2022 IEEE International Conference on Data Science and Information System (ICDSIS)

Real-Time Face Mask Detection Using Streamlit,TensorFlow, Keras and Open-CV

1st R Naga Sri Vigneswari Yalamarthi
Department of Computer Science and Technology
Lovely Professional University
Punjab, India
vignayalamarthi23@gmail.com

2nd Shareef Shaik
Department of Computer Science and Technology
Lovely Professional University
Punjab, India
shaikshareef7537@gmail.com

3rd Dalwinder singh
Department of Computer Science and Technology
Lovely Professional University
Punjab, India
dalwinder.singh@lpu.co.in

4th Manik Rakhra
Department of Computer Science and Technology
Lovely Professional University
Punjab, India
Rakhramanik786@gmail.com

2022 IEEE International Conference on Data Science and Information System (ICDSIS) | 978-1-6654-9801-2/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICDSIS55133.2022.9915817

***Abstract*—This paper is based on the protection of our health from coronavirus officially known as COVID-19. Real-time detection of a face mask can help to prevent of the coronavirus, detecting the mask with the help of machine learning and data science algorithms such as Streamlit, MoblieNetV2, OpenCV, etc., are widely used in this ideal methodology. This paper is about the method that provides an accuracy of 99.78% in detecting the mask with live video stream. The method proposes building accurate model and integrating the model with a graphical interface which can improve the experience of the user.**

Keywords—Covid-19, Coronavirus, Real-time, Facemask Detection, Streamlit, Mobile network.

I. INTRODUCTION

The coronavirus, which is around for the past few years, has been the world's biggest health crisis. This is a hazardous virus[1] that infects individuals in a variety of ways, including from person to person and via the air. According to WHOs specialists, COVID-19 infection can be prevented by the essentials of wearing a mask, maintaining a six-foot space between people, and being sanitized. This coronavirus has not been evacuated after being vaccinated also, thanks to the invention of a vaccine by various scientific laboratories. COVID-19, on the other hand, continues to disseminate its varieties. It comes as no surprise that everyone should wear a mask to shield themselves does not attack by this infection. Wearing a facemask became a part of day-to-day life.

In this paper, we are focused on real-time face mask detection with a more effective and systematic approach. Machine learning packages like Stream lit, NumPy, TensorFlow, Keras, and OpenCV are used for live detection of whether themask on a face is there or not.

The Remaining paper goes in this consecutive order as shown: Section II refers to previous work related to recognize the mask. Section III provides data set images. Section IV gives the information about using in building the proposed model. Section V is all about the method and procedure to it. Section VI is the Result and Analysis. Section VII is about the conclusion.

II. RELATED WORK

A-few methods are proposed by other authors for detecting the face mask. Based on [2] of this research, the proposed technique to recognize the face from the photo effectively and then to results whether that person in that image have a mask on face or not, by using machine learning algorithms. It takes time to update the dataset with new images and then to detect the facemask. As the Authors in [3], proposed real-time face mask detection they came across their algorithm it used to apply this work includes a detector of two-level and it must be installed device in a particular location. Other work in [4],[5],[6] research found that the authors proposed the method with real-time image classification using the MobileNetV2 for detecting the facemask, but the outputs of these methods are in CLI mode and need more resources to execute.

Facemask detection using images and real-time detection without GUI has already been proposed methods. Taking the methods as revelation, moving forward to the problem which challenged in detecting and resolving it in our way of approach that real-time detecting the face mask with GUI and Making it easy to accessible and achievable with this proposed method of application.

III. DATASET

In a practical way of understanding to the system, given the data set with images. To identify, how it will look when a person covering the face with mask or without mask. People having mask on their face and people not having the mask on their face both together are collected more than three thousand random images [7] for this method and categorized the data as Fig. 1 shows Images set in "without a mask" and Fig. 2 shows Images set in "with a mask".

Fig. 1. Images without facemask

Fig. 2. Images with facemask

IV. Incorporated Packages

A. Streamlit

Streamlit is free software application framework for machine learning and data science means in python which allows the user to create a web application with a simple and pure python script [8]. Using Streamlit the user can create an interactive web application without spending much time on the development of the app, it is easily helpful to build the dashboard.

