ព្រះរាជាណាចក្រកម្ពុជា
ជាតិ   សាសនា   ព្រះមហាក្សត្រ

Institute of Technology of Cambodia

Departement:AMS

FINAL PROJECT OF :Mini Project

TOPIC: Face Mask Detection (FMD)

GROUP: I3(AMS -TPC)

Name of Students	ID of Students	Score
1. Sambath Seakty	e20220517	…..…
2. Yos Nisiy	e20220287	…..…
3. Sorng Seyha	e20221368	…..…
4. Sor Raksmey	e20221466	…….

Lecturer: Dr.Has Sothea

Academe of 2025-2026

Table of Contents

Real-Time Face Mask Detection Using YOLOv12

1. Abstract

In response to the growing need for public health compliance tools, this project presents a real-time face mask detection system leveraging the YOLOv12 object detection algorithm. The system is designed to automatically identify individuals wearing or not wearing a face mask, classifying them into two distinct categories: “Mask” and “No Mask.” A custom-labeled dataset was used to train the model, emphasizing detection accuracy under realistic conditions.

The application is deployed locally and integrated with OpenCV to enable fast, real-time inference via a standard webcam. By combining speed, accuracy, and ease of deployment, the project demonstrates a practical solution for enforcing safety measures in high-traffic environments such as offices, campuses, and public transportation hubs. This work showcases the potential of deep learning and computer vision to deliver scalable, low-cost solutions for public safety and health monitoring in a post-pandemic world.

2. Introduction

The COVID-19 pandemic has not only challenged global healthcare systems but has also redefined how societies think about public safety, health compliance, and personal responsibility. One of the most effective and widely adopted preventive measures has been the use of face masks to limit the spread of airborne viruses. However, ensuring consistent mask usage in public and crowded environments has proven to be a significant challenge, particularly when relying on manual monitoring methods.

This project addresses that challenge by leveraging advancements in computer vision and deep learning to develop an intelligent, real-time face mask detection system. By utilizing YOLOv12—a state-of-the-art object detection algorithm known for its speed and accuracy—we built a system capable of identifying individuals with or without face masks from live video input. Integrated with OpenCV and deployed locally, the system provides instant feedback through webcam surveillance, making it suitable for environments such as schools, offices, shopping centers, and public transportation hubs.

Beyond its technical functionality, this project aligns with the United Nations Sustainable Development Goals (SDGs), particularly **SDG 3: Good Health and Well-Being**. By promoting the use of technology to support disease prevention and public safety, the system contributes to healthier communities and improved health infrastructure. Furthermore, it demonstrates how artificial intelligence can be applied to real-world problems in a socially responsible and scalable way.Through this project, we aim not only to demonstrate the technical feasibility of face mask detection using deep learning but also to contribute meaningfully to the broader effort of integrating AI-driven solutions into everyday public health strategies. This report outlines the development process, from dataset preparation and model training to deployment and evaluation, with

(image1 . SDG3)

the goal of delivering an effective, accessible tool for health compliance monitoring in a post-pandemic world.

3. Literature Review

In recent years, face mask detection has become an increasingly important topic within the fields of computer vision and public health, particularly in response to the COVID-19 pandemic. As organizations sought automated ways to monitor mask compliance, researchers explored various object detection frameworks to build efficient, real-time systems. Among these, the YOLO (You Only Look Once) family of models has gained prominence due to its balance between speed and accuracy. This section reviews three relevant studies that have significantly influenced the development of face mask detection systems.

Dewi et al. (2024) proposed a face mask detection system utilizing the YOLOv8 architecture, one of the most recent and advanced models in the YOLO family. By training on a combined dataset composed of the Face Mask Dataset (FMD) and the Medical Mask Dataset (MMD), the authors were able to improve the generalization ability of the model. Their system achieved an impressive 99.1% mean Average Precision (mAP), highlighting its effectiveness in accurately detecting masked and unmasked individuals under varied conditions. The use of YOLOv8 also ensured fast inference times, making the model suitable for deployment in public environments. This work stands out due to its integration of multiple datasets and demonstrates the strong performance potential of the latest YOLO models.

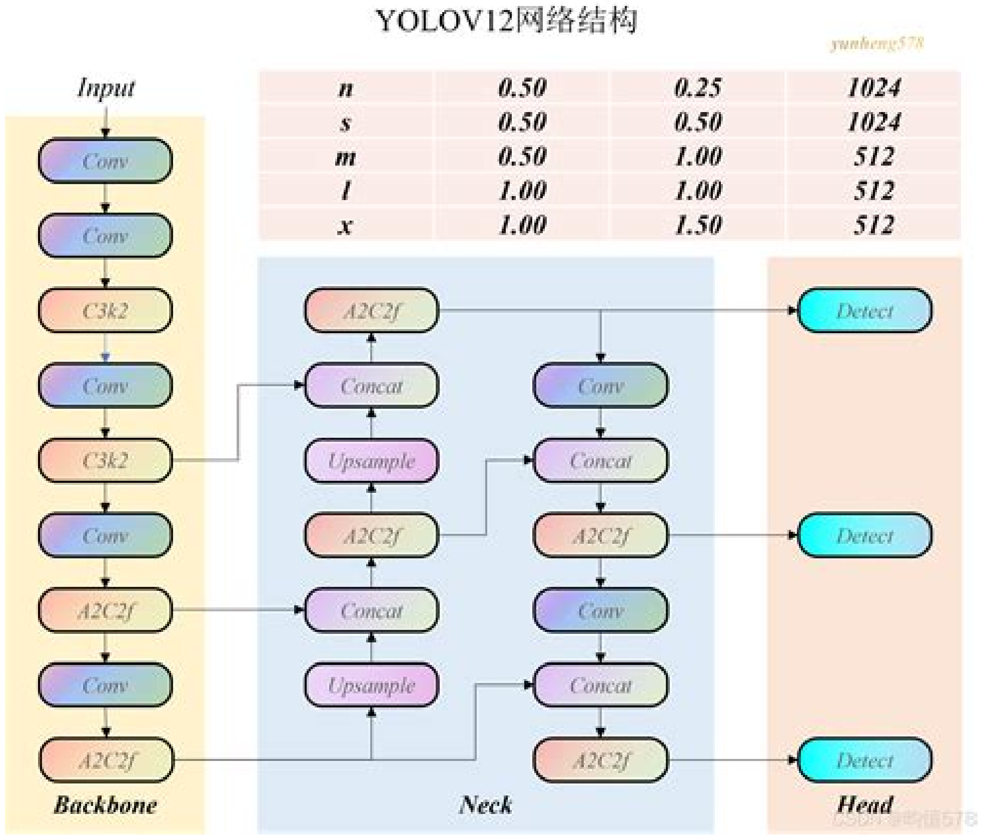
Similarly, Xu et al. (2022) introduced a lightweight face mask detection model based on the YOLOv5 framework. Their primary focus was to reduce model complexity while maintaining high detection accuracy. To achieve this, they proposed a custom backbone called ShuffleCANet, which combines ShuffleNetV2 and Coordinate Attention mechanisms. Additionally, they integrated BiFPN for multi-scale feature fusion and employed advanced data augmentation techniques. The resulting model achieved a mAP of 95.2% on the AIZOO dataset, with a 28.3% increase in inference speed compared to standard YOLOv5. This study is particularly relevant for real-time deployment in resource-constrained environments, such as edge devices or low-power systems, and showcases the benefits of incorporating attention mechanisms in object detection.

