DEVELOPMENT OF A REAL TIME FACE MASK DETECTION METHOD BASED ON YOLOV3

Jane Frank[1], Abisola Olayiola[2], Godwin Ansa[3], Olayemi Ariyo[4] and Aloysius Akpanobong[5]

[1,4]Department of Computer Science, Caleb University, Imota Lagos State, Nigeria

[2]Department of Computer Engineering, Olabisi Onabanjo University, Ago-Iwoye, Ogun State, Nigeria

[3,5] Department of Computer Science, Akwa Ibom State University, Mkpat Enin, Akwa Ibom State, Nigeria

myjanefrank@gmail.com[1], olayiwola.abisola@oouagoiwoye.edu.ng[2], godwinansa@aksu.edu.ng[3], ariyoyemi@gmail.com[4], aloysiusakpanobong@aksu.edu.ng[5]

Keyword:

Detection, Face Mask, YOLO, Model, Datasets.

ABSTRACT

This research project is focused on developing a real-time face mask recognition system based on the YOLOv3 deep learning architecture due to the importance of mask use in minimizing the COVID-19 epidemic. The system's primary objective is to accurately classify individuals into three groups: those who correctly wear masks, those who do not, and those who do so incorrectly. Rapid deployment is made possible by real-time processing capabilities in a number of situations. Deep learning is used to create the most modern and sophisticated face mask identification methods. The single-shot detection technique YOLOv3 is applied in this work to address this issue. Transfer learning techniques were utilized to train the model on a particular dataset rather of collecting a lot of data. The user-friendly interface significantly aids to pandemic preparedness by supporting authorities in successfully monitoring and enforcing mask compliance by giving current information on mask-wearing habits. The technique employs bounding boxes (colored in red or green) to differentiate people's faces depending on whether or not they are wearing masks. In order to take the proper action when a person is not wearing a mask, the system snaps a picture of them and transmits it to the relevant authorities and potential victims.

Corresponding Author: Email: ariyoyemi@gmail.com[4]

INTRODUCTION

Object detection in computer vision is the process of locating all objects of interest in a picture. Estimates of the position, size, and class of each object discovered are produced using an object detection algorithm. A bounding box, which is a rectangular box encircling the object and identifying it according to its membership in many categories, is often used to describe the position and size of identified things. An alternative method for defining a recognized item's range is to use a segmentation mask, which is a pixel-level mask of the object. Due of the global COVID-19 corona virus outbreak, wearing face masks in public is becoming more common. Before the epidemic started, people used masks to protect their health from air pollution or because they felt or looked self-conscious. When someone talks, sneezes, or coughs, there is a chance that they will infect those nearby. Utilizing facemasks exclusively will significantly reduce the disease's spread, flatten secondary and tertiary waves, and put an end to the outbreak. Therefore, until the infection is totally eradicated, it is essential to often wear face masks.

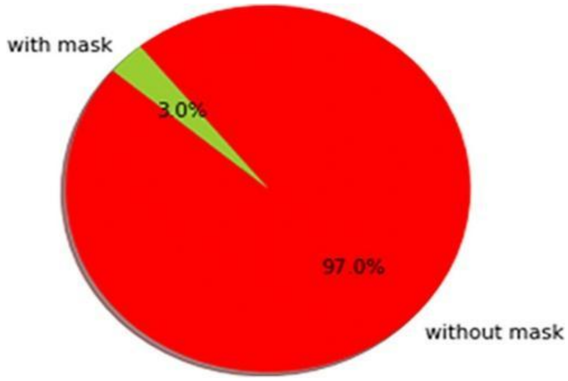


Figure 1. Percentage with Mask and without Mask

Our research is a response to earlier findings, as seen in fig. 1 highlighting the critical role that wearing a mask has in preventing the transmission of infections. However, this detection process involves problems such as false positives and negatives (FP and FN) results. There exists a substantial distinction between recognizing a face behind a mask and recognizing a mask itself. To develop a comprehensive real-time face mask recognition system that combines the most recent knowledge for exact classification of mask use, we plan to apply YOLOv3 deep learning [1]. This technology enhances public health safety for both the ongoing pandemic and prospective health emergencies [2] by utilizing cutting-edge transfer learning algorithms and a user-friendly interface design. This article's models are made using transfer learning, a machine learning technique where a model developed for one task is applied to another task as the basis for a new model. Given that developing neural network models from scratch requires a significant amount of computational and time resources, pre-trained models are often used as the basis for computer vision and natural language processing tasks in deep learning. This is because these models offer significant improvements in skill on related problems [36]. Tasks related to image processing, picture categorization, object identification, and image recognition are typical in computer vision. One appropriate solution for the mentioned above issue is object detection, which may find instances of visual objects of a particular class in the photos [37]. Recent studies on mask detection include RetinaMask [38], which enhanced single shot detection with a lighter backbone network to perform real-time mask detection and D. Chiang et al. [39], this improved RetinaNet by 1.9% on average accuracy by adding a context attention module and a transfer learning approach. However, there is still room for improvement in terms of both accuracy and speed with these methods. While most relevant studies addressed whether or not to wear masks, they did not address the scenario of inaccurate mask wearing, hence creating an acceptable dataset for mask detection and constructing a corresponding detector are considered critical issues. To steer the public away from the pandemic, detection of incorrect mask wearing needs to be discussed. The use of YOLOv3 model for this research is because it has new features, such as a better backbone classifier and multiscale prediction. YOLOv3 demonstrates a strong competitive edge in terms of precision and speed compared to other algorithms and can also correctly identify a wide range of objects [40]. Our research shows that wearing a mask, wearing the wrong mask, and not wearing a mask all continue to be important ways to stop the transmission of COVID-19.

LITERATURE REVIEW

Giving distinct identities to a range of objects in a video clip is a challenge for multiple item detection [3]. This inquiry focuses on the face mask detection method. These tasks include object detection, classification, and recognition. These computer vision approaches focus on object detection, identification, and categorization. Segmenting the face object is the first stage in addressing the face mask detection challenges, followed by the detection of the mask. YOLO represents an advanced convolutional neural network (CNN) designed for on-the-fly object detection. This technique uses a single neural network to process the entire image, divide it into different regions, and then forecast the bounding boxes and probabilities for each region [4]. Training these models becomes extremely challenging due to the huge variety of camera angles in the images and the many types of masks, creating a significant barrier for our research efforts.

Machine Learning

Machine learning, a branch of artificial intelligence (AI), is the development of methods and models that enable computers to acquire knowledge and form conclusions or predictions based on data without being explicitly programmed. Instructing a computer to learn from examples in a way that is analogous to how people learn through experience is a common comparison made to this method. Algorithms are trained for machine learning on large datasets, where they look for patterns, connections, and features to understand the underlying structure of the data. This understanding allows the model to predict or classify as yet unseen data [5]. As the model is exposed to additional data, it becomes more accurate in its predictions. A machine learning model can be trained on hundreds of images of cats and dogs to distinguish between them using features like forms, colors, and patterns, for example. Once trained, the model can tell the difference between cats and dogs in brand-new images.

Deep Learning

Since they have a finite number of parameters and frequently struggle with managing lots of training samples, machine learning methods do not scale well to large datasets. One of a larger group of machine learning techniques focused on learning data representations is deep learning. Similar to neural networks, deep learning architectures have been used in a variety of industries, such as computer vision, speech recognition, and natural language processing. The phrase "deep" refers to how many layers of change the data goes through. The capacity to represent the incoming data abstractly increases with each level of deep learning. A pixel matrix, for instance, might be the input for an image recognition task; the first layer might encode limits and edges, the second might represent the nose and eyes, and the third might identify the entire face. There is only one sort of neural network because there are so many topologies with various purposes. Convolutional neural networks (CNN) are better suited for image understanding and computer vision applications than recurrent neural networks (RNN) are for natural language processing [6]. In the literature, there are various methods for classifying and counting people that fall into three categories: methods based on detection, methods based on regression, and methods based on density estimation.

Detection-based Approaches

Techniques based on detection aim to count the number of individuals present in the scene using detector algorithms based on a backbone network. The classifier may be monolithic [7], collecting data from the entire body, or parts-based [8], concentrating on particular body parts like the head and shoulder]. These techniques perform best in low-density crowd environments since high-density areas skew the prediction and make it difficult for the model to distinguish between the various body parts of each individual in the image. Convolutional Neural Networks (CNNs) can be used to create face mask detectors by integrating them with popular models such as ResNet or Inception networks. The sigmoid activation function can then be used in the top layer of the fully connected dense network. By assigning them a class label, objects are found in photos using CNN image classification. Object localization is the process of drawing a bounding box around an object in a photograph. Combining these two challenges, a bounding box is drawn around each object of interest in the image, and a class name is given, making the object detection challenge more difficult.

Regression-based Approaches

The Problem with detection-based methods include too-dense crowds and occlusions. With the intention of discovering a mapping between features collected from the input image, counting by regression was developed to address these problems [8]. The key benefit of this approach is that it avoids the use of a detector, which makes the model more complex. Low-level feature extraction and regression modelling are the two main parts of this model. Using conventional background removal methods, such as histogram-oriented gradients (HOG), foreground and local features are extracted from the input image in a movie. Following the extraction of these features, many regression techniques—including ridge regression [9] and linear regression [10]—are used to ascertain how to map low-level features to the population size. The goal of this research is to use video security camera frames to estimate the real-time people count in specific interior regions, particularly in retail situations with plenty of furniture and wall occlusions. To benefit from the spatio-temporal coherence between the scenes, they use a supervised learning strategy based on a Long-term Recurrent Convolutional Network (LRCN) regression model combined with a foreground recognition technique. They contrasted it with a YOLOv3 model. pre-trained on the COCO dataset with a detection confidence level of 25%. Each photograph contains the same amount of people as there are things that YOLO categorized as human beings. They discovered instances where

YOLO both missed a number of people who were clearly visible and predicted others who weren't present when they thoroughly inspected some of the expected photographs. This may be the case because YOLO is unprepared to handle the unusual postures and hidden people that are typical of a shop environment. Second, YOLO lacks the spatiotemporal understanding that LRCN-RetailNet possesses and only uses one image to generate its prediction. However, YOLOv3 leaves a significant gap in the prediction time. This aspect must be considered if the final model is to be utilized in real-time applications, and accuracy must be given up in favor of responsiveness.

