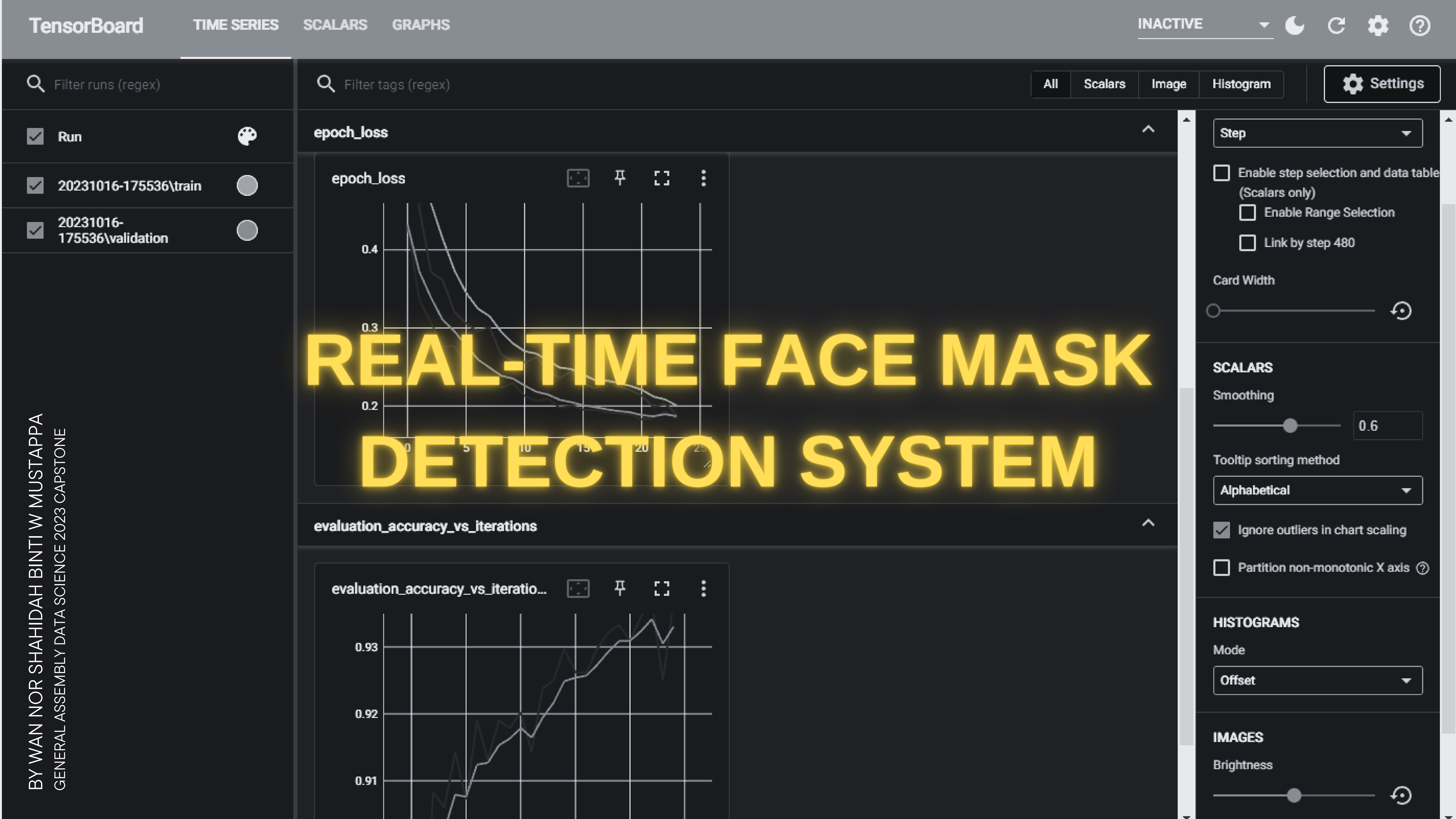
REAL-TIME FACE MASK DETECTION SYSTEM

BY WAN NOR SHAHIDAH BINTI W MUSTAPPA
GENERAL ASSEMBLY DATA SCIENCE 2023 CAPSTONE



PROJECT OVERVIEW/INTRODUCTION

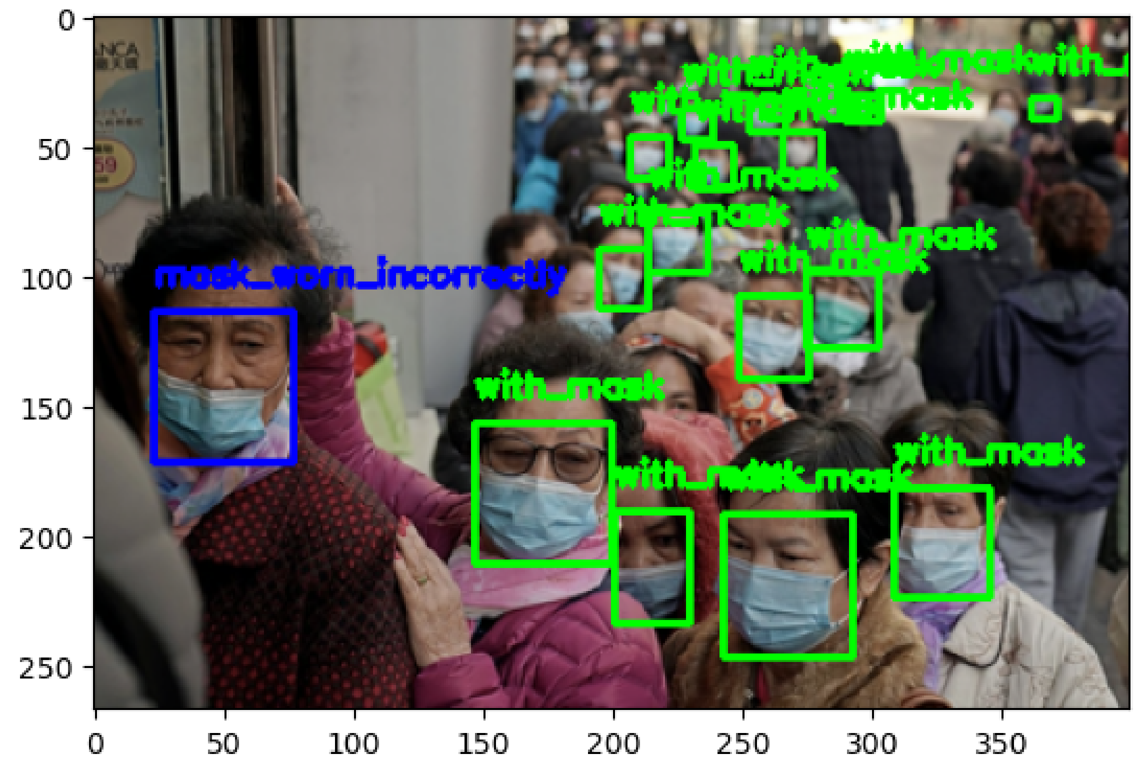
PROBLEM STATEMENT

"In the wake of the global pandemic, ensuring that individuals wear face masks correctly is paramount to public health safety. This analysis aims to explore a dataset of images annotated with individuals wearing masks, not wearing masks, or wearing masks incorrectly. We aim to understand the dataset's structure, diversity, and potential challenges to inform the development of a machine learning model for real-time face mask detection and classification."

Objective:
"Develop a Real-Time Face Mask Detection System"
Aim to create an efficient and accurate system for detecting face masks on individuals in real-time, utilizing state-of-the-art deep learning techniques

Importance:
"Enhancing Safety and Reducing the Spread of Airborne Diseases"
Automating the detection process will bolster efforts to mitigate the spread of COVID-19 and other airborne diseases, ensuring a safer environment for all.

Methodology Overview:
"Harnessing the Power of Convolutional Neural Networks (CNNs)"
Approach involves training a CNN to identify and classify individuals based on their adherence to face mask protocols, enabling real-time monitoring and reporting.



MODEL TEST PREP: EVALUATING FACE MASK DETECTION EFFICACY

Face Mask Detector Development

Face mask detection unfolds in two pivotal steps: the initial detection of faces within an image, followed by the classification of each detected face based on the presence and correctness of mask wearing.

1. Process Overview:

- Step 1: Utilize advanced face detection algorithms to accurately identify and isolate faces within diverse images.
- Step 2: Implement a classification model to categorize the detected faces into one of three distinct classes - 'no mask', 'mask worn incorrectly', or 'with mask'.

2. Model Architecture:

- Base Model: Leverage MobileNet for its efficiency and accuracy as the foundational architecture.
- Custom Head: Engineer and train a custom head layer tailored to classify faces into the three predefined classes, enhancing the base model's applicability to the specific task of mask detection.

3. Reference and Adaptation:

- **Inspiration Source**: The model's structure and training approach is inspired and adapted from a previously established model.
- **Customization**: Modifications and tuning have been meticulously executed to accommodate a three-class classification system and optimize the model's performance on the specific dataset at hand.

Insights:

- **Comprehensive Detection**: The two-step process ensures a thorough analysis, from accurate face detection to nuanced classification based on mask-wearing status.
- **Optimized Performance**: The incorporation of MobileNet and a custom-trained head ensures a balance of efficiency and accuracy, yielding reliable results in real-time applications.
- **Adaptive Approach**: Drawing inspiration from proven models and adapting to specific requirements ensures a tailored solution, optimized for performance and reliability in diverse scenarios.

DATA GATHERING AND PREPARATION

Hypothesis 1: Data Distribution

- H_0: Classes (mask, no mask, incorrect mask) are uniformly distributed in the dataset.
- H_a: Classes are not uniformly distributed.