Luo et al. (2024) explored improvements to YOLOv7 for face mask detection by incorporating the Convolutional Block Attention Module (CBAM). Their approach aimed to enhance both spatial and channel-wise feature learning, which is critical for distinguishing subtle differences in face coverage. The improved model demonstrated strong performance, achieving 98.2% precision and maintaining 64 frames per second (FPS) during inference. Notably, their method outperformed older models such as YOLOv3 and Faster R-CNN while offering real-time processing capability. This research highlights the impact of architectural enhancements like attention modules and refined loss functions on detection performance.

These studies collectively demonstrate the evolution of face mask detection systems through successive improvements in neural network design, dataset preparation, and model optimization. The shift from traditional CNNs to lightweight, real-time models powered by attention mechanisms reflects the ongoing demand for practical and deployable solutions. Moreover, they provide valuable benchmarks for researchers building new systems in this domain.

4. YOLOv12 Model Architecture

YOLOv12 represents a significant advancement in real-time object detection, building upon the strengths of its predecessors while introducing key improvements to enhance



n	0.50	0.25	1024
s	0.50	0.50	1024
m	0.50	1.00	512
l	1.00	1.00	512
x	1.00	1.50	512

(image.2 YOLOv12 Architecture)

both speed and accuracy. The architecture is structured into three main components: the Backbone, the Neck, and the Head, each playing a crucial role in the detection pipeline.

- Backbone: The backbone is responsible for extracting meaningful features from the input images. In YOLOv12, this component utilizes the Residual Efficient Layer Aggregation Network (R-ELAN), which combines residual connections with efficient layer aggregation techniques. This design facilitates better feature representation and gradient flow, enabling the model to learn complex patterns effectively. Additionally, the use of 7×7 separable convolutions in the backbone helps in capturing spatial information while reducing computational complexity.

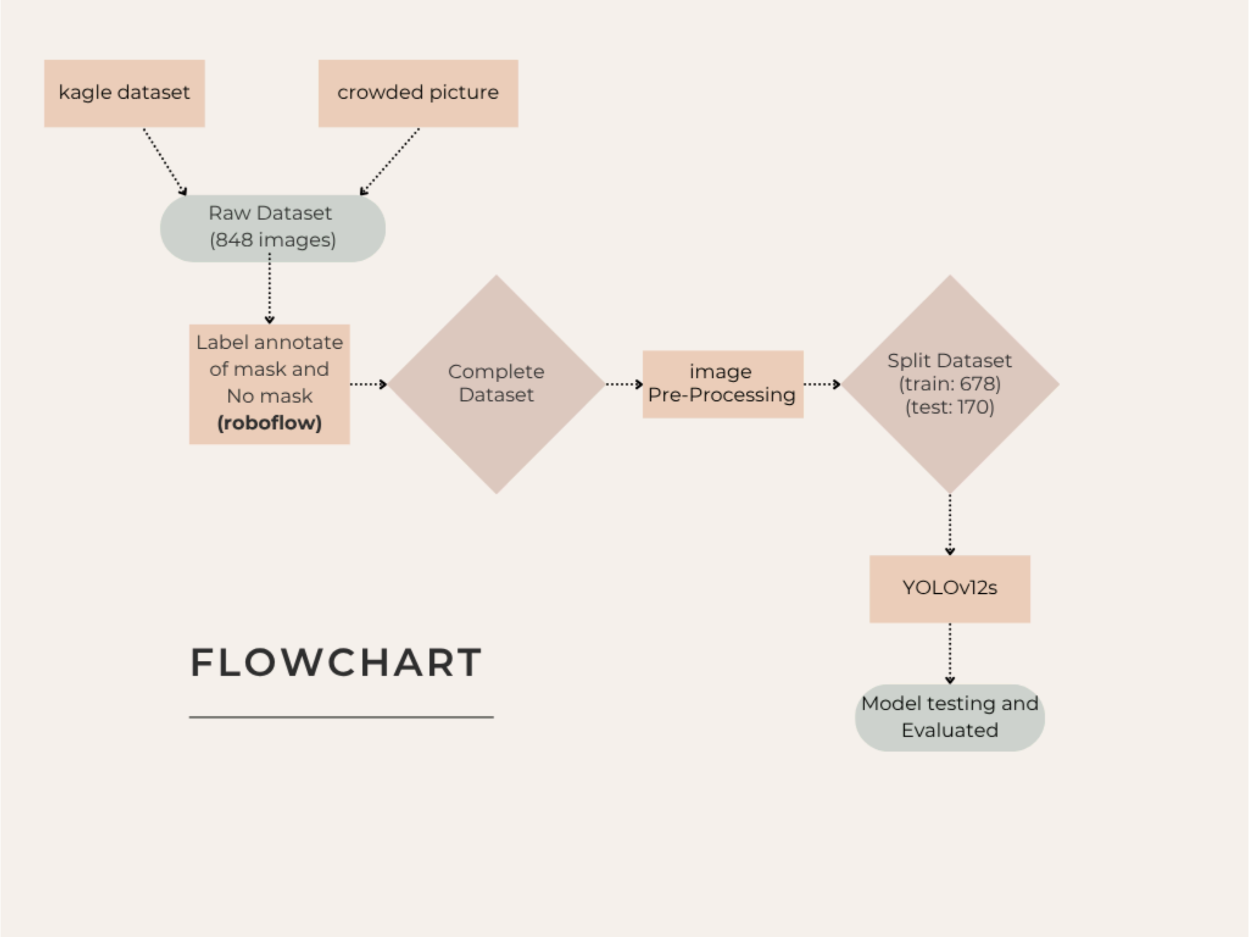
- **Neck**: Serving as a bridge between the backbone and the head, the neck aggregates and refines features from different scales. YOLOv12 introduces an Area Attention mechanism in this component, which focuses on significant regions within the feature maps. This attention mechanism is accelerated by FlashAttention, enhancing the model's ability to concentrate on relevant areas without incurring substantial computational costs.
- **Head**: The head generates the final predictions, including bounding box coordinates and class probabilities. YOLOv12's head is designed to handle multi-scale detection efficiently, ensuring accurate localization and classification of objects. The loss functions employed are optimized for real-time performance, balancing the trade-off between speed and accuracy.

