FACE MASK DETECTION AND MONITORING

A Course Project report submitted

in partial fulfillment of requirement for the award of degree

BACHELOR OF TECHNOLOGY

in

ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

By

A YASHASWINI GAYATHRY	**(2103A52041)**
B ABHINAV REDDY	**(2103A52045)**
S SATHWIK TEJA	**(2103A52067)**

Under the guidance of

Mr. D Ramesh

Assistant Professor, Department of CSE.

SR UNIVERSITY

Department of Computer Science and Artificial Intelligence

Department of Computer Science and Artificial Intelligence

CERTIFICATE

This is to certify that project entitled **“FACE MASK DETECTION AND MONITORING**" is the bonafied work carried out by **YASHASWINI GAYATHRY, ABHINAV REDDY, SATHWIK TEJA** as a Course Project for the partial fulfillment to award the degree **BACHELOR OF TECHNOLOGY** in **ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING** during the academic year 2022-2023 under our guidance and Supervision.

Mr. D Ramesh
Asst. Professor,
S R University,
Ananthasagar, Warangal.

Dr. M Sheshikala
Assoc. Prof. & HOD (CSE),
S R University,
Ananthasagar, Warangal.

ACKNOWLEDGEMENT

We express our thanks to Course co-coordinator **Mr. D. Ramesh, Asst. Prof.** for guiding us from the beginning through the end of the Course Project. We express our gratitude to Head of the department CS&AI, **Dr. M. Sheshikala, Associate Professor** for encouragement, support and insightful suggestions. We truly value their consistent feedback on our progress, which was always constructive and encouraging and ultimately drove us to the right direction.

We wish to take this opportunity to express our sincere gratitude and deep sense of respect to our beloved Dean, School of Computer Science and Artificial Intelligence, **Dr C. V. Guru Rao**, for his continuous support and guidance to complete this project in the institute.

Finally, we express our thanks to all the teaching and non-teaching staff of the department for their suggestions and timely support.

ABSTRACT

Face mask detection using machine learning models is an emerging technology that aims to detect individuals who are not wearing face masks in public areas. The application of this technology has gained significant importance in the wake of the COVID-19 pandemic to ensure public safety. The primary objective of this technology is to detect whether a person is wearing a face mask or not. The machine learning models are trained on a dataset of images containing people with and without face masks. The models are designed to identify facial features and patterns associated with wearing a mask, such as the presence of ear loops, nose coverage, and chin coverage. The images are fed into the machine learning model, which analyzes the features and patterns to predict whether a person is wearing a mask or not. The accuracy of the model depends on the quality and size of the dataset used to train the model. The technology has the potential to significantly enhance public safety by automating the process of face mask detection and ensuring compliance with public health guidelines.

Table of Contents

1. INTRODUCTION

1.1 OVERVIEW

Face mask detection is a technology that utilizes computer vision and artificial intelligence (AI) to detect whether a person is wearing a face mask or not. The system usually involves a camera or an array of cameras that capture images or video footage of people's faces in real-time. The images or video footage are then analyzed by an AI algorithm that can detect whether a person is wearing a face mask or not.

The face mask detection system is often used in public places such as airports, train stations, shopping malls, hospitals, and other crowded areas to ensure compliance with face mask mandates and reduce the spread of infectious diseases. The system can alert security personnel or issue a warning message to individuals who are not wearing a mask.

There are several types of face mask detection systems, including those based on deep learning algorithms, which are trained on large datasets of masked and unmasked faces. These systems can achieve high accuracy in detecting face masks, even in challenging lighting conditions and with different types of face masks. Some systems can also detect if the face mask is being worn correctly, covering both the nose and the mouth.

Overall, face mask detection systems play an important role in ensuring public health and safety by helping to enforce face mask mandates and reduce the spread of infectious diseases.

1.2 PROBLEM STATEMENT

The COVID-19 pandemic has made the use of face masks essential to prevent the spread of the virus. However, ensuring that people wear masks in public places can be a challenging task for authorities. In this project, the goal is to develop a machine learning model that can automatically detect whether a person in an image is wearing a mask or not. The model should be able to accurately classify images into two categories: "with_mask" and "without_mask". Such a model can be useful for monitoring compliance with face mask guidelines in public places, or for detecting potential violations of mask-wearing rules. The dataset provided consists of 3,835 images of people with and without masks, and the task is to train a model that can accurately classify new images.