B. TensorFlow

TensorFlow is nothing but the flow of tensors. It consists of multiple neurons where each one of them has its own specific function goal etc[9]. They are all interconnected in a flow-like manner to get desired output or prediction. It was actually developed by Google for their own in-house purposes and now it was built to be accessible for everyone. TensorFlow supports GPU is this very essential in deep learning as it helps to perform computations and train data sets to build a model [10].

C. Keras

Keras is a neural network high-level API developed in Python. It facilitates the implementation of neural networks by supporting various backend neural network calculations [12].

D. Open-CV

OpenCV is a library of programming capabilities especially geared toward real-time computer vision. It can be used for image processing which includes machine learning algorithms. Face in the image or video, handwriting and objects can be detected using this application [13].

V. Proposed Method

There are two models in the suggested method: the base model and the head model. The head model receives the output of the base model. The following is the proposed training algorithm for the mask detection model:

ALGORITHM I: Training the Face mask detection algorithm

INPUT: Dataset of images with masks and without masks.

OUTPUT: A model that can predict someone who is not wearing a mask.

The Steps as follows:-

1. *Data Preprocessing:*
   - 1.1. Load images with the size of 224X224 using the load_img () function from the Keras package.
   - 1.2. Convert the images into an array.
   - 1.3. Pass the array to Preprocess input () method to convert them into the format the model requires.
   - 1.4. Append the images and their corresponding labels to empty lists and convert them into NumPy arrays.
2. *Training the Model:*
   - 2.1. Split the images into train and test data.
   - 2.2. Create a base model with Mobile Net with weights as ImageNet.
   - 2.3. Pass the base model output to the head model.
   - 2.4. create pooling using AveragePooling2D with the size of 7X7.
   - 2.5. Insert the Flatten layer.
   - 2.6. Add a dense layer of 128 neurons with an activation layer as “ReLu”.
   - 2.7. Use the Dropout function to avoid overfitting the model.
   - 2.8. Add the last dense layer with two neurons as output for our two options i.e., mask and without the mask and activation layer as “SoftMax.”
3. Train the model using the Adam optimizer.
4. Save the finished model for use in creating a real- time facemask detector.

A. Data Preprocessing

Data pre-processing is one of the crucial steps when building any type of machine learning model. Before building the model, we need to make sure, that the data contains noises and is it in the same formats required by the model. So, Data pre-processing is a necessary step before building the machine learning model. With right amount of data and Data pre-processing we can get more accurate and efficient models.

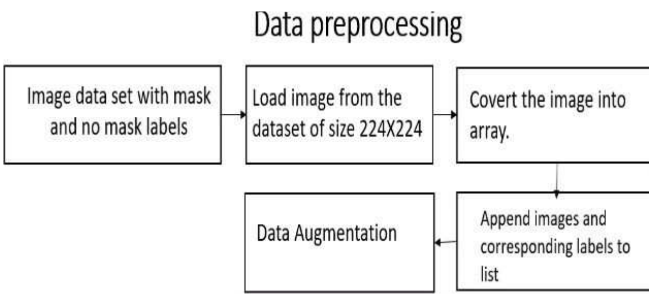

Fig. 3. Overview of Data Preprocessing

In the phase of data preprocessing Fig. 3, the images are loaded of size 224 x 224 and converted into arrays. The arrays are then added to an empty list and converted into NumPy arrays. At the end of the phase to improve the performance and efficiency of the model data augmentation is performed. Data Augmentation is a process where different images are produced from an image by changing the parameter like rotation angle, height, width, etc.

B. Training the model

The Model uses the architecture of the Mobile networks to determine whether or not the person is wearing a mask. The method has two models: the base model and the head model. The input data set is processed and converted into arrays, which are then passed to Mobile Nets [15] as a base model. The base model is used as input to the head model. In the head model, we use max-pooling and add dense layers to achieve better performance. The base model [11] is using the shape of the input image as 224 x 224 x 3, where '3' represents that the image is colored (RGB).

