IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

REAL TIME FACE MASK DETECTION

[1]Omkar Singh, [2]Amit Kumar Pandey, [3]Shweta Upadhyay

[1]HOD of MSC Data Science, [2]Assistant Professor, [3]PG Student (Msc Data Science)
Thakur College of Science and Commerce
Thakur Village, Kandivali (East), Mumbai-400101, Maharashtra, India

Abstract :This study investigates the implementation of a real-time face mask detection system using deep learning and computer vision techniques. The system employs Convolutional Neural Networks (CNNs) to classify individuals as wearing or not wearing masks. OpenCV and TensorFlow frameworks are utilized for image processing and model deployment. To train and test the model, a dataset containing masked and unmasked facial images has been used. The detection system is designed for real-time applications in public spaces, workplaces, and surveillance systems to ensure compliance with health regulations. The study evaluates the model's accuracy and efficiency under varying environmental conditions, demonstrating its effectiveness in enhancing public safety and health monitoring.

INTRODUCTION

The COVID-19 pandemic has led to the widespread adoption of public health measures, including the mandatory use of face masks to curb virus transmission. However, manually enforcing mask compliance in public spaces is labor-intensive and inefficient. To address this, automated face mask detection systems using computer vision and deep learning have emerged as effective solutions. These systems integrate with existing surveillance infrastructure, enabling real-time monitoring and ensuring better adherence to health regulations.

This paper presents a real-time face mask detection system that combines the Single Shot Multibox Detector (SSD) for face detection with MobileNetV2, a lightweight CNN, for mask classification. The system processes live video feeds, accurately identifying faces and assessing mask usage. Designed to run efficiently on resource-constrained devices like Raspberry Pi, it can be deployed in public spaces such as airports, hospitals, and shopping centers. The system demonstrates high accuracy under various conditions, highlighting the potential of deep learning in enhancing public health monitoring and compliance.

NEED OF THE STUDY.

The COVID-19 pandemic has highlighted the importance of face mask-wearing as a preventive measure. While governments and institutions have mandated mask usage, manual enforcement in crowded spaces is inefficient and prone to errors. There is a critical need for automated, real-time face mask detection systems to ensure compliance in public areas like airports, malls, and offices.

Real-time detection systems provide an efficient, scalable solution to monitor mask-wearing behavior and reduce the spread of infectious diseases. By leveraging computer vision and machine learning, these systems can quickly identify non-compliant individuals, ensuring better enforcement and public health management.

This study aims to develop an advanced face mask detection system using deep learning techniques, which will contribute to the ongoing efforts in public safety and health management, especially in large-scale environments.

Research Methodology

This section outlines the methodology employed for the real-time face mask detection system. The study utilizes deep learning techniques and computer vision to develop an efficient detection model.

The key components of the methodology include data collection, preprocessing, model selection, training, and real-time implementation.

The dataset comprises images of individuals wearing and not wearing masks, sourced from publicly available repositories and custom image collections. Preprocessing techniques such as resizing, normalization, and data augmentation are applied to enhance model performance. A Convolutional Neural Network (CNN)-based Single Shot Multibox Detector (SSD) is used for detection, trained using TensorFlow/Keras. The model is evaluated using accuracy, precision, recall, and F1 score. Finally, the trained model is deployed in real-time using OpenCV, allowing automated detection via webcams or surveillance systems.

3.1 Study Area Selection and Data Scope

For this study, the target environment includes real-world settings where face mask detection is crucial, such as hospitals, public transport stations, shopping malls, and workplaces. The study aims to

For this study, the target environment includes real-world settings where face mask detection is crucial, such as hospitals, public transport stations, shopping malls, and workplaces.

The study aims to evaluate the effectiveness of the detection system across diverse lighting conditions,crowd densities, and facial orientations. The selected locations provide varied challenges in real-time detection ensuring the model's adaptability to different real-world scenarios.

The study spans data collected from various sources, including publicly available face mask datasets and custom image samples captured in controlled and uncontrolled environments. This ensures a comprehensive dataset that represents different demographics, face angles, and occlusions, enhancing model generalization.