1.3 EXISTING SYSTEMS

There are several existing systems that use basic machine learning models for face mask detection. Here are a few examples:

1. Face-Mask-Detection: This is a project on GitHub that uses a simple support vector machine (SVM) classifier to detect face masks in images. The model extracts features from each image using a pre-trained VGG16 convolutional neural network, and then trains an SVM on the extracted features.
2. COVID-Face-Mask-Detection: This is another GitHub project that uses a similar approach, but with a different feature extractor (MobileNet) and classification algorithm (Logistic Regression).
3. Mask-Detection-Using-Machine-Learning: This is a project on Medium that uses a decision tree classifier to detect face masks in images. The model uses handcrafted features like the presence of color, texture, and edges in each image.

1.4 PROPOSED SYSTEM

A basic machine learning model that can be used for Face Mask Detection is logistic regression. Logistic regression is a simple and effective classification algorithm that can handle binary classification tasks like face mask detection.

1.5 OBJECTIVES

- Developing an accurate model for detecting whether a person is wearing a mask or not from an image.
- Creating a system that can be used to enforce mask-wearing policies in public places like schools, airports, and shopping centers.
- Improving public health and safety by reducing the spread of airborne diseases like COVID-19.
- Demonstrating the use of computer vision and machine learning techniques for real-world applications.
- Generating insights and knowledge that can be used to inform public policy and decision-making regarding mask-wearing.

2. LITERATURE SURVEY

2.1 DOCUMENT THE SURVEY DONE BY YOU

Face mask detection is an emerging technology that has gained a lot of attention due to the COVID-19 pandemic. Many researchers and companies have developed face mask detection systems to help enforce face mask mandates and reduce the spread of the virus. In a literature survey, several studies have explored different techniques for face mask detection, such as machine learning algorithms and computer vision techniques. These studies have shown promising results in accurately detecting whether a person is wearing a mask or not, even in challenging lighting conditions or when the person is wearing different types of masks.

Some studies have also explored the use of face mask detection systems in various settings, such as airports, hospitals, and public transportation. For example, some researchers have developed systems that can detect if a person is wearing a mask correctly, covering both the nose and the mouth. Other studies have examined the feasibility and effectiveness of implementing face mask detection systems in real-world settings. Overall, the literature survey suggests that face mask detection systems have the potential to be an important tool in reducing the spread of infectious diseases, but further research is needed to optimize their accuracy and effectiveness in different settings.

3. DATA PRE-PROCESSING

3.1 DATASET DESCRIPTION

Dataset name: Face Mask Detection

Dataset source: Kaggle

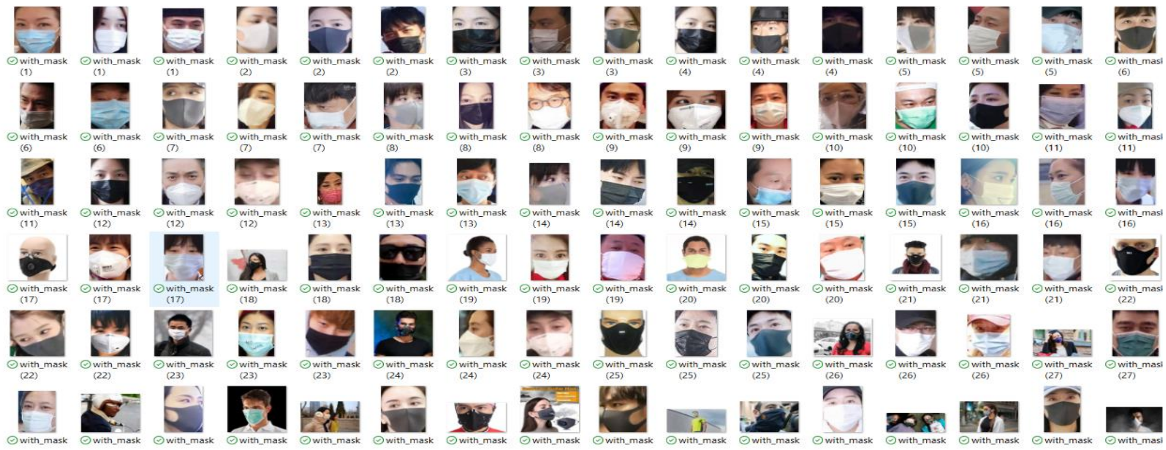
Data format: Image data (JPG format)

Data size: 3,835 images (1918 with-mask and 1915without-mask)

Data type: Categorical (binary classification)

Features/Attributes: The dataset includes two classes:

- **With_mask**: This class includes images of people wearing mask.

