Face Mask Detection 

Overview:

This project aims to create a CNN model to identify whether the given person is wearing a mask or not. The project aims to develop a trained CNN model capable of accurately detecting face masks in images and video streams by:

- Achieving a high accuracy and F1-score.
- Handling diverse and large-scale datasets efficiently.
- Providing reliable predictions that can be integrated into various surveillance systems.

Motivation:

The COVID-19 pandemic has underscored the importance of wearing face masks to mitigate the spread of the virus. Automated face mask detection systems can play a crucial role in ensuring compliance in public spaces such as airports, malls, hospitals, and workplaces. This project aims to develop a robust and accurate face mask detection system using Convolutional Neural Networks (CNN). The primary objective of this project is to design, implement, and deploy a deep learning model capable of detecting whether individuals in an image or video stream are wearing face masks. The model should be efficient and accurate to facilitate real-time detection.

Technical Tags:

Deep Learning, Convolutional Neural Networks, CNN, Transfer Learning, VGG16, ResNet50, Data Collection, Data Preprocessing, Image Resizing, Normalization, Data Augmentation, Categorical Cross-Entropy, Adam Optimizer, Model Training, Model Evaluation, Accuracy, Precision, Recall, F1-Score, Confusion Matrix, Hyperparameter Tuning, Python, TensorFlow, Keras, PyTorch, OpenCV, Real-Time Detection, Image Classification, Face Mask Detection, Computer Vision

Solution / Approach:

To tackle the problem statement, we followed a series of well-defined steps:

1. Data collection and preprocessing:
    a) Collection: Gather a comprehensive dataset with labels indicating the presence or absence of a face mask.
    b) Preprocessing: Resize images, normalize pixel values, and apply data augmentation to improve model generalization.
2. Model development and architecture
    a) CNN Design: Design a Convolutional Neural Network architecture with layers optimized for feature extraction and classification.
    b) Transfer Learning: Utilize pre-trained models (e.g., VGG16, ResNet50) to leverage existing feature extraction capabilities.

c) Fine-tuning: Fine-tune the pre-trained model on the face mask detection dataset.

3. Model training:
    a) Training: Train the CNN model on the pre-processed dataset using a suitable loss function and optimizer (e.g., categorical cross-entropy, Adam optimizer).
    b) Validation: Use a validation set to tune hyperparameters and avoid overfitting.
4. Model evaluation:
    a) Metrics: Evaluate the model using metrics such as accuracy, precision, recall, F1-score, and confusion matrix.
    b) Testing: Test the model on an unseen test set to assess its real-world performance.
5. Model Selection: Compare various models to choose the best using an appropriate evaluation metrics
6. Hyper-parameter tuning