Density Estimation-based Approaches

Regression approaches successfully dealt with occlusion and clutter issues, but spatial information was mostly disregarded. A previous attempt detailed by [10] proposed learning a linear mapping between local patch properties and associated object density maps by include spatial information in the learning process. It is no longer essential to identify and discover particular items or subsets of them in this method since image density estimation, which is integral across any region in the density map that provides the counting, takes their place. Convex optimization is the framework used to describe the entire issue. Despite the fact that this tactic worked, [11] noted that it was slow in terms of computer complexity. They suggested a subspace learning-based technique for fast estimating density. [12] observed that the earlier crowd density estimation systems' inability to predict crowd density effectively was due to the smaller number of parameters they used. The random forest regression model, which is faster and more scalable than earlier methods (based on ridge regression or Gaussian process regression), was suggested as a way to extract a broader collection of features.

Face Mask Detection

Face masks also help society by permitting other preventive measures, such as strict quarantining and isolation, to be relaxed. Face masks have positive social effects in addition to their obvious medical advantages because they prevent the infection from spreading to those who are most vulnerable [13]. Face masks by definition also cover a significant portion of the human face, which can significantly affect social interactions. The WHO promotes the use of face coverings because they can shield wearers from disease and stop those who are unwell from spreading their illness to others. All at-risk individuals (those 60 and older or those with underlying medical conditions) and anyone exhibiting COVID-19 symptoms are advised to use masks when in crowded areas. However, the WHO advises the general public to wear fabric masks rather than medical ones in circumstances when physical separation is impossible. A face mask dataset was created by Chiang, 2020 using WIDER Face, MAFA, and SSD for detection. However, the dataset accuracy was only 89.6% since the number of parameters was decreased for real-time efficacy. In order to function on both high- and low-computation hardware, RetinaMask [14] created a Feature Pyramid Network (FPN) together with a content attention mechanism and selected ResNet or Mobile Net as the backbone network. Improved feature extraction and classification were achieved by [15] by the use of hybrid transfer learning model and machine learning approaches. The Real-Time Face Dataset (RMFD)'s final accuracy was 99.64%, whereas the Masked Face Dataset's final accuracy was 99.49%. Further training data was provided by the large dataset of 137,016 masked face photos generated by [16]'s suggested masked face images based on facial feature landmarks. Additionally, they developed a smartphone app to demonstrate how to correctly use a mask by determining whether the wearer's mask covered their nose and mouth at the same time. The detecting speed was not addressed, and it's possible that the data won't fully translate to a practical setting. We expanded the PWMFD dataset by utilizing the concept from [16]. The two main methods for facial recognition are:

i. Feature Base Approach
ii. Image Base Approach

Feature Base Approach

A human face has a lot of distinguishing characteristics that set it apart from a wide range of other items. It finds faces by deleting important features like the eyes, nose, and mouth, among others, and uses these faces to determine what kind of face they are. Utilizing a fact classifier that can distinguish between facial and non-facial elements is the most popular technique. Because of their distinctive surfaces, human faces may be identified from other objects. You can tell faces from objects by looking at the highlights' edges. In the part that follows, we will use Open CV to implement a component-based strategy.

Image Base Approach

Image-composition techniques ultimately rely on techniques from factual analysis and AI to identify the key components of both photographs with and without faces. The acquired traits work as discriminant skills or appropriation models for locating faces. In this method, we use a variety of calculations, such as neural networks, HMM, SVM, and AdaBoost learning. In the following section, we'll explore how the Multi-Task Cascaded Convolutional Neural Network, an image-based approach to face identification, can help distinguish between distinct faces.

Algorithm for Detecting Objects

A computer software that can identify, track, and locate an object in an image or video using the computer vision technique for object detection. The unique quality of object recognition is that it can identify both the type of thing (a person, a table, a chair, etc.) and its precise placement within the image [14]. By boxing in the area where the object will go, the position is given. The object may or may not be precisely located by the bounding box. The effectiveness of the detecting system is determined by how well it can identify an object in a picture. One type of object detection is face detection. These object detection techniques can either be taught from scratch or with prior training. The majority of the time, we modify pre-trained weights from pre-trained models to satisfy our requirements and account for various use cases. The two methods for discovering items are as follows:

i. Two-shot detection
ii. Single-shot detection

Two-Shot Detection

These two actions make up the procedure. The initial stage is region proposal, followed by classification of those regions and enhanced location prediction. For two-shot models, the speedier RCNN variants are the most widely used. Here, we use a network, such as ResNet50, as a feature extractor during the region proposal step. To do this, we merely take out the top layer from the network and use the lower layers to draw out details from the images. Due to the network's prior training and ability to extract properties from the photos, this approach is frequently selected. Then a small fully connected network is slid over the feature layer to forecast class-agnostic box proposals with respect to a grid of anchors tiled in space, scale, and aspect ratio. Following that, features from the first stage's created intermediate feature map are removed using these box recommendations [17]. The last section of the feature extractor builds the prediction and regression heads on top of the network using the recommended boxes. Finally, for each suggested box, the output provides the class and class-specific box enhancements.

Single-Shot Detection

The region proposal stage is skipped in single-shot detection, which simultaneously delivers the final localization and content prediction. It should be emphasized that while single-shot detection is in the sweet spot for performance, speed, and resources, it is better suited for jobs like object tracking and object identification in live feeds where the speed of prediction is more crucial. One well-known example of this tactic is "YOLO". Two-shot detection models perform better in general. The mean classification error over the projected class labels is used to assess the effectiveness of a photo classification model. Comparing the variations between the predicted and expected bounding boxes for the expected class allows one to assess a model's efficacy for single-object localization. The feature extraction approach is utilized in object detection to extract edge or corner features that can be used to distinguish between different sorts of data [18]. These traits are included in a machine learning model, which will classify them into various groups and use this data to assess and categorize new products. An algorithm is required to solve the problem of identifying the object classes that are represented in the image. Object identification is one of the primary outcomes of machine learning and deep learning algorithms. People are skilled at recognizing various visual elements such as items, objects in scenes, and movie scenes. Teaching a computer to understand an image is the goal; this is a task that humans do with ease. Figure 2. provides the capability to locate and identify several objects inside an image and shows the typical operation of an object detection model. The three computer vision tasks that follow can therefore be identified as follows:

- Classification of Images: determine the kind or category of an object within an image.
- Object localization: involves finding objects in a picture and using a bounding box to show where they are.
- Object detection: use a bounding box to find objects in an image and identify the kinds or classes of those things.