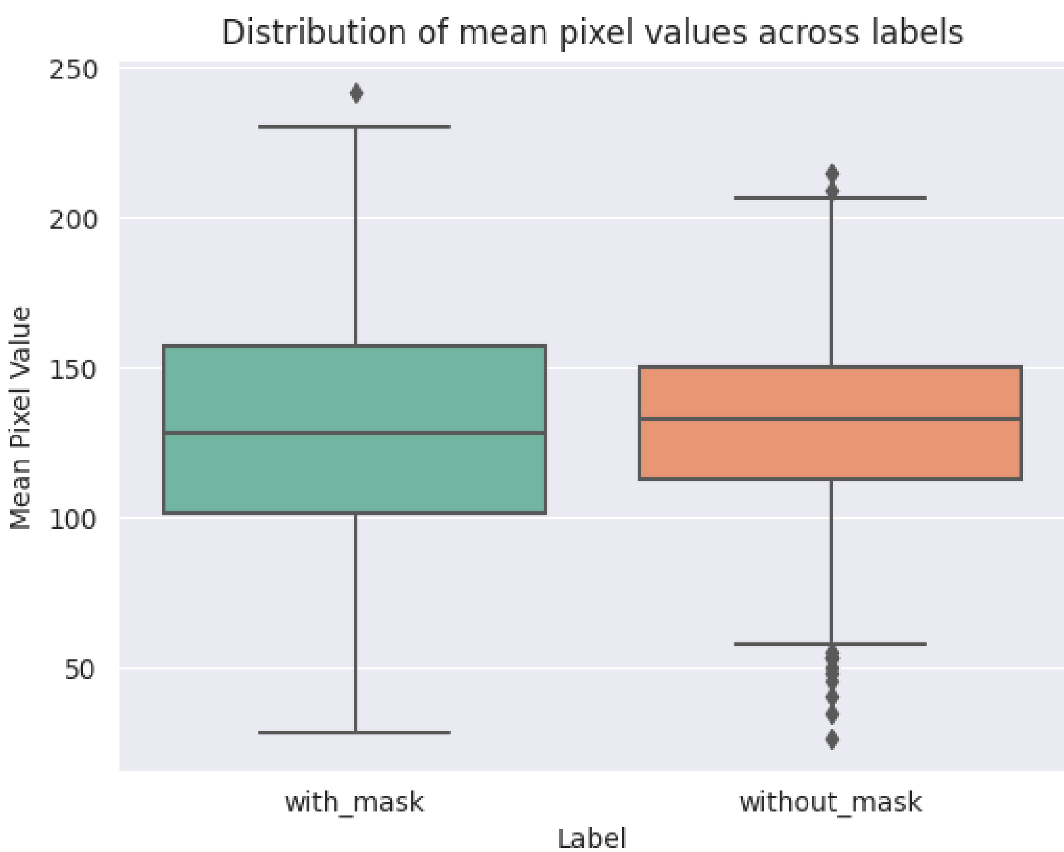


- **Without_mask**: This class includes images of people who are not wearing mask.