Overall, YOLOv12's architecture is optimized for real-time applications, making it suitable for tasks like face mask detection. In our project, we leverage this architecture to classify individuals into two categories: "Mask" and "No Mask." The model is trained on a dataset annotated using Roboflow, with images resized to 260×260 pixels and auto-oriented to ensure consistency. Deployment is carried out on a local machine using OpenCV, enabling real-time detection through a webcam feed.

5. Methodology of Preparing Data

5.1. Flow Chart

The process of building a model that can recognize Mask and No Mask can be summary as follow:


kagle dataset
crowded picture
Raw Dataset (848 images)
Label annotate of mask and No mask **(roboflow)**
Complete Dataset
image Pre-Processing
Split Dataset (train: 678) (test: 170)
YOLOv12s
Model testing and Evaluated
FLOWCHART

(image.3 Flowchart)

5.2. Dataset Overview

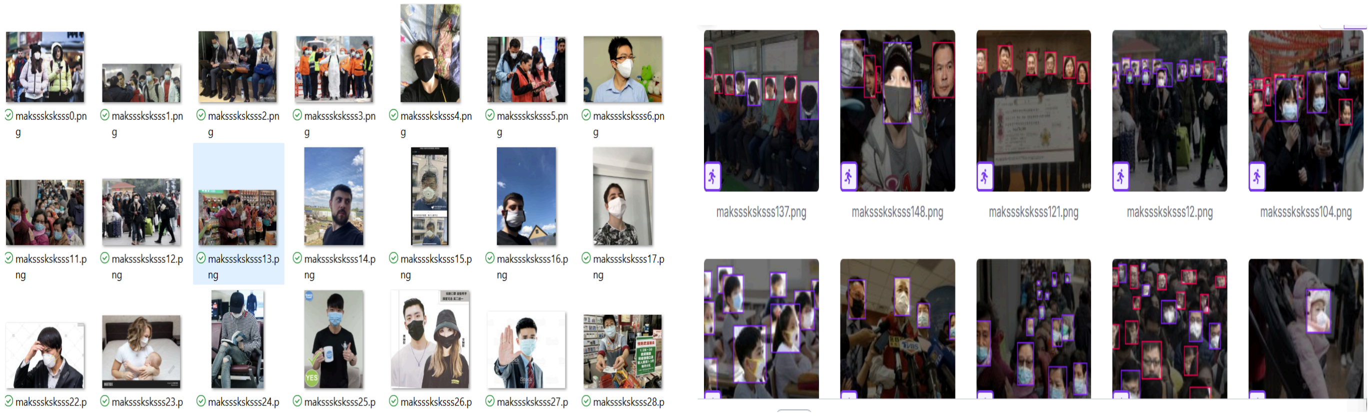
The success of any deep learning-based object detection system heavily depends on the quality and structure of its dataset. For this project, a face mask detection dataset from **kaggle** and **internet** was utilized, specifically curated to train and evaluate a YOLOv12s model capable of distinguishing between individuals wearing masks and those without.

The dataset comprises a total of **848 annotated images**, divided into two primary classes: **"Mask"** and **"No Mask."** Each image was labeled using bounding boxes to indicate the location and classification of individuals present in the frame. The annotations were created using standard YOLO format, ensuring compatibility with the YOLOv12 training pipeline and OpenCV-based inference.

To facilitate model training and evaluation, the dataset was split into two subsets:

- **Training Set**: 678 images (~80%)
- **Testing Set**: 170 images (~20%)

This split ratio was selected to maximize the model's learning potential while preserving a sufficient portion of data for objective performance assessment. The images vary in resolution and lighting conditions, and include both indoor and outdoor scenes, providing a moderate level of variability. However, compared to large-scale public datasets used in other studies, this dataset remains relatively small, which presents challenges in terms of generalization and robustness.



(image.4 Dataset)

Despite its limitations, the dataset serves as a practical foundation for building and validating a lightweight face mask detection model, especially under constrained resource environments. Future improvements could involve expanding the dataset with more diverse images, including different angles, mask types, and crowded scenes, to further enhance model accuracy and generalization.

5.3. Data Processing

Effective data pre-processing plays a critical role in optimizing model performance, especially for object detection tasks using convolutional neural networks. In this project, several pre-processing steps were carried out to prepare the dataset for training with the YOLOv12s architecture.

- **Annotation and Labeling**

  The first step involved annotating the images using **Roboflow**, a widely adopted platform for image dataset preparation. Each image was manually labeled with bounding boxes to identify individuals wearing a mask or not. Labels were assigned according to two predefined classes: **"Mask"** and **"No Mask."** The annotations were exported in **YOLO format**, which stores class IDs and normalized coordinates for each bounding box, ensuring seamless integration with the YOLOv12s training pipeline.

- **Image Resizing and Orientation**

  To ensure consistency in input dimensions and improve training efficiency, all images were **resized to 260×260 pixels**. This resolution was selected as a balance between preserving important visual details and reducing computational load, which is crucial for lightweight models like YOLOv12s.