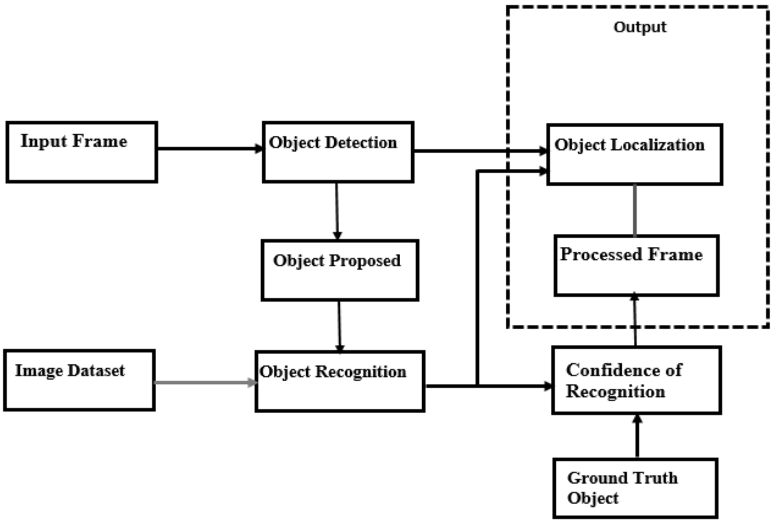


Figure 2. Object Detection Model

Object Recognition using YOLO

The "You Only Look Once" (YOLO) single shot detection method was first described by [19]. Its tremendous speed more than compensates for its less-than-perfect object detecting algorithms, giving it a fantastic balance between speed and accuracy. The YOLO method makes use of an input image to train a single neural network to predict bounding boxes and class labels for each bounding box. Despite operating at 45 frames per second (or up to 155 frames per second for a speed-optimized version of the model), the method produces fewer accurate predictions (more localization errors, for instance). The input image is initially split into a grid of cells by the model. A cell can be said to have predicted the bounding box if its center is inside its bounds. For every grid cell, a bounding box comprises the following dimensions: x, y, width, height, and confidence. Furthermore, every cell serves as the foundation for a predicted class. The image is divided into 7x7 cells for the YOLO method, and we predict a particular number of bounding boxes, a confidence score, and classes score for each cell. The NMS technique can be used during testing to get rid of unnecessary item detections. An enhanced version of YOLO and YOLOv2, YOLOv3 is available. In the first iteration, a generic architecture was offered; in the second, the design and the bounding box proposal were enhanced using preset anchor boxes; and in the third, the model architecture and training procedure were iterated. The primary modification to its network structure is the addition of residual blocks, which guarantees that the model can still converge quickly even if the YOLOv3 network deepens. The loss function merges low-level and high-level semantics using the multi-scale fusion technique, enhancing its sensitivity to small targets. The loss function incorporates binary cross entropy loss to more efficiently address the overlap problem. The Darknet-53 network, which has 53 convolutional layers altogether, is the foundation for YOLOv3. The ResNet [20] network's residual block structure is introduced. Each residual block consists of two layers of convolution and is connected by a jump link. A speed comparison experiment on the study's backbone network was advised by YOLOv3. In order to evaluate the test accuracy of a single-size image, each network is trained using the same configuration and tested using 256256 images. Here, the number of feature map channels can be decreased via 1-1 convolution, which will make the computations and model parameters simpler [21]. In comparison to earlier versions, Resnet can make the network deeper, faster, simpler to optimize, and with fewer parameters. As a result, it can deal with issues like extreme network degradation and difficult training. In YOLO V3, anchor boxes are used to forecast bounding boxes. The width and height of the anchor box are likely clues as to what it signifies. For instance, by prioritizing clustering, we may gather the data required to utilize a pixel to forecast an object. Numerous things close to this pixel have shapes that we can predict. It is not an arbitrary prediction. The size of the anchor box must be known, and this may probably be done by looking at the label. Some cluttered, overlapping, or noisy regions may be mistakenly identified as the object during object detection. By returning the area in the image that is nearest to the detected object and has the highest probability score, YOLOv3's regional proposal will address this problem. The high recall rate of this regional proposal technique, which improves the model's performance, is another advantage. Higher detection accuracy for several items in a single frame is made possible by a high recall rate, which guarantees the construction of the list of areas containing the object [22]. Image 2 describes how the YOLO model behaves when trained on data. To get the image's attributes, the loaded Darknet-53 convolutionally layers the input images. Boundary boxes and class prediction are used to generate the output following the training phase using the detector function.

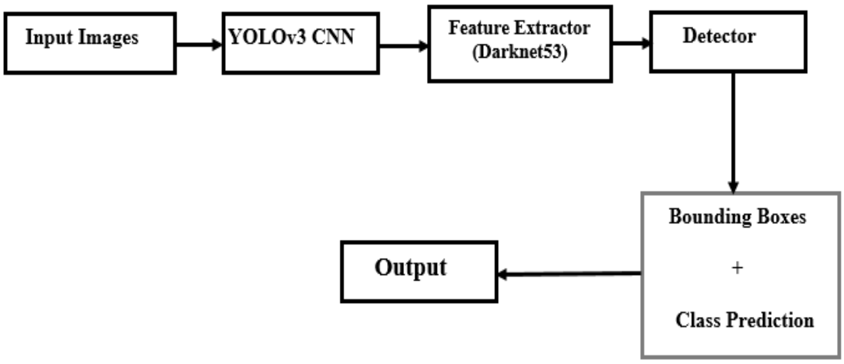


Figure 3. Standard YOLOv3 Object Detection Model

RELATED LITERATURE

Due to the impending global pandemic brought on by the coronavirus epidemic in 2020, face coverings are now required in public spaces [23Although a lot of research in this area focuses on facial expression detection and distinguishing between masked and non-masked faces, the identification of an individual's identity when wearing a face mask has not been addressed [24]. This is due to the fact that it is difficult to recognize a face when a face mask covers most of it. As a practical tactic to halt the spread of infectious diseases like COVID19, the author [25] proposed face mask recognition. Retina Face Mask combines high-level semantic data with a variety of feature maps using a feature pyramid network. The object detection network in this study is based on ResNet, a general CNN feature extractor. This model employs FPN as a neck to improve the feature map. The head end, known as SSD, serves as a classifier or predictor. The small dataset that is currently available has changed the transfer learning strategy. The work makes use of a context attention module, which can extract various receptive fields from a single feature map, to enhance the performance of face mask detection. The model's performance has increased by 4% as a result of transfer learning, which also gave insights from related tasks. [26]. This paper's main goal is to employ deep learning models to increase facial recognition accuracy. To accomplish the goal, YOLO, one of the deep learning libraries, is used. The use of ancient and modern methods for resource use is contrasted to determine which strategy is more effective. A convolutional neural network with a maximum of seven pooling layers is used in this case. The study's findings demonstrate that processing massive amounts of data requires a GPU with a high configuration. As training volume increased, better results were produced. The size of the CNN affects how quickly people learn. Lower learning rates for networks between medium and large. This paper paves the way for the detection of small and partial faces. [27] It could be difficult to identify little faces. This was accomplished by enhancing YOLOv3, which produced an anchorage box with a high average intersection ratio. Additionally, a powerful k-means algorithm technology is used. By merging target classification and detection training and instantly regressing the position and category of the target detection frame in the output layer, the enhanced YOLOv3 is applied in this instance to transform the detection problem into a regression problem. This study shows that the accuracy of dense small-scale face detection is enhanced by increasing the prediction layer's breadth from three to four dimensions. [28] created a model that, by assessing whether or not someone is wearing a helmet, can instantly spot infringement. TensorFlow, Keras, and OpenCV were also utilized for this project. When compared to some earlier algorithms that usually predicted wrong when a rider covered their face with clothing, their suggested model showed significant improvements. On the test, their overall accuracy score was 98% detection frame in the output layer. This paper discusses how the accuracy of detection of dense small-scale faces is improved by changing the width of the prediction layer from three dimensions to four. [22] developed a model that detects whether a person is wearing a helmet in real time thereby, detecting any violations. This project was also implemented with the help of TensorFlow, Keras and OpenCV. When compared to some earlier algorithms that provided inaccurate predictions anytime a rider covered their face with clothing, their suggested model demonstrated notable improvements. When tested, they obtained a 98% accuracy rate overall. The approach is however constrained by a specific number of missed detections due to the lower average precision [29]. It could be difficult to recognize someone's face if it is frequently obscured or smudged in a complicated environment. The YOLOv3 used in this inquiry has been modified. Clustering the dataset to locate the best preceding box enables this. YOLOv3 has the power to rewrite labels in face-intensive environments, stopping research from yielding subpar results. The YOLOv3 requires reasonable enhancements to some datasets in order to have a good improvement in a complicated scenario. The YOLO algorithms v4 and v3 are used in this research's methods for object recognition in video and picture files. The model was trained using a dataset with a wide range of objects and lighting conditions. YOLOv3 forecasts the class and location of objects while extracting characteristics using CNN and darknet53. Using batch normalization, leaky reLU, anchors, batch norm, and fixed padding, the model's hyper-parameter was established. Non-max suppression requires n layers of classes for feature extraction, which helps to prevent bounding box overlap. Depending on the location, the time of day, and the density of the objects, it could be difficult to detect many things during video surveillance. The research's multiple object detection technique is appropriate for traffic and surveillance applications. Facemasks may be recognized in live streaming videos and even in photos of people's faces according to a system created by [30] that uses the object recognition tool Single Shot Detector (SSD). Neural network transfer learning techniques are used to identify a facemask in video streams and images. The model performs well with 100% accuracy, 99% precision, and 99% recall, according to experimental results. Simple machine learning tools like Tensorflow, Keras, OpenCV, and Scikit-learn can be used to recognize facial masks even when they are moving. The method achieves up to 95.77% and 94.58% accuracy on two different datasets, respectively. [31] suggested using an automated system to find people wearing facemasks in public. The model is constructed on top of the trained and taught InceptionV3 modern deep learning model. The dataset is trained using the Simulated Face Mask Dataset (SMFD) dataset. In this case, the public face dataset is simulated after receiving a mask. This improves the model's training and testing processes. The model achieves 100% testing accuracy and 99.9% training precision by utilizing the picture augmentation technique to improve training and testing with little data usage. An effective real-time system approach to computer vision resulted in the development of a model that covers the areas of validating, detecting, and tracking. It was used to spot facemask infractions as well as social retreat in public settings. The Raspberry-pi4's built-in robust model is approved for use with public surveillance cameras. Lightweight neural networks are used in the analysis in the publication. When viewing real-time video surveillance cameras in public spaces to spot out violations of wearing a facemask and maintaining social distance, OpenCV and transfer learning techniques with Single Shot Detector (SSD) were used to achieve resource limitation and accuracy recognition. Based on YOLO object detection, [32] created a brand-new, trustworthy real-time ALPR (Automatic License Plate Identification) system and provided three segments with thorough explanations, higher identification rates, and their own datasets of 4500 photos. They have amassed a dataset with 800 training photographs utilizing a variety of real-world applications, such as traffic monitoring, because each form of image detection depends on the image resolution, backdrop, and camera size. 800 training image data with a plain background were processed using YOLOv2 for the ALPR approach dataset, UFPR-ALPR, in the SSIG (Segplate Database) dataset from Brazil. The basis of ALPR is the idea of DL (Deep Learning). The reprocessed 800-training data was then used to construct a huge dataset of 4500 pictures with the intention of excluding positive faces. Numerous methods to boost recognition rates were found after processing the video over the two datasets of 800 and 4500 images. For 30000 LP characters to be detected and 93.53% temporal redundancy to be reached, the recognition rate must be less than 78.33%. They asserted that there may be excellent opportunity for rate and ALPR pipeline enhancement. [33] gave an illustration of how video scenes could be utilized to find, identify, and pinpoint YOLO action. Every video element needs to be fixed for each frame in order to visualize data. By using their own, original dataset, it might have demonstrated that the YOLO strategy is quicker and more efficient than other approaches. 30 single frames from a dataset are fed into a model to predict the action level during the first stage of the execution process. When the system was tested, a single CNN query had an 88% performance capability.