In the head model using the AveragePooling2D()[18] function to calculate the maximum element from the region of the feature map covered by the filter. Flattening the feature map using the Flatten() function reduces the amount of detail in the map. After leveling the map of landmarks, a dense layer containing 128 neurons and an activation layer "ReLu" is added [16]. Reducing overfitting can be done by using the Dropout() function with a '0.5' rate. Finally, a layer of neurons is added with the SoftMax activation function. During the training process, the model is the base model of the layers should be frozen in order to avoid the layer update.

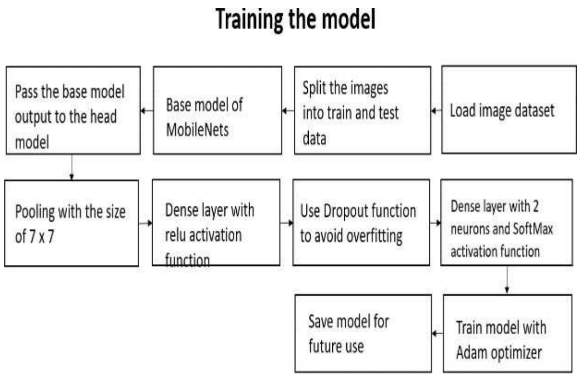

Fig. 4. Overview of Training the model

In Fig. 4 shows the flow of how the training the model works in this proposed method, respectively.

To compile the Mobile Net model [17] two parameters are required which are optimizer and loss. In this case to optimize the model "Adam" optimizer is used. Adam optimizer is oneof the best optimizers available and can adjust to different scenarios. The loss function used in this case is binary_crossentropy which helps in calculating the loss between the true and anticipated labels. In the end, the modelis trained for twenty epochs using the fit () function as shownFig. 5. The model is saved on the disk and used for future purposes.

```
Epoch 1/20
95/95 [==============================] - 66s 672ms/step - loss: 0.3710 -
accuracy: 0.8691 - val_loss: 0.1343 - val_accuracy: 0.9870
Epoch 2/20
95/95 [==============================] - 64s 667ms/step - loss: 0.1428 -
accuracy: 0.9651 - val_loss: 0.0702 - val_accuracy: 0.9909
Epoch 3/20
95/95 [==============================] - 69s 728ms/step - loss: 0.0962 -
accuracy: 0.9746 - val_loss: 0.0519 - val_accuracy: 0.9909
Epoch 4/20
95/95 [==============================] - 67s 705ms/step - loss: 0.0763 -
accuracy: 0.9792 - val_loss: 0.0440 - val_accuracy: 0.9909
Epoch 5/20
95/95 [==============================] - 63s 663ms/step - loss: 0.0677 -
accuracy: 0.9832 - val_loss: 0.0402 - val_accuracy: 0.9922
Epoch 6/20
95/95 [==============================] - 64s 676ms/step - loss: 0.0511 -
accuracy: 0.9868 - val_loss: 0.0369 - val_accuracy: 0.9909
Epoch 7/20
95/95 [==============================] - 65s 688ms/step - loss: 0.0518 -
accuracy: 0.9832 - val_loss: 0.0342 - val_accuracy: 0.9922
Epoch 8/20
95/95 [==============================] - 68s 717ms/step - loss: 0.0538 -
accuracy: 0.9845 - val_loss: 0.0344 - val_accuracy: 0.9909
Epoch 9/20
95/95 [==============================] - 63s 664ms/step - loss: 0.0408 -
accuracy: 0.9901 - val_loss: 0.0319 - val_accuracy: 0.9935
Epoch 10/20
95/95 [==============================] - 66s 695ms/step - loss: 0.0429 -
accuracy: 0.9871 - val_loss: 0.0312 - val_accuracy: 0.9922
Epoch 11/20
95/95 [==============================] - 62s 650ms/step - loss: 0.0414 -
accuracy: 0.9885 - val_loss: 0.0301 - val_accuracy: 0.9922
Epoch 12/20
95/95 [==============================] - 61s 636ms/step - loss: 0.0403 -
accuracy: 0.9895 - val_loss: 0.0294 - val_accuracy: 0.9935
Epoch 13/20
95/95 [==============================] - 60s 630ms/step - loss: 0.0336 -
accuracy: 0.9888 - val_loss: 0.0299 - val_accuracy: 0.9935
Epoch 14/20
95/95 [==============================] - 60s 629ms/step - loss: 0.0319 -
accuracy: 0.9908 - val_loss: 0.0296 - val_accuracy: 0.9922
Epoch 15/20
95/95 [==============================] - 60s 630ms/step - loss: 0.0361 -
accuracy: 0.9875 - val_loss: 0.0299 - val_accuracy: 0.9909
Epoch 16/20
95/95 [==============================] - 60s 633ms/step - loss: 0.0275 -
accuracy: 0.9918 - val_loss: 0.0275 - val_accuracy: 0.9922
Epoch 17/20
95/95 [==============================] - 61s 637ms/step - loss: 0.0325 -
accuracy: 0.9878 - val_loss: 0.0279 - val_accuracy: 0.9909
Epoch 18/20
95/95 [==============================] - 60s 629ms/step - loss: 0.0276 -
accuracy: 0.9931 - val_loss: 0.0267 - val_accuracy: 0.9935
Epoch 19/20
95/95 [==============================] - 60s 632ms/step - loss: 0.0314 -
accuracy: 0.9908 - val_loss: 0.0269 - val_accuracy: 0.9922
Epoch 20/20
95/95 [==============================] - 60s 628ms/step - loss: 0.0291 -
accuracy: 0.9904 - val_loss: 0.0265 - val_accuracy: 0.9935
```

