Mask detection system

Chosen Field and Motivation

The chosen field is Object Detection in computer vision, with a specific focus on face mask detection. The motivation for this project is the global health crisis due to COVID-19, which has heightened the need for automated systems that can detect whether people are wearing masks in public spaces. Face mask detection can play an important role in ensuring public health safety and compliance with regulations in various environments, including airports, stores, schools, and offices.

Dataset and Why it was Selected

The dataset used in this project consists of images and corresponding annotations in XML format. The dataset includes three classes:

1. **With Mask**
2. **Without Mask**
3. **Mask Worn Incorrectly**

The dataset was selected because it directly addresses the need for automated mask detection, a problem that has gained relevance due to the pandemic. Additionally, it is structured to include labeled bounding boxes around faces, which is crucial for object detection tasks.

Why this dataset?

- **It includes diverse scenarios of face mask usage.**
- **It contains enough labeled data to train a deep learning model.**
- **The dataset provides ground truth bounding boxes, which are essential for training an object detection model.**

Detailed Explanation of Each Step in the Pipeline

1. **Data Loading and Preprocessing**
    - **Input: Raw image files (PNG format) and corresponding annotations in XML format (PASCAL VOC format).**
    - **Process:**
        - **The images are read using OpenCV.**
        - **Annotations (bounding boxes and class labels) are extracted from XML files using the xml.etree.ElementTree module.**
        - **Each image is resized to a target size of 224x224 pixels and normalized (pixel values scaled between 0 and 1).**
    - **Output: Preprocessed images and normalized bounding boxes.**
2. **Dataset Splitting**
    - **Input: The preprocessed images and their associated labels.**
    - **Process: The dataset is split into training and validation sets using train_test_split from sklearn.model_selection.**
    - **Output: Training and validation sets for both images and labels.**
3. **Model Building**
    - **Input: The training dataset (images and labels).**
    - **Process: A Convolutional Neural Network (CNN) architecture is used for feature extraction. The model consists of multiple convolutional layers, followed by max-pooling, and a fully connected layer to predict both the class and bounding box coordinates of the object.**
    - **Output: The model architecture, which can predict the class (mask or no mask) and the bounding box (coordinates of the face) for each image.**

4. **Model Training**
    - **Input: Training images and corresponding class labels and bounding box labels.**
    - **Process: The model is trained using the Adam optimizer and a combination of sparse categorical cross-entropy loss for class prediction and mean squared error (MSE) for bounding box regression.**
    - **Output: Trained model with learned weights.**
5. **Model Evaluation and Visualization**
    - **Input: Validation dataset.**
    - **Process: The trained model makes predictions on the validation set. The predicted bounding boxes and classes are compared with the true values to evaluate the model's performance.**
    - **Output: Visualizations that show predictions overlaid on the original images, with bounding boxes for detected faces.**

Input and Output of Each Step

1. **Data Loading and Preprocessing**
    - **Input: Raw image files and XML annotations.**
    - **Output: Preprocessed images (224x224, normalized) and bounding boxes (normalized).**
2. **Dataset Splitting**
    - **Input: Preprocessed images and labels.**
    - **Output: Training and validation sets for images, class labels, and bounding boxes.**
3. **Model Building**

- **Input: Preprocessed images and labels (from the training set).**
- **Output: A trained object detection model.**

4. **Model Training**
    - **Input: Training images, class labels, bounding box labels.**
    - **Output: Trained model.**
5. **Model Evaluation and Visualization**
    - **Input: Validation images, predicted bounding boxes, and class labels.**
    - **Output: Visualized images showing predicted vs. true bounding boxes and class labels.**

Techniques and Justification for Each Technique's Selection

1. **Convolutional Neural Networks (CNNs):**
    - **Justification: CNNs are the go-to technique for image classification and object detection tasks. Their ability to automatically learn spatial hierarchies of features (edges, textures, shapes, etc.) makes them ideal for face mask detection in images.**

2. Bounding Box Regression:

- **Justification: Object detection tasks not only require classifying objects (faces, in this case) but also locating them within the image using bounding boxes. Using bounding box regression with a sigmoid activation function**

allows the model to predict the coordinates of the face in the image.

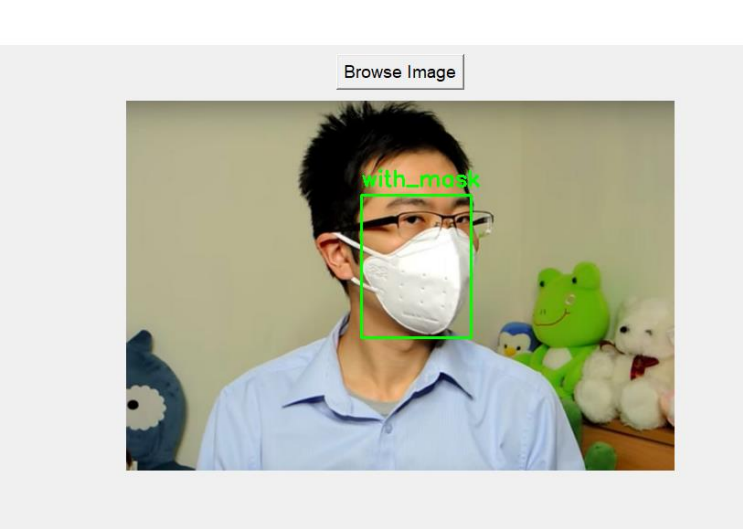
3. **Cross-Entropy Loss for Classification:**
    - **Justification: Cross-entropy loss is widely used for classification tasks as it measures the difference between the true class labels and the predicted probabilities.**
4. **Mean Squared Error (MSE) for Bounding Box Regression:**
    - **Justification: MSE is used here because it effectively penalizes deviations between the predicted bounding box coordinates and the ground truth, helping the model improve the precision of object localization.**