RESEARCH METHOD

The creation of a face mask identification system based on YOLOv3 is highlighted in this section. Additionally, it describes the transfer learning-based special model, datasets, model architecture, and datasets for the proposed system.

- **Custom Datasets Model**

Every project is built on data; hence it is crucial to properly understand the data before building a model. The dataset used for the study will contain the facial data of 30 distinct users, 10 faces for each category. The model will be trained using each person's clear face, masked face, and erroneous masked face. As a result, the person could be recognized by the model both with and without a face mask. The number of successfully and incorrectly masked, non-masked, and masked face photographs in each user's face data is identical. After pre-processing the photos to make them smaller, the entire dataset was divided into a train, validation, and test set. Each set includes a._darknet, the.jpg photos, and the corresponding.txt files. The correct chronological order is maintained for labels in label files. With each layer entirely frozen, the dataset was utilized to train the model for around 10 epochs. After the model had been trained and evaluated on a camera, it needed to be adjusted by having all its layers unfrozen. Once the learning rate was slowed down, successful outcomes were occasionally achieved. The dataset's details are provided below:

a) Number of persons’ face data: 30
b) Number of masked faces of each user (training set): 10
c) Number of non-masked faces of each user (training set): 10
d) Number of improper masked faces of each user (training set): 10
e) Number of masked faces of each user (testing set): 5
f) Number of non-masked faces of each user (testing set): 5
g) Number of improper masked faces of each user (testing set): 5
h) Number of faces in training set: 30
i) Number of faces in testing set: 15

- **Yolov3 Algorithm**

The YOLOv3 algorithm, which is visually depicted in figure 4, served as the foundation for the algorithm employed in this work. The 53 convolutional layers of YOLOv3's architecture were discovered on ImageNet. For YOLO v3, the detection-focused addition of fifty-three

(53) new levels brings the total number of layers in the underlying architecture to 106 (106). YOLOv3 locates objects by applying detection kernels to feature maps of three (3) different sizes in three (3) distinct locations throughout the network. The first identification is done through layer 82. The network down samples the image for the first eighty-one (81) layers in order to give the 81st layer a stride of thirty-two (32) pixels. If the starting image had been 416 416 in size, the final feature map would have been 13 13. The 13 13 255 detection feature map is what we get in this case after using the 1 1 detection kernel for one detection. Layer 79's feature map is upsampled by 22 to dimensions of 26, 26 after a few convolutional layers. Next, depth concatenates this feature map with the layer 61 feature map. Subsequently, following the merging of the feature maps, a few 1 1 convolutional layers are added to integrate the features from sixth- one layer (61).

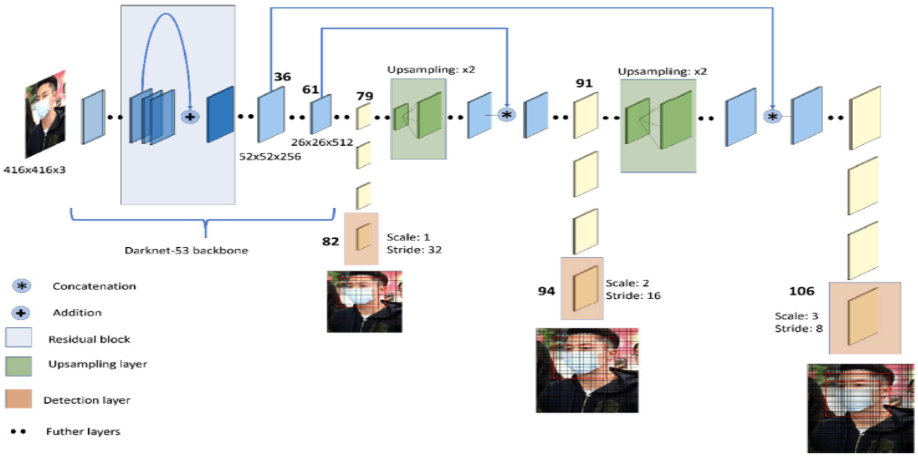


Figure 4. The network architecture of YOLOv3

The second detection is then performed using the 94th layer, resulting in a detection feature map that is 26 by 26 by 255 in size. The whole architectural layout of YOLOv3. The model is trained using the face-mask dataset for the face-mask identification challenge, leaving the final three (3) layers unfrozen. The complete model is subsequently trained for a later weight modification. They employed the same nine (9) anchors as the training model. The YOLOv3 algorithm employs five (5) different types of layers, and they are as follows:

- **Convolution Layer**: It is necessary to learn the parameters of the several filters that make up the convolution layer. The input volume is less than the filters' height and width. The following formula will be used to determine the shape of the detection kernel in this kind of layer:

  Shape of the detection kernel = 1 * 1 * (B * (5 + C))

  Where:

  ❖ B is the number of bounding boxes that can be predicted by a single cell.

  ❖ The number "5" represents one item confidence and the four bounding box characteristics.

  ❖ C will determine the number of classes.

For this project, B = 3, C = 2 (MASK and NO_MASK). Hence, the kernel size will be 1 * 1

* 21. The feature map created by this kernel will have the same width and height as the prior feature map, as well as the previously mentioned detection properties along the depth.

- **Shortcut Layer:** is a skip connection similar to the one that is used in the Resnet. The output of the shortcut layer is obtained by adding feature maps from the previous layer and the form parameter that is defined (in the configuration file) backward from the shortcut layer.

- **Residual Block:** Residual block also known as an identity block; a residual block is a ResNet building block. A residual block merely occurs when an activated layer in the neutral network is quickly transmitted to a deeper layer.

- **Upsample Layer:** The Upsample layer operates in a rather straightforward manner. Bilinear upsampling is used to increase the feature map from the previous layer by a factor of stride. As we go deeper into the network, the image size keeps growing smaller, necessitating the usage of upsampling to increase the image size so that it may be added to more layers.

- **YOLO Layer:** The YOLO Layer is fairly similar to the detection layer that was previously stated. Only the anchors that are referred to by the mask stage's characteristics are used out of the nine anchors listed in the YOLO layer.

Model for Detecting Face Masks

In this study, YOLOv3 is used to search for face masks. The input image, which is a frame from a video, is examined by the neural network, which divides it into three categories: mask, no mask, and wrongly worn mask. A total of 30 photos were used to train the model, 10 of which were split up into the categories of masked faces, unmasked faces, and incorrect masks. We must therefore filter and eliminate any unnecessary boxes. Getting rid of any boxes with a low likelihood of having something found inside of them. This can be done by using a confidence threshold and only maintaining the boxes whose probability is higher than the threshold. It eliminates strange object detections. Non-max suppression, which employs the "intersection over union" strategy, is followed by Thresholding. As implied by the name, it calculates the ratio of the intersection and union of the two boxes given two boxes as input. This kind of filtering ensures that each detected object only receives one bounding box. The class label is not considered throughout this non-max suppression process. while merging, to see whether any of the neighboring boxes have the class label set to MASK. The subsequent pseudocodes were run.
If YES: Final merged bounding box will be labeled as MASK.

If NO: Final merged bounding box will be labeled as NO_MASK.

Else: Merged bounding box will be labeled as INCORRECT_MASK

The final output layer's conversion of the values into a probability distribution is shown in Figure 5. The softmax function is used to do this. When dealing with classification difficulties or multi-class classification problems, this feature is beneficial. The item on the list with the greatest confidence score represents the last predicted class [34]. The class with the highest likelihood will serve as the foundation for the final forecast.

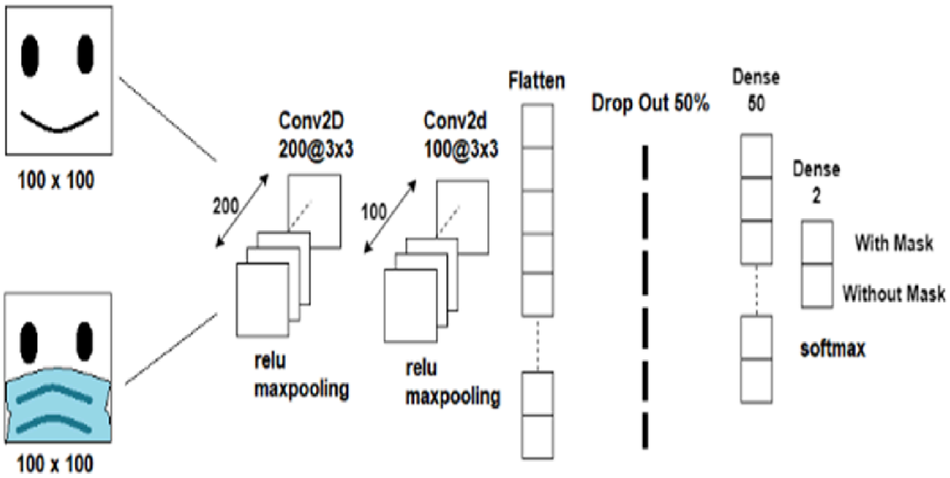


Figure 5. Network Architecture

The Proposed System

We suggested a single-shot object detection technique utilizing Yolov3 that is suitable for detecting numerous people in the image to achieve real-time face mask detection. The architecture that best illustrates how our technology prevents the spread of COVID19 is shown in Figure 6. Pictures of three groups of people—those who are wearing masks, those who are not, and those who are wearing the wrong kind of mask—make up the models on a dataset. This project offers a way to classify images and keep tabs on the number of people who wear face masks every day by drawing bounding bars (red or green) around people's faces depending on whether or not they are wearing masks. subsequently sends a message to someone who is not wearing a mask to inform them. Having a dataset with labels and annotations was necessary. To fulfill this task, we personally created a tiny custom dataset, meticulously gave labels, and used transfer learning. The same dataset, which consisted of 30 total images from the three categories and around 10 photos, was used to train our models. Each face's bounding box in each image was drawn with great precision. Annotation records are created with all the data needed for the three models taken into consideration in this work, including bounding box information, picture names, classification, and labels in various formats. The hand-edited pictures MAFA, WIDER FACE, and others make up the dataset. Because Keras supports practically all neural network models, it has been used to implement the models on top of Python's TensorFlow machine learning library. The filename is in the grid-based ".h5" file format, which is solely supported by Keras and is suited for storing multidimensional arrays of numbers. The YOLOv3 algorithm type has been applied to the classification of the images. Each image will have a label file with the same name as the image file but a.txt extension. A person is shown clutching a box with square corners in the video output. When a person without a mask is detected by this technology, a photo of their face is captured and forwarded by email to the appropriate higher-ups.