Fig. 5. Training the Model with twenty epochs

ALGORITHM II: Deploying the mask detector model with the Streamlit

INPUT: Face and mask detector models.

OUTPUT: Real-time face mask detection web application

The Steps as follows:-

1. Load the mask model of the detector from the hard disk.
2. Again, load the face model of the detector from the hard disk.
3. Create a function that takes a frame as input and returns the predictions and the locations of the face in the frame. def detect_mask_face ()
4. Initialize streamlit components like menu, and buttons.
5. Use VideoCapture () function to capture the video stream.

6. Pass every frame from the video stream to the function detect_mask_face ().

7. Get a prediction about the mask and location from the function detect_mask_face ().

8. Use PutText () function from the OpenCV package [14] to show the prediction onto the video stream.

C. *Integrating the model with Streamlit*

Developing a model is a crucial step but integrating with the user interface and making it available to the user to use in a friendly manner is also an important phase. Streamlit is an open-source library in python. Applications related to Data Science and machine learning can use the Streamlit features and provide the User interface. The applications which build on streamlit [19] can be easily deployed on the web in small number of steps.

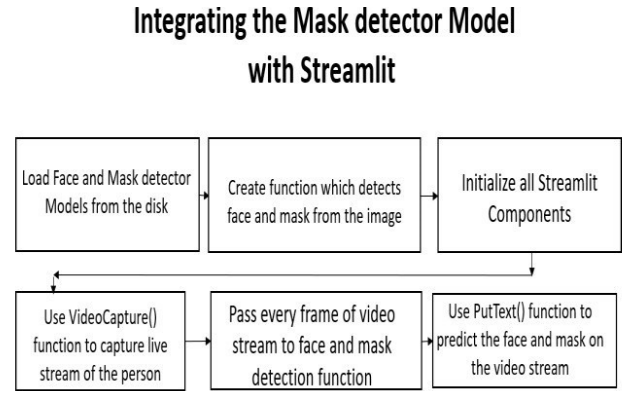

Fig. 6. Flowchart of Integrating the model with Streamlit

This Fig. 6 phase has helped to create the user- interference for detecting the face mask in real-time. The web applications take the live stream of the video, and each frame of the video is passed to a function that gives the locations of the face and prediction about whether they are wearing masking or not. The predictions and locations are displayed in the web application in real-time.

VI. Result And Analysis

The proposed model has more accuracy compared to other different models. As we are using Mobile Networks which are lighter in weight and equally accurate to other methods like convolutional neural networks (CNN). Mobile Networks can be considered the best alternative to the CNN models. In the proposed method, MobileNet-v2 is used which is a CNN but it is small, has low latency compared to CNN and it is the best fit if there are resources constraints. The below table-1 represents the accuracy of the model.