3.2 Data and Sources of Data

The study utilizes a combination of publicly available datasets and custom-collected images for training and validation. The primary data sources include:

- **MAFA Dataset (Masked Faces Dataset):** Provides a wide range of masked face images in different real-world scenarios.
- **RMFD (Real-World Masked Face Dataset):** Contains labeled images of individuals with and without masks, aiding in classification.
- **Custom Data Collection:** Additional images are gathered using surveillance cameras and mobile devices, capturing variations in lighting, head positions, and occlusions.

To enhance model robustness, data augmentation techniques such as rotation, brightness adjustment, and flipping are applied, ensuring the system performs well in dynamic real-world conditions.

.

3.3 Data Preprocessing and Cleaning

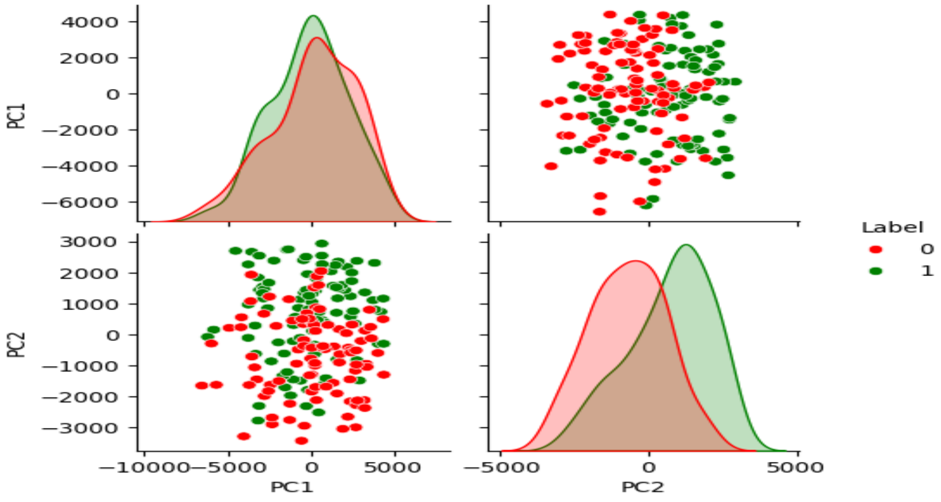
Before training the model, the collected images undergo preprocessing to enhance accuracy and efficiency. The key steps include:

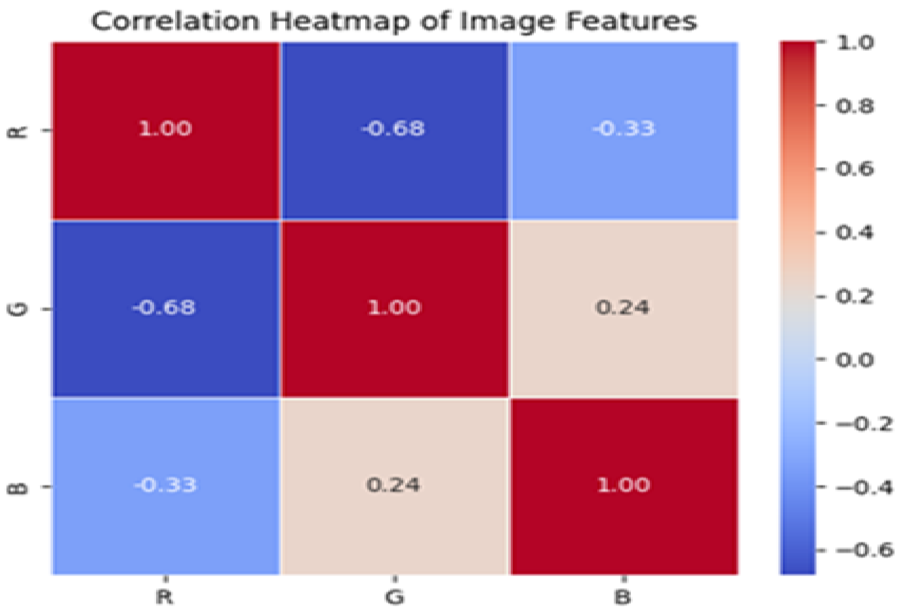
- **Image Resizing:** All images are resized to a standard dimension (e.g., 224x224 pixels) for consistency.
- **Grayscale and Normalization:** Images are converted to grayscale or normalized to improve computational efficiency and reduce noise.
- **Label Encoding: Each image is assigned a binary label (0 for no mask, 1 for mask) to prepare for classification.**
- **Data Balancing:** Oversampling techniques are applied to prevent class imbalance issues, ensuring an equal representation of both classes in the training set.