Modules for the Proposed System

The suggested method makes advantage of various methods for identifying magician, including:

- **Datasets Collecting:** Transfer learning was used to collect data sets, enabling the acquisition of a number of data sets with face masks, without any masks, and with the wrong masks. We can obtain results that are fairly accurate depending on the number of images taken**.**
- **Datasets Extracting:** extraction of features from datasets that have and do not have masks.
- **Models Training** The models should be trained using openCV and the Python package keras**.**
- **Face mask detector and classifier:** The entire system relies on the YOLOv3 algorithm, which combines keras and opencv to recognize face masks from pre-processed human faces in bounding boxes. It is the finest choice for our needs based on its attributes for speed and precision.
- **Tracker:** In order to alert a person without a mask, the system must keep track of precise statistics about each image and person during the entire time they appear in the video sequence.

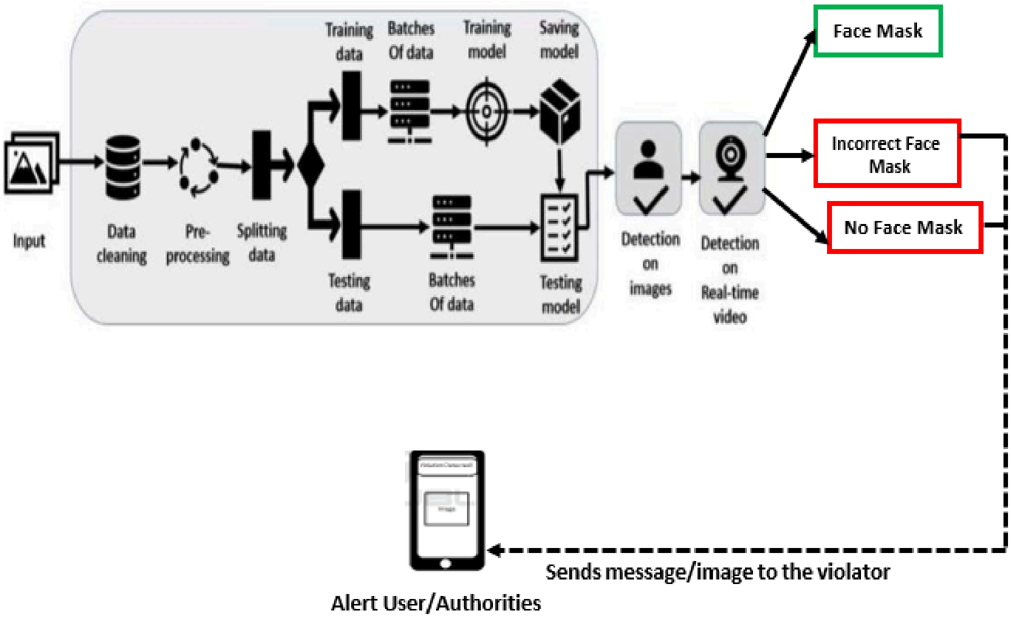


Figure 6. Proposed System Architecture

Training Datasets

The positions of the training and validation images, the number of labels, and the names of the labels in the training data must all be recorded in a data file called.h5 before the model can be trained. Launch train.py, which accepts the following inputs, to train the model.

- **img:** input image size
- **batch:** batch size
- **epochs:** number of epochs
- **data:** path to the data.h5 model file
- **cfg:** model to choose among the preexisting in **models**
- **weights:** initial weights path, defaults to yolov3.pt
- **name:** renames output folder
- **device**: Whether to train on cpu or gpu.

We loaded the weights of the pre-trained model using transfer learning rather than training the model from scratch after accounting for the aforementioned criteria. Put 80% of the images in the train.txt file and 20% in the test.txt file. Depending on how many batches you choose to perform the activity in, the weights are continually saved as the training begins.

Training of YOLOv3

To train the YOLO v3 model, we started with the weights of a previously trained model. We used the same weights as were listed on Joseph Redmon's initial website. Using the dataset, we trained the model for approximately ten epochs, freezing all but the top three layers. The model was then put to the test using a webcam, but the outcomes need improvement because they weren't impressive. Once every layer of our model had been unfrozen, we trained it for further epochs to improve it. We finally had some success slowing down the learning rate after several failed attempts.

Pseudocode Training:

```
1. Create your own class names and annotation files using the syntax shown below: Row syntax: image_file_path box 1, box 2,... boxN. Box format: class_id, x_min, y_min, and x_max
2. Convert the pre-trained weights into the ".h5" format suggested by Keras.
3. Train for a few epochs (until the no improvement stage) is reached while freezing all layers save the last ones.
4. Reduce the learning rate, unfreeze each layer, and train each weight until you reach a new limit.
5. End
```

Testing on webcam or video in keras using OpenCV: Using OpenCV, test Keras with a camera or video:

- Use a webcam or another previously saved video testing file to capture the footage.
- The boxes, scores, and classes that were established should be used to draw the boxes on the frame in the appropriate classes (green or red, depending on the output).
- Count the total number of boxes in each of the two classes and put the total number in a variable at the bottom of the screen.
- Following each frame iteration, we can obtain the final video with the necessary effects.

RESULTS AND DISCUSSION

The system, an image detection system, employs real-time face mask detection. One of the few COVID-19 prevention methods available in the absence of immunization, masks are crucial for safeguarding people's health from viral airborne transmission. Therefore, it is crucial for us to identify whether someone is using a mask and whether they are adequately using it in order to track the infection [35]. At the moment, a dataset must be fitted into data-driven detection and classification algorithms for them to work successfully. The transfer learning technique was applied to the dataset for this study in order to identify and classify masks. In this collection, 30 photographs from the following three categories are featured, together with their bounding boxes in the opencv file format:

i. With mask
ii. Without mask
iii. Mask worn incorrectly

This dataset is ideal for detection and classification algorithms because to the possibility of several targets of various classes showing in a single image. Using the Yolov3 algorithm and transfer learning, this challenge has been completed. Following that, these datasets are transformed into.h5 file types. The finished output depicts the video's subject clutching a square-bound box. An image of the person's face is taken and provided to both them and the higher authorities if this equipment notices someone wandering around without a mask. This suggested technique can be used to track face mask users in public in real time due to the appearance of a new Corona Virus. By automatically keeping an eye on people in public spaces, our approach assists people who manually or physically keep an eye on people. We have implemented the following three steps in the proposed system:

a. Data gathering and pre-processing.
b. Creating and preparing the
c. Application of the Model

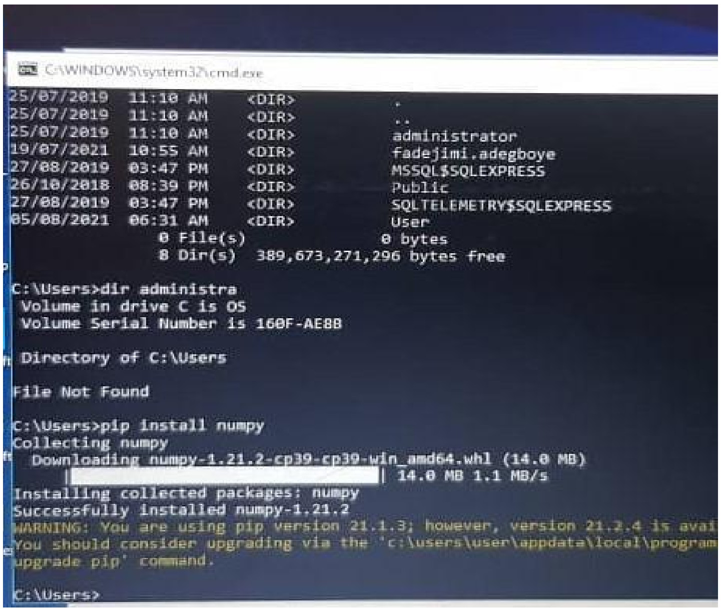
a. Collection of Data and pre-processing

The model was trained with face cropped data, which consists of pictures of faces in different positions and perspectives, both with and without masks. With Keras and OpenCV, a real-time automatic face mask detection system was created. 30 photos make up the dataset used to train the recommended model. Three categories are used to arrange the data: faces with masks, faces without masks, and faces wearing the incorrect masks.