Hypothesis 2: Face Detection Accuracy

- H_0: MTCNN and OpenCV DNN have similar face detection accuracy.
- H_a: There's a significant difference in accuracy between MTCNN and OpenCV DNN.

Hypothesis 3: Model Performance

- H_0: The mask detection model performs no better than chance in classifying mask-wearing status.
- H_a: The model performs significantly better than chance.

Hypothesis 4: Mask Wearing Trends (if additional data is available)

- H_0: No significant association between demographic/environmental factors and correct mask-wearing.
- H_a: A significant association exists between specific factors and mask-wearing trends.

Using the Hypotheses:

- **Data Exploration and Preparation:** Use Hypothesis 1 to understand and prepare the dataset for modeling.
- **Face Detection:** Evaluate and compare face detection methods using Hypothesis 2.
- **Model Training and Evaluation:** Use Hypothesis 3 to evaluate the performance of the mask detection model.
- **Insights and Recommendations:** If applicable, use Hypothesis 4 to derive insights and make recommendations for targeted interventions to improve mask-wearing compliance.

Target (Label):
The target variable is the classification of each detected face based on mask-wearing status. It is a categorical variable with three possible classes:

With Mask: The face is wearing a mask correctly.
Without Mask: The face is not wearing a mask.
Mask Worn Incorrectly: The face is wearing a mask, but it is not covering the face properly.

This target is derived from the annotations in dataset, which label each face according to these categories.

Features:
The features are the input data used to make predictions about the target variable. In this project, the features are primarily the images of detected faces, processed and prepared for input into the machine learning model. Specifically, the features include:

Images of Detected Faces:
These are extracted from the original images using face detection methods (MTCNN or OpenCV DNN as per your notebooks). Each detected face is an individual data point that the model will classify.
These images may be preprocessed (e.g., resized, normalized) to meet the input requirements of the model.

Bounding Box Coordinates:
The coordinates of the bounding box around each detected face could potentially be used as features, although this is more common in object detection tasks rather than classification tasks.

Data Preparation for Model Training:

- ***Face Images***: Each face image can be preprocessed and transformed into a numerical format (e.g., pixel intensity values) suitable for machine learning. The preprocessing steps might include resizing the images to a standard size, normalizing the pixel values, and augmenting the data to increase the dataset's diversity and robustness.

- ***Annotations***: The annotations that indicate whether a face is wearing a mask, not wearing a mask, or wearing a mask incorrectly will be encoded into numerical labels that the model can predict.

Model Training:
With the features and target defined, we can proceed to train a machine learning model (like the MobileNetV2 model mentioned in *model-training.ipynb notebook*) to classify the faces based on mask-wearing status. The model will learn the patterns and features associated with each class from the training data and will then be able to make predictions on new, unseen data.

Exploratory Data Analysis (EDA)

A thorough exploratory data analysis is paramount to gain insights into the dataset's structure, volume, and characteristics. This foundational step precedes the modeling phase, ensuring an informed and strategic approach to model development.

1. Data Inspection:
Objective: Initiate the EDA by quantifying the total number of images and faces available for model training, offering a preliminary insight into the dataset's volume and diversity.
Method: Employ descriptive statistics and visualization techniques to facilitate a comprehensive and intuitive understanding of the data.

2. Dataframe Construction:
Objective: Create a structured dataframe that encapsulates key information, ensuring efficient image referencing and data manipulation.
Process: Integrate image data and annotations into a cohesive dataframe, streamlining data retrieval and analysis.

3. Image and Annotation Verification:
Objective: Validate the accuracy and consistency of image annotations to ensure reliable data for model training.
Method: Visual inspection of a subset of images alongside their annotations, confirming the precision and reliability of the annotated data.

Insights and Observations:

Data Volume and Quality: A preliminary assessment of the dataset’s volume, diversity, and quality, laying the groundwork for subsequent data preprocessing and modeling.
Structured Representation: The dataframe offers a systematic and accessible representation of the data, enhancing efficiency in data handling and analysis.
Annotation Integrity: Verification processes confirm the integrity and accuracy of annotations, ensuring that the model is trained on reliable and precise data.

Key Takeaways:
Informed Approach: EDA results are instrumental in shaping the modeling strategy, offering insights into data characteristics and potential challenges.
Data Integrity: Ensuring the accuracy and consistency of annotations is pivotal to develop a model that is both reliable and accurate.
Strategic Foundation: The insights gleaned from EDA inform subsequent steps in data preprocessing, feature engineering, and model development, ensuring an informed and strategic approach.

EDA IN CONVOLUTIONAL NEURAL NETWORKS (CNNS)

EDA for image data, especially involving Convolutional Neural Networks (CNNs), can include tasks like:

1. Visual Inspection:
Displaying Images: Visualize sample images from the dataset to understand their quality, variations, and challenges (like lighting conditions, occlusions, etc.).
Face Detection: Use a face detector to locate and visualize faces in images.

2. Data Quality Assessment:
Missing Data: Check for missing or corrupted images.
Label Quality: Assess the quality and consistency of annotations/labels.