TABLE I. Model Evaluation

	Precision	Recall	F1- score	support
With mask	0.99	0.99	0.99	383
Without mask	0.99	0.99	0.99	384
Accuracy			0.99	767
Macro Average	0.99	0.99	0.99	767
Weighted Average	0.99	0.99	0.99	767

The above Table 1Model with twenty epochs of training. The below Fig. 7 and Fig. 8 represents the accuracy and loss for every epoch.

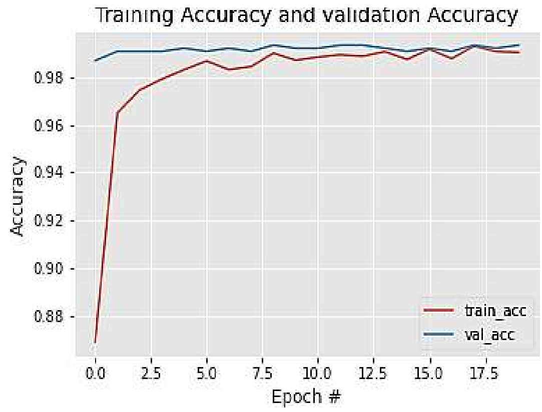

Fig. 7. Training Accuracy

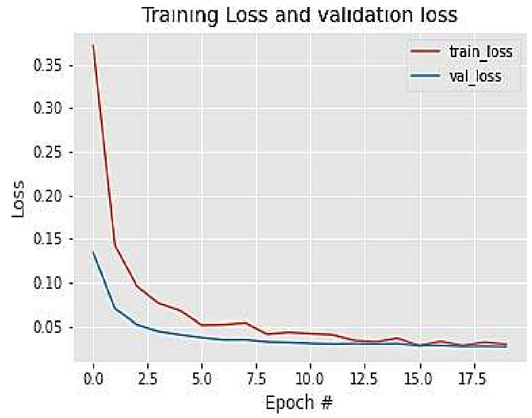

Fig. 8. Training Loss

In Fig. 9 and Fig. 10 shows the snap shots of the livedetection of a person with mask and without mask.

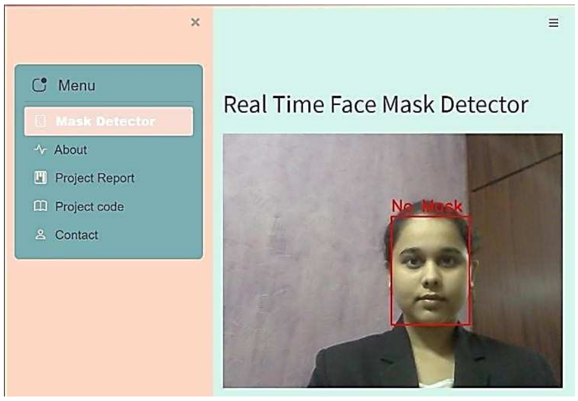

Fig. 9. Detection of face when Mask off

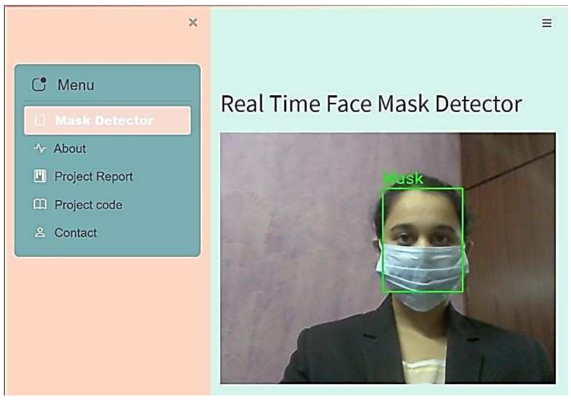

Fig. 10. Detection of face when Mask On

The other models build on CNN are heavier and require more processing power to use. The approach building model with MobileNet which is accurate and integrates with streamlit architecture will be a useful and ideal approach.