b. Building and Training the Model

The suggested approach loads a custom dataset and trains the algorithm using captioned photos. The photos are now reduced in size and changed to.h5 file format. TenserFlow is used for training the Opencv-based model. Following 10 iterations of improvement, the batch size in the first phase is set to 16. We employ a learning rate of 0.001 and a scheduling factor of 0.1. During this stage, a steady loss field converges faster. Ten different epochs were used for the training procedure. The model uses a webcam to spot face masks, and once they are located, they are marked with a square box. Then, using the.pip install syntax, the following python libraries were installed and used on a command line:

numpy, pillow, torch, tqdm, terminal_tables, torch_vision, matplotlib, tensor_board, and opencv-python (as seen in figure 7). Finally, use python video.py --weights_path checkpoints/yolov3_ckpt_35.pth to carry out the detection.



```
25/07/2019  11:10 AM    <DIR>          .
25/07/2019  11:10 AM    <DIR>          ..
25/07/2019  11:10 AM    <DIR>          administrator
19/07/2021  10:55 AM    <DIR>          fadejimi.adegboye
27/08/2019  03:47 PM    <DIR>          MSSQL$SQLEXPRESS
26/10/2018  08:39 PM    <DIR>          Public
27/08/2019  03:47 PM    <DIR>          SQLTELEMETRY$SQLEXPRESS
05/08/2021  06:31 AM    <DIR>          User
               0 File(s)              0 bytes
               8 Dir(s)  389,673,271,296 bytes free

C:\Users>dir administra
 Volume in drive C is OS
 Volume Serial Number is 160F-AE8B

 Directory of C:\Users

File Not Found

C:\Users>pip install numpy
Collecting numpy
  Downloading numpy-1.21.2-cp39-cp39-win_amd64.whl (14.0 MB)
     |████████████████████████████████| 14.0 MB 1.1 MB/s
Installing collected packages: numpy
Successfully installed numpy-1.21.2
WARNING: You are using pip version 21.1.3; however, version 21.2.4 is avai
You should consider upgrading via the 'c:\users\user\appdata\local\program
upgrade pip' command.

C:\Users>
```

Figure 7. Installing python libraries

c. Implementing the Model

Our method takes input footage from any camera equipment and a unique dataset. We can use OpenCV to read files from the local file system and read streams from remote devices to display the prediction results in movies. TensorFlow was employed to load the trained model. Using OpenCV, the weights were initialized from the configuration file. The output of the model includes at least the following fields after the mask detection dataset has been trained:

- An array of images used in the prediction
- An array of predictions generated by the model, of tuples of the following format

(a) The top-left corner's x and y coordinates, normalized to the image's width and height.
(b) The bounding box's bottom right corner's x and y coordinates, normalized to the width and height of the image.
(c) a floating-point confidence levels
(d) a number indicating the predicted class 3. An array of label names

An iterable stream of picture frames is read by the video source. Each frame of an image is sent to our model at its original resolution (1080 pixels wide by 1920 pixels high, for instance). The inference findings produced by our method follow the aforementioned guidelines. In order to predict the class names and degrees of confidence for each object visible in this frame of the image (facial masks, no face masks, incorrectly worn face masks, etc.), the results are utilized to construct bounding boxes. The region of interest is cropped out of the main frame, reshaped, and then fed to the model after the bounding boxes for the faces in the frame have been determined. A video encoder receives the drawn frame and saves it to use as a frame in the final video. Figure 8 demonstrates how the system automatically determines whether people are wearing face masks while streaming real-time video in public spaces. Every time someone is spotted without a mask on, a photo of them is taken and forwarded to the victim and higher authorities so that they can take the necessary action.

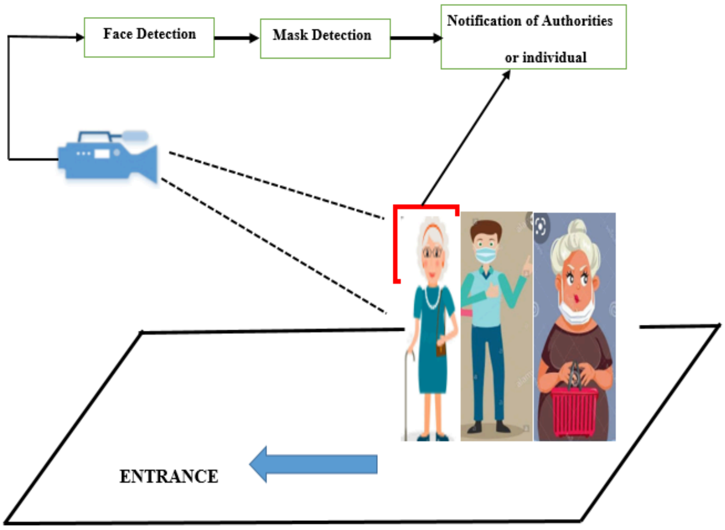
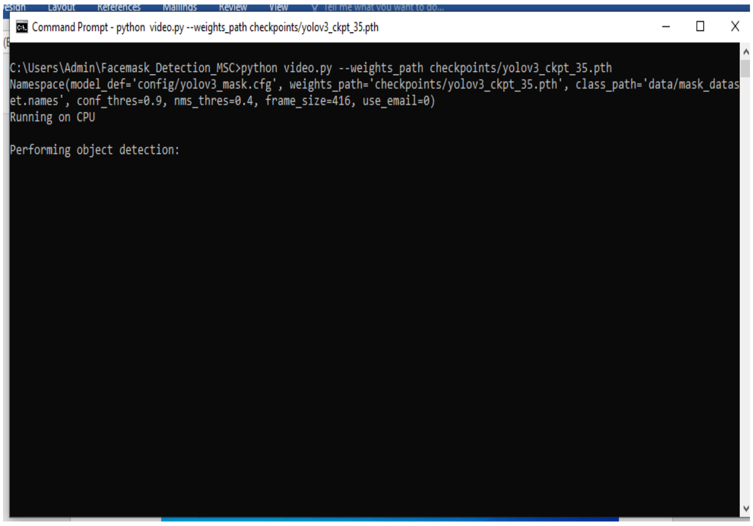


Figure 8. Demonstration of the System

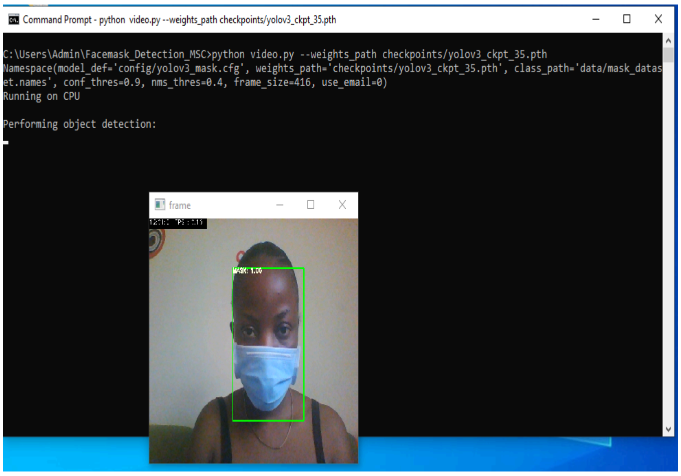
This automatic approach notifies higher authorities and the subject of the subject's appearance and assists in determining whether the subject is wearing a mask. The training set consists of thirty (30) images, of which ten (10) have masks on, ten (10) do not, and ten (10) do but improperly. An iterable stream of picture frames is read by the video source. Each frame of an image is sent to our model at its original resolution (1080 pixels wide by 1920 pixels high, for instance). The inference findings produced by our method follow the aforementioned guidelines. We project class names and confidence levels on the input video, which is left unaltered, and utilize the findings to build the bounding boxes. As a result of OpenCV video processing, the computing cost of model prediction increases. Overhead is produced by rendering the visualizations, reading frames from the input video, and writing the rendered frame to the output video. The model is tested by showing a bounded box with the confidence score on top of the bonded box after it has fully trained via transfer training. The camera is used by our proposed technique to identify each person's face, and it then displays a bounding box in green or red (Figure 12) to indicate whether they are appropriately wearing, are not wearing, or are both (Figure 12). The device will take a photo of anyone who is not wearing a mask and send the photo to the victim and higher authorities. for every face, face mask, or face mask worn improperly that can be spotted in this frame of the picture. A video encoder receives the drawn frame and saves it to use as a frame in the final video. The email alert that is sent whenever a person is seen on camera without a face mask is shown in Figure 14. A red bounding box that states there isn't a mask present informs the administrator. Figure 8 shows the system's command-line interface for item detection. carry out an order. Figure 10 shows how, when the algorithm detects a mask-wearing face, a green bounding box with a confidence score at the right top of the bounding box is displayed. Figure 11 demonstrates how the system sends an email to the administrator whenever it notices a face without a mask. Figure 12 shows the three groups with a lot of faces in a single frame. The outcomes, which are displayed below, include an email notification and a new video. The model was built using transfer learning and trained for 10 iterations at a learning rate of 0.001 to obtain a validation set accuracy of 96.35%. The confidence level is displayed as 0.99, 1.00, 0.98, etc. in the graphics below.