3. Data Distribution:
Class Distribution: Analyze the distribution of classes (faces with masks, faces without masks, etc.) to identify any imbalance.
Feature Distribution: Extract features using CNN and analyze their distributions.

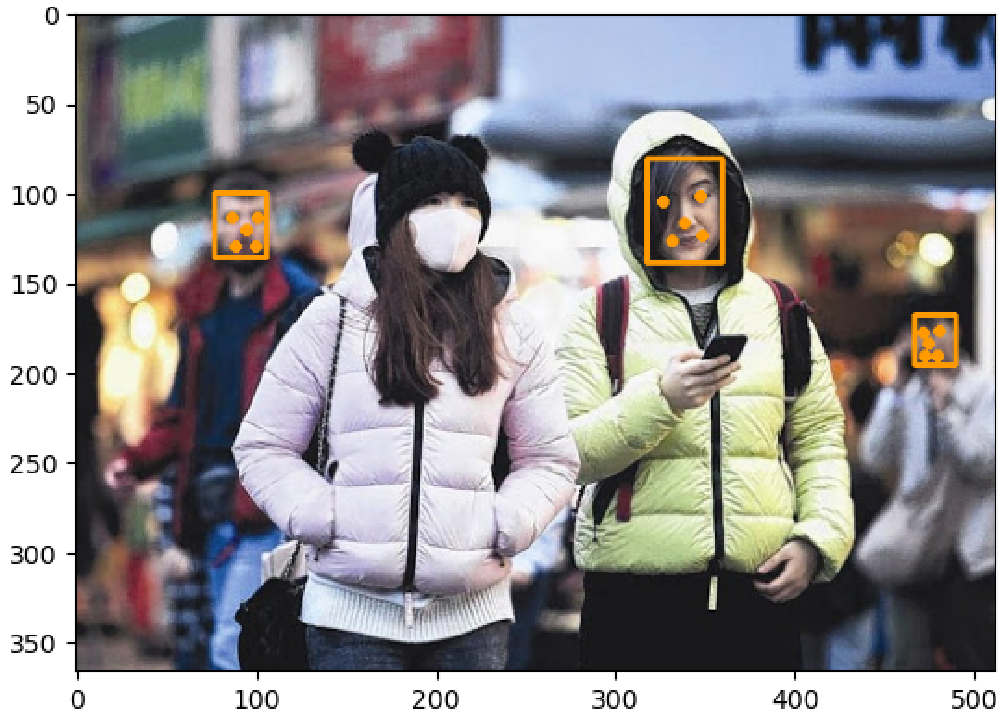
4. Model Predictions:
Face Mask Detection: Use a CNN model to predict whether faces in images are wearing masks or not and analyze the predictions' distribution and quality.

5. Error Analysis:
False Positives/Negatives: Analyze cases where the model made incorrect predictions to understand its weaknesses.

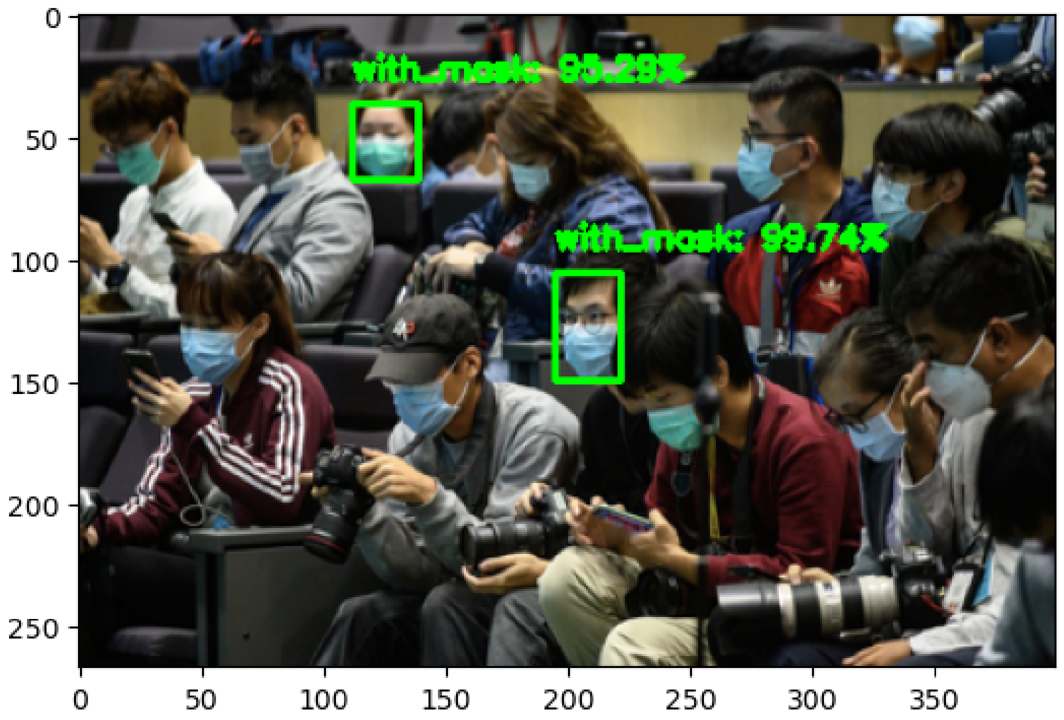
Raw datasets:

EDA:

<matplotlib.image.AxesImage at 0x1c7d1df8410>



Example of one function definition:
Face detector to locate faces in an image at a given confidence minimum. Predictions are made using the face mask detector model



MODEL DEVELOPMENT AND ARCHITECTURE

[MOBILENETV2 (FROZEN LAYERS)]
↓
[AVERAGE POOLING 2D]
↓
[FLATTEN]
↓
[DENSE - 64, RELU]
↓
[DROPOUT - 0.5]
↓
[DENSE - 3, SOFTMAX]

Harnessing MobileNetV2 and Custom Layers:

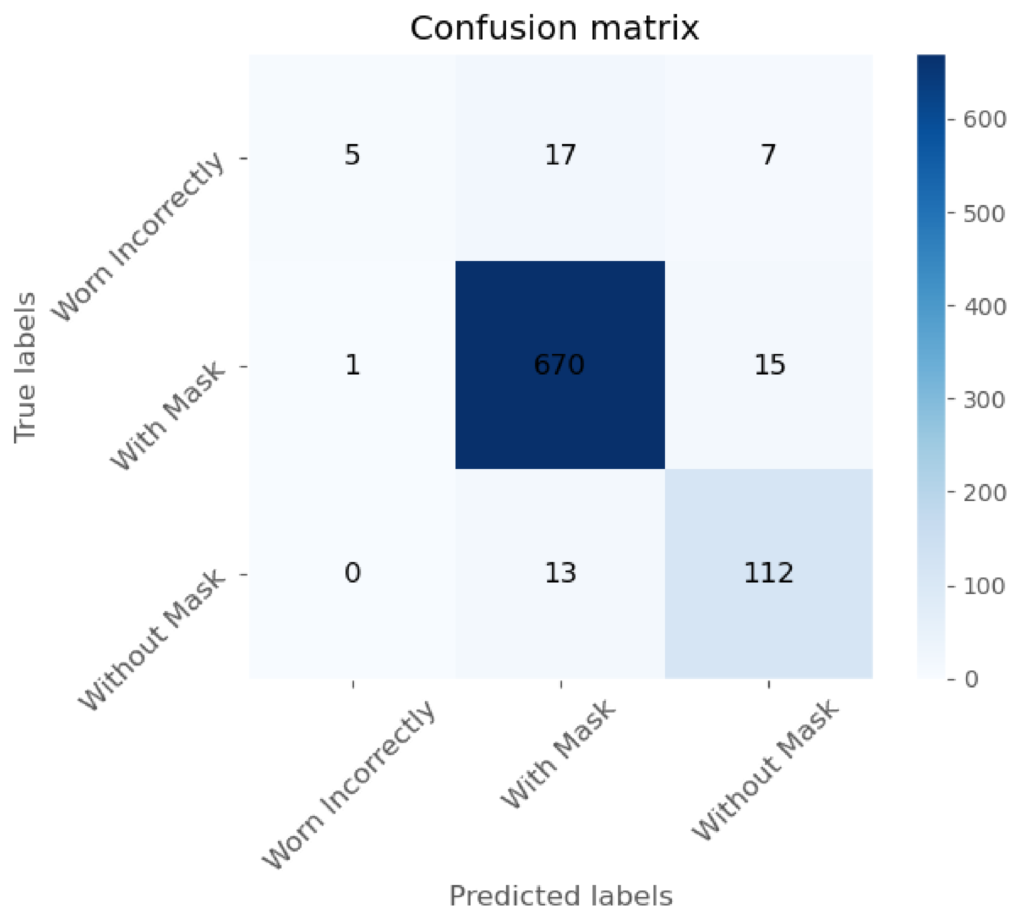
Overview:
Mask classifier model ingeniously combines the architectural prowess of MobileNetV2 with tailored custom layers, ensuring both efficiency and precision in identifying and classifying face masks in real-time.

Details:
The MobileNetV2 base, renowned for its speed and performance, is enhanced with a custom head layer. This synergy facilitates the distinct classification of individuals into three categories: with mask, without mask, and mask worn incorrectly, with an emphasis on real-time processing and accuracy.

Optimizing Data for Peak Performance:

Overview:
Every image is a repository of data waiting to be harnessed. Through meticulous preprocessing and innovative augmentation techniques, each image is transformed into an optimized data entity, ready to contribute to the model's learning journey.

