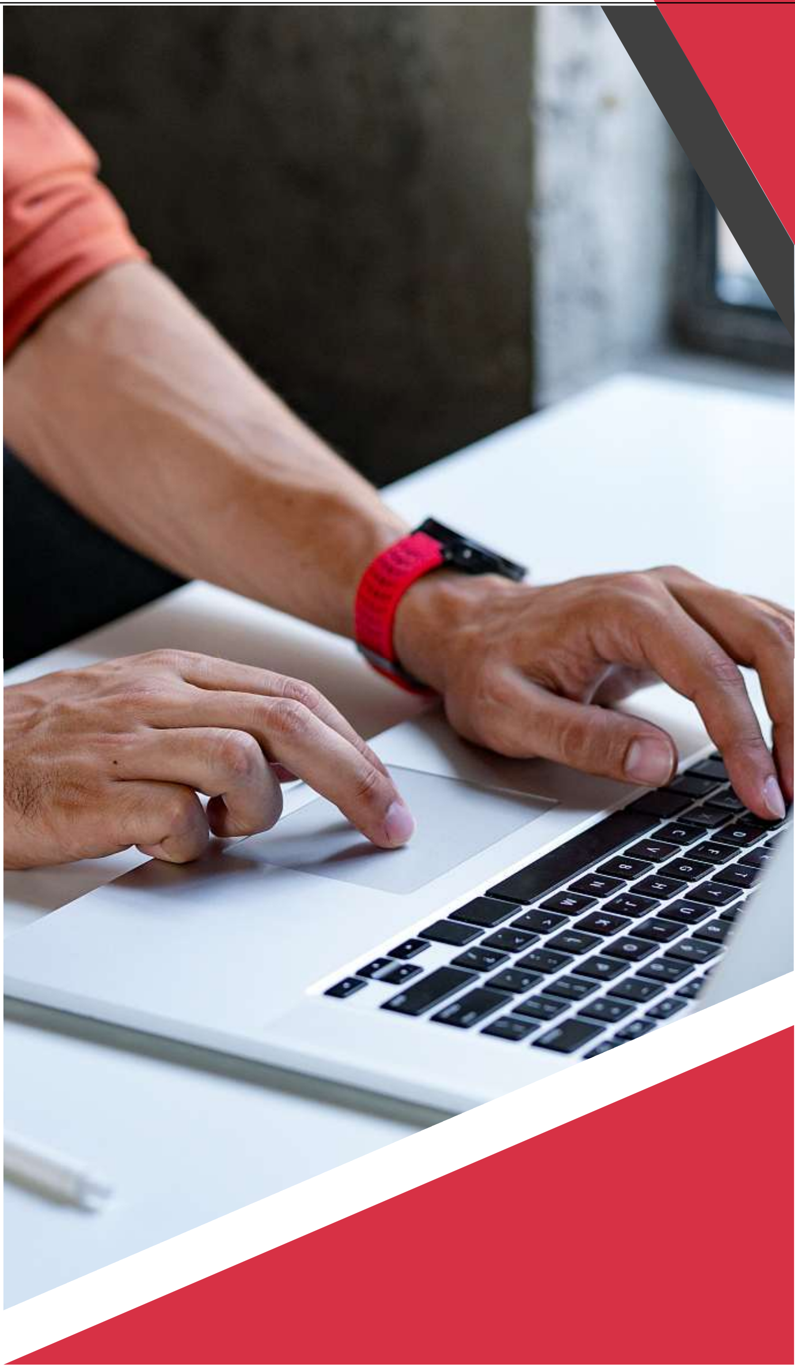
MODEL SELECTION



BLIND GLASS

BEST MODEL FOR OBJECT SELECTION

This document will serve as a report of openCV team in which we would choose the best model for object detection and classification to be intergrated in our project.

Submitted by: OPENCV TEAM

FEBRARY 10 2025

1. Introduction

Blind glass applications require an AI model that is lightweight, efficient, and capable of real-time object recognition or segmentation. Choosing the right model impacts both the user experience and device performance. This document compares four prominent deep learning models—**MobileNet, YOLOv11, EfficientNet, and SAM**—based on their **accuracy, precision, recall, F1-score, computational efficiency, and suitability for mobile deployment**.