Command Prompt - python video.py --weights_path checkpoints/yolov3_ckpt_35.pth

```
C:\Users\Admin\Facemask_Detection_MSC>python video.py --weights_path checkpoints/yolov3_ckpt_35.pth
Namespace(model_def='config/yolov3_mask.cfg', weights_path='checkpoints/yolov3_ckpt_35.pth', class_path='data/mask_datas
et.names', conf_thres=0.9, nms_thres=0.4, frame_size=416, use_email=0)
Running on CPU

Performing object detection:
```

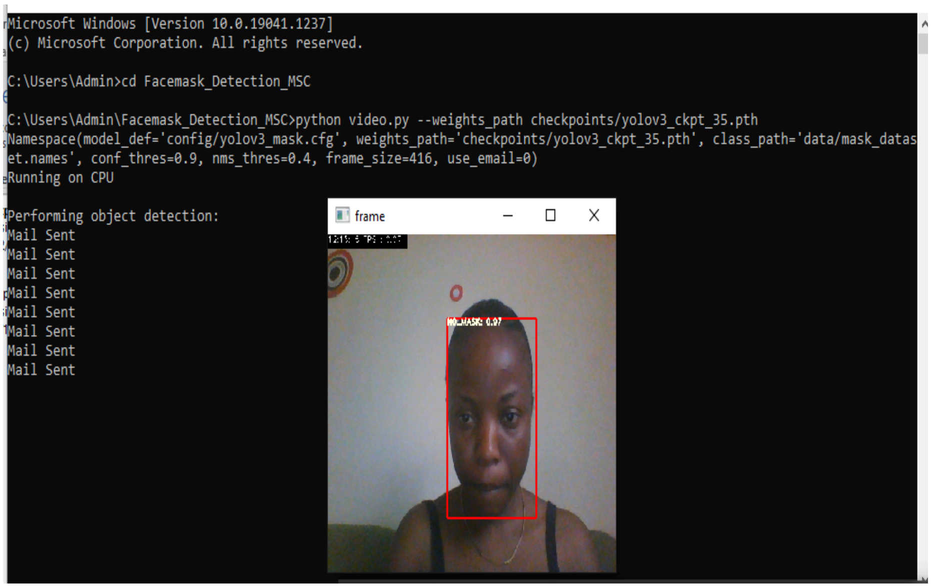
Figure 9. Performing Object Detection


Command Prompt - python video.py --weights_path checkpoints/yolov3_ckpt_35.pth
frame
MASK: 1.00

```
C:\Users\Admin\Facemask_Detection_MSC>python video.py --weights_path checkpoints/yolov3_ckpt_35.pth
Namespace(model_def='config/yolov3_mask.cfg', weights_path='checkpoints/yolov3_ckpt_35.pth', class_path='data/mask_datas
et.names', conf_thres=0.9, nms_thres=0.4, frame_size=416, use_email=0)
Running on CPU

Performing object detection:
```

Figure 10. Face Mask Detection



```
Microsoft Windows [Version 10.0.19041.1237]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin>cd Facemask_Detection_MSC

C:\Users\Admin\Facemask_Detection_MSC>python video.py --weights_path checkpoints/yolov3_ckpt_35.pth
Namespace(model_def='config/yolov3_mask.cfg', weights_path='checkpoints/yolov3_ckpt_35.pth', class_path='data/mask_datas
et.names', conf_thres=0.9, nms_thres=0.4, frame_size=416, use_email=0)
Running on CPU

Performing object detection:
Mail Sent
Mail Sent
Mail Sent
Mail Sent
Mail Sent
Mail Sent
Mail Sent
Mail Sent
```

Figure 11. No Face Mask Detection

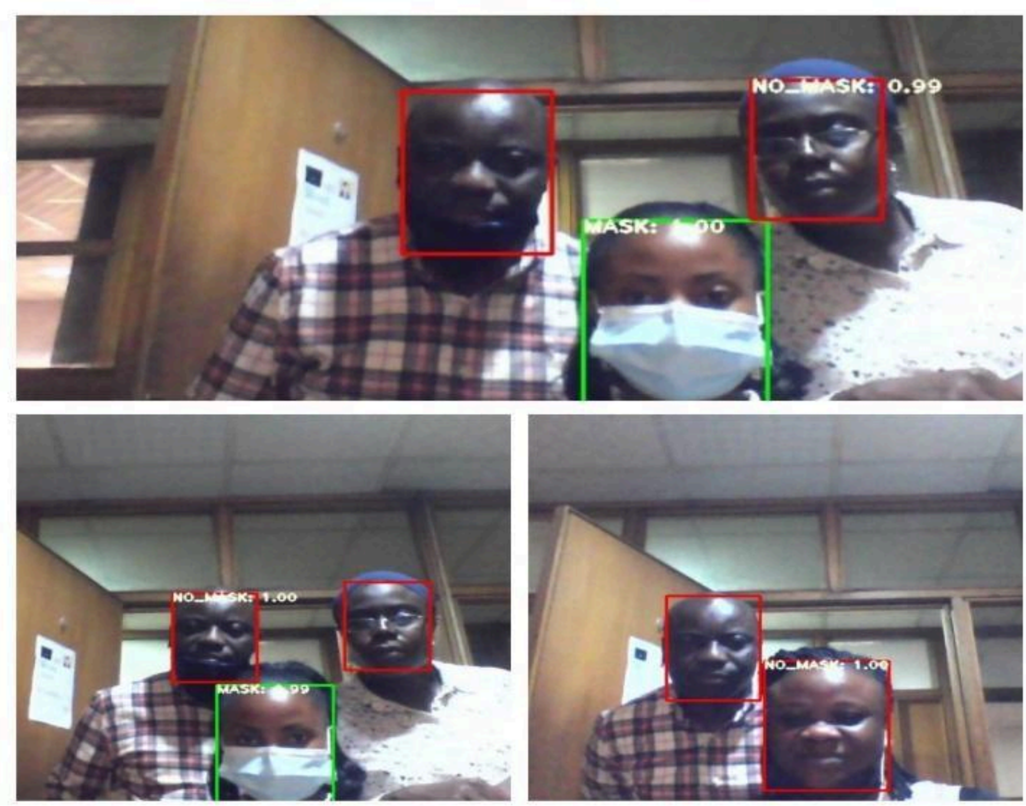


Figure 12. Detecting Multiple Faces with Mask and without Mask

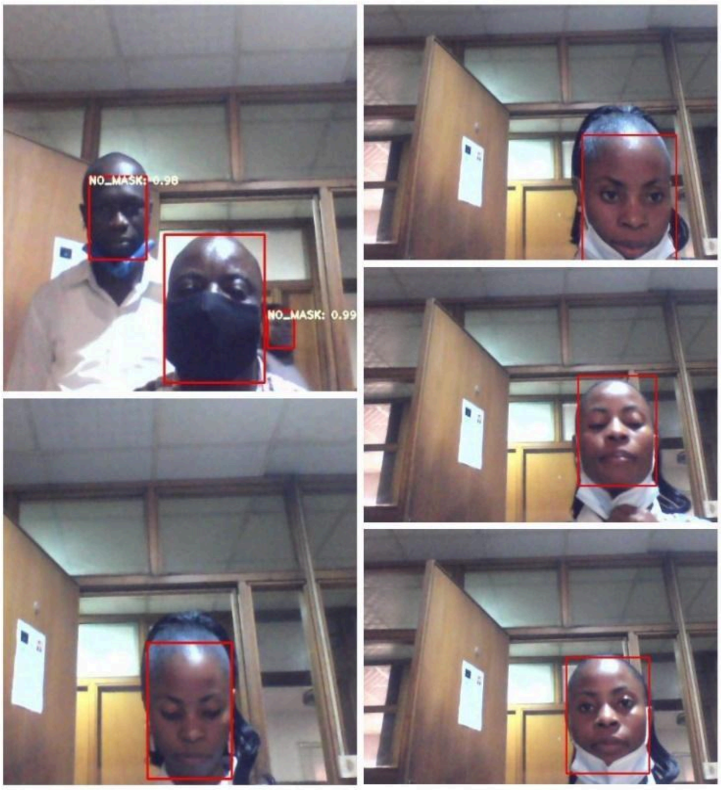
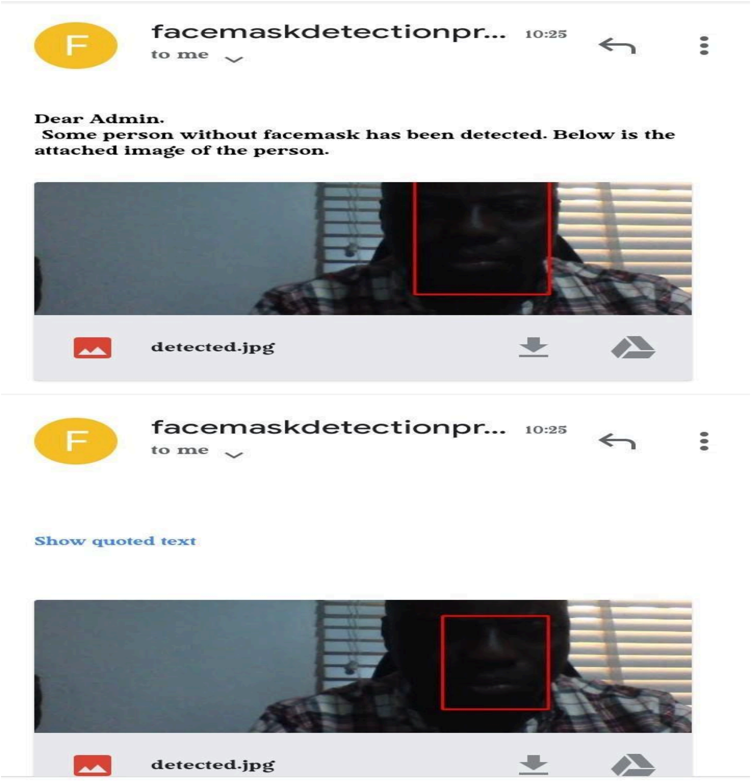


Figure 13. Improper Wearing of Face Mask



facemaskdetectionpr... 10:25
to me

Dear Admin.
Some person without facemask has been detected. Below is the attached image of the person.

detected.jpg

facemaskdetectionpr... 10:25
to me

Show quoted text

detected.jpg

Figure 14. Email notification

CONCLUSION:

In our research, we have put forth a device that can instantly ascertain whether someone is wearing a face mask and alert higher authorities if they are not. Single shots were found using the YOLOv3 approach, which also provides a far greater real-time frame rate. Our research helps law enforcement or higher authorities identify whether someone is wearing a mask; if not, they will also have the victim's photo and can take additional action. The public can utilize Keras OpenCV to ensure that individuals are wearing face masks and stop the COVID-19 virus from spreading by using this suggested solution. The suggested approach can be applied in a range of locations, including offices, shopping centers, train stations, and airports. The model was built using transfer learning and trained for 10 iterations at a learning rate of 0.001 to obtain a validation set accuracy of 96.35%.