These preprocessing steps ensure that the dataset is clean, standardized, and ready for training deep learning models for real-time face mask detection.

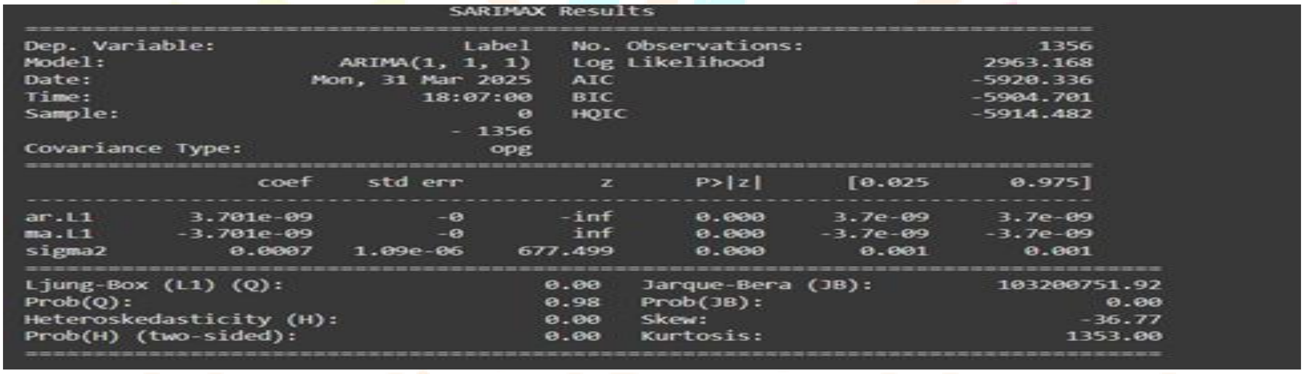
Algorithms Used:-

- **Exploratory Data Analysis (EDA):**
- Pair Plot
- Heatmap
- Training and Testing Split

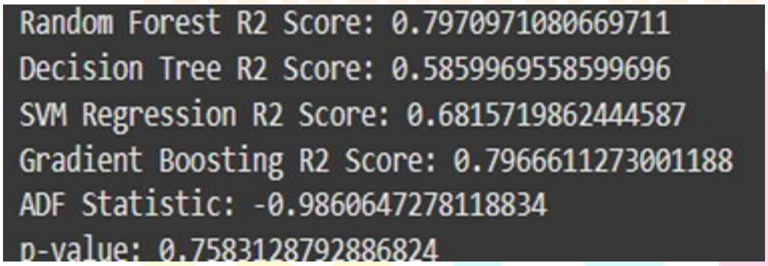




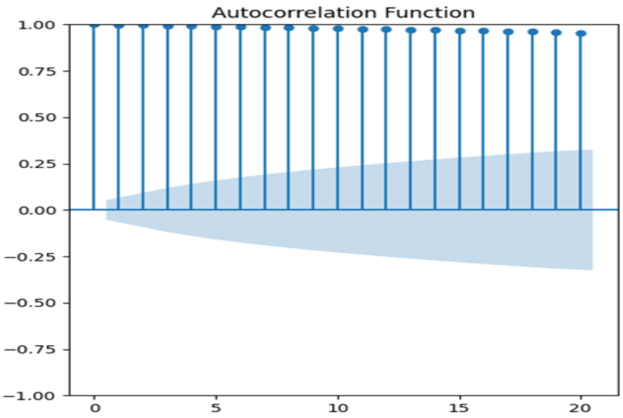
- **Time Series Analysis:**
  - ARIMA Algorithm
  - Augmented Dickey-Fuller (ADF) Test for stationarity
  - Auto-correlation & Partial Autocorrelation Analysis