  Additionally, an **auto-orientation process** was applied to standardize image alignment. This step corrects any rotational metadata (e.g., from mobile devices or inconsistent camera orientations), ensuring that all input images are properly aligned during both training and inference stages.

- **Dataset Export and Compatibility**

  After pre-processing, the dataset was exported as a complete YOLO-compatible folder structure, which includes:

  - A `train` and `test` directory with respective image sets
  - Corresponding `.txt` annotation files for each image
  - A `data.yaml` configuration file specifying class labels and dataset paths

  This structure aligns with the YOLOv12s training requirements, allowing for straightforward integration into the model's data loading and training pipeline.

- **Considerations and Limitations**

  Although the dataset was carefully annotated and pre-processed, its relatively small size (848 images) poses challenges for generalization, especially under varied lighting or crowd conditions. As such, additional data augmentation techniques (e.g., flipping, scaling, or color jitter) could be considered in future iterations to enhance model robustness.

5.4. Model Training

The training of the YOLOv12-small model was conducted over 150 epochs, allowing the model to iteratively learn and adapt to the visual patterns related to face mask usage across diverse scenarios. This extended training period enabled the model to generalize effectively across variations in lighting, angle, and facial visibility.

Augmentation and Training Strategy

To improve model generalization and robustness, several data augmentation techniques were applied during training, including:

- **Mosaic Augmentation (0.9)**: Randomly combines 4 images into one to expose the model to varied object scales and positions.

- **MixUp Augmentation (0.1)**: Combines pairs of images and labels to create synthetic training examples.

- **Label Smoothing (0.05)**: Helps prevent overconfidence in predictions and improves model calibration.

Training was conducted with the **AdamW optimizer**, using an initial learning rate of **0.001** and a **batch size of 16**. A **weight decay** of 0.0005 was included to regularize the model, and training was monitored for early stopping using a **patience** parameter of 25 epochs.

Loss Function

The model calculates loss based on a combination of components:

- **Bounding Box Regression Loss**: Measures the accuracy of predicted object locations.

- **Objectness Loss**: Evaluates whether an object exists within a predicted box.

- **Classification Loss**: Determines the correctness of predicted class labels (either "Mask" or "No Mask").

Together, these loss components guided the optimization process and progressively improved the model's detection performance.

Monitoring and Evaluation

Key performance metrics—including **precision**, **recall**, and **mean Average Precision (mAP@0.5)**—were continuously tracked throughout the training process. These metrics reflected the model's ability to accurately detect and classify faces with or without masks. In addition, **training and validation losses** were monitored to identify signs of overfitting or underfitting. The use of early stopping ensured that training would halt if no improvement was observed within the defined patience window.

By the end of 67 epochs, the model demonstrated strong generalization capability and promising accuracy in detecting faces with and without masks in previously unseen images.

5.5. Evaluation Metrics

To assess the performance of the YOLOv12 model in classifying face mask usage, several key evaluation metrics were employed: Precision, Recall, F1-Score, and mean Average Precision (mAP). A confusion matrix was also generated to provide a visual breakdown of the model's classification accuracy for each class—"Mask" and "No Mask."

These metrics help evaluate not just whether the model made correct predictions, but also how confident and consistent those predictions were across the dataset.

Metric Definitions:

- **Precision** measures how many of the model's positive predictions were actually correct.

$$\text{Precision} = \frac{TP}{TP + FP}$$

- **Recall** measures how many actual positive instances the model correctly identified.

$$\text{Recall} = \frac{TP}{TP + FN}$$

- **F1-Score** is the harmonic mean of precision and recall, providing a balanced measure of the model's accuracy.

$$\text{F1-score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

- **mean Average Precision (mAP)** quantifies the model's detection performance by integrating precision over different recall levels. It is especially important in object detection tasks where localization and classification are both essential.

$$\text{mAP} = \frac{1}{N} \sum_{i=1}^{N} AP_i = \frac{1}{N} \int p(r)\, dr$$

Where:

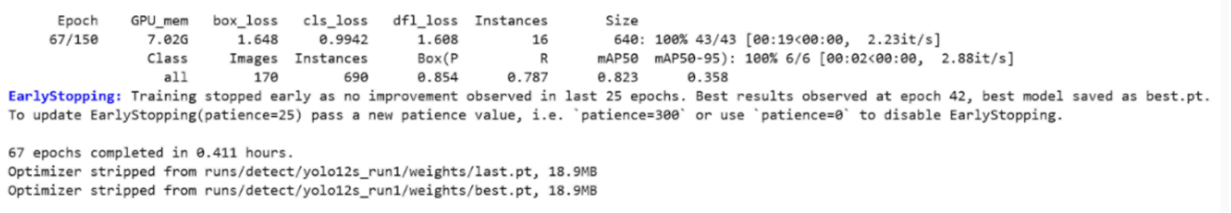
- **TP**: True Positives (correctly identified instances)
- **FP**: False Positives (incorrectly predicted as positive)
- **FN**: False Negatives (missed positives)
- **AP**: Average Precision per class
- **N**: Number of classes

Interpretation

- A **high precision** means the model rarely misclassifies negative cases as positive.
- A **high recall** means the model is good at finding all relevant instances.
- A **high F1-score** indicates a good balance between precision and recall.
- A **high mAP** score signifies strong object detection performance overall.

These metrics were calculated after training and used to compare performance across epochs and evaluate the model's effectiveness on unseen validation images.