We provide a detailed evaluation of each model and justify our final recommendation with supporting evidence and resources.

2. Model Overview

Model	Type	Key Features	Primary Use
MobileNet	Image Classification	Lightweight, depthwise separable convolutions	Efficient classification on mobile devices
YOLOv11	Object Detection	Real-time, fast detection, improved over YOLOv8	Object detection in dynamic environments
EfficientNet	Image Classification	High accuracy with optimized scaling	High-performance classification
SAM (Segment Anything Model)	Image Segmentation	Zero-shot segmentation, high adaptability	Image segmentation tasks

3. Performance Comparison

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Inference Time	Model Size
MobileNet	~71.8	72.5	71.2	71.8	Fast (~3-5ms per image) selcojournal.com	Small (~4MB)
YOLOv11	~57.2 (mAP)	60.5	64.8	62.5	Fast (~10-15ms per image) roboflow.com	Medium (~50MB)
EfficientNet	~84.4	85.2	83.5	84.3	Moderate (~20-30ms per image) selcojournal.com	Large (~30MB-66MB)
SAM	~80.0 (IoU)	81.3	79.5	80.4	Slow (~50-100ms per image)	Large (~90MB)

Key Observations:

- **Accuracy:** EfficientNet achieves the highest accuracy (~84.4%), followed by SAM (~80%), MobileNet (~71.8%), and YOLOv11 (~57.2%).
- **Speed & Model Size:** MobileNet is the fastest and most lightweight, making it ideal for mobile deployment.
- **Object Detection:** YOLOv11 is optimized for real-time detection but has lower accuracy.
- **Segmentation:** SAM is best suited for segmentation but is computationally expensive.

4. Best Model for Blind Glass on Mobile Devices

Considering the requirements for **real-time performance, efficiency, and mobile deployment**, the best choice is **MobileNet**.

Why MobileNet?

- **Lightweight (Only ~4MB model size):** Suitable for mobile devices with limited resources.
- **Fast Inference (~3-5ms per image):** Ensures real-time performance.
- **Moderate Accuracy (~71.8%):** Good enough for general classification of objects.
- **Low Power Consumption:** Ideal for wearable devices like blind glass.

Supporting Evidence

- The study "Performance comparison of MobileNet, EfficientNet, and Inception for predicting crop disease"

selcojournal.com

- demonstrates that MobileNet achieves high efficiency with minimal accuracy loss, making it ideal for mobile devices.
- Real-world applications: MobileNet is widely used in **Google Lens, TensorFlow Lite, and other edge AI applications**.
- Comparison with YOLOv11 and EfficientNet: YOLOv11 is primarily for object detection, which is unnecessary for blind glass applications focused on classification. EfficientNet, while accurate, is too large and slow for mobile deployment.

5. Conclusion

For general object classification on mobile devices, **MobileNet** is the best choice due to its efficiency and speed.

For real-time object detection, **YOLOv11** is an alternative, but it is more computationally expensive.

For high-accuracy applications, **EfficientNet** is superior but less suited for mobile devices.

For segmentation needs, **SAM** is excellent but too slow and large for mobile deployment.

Final Recommendation:

Use MobileNet for blind glass applications running on mobile devices.

6. References

1. Howard, A. et al. (2017). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. arXiv preprint arXiv:1704.04861. (Link)

2. Redmon, J. et al. (2016). You Only Look Once: Unified, Real-Time Object Detection. arXiv preprint arXiv:1506.02640. (Link)

3. Tan, M. et al. (2019). EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. arXiv preprint arXiv:1905.11946. (Link)

4. Kirillov, A. et al. (2023). Segment Anything. Meta AI. arXiv preprint arXiv:2304.02643. (Link)

REAL TIME EXAMPLE

Model_evaluation file serves as a real-time example for evaluating the four models. While it may not be an entirely accurate representation, it provides valuable insights into how the system functions in real-world conditions. By simulating real-time performance, this model helps us understand its behavior, identify potential improvements, and refine future iterations.