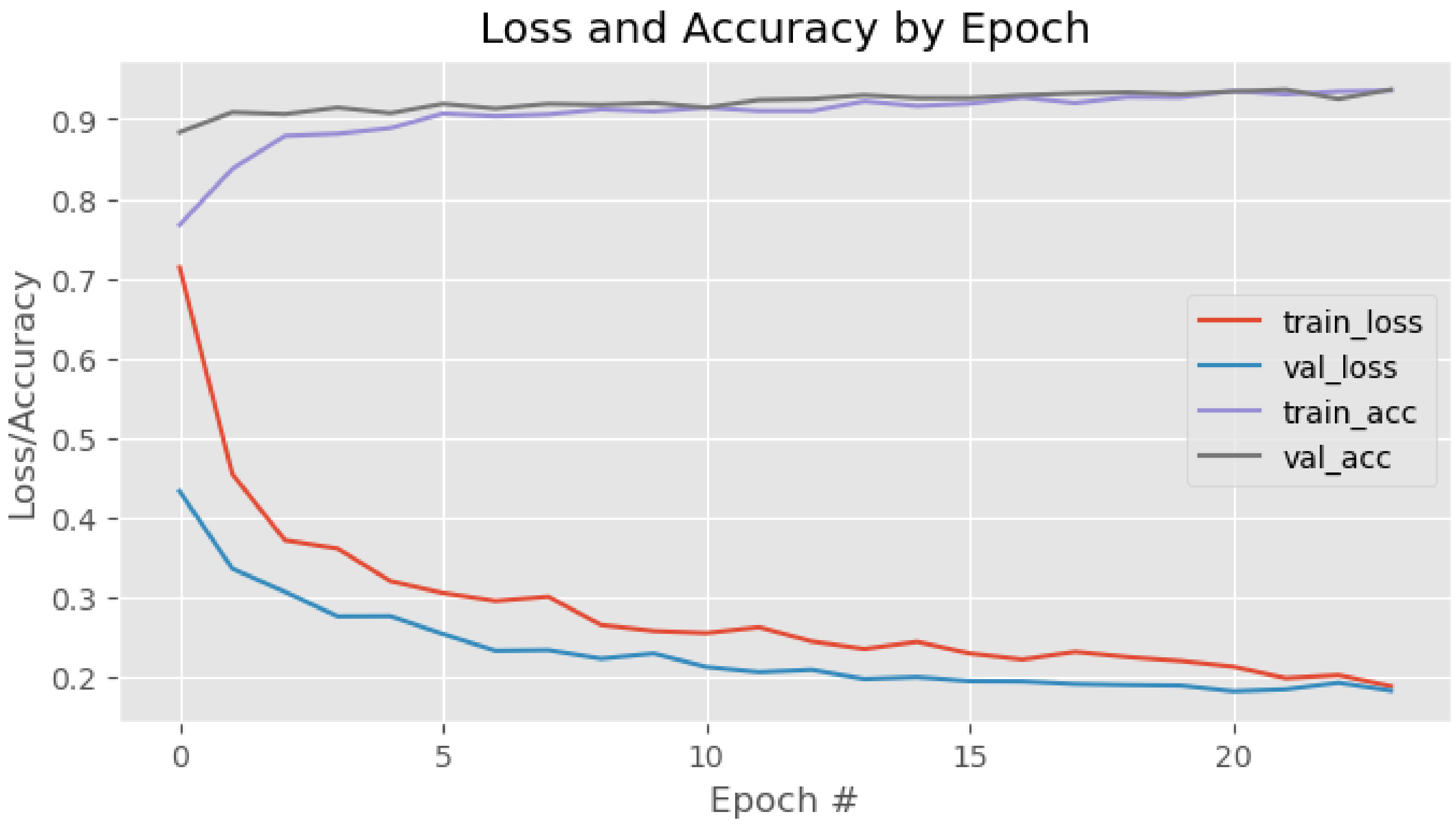
Details:
The preprocessing phase sees each image resized and encoded, ensuring uniformity and consistency. Real-time data augmentation, utilizing ImageDataGenerator, applies a series of transformations, including rotation, zoom, and flip, enhancing data diversity and model generalization.



Leveraging Pretrained Weights and Architectures:

Overview:
In the world of deep learning, standing on the shoulders of giants often yields the most significant results. The model does just that, applying pretrained ImageNet weights and freezing layers to leverage pre-learned features.

Details:
With 17 bottleneck residual blocks, each with 3 layers, and a final 1x1 convolution layer, feature extraction becomes a forte of this model. The application of transfer learning amplifies its ability to discern and classify with increased accuracy, ensuring each prediction is both reliable and swift.



Tailoring the Model for Specific Classifications:

Overview:
Every model bears the hallmark of its creation. The modified head of our model, complete with tailored layers and activations, stands as a testament to innovation, ensuring each classification is both precise and reliable.

Details:
The architecture includes Average Pooling 2D, Flatten, a Dense 64-node layer with ReLU activation, and a final Dense 3-node layer with softmax activation. A dropout of 0.5 ensures the model remains general and versatile, avoiding the pitfalls of overfitting.