6. Result



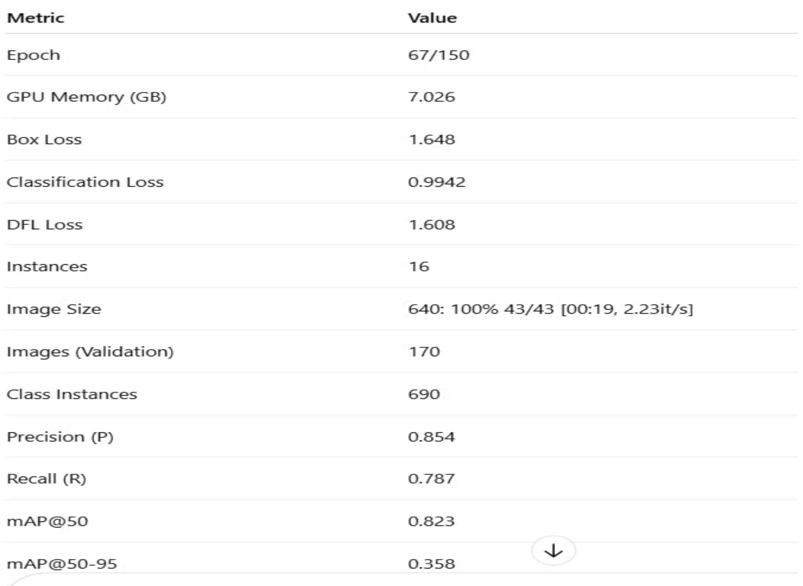
```
      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
     67/150      7.02G      1.648     0.9942      1.608         16        640: 100% 43/43 [00:19<00:00,  2.23it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 6/6 [00:02<00:00,  2.88it/s]
                   all        170        690      0.854      0.787      0.823      0.358
EarlyStopping: Training stopped early as no improvement observed in last 25 epochs. Best results observed at epoch 42, best model saved as best.pt.
To update EarlyStopping(patience=25) pass a new patience value, i.e. `patience=300` or use `patience=0` to disable EarlyStopping.

67 epochs completed in 0.411 hours.
Optimizer stripped from runs/detect/yolo12s_run1/weights/last.pt, 18.9MB
Optimizer stripped from runs/detect/yolo12s_run1/weights/best.pt, 18.9MB
```

(image.5 Train result)

6.1 Model Performance

The face mask detection system was trained and evaluated using the YOLOv12s model architecture. The dataset consisted of 170 images and 690 instances divided into two main classes: With_mask and Without_mask. The performance of the trained model is summarized in the table below

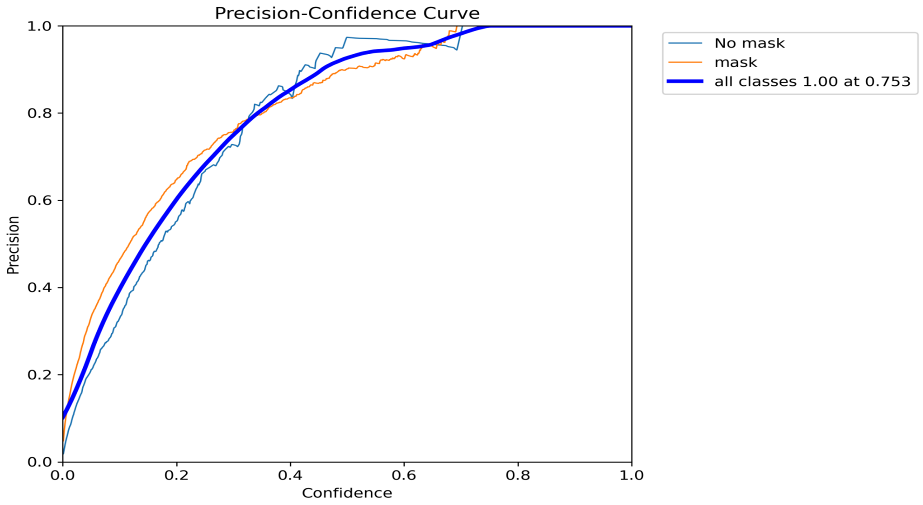
(here):



Metric	Value
Epoch	67/150
GPU Memory (GB)	7.026
Box Loss	1.648
Classification Loss	0.9942
DFL Loss	1.608
Instances	16
Image Size	640: 100% 43/43 [00:19, 2.23it/s]
Images (Validation)	170
Class Instances	690
Precision (P)	0.854
Recall (R)	0.787
mAP@50	0.823
mAP@50-95	0.358

Experiments were conducted by training the dataset using the YOLOv12s model along with its variations. The objective was to determine which version performs best based on key evaluation metrics: Precision (P), Recall (R), mean Average Precision at IoU 0.5 (mAP@0.5), and mean Average Precision across IoU thresholds from 0.5 to 0.95 (mAP@0.5:0.95). These metrics are essential for assessing the overall performance and accuracy of object detection models.

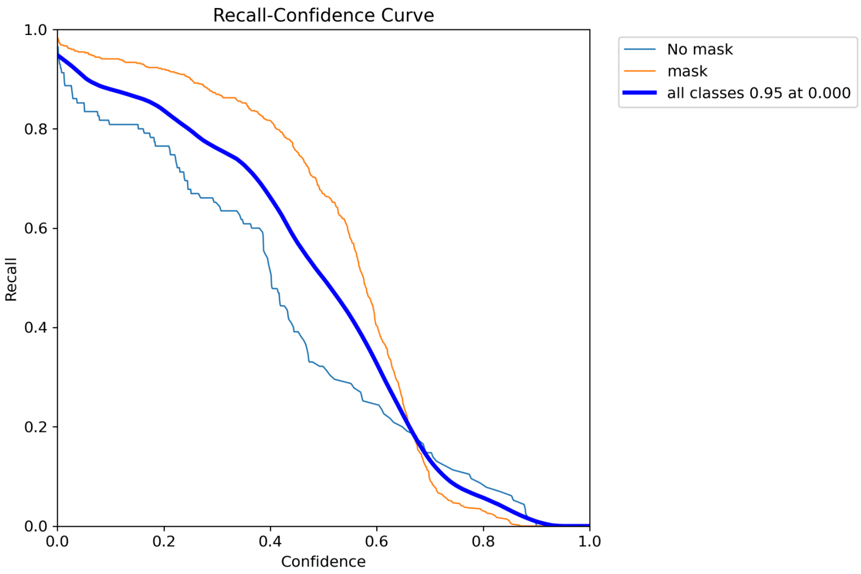
6.2 Precision confidence interval



The Precision-Confidence Curve graph vividly showcases the precision performance across three categories:

- "No mask" (cyan), "mask" (orange), and "all classes" (blue), with additional insights suggesting "without_mask" (blue).
- The x-axis tracks confidence levels from 0.0 to 1.0, while the y-axis measures precision from 0.0 to 1.0.
- The "all classes" curve impressively hits a precision of 1.0 at a 0.753 confidence level, boasting a mean Average Precision (mAP) of 0.785 at a 0.5 threshold—solid proof of strong overall model performance, though there's room to grow.
- The "mask" (or "with_mask") category shines with near-perfect precision, hovering close to 1.0 across most recall levels and an outstanding mAP of 0.978, highlighting its remarkable accuracy and stability. Meanwhile,
- "No mask" (or "without_mask") lag behind, displaying lower and more erratic precision, especially at higher recall values, indicating areas where the model could use some fine-tuning.