Visual Examples of Detection Output

Here are visual examples of the detection output from the model:

- **Predicted output: A green bounding box representing the predicted face with a label indicating whether the person is wearing a mask.**

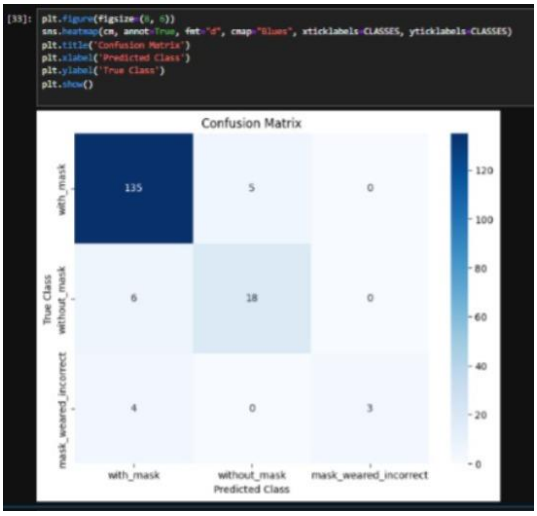
- **Ground truth: A red bounding box indicating the true position of the face.**



Evaluation Metrics and Discussion

Evaluation Metrics:

1. **Accuracy: This metric is used for classification to measure the percentage of correct predictions.**
2. **Precision, Recall, and F1-Score: These metrics are critical in evaluating classification models, especially when dealing with imbalanced classes (e.g., more "with_mask" than "without_mask").**
    - **Precision measures how many of the predicted positive cases (e.g., "with_mask") are actually correct.**
    - **Recall measures how many of the actual positive cases (e.g., "with_mask") were correctly identified by the model.**
    - **F1-Score is the harmonic mean of precision and recall, giving a balance between the two.**
3. **Intersection over Union (IoU): This metric is used to evaluate the accuracy of the predicted bounding boxes by comparing the overlap between predicted and ground truth boxes. A higher IoU indicates better localization of objects.**



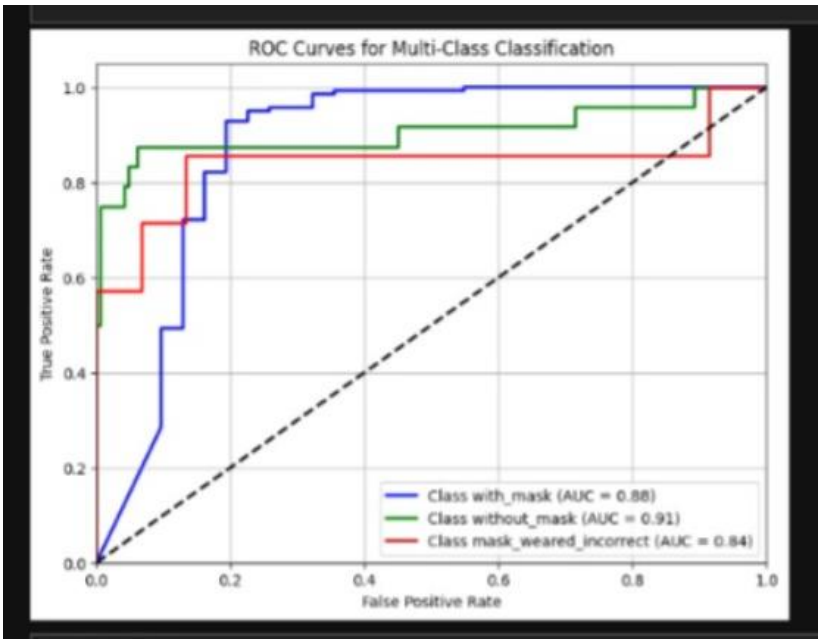
```
[17]: from sklearn.metrics import classification_report

      print("Classification Report:")
      print(classification_report(cls_test, pred_class, target_names=CLASSES))

      Classification Report:
                           precision    recall  f1-score   support

                with_mask       0.93      0.96      0.95       140
             without_mask       0.78      0.75      0.77        24
mask_weared_incorrect       1.00      0.43      0.60         7

                 accuracy                           0.91       171
                macro avg       0.90      0.71      0.77       171
             weighted avg       0.91      0.91      0.91       171
```



Limitations or Challenges Faced

1. **Imbalanced Dataset: The dataset may have an imbalance in the number of samples for each class, leading to biases in the model. For example, "with_mask" might have more images than "without_mask," which could affect the model's performance for less represented classes.**
2. **Bounding Box Regression Accuracy: The accuracy of predicting bounding boxes can be difficult to perfect, especially when faces are occluded or at different angles.**
3. **Overfitting: With limited data and complex models, the model might overfit to the training data and fail to generalize well to unseen images. This can be mitigated by using data augmentation, dropout, or early stopping techniques.**

4. **Face Detection in Different Lighting/Angles: Variations in lighting, face orientation, and occlusions could impact the model's ability to detect faces and their corresponding masks.**

References :

Tensorflow documentation

Opencv documentation

dataset : https://www.kaggle.com/datasets/andrewmvd/face-mask-detection