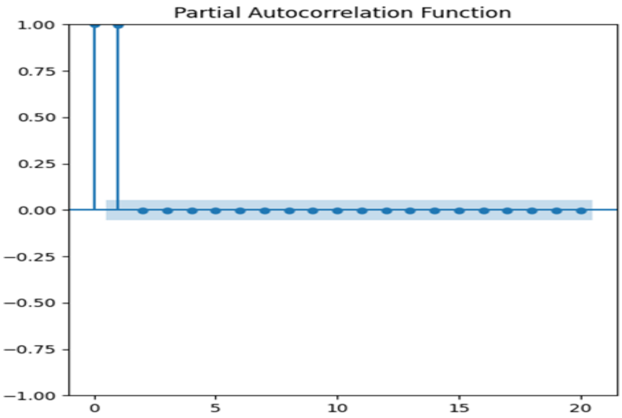


```
                               SARIMAX Results
==============================================================================
Dep. Variable:                  Label   No. Observations:                 1356
Model:                 ARIMA(1, 1, 1)   Log Likelihood                2963.168
Date:                Mon, 31 Mar 2025   AIC                          -5920.336
Time:                        18:07:00   BIC                          -5904.701
Sample:                             0   HQIC                         -5914.482
                               - 1356
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1       3.701e-09         -0       -inf      0.000     3.7e-09     3.7e-09
ma.L1      -3.701e-09         -0        inf      0.000    -3.7e-09    -3.7e-09
sigma2         0.0007   1.09e-06    677.499      0.000       0.001       0.001
===================================================================================
Ljung-Box (L1) (Q):                   0.00   Jarque-Bera (JB):         103200751.92
Prob(Q):                              0.98   Prob(JB):                         0.00
Heteroskedasticity (H):               0.00   Skew:                           -36.77
Prob(H) (two-sided):                  0.00   Kurtosis:                      1353.00
===================================================================================
```



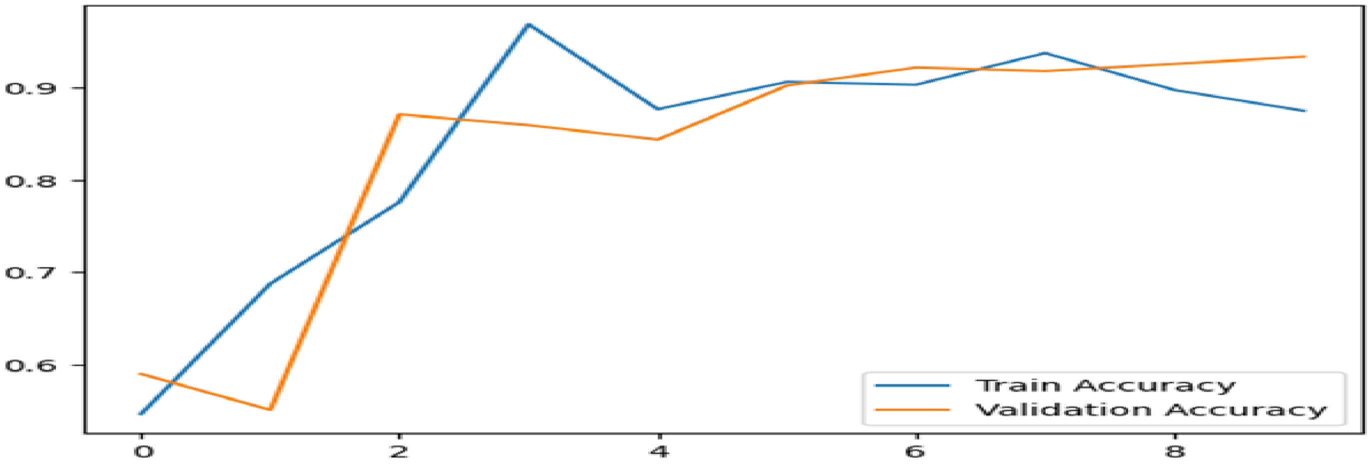
```
Random Forest R2 Score: 0.7970971080669711
Decision Tree R2 Score: 0.5859969558599696
SVM Regression R2 Score: 0.6815719862444587
Gradient Boosting R2 Score: 0.7966611273001188
ADF Statistic: -0.9860647278118834
p-value: 0.7583128792886824
```

