 (2012)
[5] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv:1409.1556 (2014)
[6] Z.P. Jiang, Y.Y. Liu, Z.E. Shao, K.W. Huang, An improved VGG16 model for pneumonia image classification, Appl. Sci., 11(23), 11185 (2021)
[7] A.M. Lad, A. Mishra, A. Rajagopalan, Comparative analysis of convolutional neural network architectures for real time COVID-19 facial mask detection, J. Phys.: Conf. Ser., 1969(1) (2021)
[8] M. Bansal, M. Kumar, M. Sachdeva, A. Mittal, Transfer learning for image classification using VGG19: Caltech-101 image data set, J. Ambient Intell. Humaniz. Comput., 1–12 (2021)
[9] V. Sudha, T.R. Ganeshbabu, A convolutional neural network classifier VGG-19 architecture for lesion detection and grading in diabetic retinopathy based on deep learning, Comput. Mater. Continua, 66(1) (2021)
[10] A.F. Ibrahim, S.P. Ristiawanto, C. Setianingsih, B. Irawan, Micro-expression recognition using VGG19 convolutional neural network architecture and random forest, in 2021 4th Int. Symp. Agents, Multi-Agent Syst. Robot. (ISAMSR), 150–156 (IEEE, 2021)
[11] M.J. Ahmed, P. Nayak, Detection of lymphoblastic leukemia using VGG19 model, in 2021 Fifth Int. Conf. I-SMAC (IoT in Social, Mobile, Analytics and Cloud), 716–723 (IEEE, 2021)
[12] Z. Cao, J. Huang, X. He, Z. Zong, BND-VGG-19: A deep learning algorithm for COVID-19 identification utilizing X-ray images, Knowl.-Based Syst., 258, 110040 (2022)
[13] J. Xiao, J. Wang, S. Cao, B. Li, Application of a novel and improved VGG-19 network in the detection of workers wearing masks, J. Phys.: Conf. Ser., 1518(1), 012041 (2020)
[14] M.A. Marjan, M. Hasan, M.Z. Islam, M.P. Uddin, M.I. Afjal, Masked face recognition system using extended VGG-19, in 2022 4th Int. Conf. Electr., Comput. Telecommun. Eng. (ICECTE), 1–4 (IEEE, 2022)
[15] P. Viola, M. Jones, Rapid object detection using a boosted cascade of simple features, in Proc. 2001 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR), 1–I (IEEE, 2001)
[16] P.Y. Kumbhar, M. Attaullah, S. Dhere, S. Hipparagi, Real time face detection and tracking using OpenCV, Int. J. Res. Emerg. Sci. Technol., 4(4), 39–43 (2017)
[17] J. Yao, J. Qi, J. Zhang, H. Shao, J. Yang, X. Li, A real-time detection algorithm for Kiwifruit defects based on YOLOv5, Electronics, 10(14), 1711 (2021)
[18] J. Ieamsaard, S.N. Charoensook, S. Yammen, Deep learning-based face mask detection using yolov5, in 2021 9th Int. Electr. Eng. Congr. (iEECON), 428–431 (IEEE, 2021)