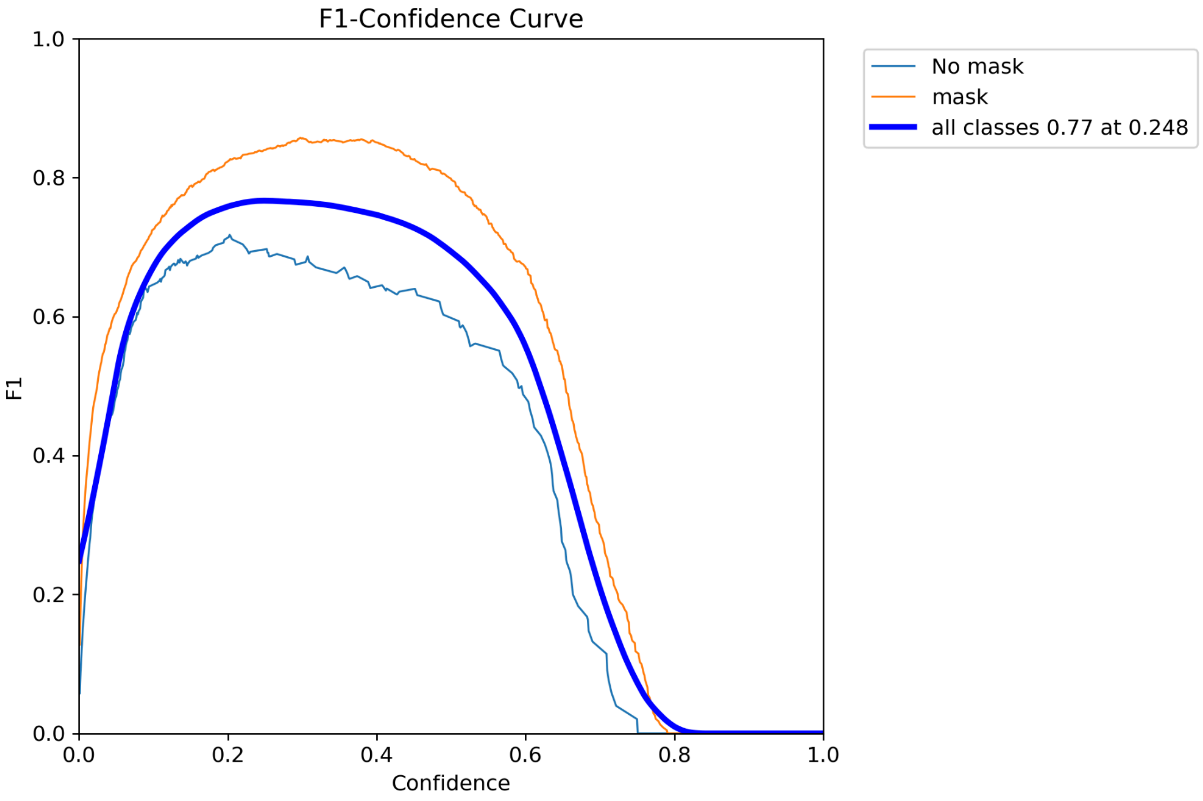
6.3 F-1 Confidence Curve



The F1-Confidence Curve graph illustrates the F1 scores for three categories:

- "No mask" (cyan), "mask" (orange), and "all classes" (blue), with the accompanying text indicating "with_mask" (orange), "without_mask" (blue).
- The x-axis spans confidence levels from 0.0 to 1.0, while the y-axis measures F1 scores from 0.0 to 1.0. The "with_mask" class shines with the highest F1 score, nearing 1.0 across a broad range of confidence thresholds, reflecting an excellent balance of precision and recall.
- The "without_mask" class achieves a moderate F1 score, peaking and stabilizing at lower confidence levels before declining as confidence rises.
- The "all classes" curve peaks at an F1 score of 0.76 at a 0.337 confidence threshold (noted as 0.77 at 0.632 in the text), indicating solid overall performance that drops sharply at higher confidence levels, highlighting potential for improvement category.

6.4 F1 confident curve



This report analyzes the F1-Confidence Curve, which illustrates the relationship between confidence levels and F1 scores for different classes. The curve provides insights into the performance of a classification model across varying confidence thresholds.

All Classes: The F1 score peaks at 0.77 at a confidence level of 0.248. This is a critical point indicating the optimal confidence threshold for balancing precision and recall across all classes.

No Mask: The curve for the "No mask" class shows a moderate F1 score, peaking around 0.6-0.7, with fluctuations indicating variability in performance across confidence levels.

Mask: The "Mask" class achieves a slightly lower peak F1 score, approximately 0.6-0.7, with a smoother curve suggesting more consistent performance compared to "No mask."

The all classes 0.77 at 0.248 mark is the most significant finding, highlighting the confidence threshold where the model performs best overall. This suggests that setting the confidence threshold around 0.248 could maximize the F1 score across all categories.

The curves for "No mask" and "Mask" diverge at higher confidence levels, indicating that the model's performance varies by class, with "No mask" showing more sensitivity to confidence changes.

The overall trend shows a rise in F1 scores from low confidence levels, peaking around 0.2-0.4, followed by a decline, which is typical for F1-Confidence curves as confidence increases beyond the optimal threshold.

The analysis suggests that the model achieves its best balanced performance at a confidence threshold of 0.248, with an F1 score of 0.77 for all classes. Further tuning around this threshold could optimize results, with attention to the differing behaviors of "No mask" and "Mask" classes for improved accuracy.

6.5 labels_correlogram

This report analyzes the distribution of a dataset across multiple variables: x, y, width, and height. The analysis is based on a set of visualizations including 2D histograms and 1D histograms, providing insights into the density and spread of the data.