REFERENCES

[1] Ashwin, A., et al., (2022) "The Impact of Mask Use on Reducing COVID-19 Spread During the Early Stages of the Pandemic," J. P. H., 10.1371/journal.pgph.0000954. PLOS Public Health. 2022; 2(9): e0000954.

[2] Cheng C, Barceló J, Hartnett AS, Kubinec R, Messerschmidt L. (2020) "COVID-19 government response event dataset (CoronaNet v. 1.0). Nature Human Behaviour". :756–68. doi: 10.1038/s41562-020-0909-7

[3] Anderson, T., Niclas, A. & Michael, F. (2019). "Evaluation of Multiple Object Tracking in Surveillance Video".

[4] Sunil, S. et., al. (2021)."YOLOv3 and faster R-CNN models for face mask detection in the COVID-19 environment." Springer Nature doi: 10.1007/s11042-021-10711-8

[5] Xiaohua, Q., Min, L., Liqiong, Z. & Rui Zhao (2020). "Deep Convolutional Feature Fusion Model for Multispectral Maritime Imagery Ship Recognition"; Journal of Computer and Communications, Vol.8 No.11.

[6] Ji, L.K. (2019) "Research on Text Sentiment Analysis Technology Based on Deep Learning. Beijing University of Posts and Telecommunications, Beijing".

[7] Sun L, Zhao C, Yan Z, Liu P, Duckett T, Stolkin R. (2019). "A new method for RGB-D-based nuclear waste object detection that is weakly supervised". IEEE Sensors" J.; 19:3487–3500. doi: 10.1109/JSEN.2018.2888815. [Google Scholar]

[8] Valério, N., Hugo, O., José, S., Thales, V. & Krerley, O. (2019). "RetailNet: A Deep Learning Approach for People Counting and Hot Spots Detection in Retail Stores". In: July. doi: 10.1109/SIBGRAPI.2019.00029 (cit. on p. 21).

[9] Lucas, M., Adriano, B., Krerley, O., & Thales, V. (2020). "LRCN-RetailNet: A recurrent neural network architecture for accurate people counting". (cit. on pp. 21, 23).

[10] Valério, N., Hugo, O., José, S., Thales, V. & Krerley, O. (2019). "RetailNet: A Deep Learning Approach for People Counting and Hot Spots Detection in Retail Stores". In: July. doi: 10.1109/SIBGRAPI.2019.00029 (cit. on p. 21).

[11] Guangshuai, G., Junyu, G., Qingjie, L., Qi, W. & Yunhong Wang (2020). "CNN-based Density Estimation and Crowd Counting: A Survey". arXiv:2003.12783v1 [cs.CV]

[12] Xu, B. & Qiu, G. (2016). "Crowd density estimation based on rich features and random projection forest". pp. 1–8. doi: 10.1109/WACV.2016.7477682 (cit. on p. 23).

[13] Wu, Z., & McGoogan, M. (2020) "Summary of a report of 72314 cases from the Chinese Center for Disease Control and Prevention: characteristics of and key takeaways from the coronavirus disease 2019 (COVID-19) outbreak in China." JAMA 323, 1239–1242. doi: 10.1001/jama.2020.2648.

[14] Jiang, M. & Fan, X. (2020)." RetinaMask: A Face Mask detector". arXiv:2005.03950.

[15] Loey, M. Manogaran, G. Taha, M. & Khalifa, N. (2020). "For face mask detection in the COVID-19 pandemic timeframe, a hybrid deep transfer learning model incorporates machine learning techniques. Assessment" 2020, 167, 108288.

[16] Cabani, A.; Hammoudi, K.; Benhabiles, H.; Melkemi, M. (2020). "MaskedFace-Net: A Collection of Accurate and Inaccurately Masked Face Pictures under the Framework of COVID-19". arXiv 2020, arXiv:2008.08016.

[17] Hussain, M. (2020). “YOLO object detection usingOpenCV”https://www.mygreatlearning.com/blog/yolo-object-detection- using- opencv/

[18] Mathworks (2021) https://neuravisio.com/object_detection?gclid=EAIaIQobChMIw6S9-4

[19] Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. (2016). “Unified, Real-Time Object Detection: You Only Need to Look Once. In IEEE Conference on Computer Vision and Pattern Recognition Proceedings” (CVPR), Las Vegas, NV, USA, 27–30, pp. 779–788.

[20] He, K., Zhang, X., Ren, S. & Sun, J. (2016). “Deep residual learning for picture identification, in IEEE Computer Vision and Pattern Recognition Proceedings,” Hong Kong, China, pp. 770–778.

[21] Jing, L., Jinan, G., Zedong, H., & Jia, W. (2019). “Application Research of Improved YOLO V3 Algorithm in PCB Electronic Component Detection”. Appl. Sci., 9, 3750; doi:10.3390/app9183750.

[22] Redmon J, Farhadi A. (2018). “YOLOv3: an incremental improvement”. arXiv:1804.02767 [cs], Apr. 2018, [Online]. Available: http://arxiv.org/abs/1804.02767.

[23] Adarsh, D. (2021). “Applying Deep learning techniques - Masked facial recognition in Smartphone security systems using transfer learning”.

[24] Adeyanju, A., Omidiora, O. & Oyedokun, F. (2015). “Performance evaluation of different support vector machine kernels for face emotion recognition. SAI Intelligent Systems Conference (IntelliSys)”, London, 2015, pp. 804-806, doi: 10.1109/IntelliSys.2015.7361233. [accessed: 22-122020].

[25] Mingjie, J., Xinqi, F. & Hong, Y. (2020). “RetinaFaceMask: A face mask detector”.

[26] Dweepna, G. Parth, G, Sharnil, P., Amit, G. & Ketan, K. (2020). “A Deep Learning Approach for Face Detection using YOLO”. Computer Engineering Department Devang Patel Institute of Advance Technology and Research, CHARUSAT Changa, Anand, India.

[27] Zhenrong, D., Rui, Y., Rushi, L., Henbing, L., & Xiaonan, L. (2020). SE-IYOLOV3: “An Accurate Small Scale Face Detector for Outdoor Security”. DOI:10.3390/math8010093.

[28] Soni, A., & Singh, A. (2020). “Automatic Motorcyclist Helmet Rule Violation Detection using Tensorflow & Keras in OpenCV”. In 2020 IEEE International Students Conference on Electrical, Electronics and Computer Science (SCEECS).

[29] Lin, Z., Chun, D., Jiang, Y., Wang, J., & Chao, Z. (2020). “Yolov3: Face Detection in Complex” Environments.https://doi.org/10.2991/ijcis.d.200805.002. Volume 13, Issue 1, 2020, Pages 1153 – 1160. Published by Atlantis Press B.V.

[30] Suresh, K. & Praveen, O. (2020)."Extracting of Patterns Using Mining Methods Over Damped Window”. Second International Conference on Invent Live Research in Computing Applications (ICIRCA), Coimbatore, India, 2020, pp. 235-241, doi: 10.1109/ICIRCA48905.2020.9182893.

[31] Akshay S., Bhat, M. & Rao, A. (2019). “Facial Expression Recognition using Compressed Images.” International Journal of Recent Technology and Engineering. DOI:10.35940/ijrte. B1041.078219.

[32] Rayson, L., (2018). “YOLO Detector-Based Robust Real-Time Automatic License Plate Recognition, 2018 International Joint Conference on Neural Networks” (IJCNN).

[33] Shubham S., (2018). “Human Action Recognition and Localization based on YOLO. International Conference on Intelligent Manufacturing and Robotics”, pp. 831-838, 2018.

[34] Gokul, K., & Sujala, D. (2021). “Development of Convolutional Neural Network-Based Applications for Mask and Social Distancing Violation Detection”. DOI: 10.5220/0010483107600767 In Proceedings of the 23rd International Conference on Enterprise Information Systems (ICEIS 2021) - Volume 1, pages 760-767 ISBN: 978-989-758-509-8.

[35] Ren, L. & Ziang, R. (2020). “Utilizing Yolo for Mask Recognition Task”.

[36] Singh, S., Ahuja, U., Kumar, M. *et al.* (2021). “Identification of face masks in the COVID-19 environment using YOLOv3 and faster R-CNN models” . *Multimed Tools Appl* **80**, 19753–19768 https://doi.org/10.1007/s11042-021-10711-8.

[37] Zhengxia, Z.; Shi, Z.; Yuhong, G.; Jieping, Y. (2019). “A Survey of Object Detection in 20 Years”. arXiv 2019, arXiv:1905.05055. [Google Scholar]

[38] Jiang, M.; Fan, X. (2020). **“**"RetinaMask: A Face Mask Finder." *arXiv* , arXiv:2005.03950. [**Google Scholar**]

[39] “Examine Faces to See If Anyone Is Mask Wearing”. Available online: **https://github.com/AIZOOTech/FaceMaskDetection** (accessed on 21 March 2020).

[40] Salama A., Sharran R., Dilovan Z., Nechirvan A., Mazin M., Jan N., Radek M., Muhammet D., Weiping D., (2024). “A deep learning algorithm based on YOLO enabling real-time identification of face masks using drone monitoring in public areas”. Information Sciences, Volume 676, 2024, 120865, ISSN 0020-0255, https://doi.org/10.1016/j.ins.2024.120865.