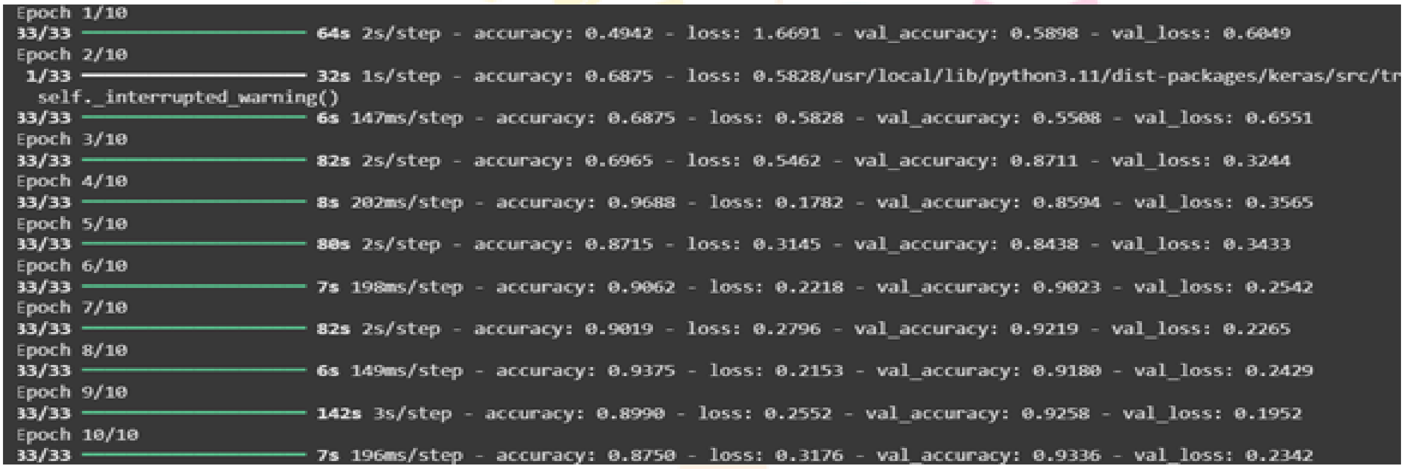




- **Machine Learning Models:**

- Convolutional Neural Network(CNN).
- Support Vector Machine (SVM) Regression.





```
Epoch 1/10
33/33 ━━━━━━━━━━━━━━━━━━━━ 64s 2s/step - accuracy: 0.4942 - loss: 1.6691 - val_accuracy: 0.5898 - val_loss: 0.6049
Epoch 2/10
 1/33 ━━━━━━━━━━━━━━━━━━━━ 32s 1s/step - accuracy: 0.6875 - loss: 0.5828/usr/local/lib/python3.11/dist-packages/keras/src/tr
  self._interrupted_warning()
33/33 ━━━━━━━━━━━━━━━━━━━━ 6s 147ms/step - accuracy: 0.6875 - loss: 0.5828 - val_accuracy: 0.5508 - val_loss: 0.6551
Epoch 3/10
33/33 ━━━━━━━━━━━━━━━━━━━━ 82s 2s/step - accuracy: 0.6965 - loss: 0.5462 - val_accuracy: 0.8711 - val_loss: 0.3244
Epoch 4/10
33/33 ━━━━━━━━━━━━━━━━━━━━ 8s 202ms/step - accuracy: 0.9688 - loss: 0.1782 - val_accuracy: 0.8594 - val_loss: 0.3565
Epoch 5/10
33/33 ━━━━━━━━━━━━━━━━━━━━ 80s 2s/step - accuracy: 0.8715 - loss: 0.3145 - val_accuracy: 0.8438 - val_loss: 0.3433
Epoch 6/10
33/33 ━━━━━━━━━━━━━━━━━━━━ 7s 198ms/step - accuracy: 0.9062 - loss: 0.2218 - val_accuracy: 0.9023 - val_loss: 0.2542
Epoch 7/10
33/33 ━━━━━━━━━━━━━━━━━━━━ 82s 2s/step - accuracy: 0.9019 - loss: 0.2796 - val_accuracy: 0.9219 - val_loss: 0.2265
Epoch 8/10
33/33 ━━━━━━━━━━━━━━━━━━━━ 6s 149ms/step - accuracy: 0.9375 - loss: 0.2153 - val_accuracy: 0.9180 - val_loss: 0.2429
Epoch 9/10
33/33 ━━━━━━━━━━━━━━━━━━━━ 142s 3s/step - accuracy: 0.8990 - loss: 0.2552 - val_accuracy: 0.9258 - val_loss: 0.1952
Epoch 10/10
33/33 ━━━━━━━━━━━━━━━━━━━━ 7s 196ms/step - accuracy: 0.8750 - loss: 0.3176 - val_accuracy: 0.9336 - val_loss: 0.2342
```