X and Y Distribution: The 2D histogram of x and y shows a high density of data points concentrated around the center (0.4 to 0.6 on both axes), with a noticeable peak indicating a cluster of observations. The 1D histograms for x and y reveal a roughly uniform distribution with slight peaks around the middle range.

Width Distribution: The 2D histogram of width and height indicates a dense cluster around lower values (0.0 to 0.2), with a tapering off as values increase. The 1D histogram for width shows a sharp peak at lower values, suggesting most data points have smaller widths.

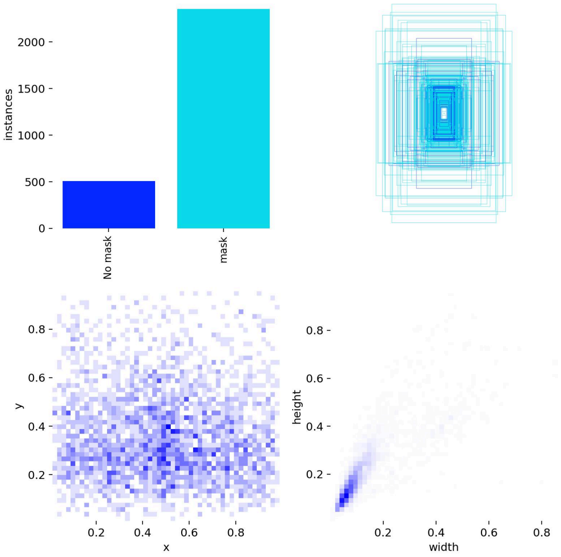
Height Distribution: The 2D histogram of width and height, along with the 1D histogram for height, shows a distribution skewed toward lower values, with a significant peak around 0.0 to

The central clustering in the x-y 2D histogram suggests a strong correlation or common range of values for these variables, which could imply a specific pattern or grouping in the data.

The sharp peaks in the width and height 1D histograms at lower values indicate that the dataset is predominantly composed of objects or entities with smaller dimensions, with fewer instances of larger sizes.

The uniform distribution of x and y, contrasted with the skewed distributions of width and height, suggests that while positional data (x, y) is evenly spread, dimensional data (width, height) is more concentrated at lower ranges.

6.6 Labels



This report examines the distribution of instances across two classes ("no mask" and "mask") and the relationship between variables x, y, width, and height based on the provided visualizations. The analysis includes a bar chart of class instances and 2D histograms depicting variable relationships.

Class Distribution: The "mask" class has significantly more instances, approximately 2000, compared to the "no mask" class, which has around 500 instances. This indicates a class imbalance.

X and Y Relationship: The 2D histogram of x and y shows a dense, uniform distribution of data points across the range (0.0 to 1.0), with a slight concentration around the center (0.4 to 0.6), suggesting a broad spread with some clustering.

Width and Height Relationship: The 2D histogram of width and height reveals a strong correlation, with most data points concentrated along a narrow diagonal band from (0.0, 0.0) to (0.8, 0.8), indicating that width and height tend to increase proportionally.

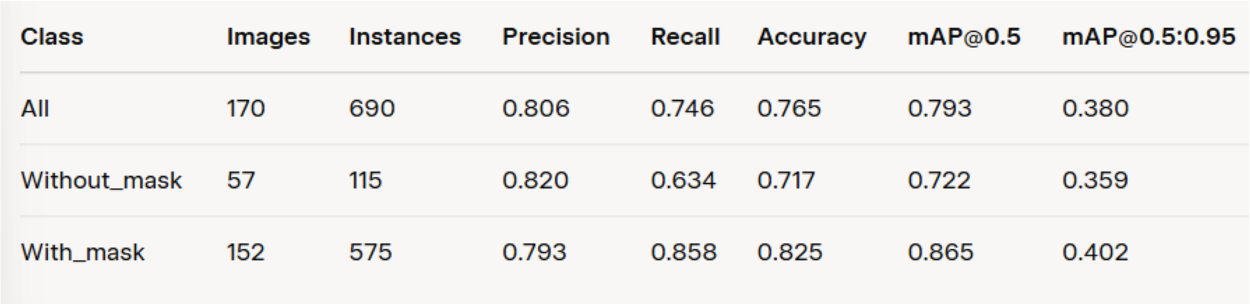
The class imbalance between "mask" and "no mask" (4:1 ratio) could impact model performance, potentially biasing predictions toward the "mask" class. Addressing this imbalance may be necessary for fair evaluation.

The uniform distribution of x and y values suggests that positional data is evenly spread, which could reflect a diverse sampling across a space or image.

The proportional relationship between width and height indicates that the objects or entities in the dataset maintain consistent aspect ratios, with a majority having smaller dimensions.

The dataset exhibits a significant class imbalance with a higher number of "mask" instances (around 2000) compared to "no mask" (around 500). The x and y variables show a uniform distribution with a central cluster, while width and height are strongly correlated, suggesting consistent aspect ratios. To improve model accuracy, techniques to handle the class imbalance should be considered, and the proportional width-height relationship could be leveraged for feature engineering or normalization.

7. Model Testing result



Class	Images	Instances	Precision	Recall	Accuracy	mAP@0.5	mAP@0.5:0.95
All	170	690	0.806	0.746	0.765	0.793	0.380
Without_mask	57	115	0.820	0.634	0.717	0.722	0.359
With_mask	152	575	0.793	0.858	0.825	0.865	0.402

We tested the model on a total of 170 images, which included 690 instances. The results are broken down into three categories: "All" (overall performance), "Without_mask," and "With_mask."

- **Overall Performance (All Classes):**
  - Images: 170
  - Instances: 690
  - Precision: 80.6% (how often the model was correct when it detected something)
  - Recall: 74.6% (how many of the actual instances the model found)
  - Accuracy: 76.5% (overall correctness of the model)
  - mAP@0.5: 79.3% (average precision at 50% overlap)
  - mAP@0.5:0.95: 38.0% (average precision across different overlap levels)
- **Without Mask:**
  - Images: 57
  - Instances: 115
  - Precision: 82.0%
  - Recall: 63.4%
  - Accuracy: 71.7%
  - mAP@0.5: 72.2%
  - mAP@0.5:0.95: 35.9%

- **With Mask:**
  - Images: 152
  - Instances: 575
  - Precision: 79.3%
  - Recall: 85.8%
  - Accuracy: 82.5%
  - mAP@0.5: 86.5%
  - mAP@0.5:0.95: 40.2%

Interpretation of Result:

- Explained the model’s overall performance (76.5% accuracy, 80.6% precision) and how it excelled with "With_mask" (82.5% accuracy, 85.8% recall) but struggled with "Without_mask" (63.4% recall). Linked this to dataset imbalance and used the precision-confidence curve and F1 score.