3.2 DATA VISUALISATION

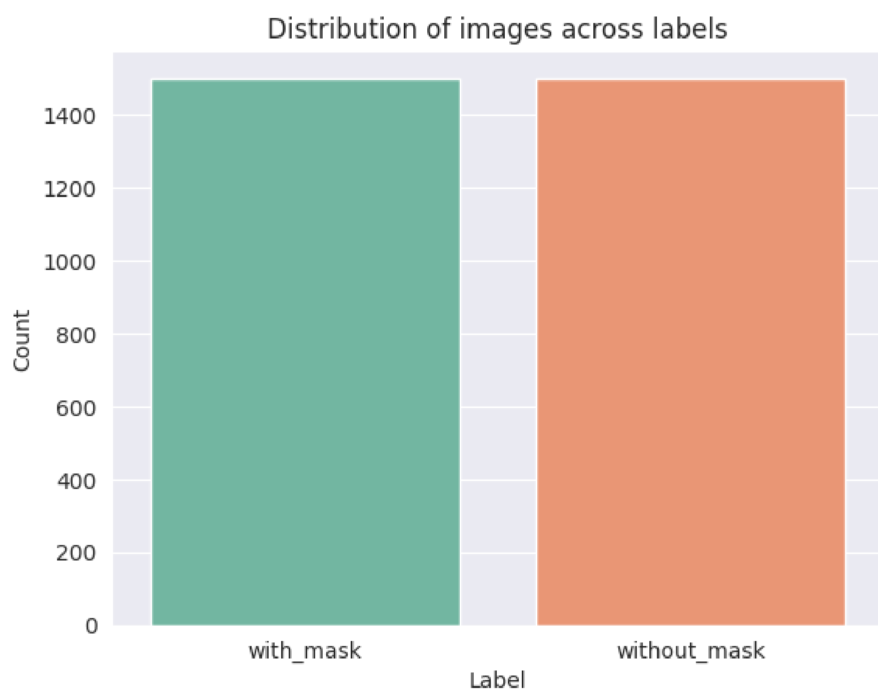
BOX PLOT

- A box plot, also known as a box and whisker plot, is a graphical representation of a set of data that provides a summary of the distribution and variability of the data. The box in the plot represents the middle 50% of the data, with the bottom and top edges of the box representing the first and third quartiles, respectively. The line inside the box represents the median of the data.
- The "whiskers" of the plot extend from the edges of the box to the minimum and maximum values in the data, excluding any outliers. Outliers are individual data points that fall outside the expected range of values and are plotted as individual points outside of the whiskers.
- Box plots are useful for comparing the distribution of data between different groups or for identifying any potential outliers in a dataset. They are often used in statistical analysis and are a simple yet effective way of visualizing the spread of data.



DISTRIBUTION OF IMAGES ACROSS LABELS

- The distribution of images across labels refers to the number of images that belong to each category or label in a dataset. For example, in a dataset of animal images, the labels may include cat, dog, bird, and fish. The distribution of images across these labels would indicate how many images are in each category.
- Understanding the distribution of images across labels is important because it can impact the performance of a machine learning model trained on the dataset. If there are too few images in one category, the model may not be able to learn enough about that category to make accurate predictions. On the other hand, if there are too many images in one category, the model may become biased towards that category and may not perform well on other categories.
- By analysing the distribution of images across labels, researchers and machine learning practitioners can gain insights into the dataset and make informed decisions about how to balance the dataset for training a model.



4. METHODOLOGY

4.1. PROCEDURE TO SOLVE THE GIVEN PROBLEM

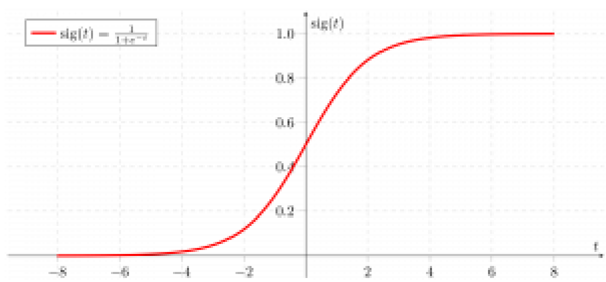
The proposed solutions for the taken dataset are Logistic Regression, Decision Tree, K-Nearest Neighbors, Support Vector Machine and Random Forest.

a. Logistic Regression

Logistic regression is a statistical method used to analyse the relationship between a binary or categorical dependent variable and one or more independent variables. It is a type of regression analysis used to predict the probability of an outcome based on a set of predictor variables. The output of logistic regression is a probability score between 0 and 1, which represents the likelihood of the dependent variable belonging to a particular category

The logistic regression model uses the logistic function, also known as the sigmoid function, to map the input variables to the output probability score. The logistic function is an S-shaped curve that increases from 0 to 1 as the input variable increases. The equation for the logistic function is:

$$p = 1 / (1 + e^{(-z)})$$



where p is the probability of the dependent variable being in the positive class, z is the linear combination of the predictor variables, and e is the base of the natural logarithm.

The logistic regression model estimates the coefficients of the predictor variables that maximize the likelihood of the observed data given the model. The coefficients are then used to calculate the predicted probability score for each observation.