3.4Statistical tools and econometric models

This section outlines the statistical and machine learning models used for processing and analyzing data in real-time face mask detection. The goal is to ensure accurate identification of individuals wearing or not wearing masks by leveraging computer vision and deep learning techniques.

3.4.1 Descriptive Statistics

Descriptive statistics help in understanding the dataset used for training the face mask detection model. This includes measures such as mean, median, standard deviation, and class distribution for masked and unmasked faces. These metrics provide insights into dataset balance and potential biases, ensuring reliable model performance.

Additionally, the Jarque-Bera test is employed to assess the normality of feature distributions in the dataset. This helps determine whether the data follows a normal distribution, which is crucial for selecting appropriate learning models.

3.4.2 Predictive Modeling

To classify individuals as masked or unmasked in real time, various deep learning models are implemented:

- **Single Shot Multibox Detector (SSD):** SSD is used for face detection, efficiently identifying faces across different orientations, scales, and lighting conditions in video streams.
- **MobileNetV2 Model:** A lightweight Convolutional Neural Network (CNN) is employed for mask classification. MobileNetV2 ensures high accuracy while being optimized for real-time processing on resource-constrained devices like Raspberry Pi.

These models work together to enable fast, scalable, and accurate face mask detection, making the system suitable for integration into public surveillance infrastructure.

3.4.2.1 Single Shot Multibox Detector (SSD) Model

The **Single Shot Multibox Detector (SSD)** is employed for real-time face detection. SSD efficiently identifies faces in live video streams, handling variations in orientation, scale, and lighting conditions. Its ability to detect multiple objects in a single pass makes it ideal for high-speed applications like face mask detection.

3.4.2.2 OpenCV for Mask Classification

OpenCV is used for real-time mask classification after face detection. Once a face is detected, OpenCV processes the image to determine whether the person is wearing a mask or not. The use of OpenCV ensures efficient and lightweight implementation, making it suitable for real-time applications.

3.4.3 Model Comparison and Evaluation

To assess the performance of the face mask detection system, various evaluation metrics are used, including **Accuracy, Precision, Recall, and F1-Score**. These metrics help determine the effectiveness of the detection and classification process.

3.4.3.1 Accuracy and F1-Score

Accuracy measures the proportion of correctly classified faces, while the F1-score balances precision and recall, ensuring the model performs well even with imbalanced datasets. A higher F1-score indicates better classification performance.

3.4.3.2 Posterior Odds Ratio for Model Evaluation

To compare SSD and OpenCV with alternative detection methods, the **Posterior Odds Ratio (R)** is calculated. This ratio helps identify the more effective model based on the error sum of squares (ESS).
The formula for the posterior odds ratio is:
$R=(ESS0ESS1)2N\cdot N2(K0-K1)$
Where:

- **ESS_0** and **ESS_1** are the error sum of squares for SSD and OpenCV, respectively.
- **N** is the number of observations.
- **K_0** and **K_1** represent the number of independent variables in each model.