Achieving Precision with Refined Techniques:

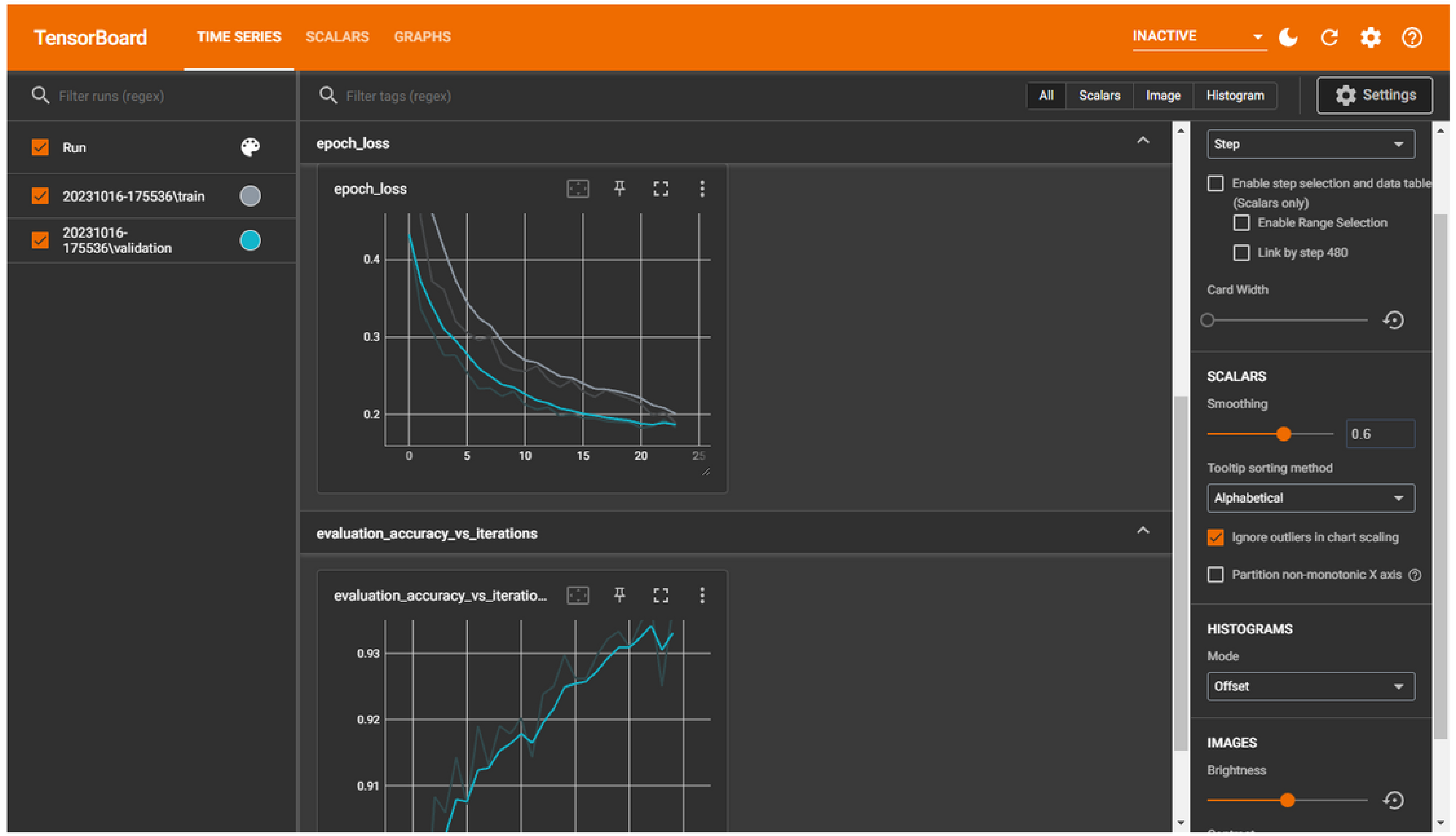
Overview:
Model training is both an art and a science. Through a series of refined techniques and evaluations, our model achieved a 94.2% accuracy rate, marking a significant milestone in the quest for optimal face mask detection.