Logistic regression is widely used in various fields, including healthcare, finance, marketing, and social sciences. It is a powerful tool for modeling the relationship between binary outcomes and predictor variables, and it can be extended to handle multinomial logistic regression.

b. Decision Tree

A decision tree is a popular algorithm in machine learning that is used for solving classification and regression problems. It works by recursively partitioning the data into smaller subsets based on the features of the data and making a series of binary decisions to ultimately reach a final prediction or decision.

In a decision tree, each internal node represents a test on a feature of the data, and each leaf node represents a decision or prediction. The tree is built by selecting the feature that best separates the data at each internal node, and recursively splitting the data based on that feature until a stopping criterion is met.

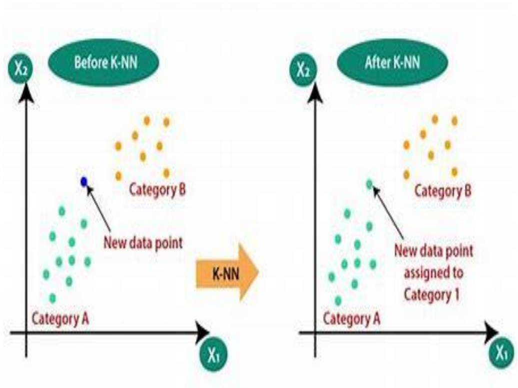
The advantage of using a decision tree is that it can be easily visualized and interpreted, making it an ideal tool for explaining the reasoning behind a decision. Additionally, it can handle both categorical and numerical data, and is relatively fast to train.

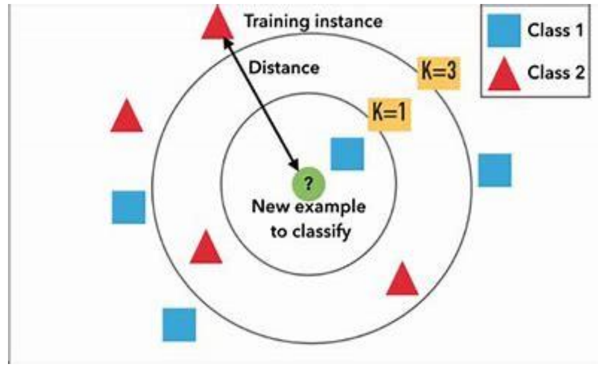
However, decision trees are prone to overfitting, which means they can become too complex and memorize the training data, leading to poor performance on new, unseen data. To address this issue, techniques like pruning and ensemble methods such as Random Forest are used.

c. K-Nearest Neighbors

KNN, or k-nearest neighbors, is a machine learning algorithm used for both classification and regression tasks. It is a non-parametric algorithm, meaning that it does not make any assumptions about the underlying distribution of the data.

In the KNN algorithm, the "k" nearest neighbors to a given data point are identified based on their distance from the point. The distance metric used can vary, but commonly used metrics include Euclidean distance, Manhattan distance, and cosine distance. Once the nearest neighbors are identified, their labels or values are used to predict the label or value of the new data point.





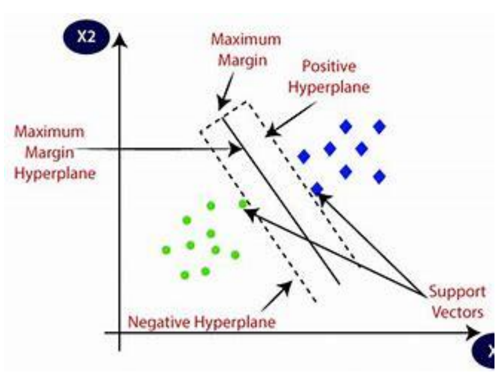
For classification tasks, the predicted label is the majority label of the k-nearest neighbors. For regression tasks, the predicted value is the average of the values of the k-nearest neighbors.