VII. CONCLUSION

In this paper, we developed a method in our research that automatically detects if a person covered their face with mask or without it. Else, alerts in red color if their mask not covering the face. This suggested solution employs Mobile networks and Streamlit to assist the people and helps as precaution to not affect by the spread of the COVID-19 virus. Building a model with any GUI architecture becomes useful. Steramlit provides a GUI interface for integrating the model and using it in a user-friendly way. The user does not need to have any prior knowledge about the commands and can easily run the applications using GUI provided by the streamlit.

The web application for detecting whether the person is wearing a mask or not. The applications built from the above-proposed method can work with low-end devices as themodel is lightweight and have more performance.

REFERENCES

[1] "Centers for Disease Control and Prevention 2019 novel coronavirus disease (COVID-19) information management: addressing national healthcare and public health needs for standardized data definitions and codified vocabulary for data exchange - PMC," PubMed Central (PMC), Sep. 01, 2020. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7543614/.

[2] A. Das, M. W. Ansari, and R. Basak, "[PDF] Covid-19 Face Mask Detection Using TensorFlow, Keras and OpenCV | Semantic Scholar," [PDF] Covid-19 Face Mask Detection Using TensorFlow, Keras and OpenCV | Semantic Scholar, Jan. 01, 2021. https://www.semanticscholar.org/paper/Covid-19-Face- Mask-Detection-Using-TensorFlow%2C-and-Das-Ansari/3689bcce061645e966fb39ae69b607013e7d435d.

[3] [3]F. A. Putra Susanto, R. Analia, and I. I. Suciningtyas, "The Face Mask Detection For Preventing the Spread of COVID-19 at Politeknik Negeri Batam | 2020 3rd InternationalConference on Applied Engineering;2020. | Web ofScience," The Face Mask Detection For Preventing theSpread of COVID-19 at Politeknik Negeri Batam | 2020 3rdInternational Conference on Applied Engineering;2020. |Webof Science. https://pesquisa.bvsalud.org/global-literature-on-novel-coronavirus-2019- ncov/resource/en/covidwho-1373735.

[4] [4]"SSDMNV2: A real time DNN-based face mask detection system using single shot multibox detector and MobileNetV2 - ScienceDirect," SSDMNV2: A real time DNN-based face mask detection system using single shot multibox detector and MobileNetV2 - ScienceDirect, Dec. 31, 2020. https://www.sciencedirect.com/science/article/pii/S2210670 720309070.

[5] [5]S. A. Sanjaya and S. A. Rakhmawan, "Face Mask Detection Using MobileNetV2 in The Era of COVID-19 Pandemic | Semantic Scholar," Face Mask Detection Using MobileNetV2 in The Era of COVID-19 Pandemic | SemanticScholar, Jan. 01, 2021. https://www.semanticscholar.org/paper/Face-Mask- Detection-Using-MobileNetV2-in-The-Era-of-Sanjaya-Rakhmawan/bd1cfb454e8cdd1820d0e8d2da0bca8a640a8ae 6#references.

[6] "Sci-Hub | Face Mask Detection by using Optimistic Convolutional Neural Network. 2021 6th International Conference on Inventive Computation Technologies (ICICT) | 10.1109/icict50816.2021.9358653," Sci-Hub |Face Mask Detection by using Optimistic Convolutional Neural Network. 2021 6th International Conference on Inventive Computation Technologies (ICICT) | 10.1109/icict50816.2021.9358653. https://www.www.sci-hub.se/10.1109/icict50816.2021.9358653.

[7] skshareef, "datset_face_mask/dataset at main · skshareef/datset_face_mask," GitHub, Apr. 05, 2022. https://github.com/skshareef/datset_face_mask/tree/main/da taset.

[8] "J. N. Cheltha C, M. Rakhra, R. Kumar, and H. Walia, "A Review on Data hiding using Steganography and Cryptography," 2021 9th Int. Conf. Reliab. Infocom Technol. Optim. (Trends Futur. Dir. ICRITO 2021, pp. 5–8, 2021, doi: 10.1109/ICRITO51393.2021.9596531.