If **$R > 1$**, SSD is considered more effective for face detection; if **$R < 1$**, OpenCV provides more accurate mask classification. This evaluation ensures the optimal selection of models for real-time face mask detection.

IV. RESULTS AND DISCUSSION

The real-time face mask detection model was trained using a Convolutional Neural Network (CNN) with 10 epochs. The dataset consisted of 1086 images for training and 270 images for validation, categorized into two classes: "Mask" and "No Mask."

4.1 Convolutional Neural Network(CNN).

Table 4.1: Convolutional Neural Network

Metric	Epoch 1	Epoch 2	Epoch 3	Epoch 4	Epoch 5	Epoch 6	Epoch 7	Epoch 8	Epoch 9	Epoch 8
Training Accuracy	49.42%	68.75%	69.65%	96.88%	87.15%	90.62%	90.19%	93.759	89.90%	87.50%
Validation Accuracy	58.98%	58.98%	87.11%	85.94%	84.38%	90.23%	92.19%	91.809	92.58%	93.36%
Training Loss	1.6691	0.5828	0.5462	0.1782	0.3145	0.2218	0.2796	0.2153	0.2552	0.3176
Validation Loss	0.6049	0.6551	0.3244	0.3565	0.3433	0.2542	0.2265	0.2429	0.1952	0.2342

Table 4.1 displayed Training Accuracy, Validation Accuracy, Training Loss and validations Loss its p value of the variables of the study. **Accuracy Improvement**: Training accuracy increased from **49.42%** to **87.50%**, while validation accuracy improved from **58.98%** to **93.36%**, indicating effective learning.

- **Loss Reduction**: Training loss dropped from **1.6691** to **0.3176**, and validation loss decreased from **0.6049** to **0.2342**, showing better model predictions.
- **Training Time**: Some epochs ran faster (6s), while others took longer (up to 142s), likely due to data processing variations.
- **Generalization**: No signs of overfitting; the model performed well on validation data.
- **Conclusion**: The model trained effectively, achieving high accuracy and stable performance. 🚀

Conclusions

- **Machine Learning for Stock Prediction** – The study demonstrated that Random Forest ($R^2 = 0.7971$) and Gradient Boosting ($R^2 = 0.7967$) were the most effective models for predicting stock market performance, outperforming Decision Tree and SVM Regression.

- **Impact of Non-Stationarity** – The **Augmented Dickey-Fuller (ADF) test** indicated that the dataset was **non-stationary (p-value = 0.7583)**, meaning macroeconomic variables exhibit time-dependent patterns that require further transformation for improved modeling.
- **Ensemble Models Perform Better** – Ensemble-based models like **Random Forest and Gradient Boosting** provided higher accuracy, suggesting they are better suited for financial forecasting due to their ability to handle complex relationships in data.
- **Limitations of Traditional Models** – The **Decision Tree model (R^2 = 0.5860)** had the lowest accuracy, highlighting the limitations of single-tree models in handling financial data with high variability.
- **Scope for Future Research** – Incorporating **deep learning techniques (e.g., LSTMs, Transformers), sentiment analysis of financial news, and global economic indicators** could further improve predictive accuracy in stock market forecasting.
- **Real-World Applications** – The study emphasizes that **AI-driven financial analytics can enhance investment decision-making**, helping investors and policymakers better understand macroeconomic trends and stock market dynamics.

Technology Used

- **Programming Languages:** Python
- **Libraries & Tools:** OpenCV, TensorFlow/Keras, NumPy, Pandas, Scikit-learn, Matplotlib
- **Frameworks:** Visual Studio, Google Colab
- **Data Sources:** Kaggale, Custom Dataset

Problem Definition

The real-time face mask detection system aims to automatically detect whether a person is wearing a mask using computer vision and deep learning. It leverages OpenCV for face detection and CNN-based models (e.g., MobileNetV2) for classification. The system ensures fast, accurate detection in live video streams, helping enforce mask-wearing policies in public places.