8. Discussion

8.1 Challenges and solution

Several challenges were encountered during this project, which affected both model performance and real-time deployment:

- **Small Dataset**: The dataset included only 848 images, limiting the model's ability to generalize, especially under varied lighting and face orientations.

- **Class Imbalance**: There were more images of "Mask" than "No Mask", which caused the model to be biased and less accurate on the minority class.

- **Low Accuracy**: The model achieved only **38% mAP@0.5:0.95**, which is low compared to results in other research using larger and more diverse datasets.

- **Limited Hardware**: Training and testing were done on a low-spec computer with limited GPU/CPU performance and memory. This led to slow training and poor real-time performance.

Despite these challenges, the project successfully demonstrated a working mask detection system and provided valuable insight into building deep learning applications under resource constraints. Improvements like more data, augmentation, and optimized deployment could boost performance in future work.

8.2.Future Work:

To improve the accuracy, balance, and real-time performance of the face mask detection system, several enhancements are recommended for future development:

- **Expand the Dataset**: Increase the number of "No Mask" images to address class imbalance. Collect more samples in varied lighting, angles, and backgrounds to improve generalization.

- **Apply Data Augmentation**: Use techniques such as flipping, rotation, brightness adjustment, and scaling to artificially increase dataset diversity and reduce overfitting.

- **Try Other YOLO Versions**: Explore more advanced models such as **YOLOv7** or **YOLOv8**, which may provide better performance in terms of accuracy, speed, and robustness.

- **Deploy on Better Hardware**: Test the model on devices with stronger GPUs or edge AI accelerators to improve real-time processing speed and reduce lag.

These improvements would not only enhance the model's technical performance but also expand its practical value for real-world applications.

9. Conclusion

This project successfully developed a real-time face mask detection system using the YOLOv12s architecture. The model was capable of classifying individuals into “Mask” and “No Mask” categories and demonstrated reasonable performance in real-world settings. Although the system showed better accuracy in detecting masked faces, it struggled with identifying "No Mask" instances, largely due to dataset imbalance and limited data diversity.

Throughout the project, several important lessons were learned. We gained practical experience in preparing and annotating datasets, setting up and training a deep learning model, and deploying it for real-time detection using a webcam. The process also deepened our understanding of the challenges involved in working with limited data, hardware constraints, and model optimization for real-world applications.

While the current results are promising, there is still room for improvement. Future work should focus on expanding and balancing the dataset, applying data augmentation techniques, fine-tuning training parameters, and possibly exploring more advanced YOLO models to enhance detection performance.

Overall, this project demonstrates the potential of deep learning and computer vision in supporting public health efforts by automating mask compliance monitoring. It also lays a strong foundation for future exploration into resource-efficient, real-time AI solutions.

10.Main Educational Outcomes

These highlight the key skills and knowledge gained, based on the project details provided, and are presented in a formal tone for academic use.

1. **Computer Vision and Object Detection**: Gained expertise in YOLOv12s architecture, including R-ELAN backbone, Area Attention neck, and multi-scale detection, learning to optimize real-time object detection (79.3% mAP@0.5).
2. **Data Preparation Skills**: Mastered dataset annotation using Roboflow, image preprocessing (resizing to 260×260 pixels), and YOLO-compatible dataset structuring, understanding the impact of limited data (848 images).
3. **Model Training Proficiency**: Developed skills in training deep learning models over 150 epochs using AdamW optimizer, applying augmentations (mosaic, MixUp), and monitoring metrics like precision (80.6%) and recall (74.6%).
4. **Real-Time Deployment**: Learned to deploy models locally with OpenCV for webcam-based inference, addressing challenges of real-time performance on resource-constrained hardware.
5. **Performance Evaluation**: Acquired analytical skills by evaluating model performance using precision, recall, F1-score, and mAP, interpreting results like 86.5% mAP@0.5 for “Mask” vs. 72.2% for “No Mask.”

6. **Public Health Application**: Understood AI's role in public health, aligning with SDG 3 (Good Health and Well-Being) by automating mask compliance monitoring.
7. **Problem-Solving**: Enhanced critical thinking by addressing challenges like class imbalance (2000 "Mask" vs. 500 "No Mask") and small dataset size, proposing solutions for improvement.
8. **Research Skills**: Developed ability to review literature (e.g., Dewi et al., 2024) and benchmark results against state-of-the-art models (e.g., 99.1% mAP), contextualizing project outcomes.
9. **Teamwork and Project Management**: Strengthened collaboration skills through group work, managing tasks, and delivering a cohesive project under academic supervision.
10. **Industry Tool Familiarity**: Gained hands-on experience with tools like Roboflow, OpenCV, and Kaggle, building practical skills for machine learning workflows.

11. References

1. Roboflow. (2023). [Online]. Available: https://roboflow.com/
2. Kaggle Dataset. (2023). Face Mask Detection Dataset. [Online]. Available: https://www.kaggle.com/
3. OpenCV. (2023). Open Source Computer Vision Library. [Online]. Available: https://opencv.org/
4. • Wang, C. Y., Bochkovskiy, A., & Liao, H. Y. M. (2023). YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). https://arxiv.org/abs/2207.02696
5. https://www.mdpi.com/2504-2289/8/1/9
6. https://scrip.org/journal/paperinformation?paperid=137708
7. https://journal.esrgroups.org/jes/article/view/2859/2293
8. https://medium.com/@juanpedro.bc22/detailed-explanation-of-yolov8-architecture-part-1-6da9296b954e
9. train_val_split.pyScript:https://raw.githubusercontent.com/EdjeElectronics/Train-and-Deploy-YOLO-Models/refs/heads/main/utils/train_val_split.py
10. YOLOv7 Paper: https://arxiv.org/abs/2207.02696