[9] L. Gundaboina, S. Badotra, S. Tanwar and Manik, "Reducing Resource and Energy Consumption in Cryptocurrency Mining by using both Proof-of-Stake Algorithm and Renewable Energy," 2022 International Mobile and Embedded Technology Conference (MECON), 2022, pp. 605-610, doi: 10.1109/MECON53876.2022.9752365.

[10] Monika Hooda, Chhavi Rana, Omdev Dahiya, Jayashree Premkumar Shet, Bhupesh Kumar Singh, "Integrating LA and EDM for Improving Students Success in Higher Education Using FCN Algorithm", Mathematical Problems in Engineering, vol. 2022, Article ID 7690103, 12 pages, 2022. https://doi.org/10.1155/2022/7690103

[11] Sushovan Chaudhury, Manik Rakhra, Naz Memon, Kartik Sau, Melkamu Teshome Ayana, "Breast Cancer Calcifications: Identification Using a Novel Segmentation Approach", Computational and Mathematical Methods in Medicine, vol. 2021, Article ID 9905808, 13 pages, 2021. https://doi.org/10.1155/2021/9905808.

[12] Murugesan G., Tousief Irshad Ahmed, Jyoti Bhola, Mohammad Shabaz, Jimmy Singla, Manik Rakhra, Sujeet More, Issah Abubakari Samori, "Fuzzy Logic-Based Systems for the Diagnosis of Chronic Kidney Disease", BioMed Research International, vol. 2022, Article ID 2653665, 15 pages, 2022.https://doi.org/10.1155/2022/2653665

[13] Manik Rakhra, Amitabh Bhargava, Deepshikha Bhargava, Ramandeep Singh, Astha Bhanot, Abdul Wahab Rahmani, "Implementing Machine Learning for Supply-Demand Shiftsand Price Impacts in Farmer Market for Tool and Equipment Sharing", Journal of Food Quality, vol. 2022, Article ID 4496449, 19 pages, 2022. https://doi.org/10.1155/2022/4496449

[14] Manik Rakhra, Sumaya Sanober, Noorulhasan Naveed Quadri, Neha Verma, Samrat Ray, Evans Asenso, "Implementing Machine Learning for Smart Farming to Forecast Farmers' Interest in Hiring Equipment", Journal of Food Quality, vol. 2022, Article ID 4721547, 17 pages, 2022. https://doi.org/10.1155/2022/4721547.

[15] Manik Rakhra, Ramandeep Singh, Tarun Kumar Lohani, Mohammad Shabaz, "Metaheuristic and Machine Learning- Based Smart Engine for Renting and Sharing of Agriculture Equipment", Mathematical Problems in Engineering, vol. 2021, Article ID 5561065, 13 pages, 2021. https://doi.org/10.1155/2021/5561065

[16] M. Rakhra, N. Verma, and S. Bhatia, "Structural, Morphological, Optical, Electrical and Agricultural Properties of Solvent/ZnO Nanoparticles in the Photodegradation of DR-23 Dye," J. Electron. Mater., vol. 49, no. 1, pp. 643–649, 2020, doi: 10.1007/s11664-019-07760-z.

[17] Rakhra, M., Verma, N. Tri-doped co-annealed zinc oxide semi-conductor synthesis and characterization: photodegradation of dyes and gas sensing applications. J Mater Sci: Mater Electron 32, 17588– 17601 (2021). https://doi.org/10.1007/s10854-021-06292-9

[18] P. Thapar, M. Rakhra and A. Singh, "Comparing Image Feature Extraction Methods Using Dermoscopy Noisy Images," 2022 International Mobile and Embedded Technology Conference (MECON), 2022, pp. 559-562, doi: 10.1109/MECON53876.2022.9751935.

[19] Shruti, R. Soumya, M. Rakhra, N. S. Chintagunti, B. Singh and D. Singh, "Modern Data Mining Approach to Handle Multivariate Data and to Implement Best Saving Services for Potential Investor," 2022 International Mobile and Embedded Technology Conference (MECON), 2022, pp. 611-616, doi: 10.1109/MECON53876.2022.97521.