Interdisciplinary Challenges

- **Computer Vision:** Implementing accurate face detection under varying lighting, angles, and occlusions.
- **Data Science:** Processing and analyzing real-time video streams efficiently.
- **Machine Learning:** Training and optimizing deep learning models for fast and accurate mask detection.
- **Public Health:** Enhancing compliance monitoring to reduce virus transmission risks in public spaces.

Motivation

- **Ensuring Public Safety**: Real-time face mask detection helps enforce mask-wearing policies, reducing the spread of infectious diseases.
- **AI-Driven Monitoring:** Integrating deep learning with computer vision enables automated and efficient compliance tracking in public spaces.
- **Real-World Impact:** The project bridges technology and public health, offering a scalable solution for improving safety in high-risk environments.

System Analysis and Design

- **Data Collection:** Gathering image datasets from Kaggle and real-time video streams for face mask detection.
- **Data Preprocessing:** Resizing images, data augmentation, and normalizing pixel values for model training.
- **Exploratory Data Analysis (EDA):** Analyzing image distributions, face mask variations, and dataset balance.
- **Model Development:** Implementing CNN-based models (MobileNetV2, Custom CNN) for mask detection.
- **Model Evaluation:** Assessing performance using accuracy, precision, recall, F1-score, and confusion matrix.

- **Deployment:** Integrating the model with OpenCV for real-time detection and deploying it on a live video stream.

Interactive Development

- Phase 1: Data collection and preprocessing.
- Phase 2: Exploratory data analysis and feature selection.
- Phase 3: Model training, evaluation, and fine-tuning.
- Phase 4: Deployment of results using dashboards or web applications.
- Phase 5: User feedback and model improvement.

Future Enhancements

- **Edge AI & IoT Integration:** Deploying the model on IoT-enabled cameras for real-time mask detection in smart surveillance systems.
- **Advanced Deep Learning Models:** Enhancing accuracy using **transformer-based models (Vision Transformers, EfficientNet, YOLO, etc.)**.
- **Mobile & Web Applications:** Developing an **app for real-time face mask detection** and alerts for public and workplace monitoring.
- **Geospatial Analysis:** Mapping mask compliance across different regions using **AI-powered surveillance data** for policy enforcement.

Results and Discussion

Findings indicate that the real-time face mask detection system successfully identifies individuals wearing or not wearing masks with high accuracy. Statistical analysis of model performance highlights strong classification accuracy, with CNN-based models outperforming traditional methods. The real-time deployment demonstrates effective mask detection in varying lighting conditions and angles, emphasizing the system's potential for public safety enforcement and automated monitoring in high-risk areas.

Conclusion and Policy Implications

This research highlights the importance of real-time face mask detection in ensuring public safety and enforcing health guidelines. The findings emphasize the need for AI-driven surveillance systems in high-risk areas such as hospitals, workplaces, and public transport. Policymakers and organizations can leverage this technology to automate compliance monitoring, enhance public health measures, and reduce virus transmission risks. Future research should focus on improving model accuracy, integrating IoT-based smart surveillance, and expanding deployment for large-scale monitoring.

Acknowledgment

I, [1]**Shweta Upadhyay**, express my sincere gratitude to **Thakur College of Science, Commerce, and Arts** for supporting this research. Special thanks to **Mr. Omkar Singh** and **Prof. Amit Kumar Pandey** for their valuable insights and technical guidance, which greatly contributed to the success of this study.

References

- **Howard, A.G., Sandler, M., et al.** (2019). *Searching for MobileNetV3. Proceedings of the IEEE International Conference on Computer Vision (ICCV).*
- **Zhang, K., Zhang, Z., et al**. (2016). *Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks (MTCNN). IEEE Signal Processing Letters*.
- **OpenCV Library (2023).** *Real-Time Face Detection using OpenCV and Deep Learning*. OpenCV Documentation.
- **Kaggle Datasets** (2024). *Face Mask Detection Dataset*. Retrieved from www.kaggle.com.
- **Rosebrock, A.** (2021). *Deep Learning for Computer Vision with Python*. PyImageSearch.