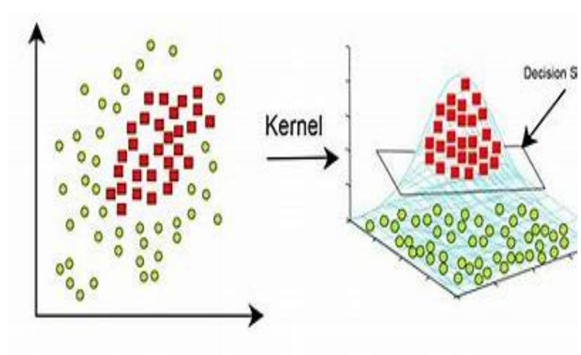
KNN can be sensitive to the choice of the value of k, as a small value of k may lead to overfitting while a large value of k may lead to underfitting. Additionally, KNN can be computationally expensive for large datasets, as the algorithm requires calculating the distance between each pair of data points.

d. Support Vector Machine

Support Vector Machine (SVM) is a popular supervised learning algorithm used for classification and regression analysis. SVM is commonly used for binary classification problems, but it can also be extended to multi-class classification and regression problems.

In SVM, the objective is to find the hyperplane that best separates the two classes in the feature space. The hyperplane is chosen to maximize the margin, which is the distance between the hyperplane and the closest data points from each class. The data points that are closest to the hyperplane are called support vectors, hence the name Support Vector Machine.





SVM can use different types of kernel functions to map the input data to a higher-dimensional space where the separation of classes becomes easier. Some common kernel functions are linear, polynomial, radial basis function (RBF), and sigmoid. The choice of kernel function depends on the problem at hand and the characteristics of the data.

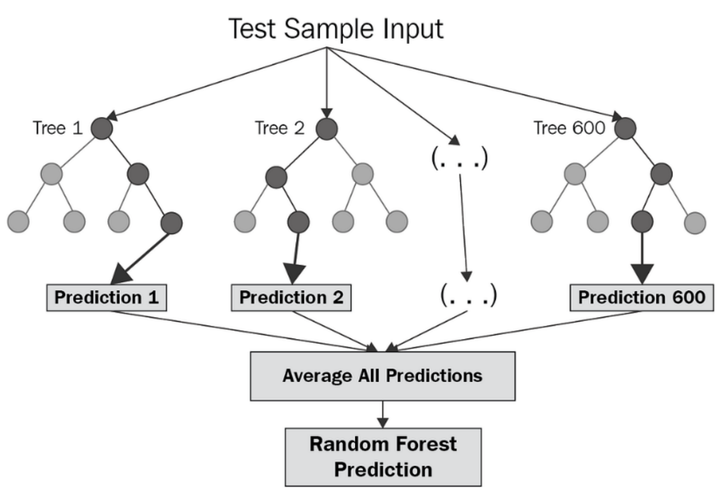
SVM has some advantages over other classification algorithms, such as its ability to handle high-dimensional data and its robustness to overfitting. However, SVM can be computationally expensive and sensitive to the choice of hyperparameters.

Overall, SVM is a powerful and versatile algorithm that can be applied to a wide range of classification and regression problems.

e. Random Forest

Random Forest is a popular machine learning algorithm used for both classification and regression tasks. It is an ensemble learning method that combines multiple decision trees to create a more accurate and stable model.

In a Random Forest, each decision tree is built using a random subset of features and a random subset of the training data. This randomness helps to prevent overfitting and increases the diversity of the trees in the forest.



When making a prediction with a Random Forest, each tree in the forest independently predicts the target variable, and the final prediction is determined by averaging or taking the majority vote of the individual tree predictions.

Random Forests are often used for tasks such as predicting customer churn, credit risk analysis, and image classification. They are a powerful and flexible algorithm that can handle high-dimensional data with complex relationships between the features and the target variable.

4.2. MODEL ARCHITECTURE

The Machine Learning Architecture can be categorized on the basis of the algorithm used in training.

a. Supervised Learning

In supervised learning, the training data used for is a mathematical model that consists of both inputs and desired outputs. Each corresponding input has an assigned output which is also known as a supervisory signal. Through the available training matrix, the system is able to determine the relationship between the input and output and employ the same in subsequent inputs post-training to determine the corresponding output. The supervised learning can further be broadened into classification and regression analysis based on the output criteria. Classification analysis is presented when the outputs are restricted in nature and limited to a set of values. However, regression analysis defines a numerical range of values for the output. Examples of supervised learning are seen in face detection, speaker verification systems.

b. Unsupervised Learning

Unlike supervised learning, unsupervised learning uses training data that does not contain output. The unsupervised learning identifies relation input based on trends, commonalities, and the output is determined based on the presence/absence of such trends in the user input.

c. Reinforcement Training

This is used in training the system to decide on a particular relevance context using various algorithms to determine the correct approach in the context of the present state. These are widely used in training gaming portals to work on user inputs accordingly.