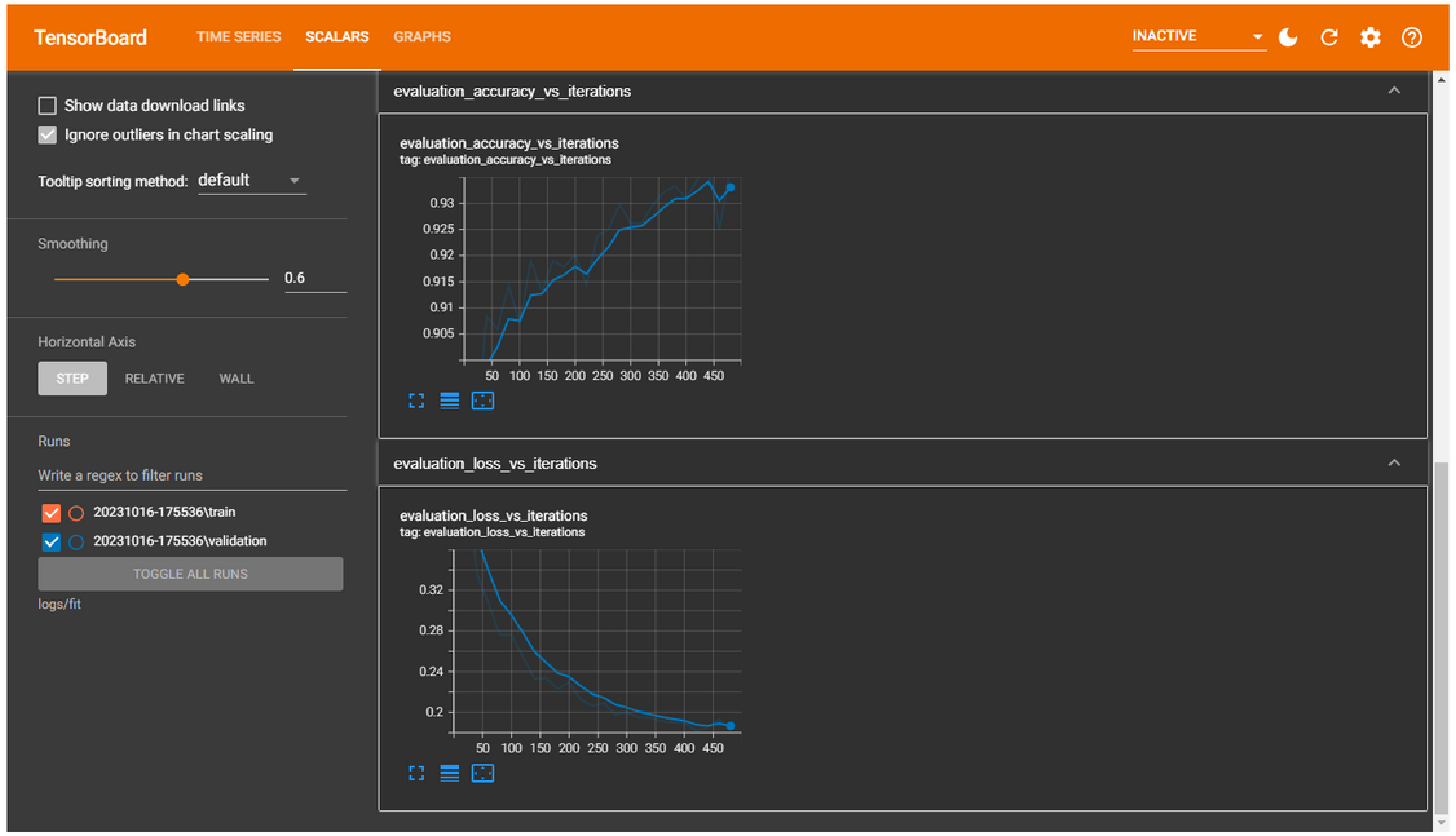
Details:
A train/test split of 0.3 ensures the model is vetted rigorously, proving its mettle before deployment. Each epoch of training brings with it enhanced precision, culminating in a model that stands ready to tackle real-world challenges with grace and accuracy.

RECOMMENDATION

Evaluation:

- **Face Detection:**
  - Two different face detection methods, MTCNN and OpenCV, have been implemented. MTCNN is generally more accurate and can detect faces at various scales, angles, and lighting conditions. OpenCV, while faster, might not be as accurate in complex scenarios.
- **Mask Detection:**
  - The mask detection model is built upon a MobileNetV2 architecture, known for its efficiency and performance. The model is adapted to classify faces into 'with mask', 'without mask', and 'mask worn incorrectly'.
- **EDA:**
  - The exploratory data analysis is comprehensive, involving a detailed inspection of the dataset, counting the number of labels per class, and preparing a structured DataFrame for efficient data manipulation and retrieval.





RECOMMENDATION

Recommendations:

Data Augmentation:
Consider employing advanced data augmentation techniques to increase the dataset's diversity and improve the model's generalization capabilities.

Hyperparameter Tuning:
Experiment with different hyperparameters and architectures for the mask detection model to enhance accuracy and reduce overfitting.

Evaluation Metrics:
Implement a detailed evaluation metric system, including precision, recall, F1-score, and ROC curves, to have a comprehensive understanding of the model's performance.

Real-time Application:
Explore the possibilities of deploying the model in a real-time application, considering the efficiency and speed of the face detection and mask classification processes.

User Interface:
Develop a user-friendly interface for real-time mask detection, allowing users to easily interact with and utilize the model for safety and compliance monitoring.

Click here to view more detail documentation:

or:

https://github.com/shamustappa/FaceMask-CNN_Detection