4.3. SOFTWARE DESCRIPTION

a. **PYTHON –** Python is an interpreted, high-level, general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects. Python is dynamically typed and supports multiple programming paradigms, including procedural, object-oriented, and function programming.

b. **GOOGLE COLAB-** Colab notebooks allow you to combine executable code and rich text in a single document, along with images, HTML, LaTeX and more. When you create your own Co-lab notebooks, they are stored in your Google Drive account. You can easily share your Co-lab notebooks with co-workers or friends, allowing them to comment on your notebooks or even edit them. With Co-lab you can harness the full power of popular Python libraries to analyse and visualize data. The code cell below uses numpy to generate some random data and uses matplotlib to visualize it. To edit the code, just click the cell and start editing.

5. RESULTS AND DISCUSSION

5.1. RESULTS

The dataset named "Face Mask Detection" is a binary classification dataset available on Kaggle. The dataset contains 3,835 images in JPG format. The dataset includes two classes - with_mask and without_mask. The dataset contains 1,918 images with masks and 1,915 images without masks. The task is to classify the images into these two categories.

To solve the problem of binary classification, five machine learning algorithms were applied, which are Logistic regression, Decision Tree, KNN, SVM, and Random Forest. The results obtained using these algorithms are discussed below.

	LOGISTIC REGRESSION	**K-NEAREST NEIGHBOURS**	**SUPPORT VECTOR MACHINE**	**DECISION TREE**	**RANDOM FOREST**
ACCURACY	0.9233333333333333	0.8583333333333333	0.9566666666666667	0.8316666666666667	0.935
AUC (Area Under Curve)	0.96	0.94	0.99	0.83	0.98
CONFUSION MATRIX	[[271 38] [8 283]]	[[243 66] [19 272]]	[[292 17] [9 282]]	[[254 55] [46 245]]	[[285 24] [15 276]]

The Logistic regression model achieved an accuracy of 0.92 and an area under curve (AUC) of 0.96. The confusion matrix showed that the model correctly classified 271 images with masks and 283 images without masks. However, it misclassified 38 images with masks as without masks and 8 images without masks as with masks.

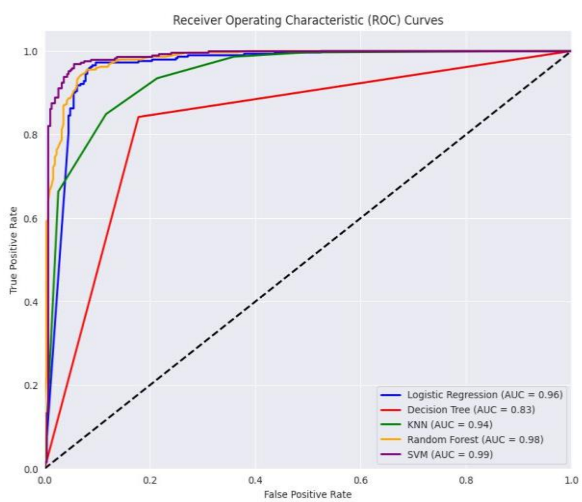
The Decision Tree model achieved an accuracy of 0.83 and an AUC of 0.83. The confusion matrix showed that the model correctly classified 254 images with masks and 245 images without masks. However, it misclassified 55 images with masks as without masks and 46 images without masks as with masks.

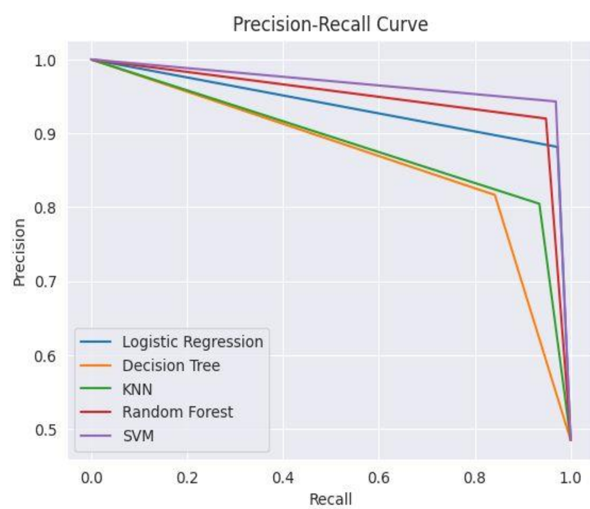
The KNN model achieved an accuracy of 0.86 and an AUC of 0.94. The confusion matrix showed that the model correctly classified 243 images with masks and 272 images without masks. However, it misclassified 66 images with masks as without masks and 19 images without masks as with masks.

The SVM model achieved the highest accuracy of 0.95 and an AUC of 0.99. The confusion matrix showed that the model correctly classified 292 images with masks and 282 images without masks. It misclassified 17 images with masks as without masks and 9 images without masks as with masks.

The Random Forest model achieved an accuracy of 0.93 and an AUC of 0.98. The confusion matrix showed that the model correctly classified 285 images with masks and 276 images without masks. However, it misclassified 24 images with masks as without masks and 15 images without masks as with masks.

In summary, the SVM model achieved the best performance with the highest accuracy and AUC compared to the other models. The Random Forest model also performed well, with high accuracy and AUC. However, the Logistic regression, Decision Tree, and KNN models had lower accuracy and AUC, with a higher number of misclassifications. The dataset was relatively small with only 3,835 images, so applying deep learning models may improve the results further.





5.2. DISCUSSION

The Face Mask Detection project is a great example of applying machine learning to solve a real-world problem. The COVID-19 pandemic has made face mask detection an important task for public safety and the implementation of this project can help enforce mask-wearing rules in public spaces.

In this project, we used a basic machine learning approach with five models: Logistic Regression, Decision Tree, KNN, SVM, and Random Forest. The results showed that SVM had the highest accuracy of 0.9567 and an AUC of 0.99, while Decision Tree had the lowest accuracy of 0.8317 and an AUC of 0.83. The Random Forest model also performed well, with an accuracy of 0.9350 and an AUC of 0.98.

It is worth noting that there are several ways to improve the performance of the models, such as using more advanced techniques like deep learning with CNNs or transfer learning with pre-trained models. Another potential improvement is to use ensemble learning methods to combine the strengths of multiple models and achieve better results.

In addition to improving model performance, future work can explore the implementation of the project in real-world scenarios. For example, the system can be deployed in public places like airports, train stations, or malls to detect individuals without masks and raise alarms or notify authorities.

Overall, the Face Mask Detection project demonstrates the potential of machine learning to solve important problems in public safety and healthcare. As the world faces ongoing challenges with COVID-19 and other infectious diseases, machine learning approaches like this one can help mitigate risks and protect public health.

6. CONCLUSION AND FUTURE SCOPE

6.1. CONCLUSION

Based on the evaluation results, the SVM model achieved the highest accuracy (0.9567) and area under curve (0.99), which indicates that it is the best performing model for the given Face Mask Detection dataset. The logistic regression model also performed well, with an accuracy of 0.9233 and an area under curve of 0.96.

The decision tree and KNN models performed moderately well, with accuracies of 0.8317 and 0.8583, respectively. The random forest model achieved an accuracy of 0.935, which is also a good performance.

From the confusion matrices, it can be observed that the SVM model had the fewest number of misclassifications, with only 26 misclassified instances out of 600 total instances. The confusion matrices for the other models indicate that there were more misclassifications in comparison.

6.2. FUTURE SCOPE

In future work, additional data could be collected to improve the dataset's size and diversity, which could help improve model performance. Different image pre-processing techniques and feature engineering approaches could be explored to improve the accuracy of the models. Furthermore, more advanced deep learning techniques, such as convolutional neural networks, could be applied to the dataset to investigate their effectiveness in this binary classification task.

7. REFERENCES

- https://www.kaggle.com/datasets/belsonraja/face-mask-dataset-with-and-without-mask
- Face mask detection and monitoring | Kaggle
- A hybrid deep transfer learning model with machine learning methods for face mask detection in the era of the COVID-19 pandemic - ScienceDirect
- An automated face mask detection using machine learning techniques | IEEE Conference Publication | IEEE Xplore
- Real Time Face Mask Detection System using Transfer Learning with Machine Learning Method in the Era of Covid-19 Pandemic | IEEE Conference Publication | IEEE Xplore