Ministry of Science and Higher Education of the Republic of Kazakhstan
L.N. Gumilyov Eurasian National University

Faculty of Information Technology
Department of Information Systems

Openpose, Yolo, example

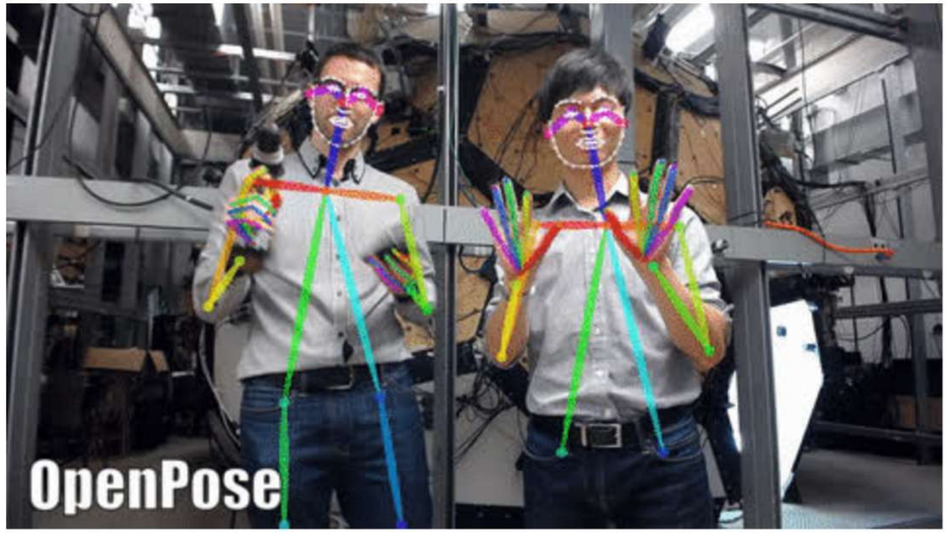
Done by: Yerassyl Iskakov

CHECKED BY: PROF. ZHUKABAYEVA T.K.

WHAT IS OPENPOSE?

OpenPose has represented the first real-time multi-person system to jointly detect human body, hand, facial, and foot keypoints (in total 135 keypoints) on single images.

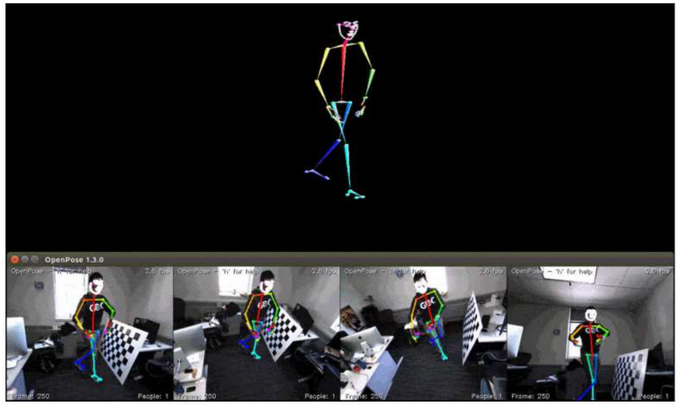
It is authored by Ginés Hidalgo, Zhe Cao, Tomas Simon, Shih-En Wei, Yaadhav Raaj, Hanbyul Joo, and Yaser Sheikh. It is maintained by Ginés Hidalgo and Yaadhav Raaj. OpenPose would not be possible without the CMU Panoptic Studio dataset. We would also like to thank all the people who have helped OpenPose in any way.



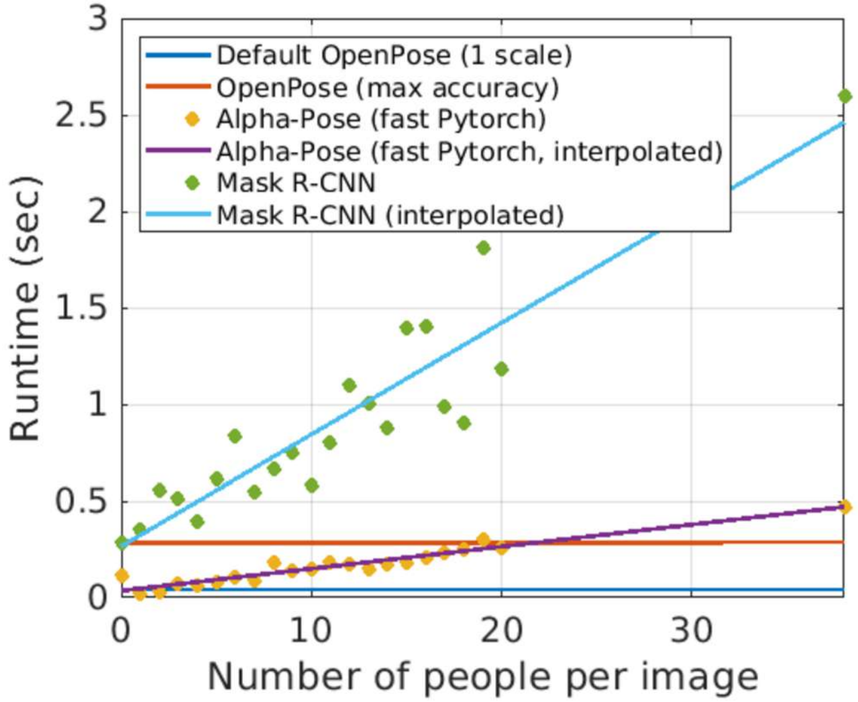
Authors Ginés Hidalgo (left) and Hanbyul Joo (right) in front of the CMU Panoptic Studio

RUNTIME ANALYSIS

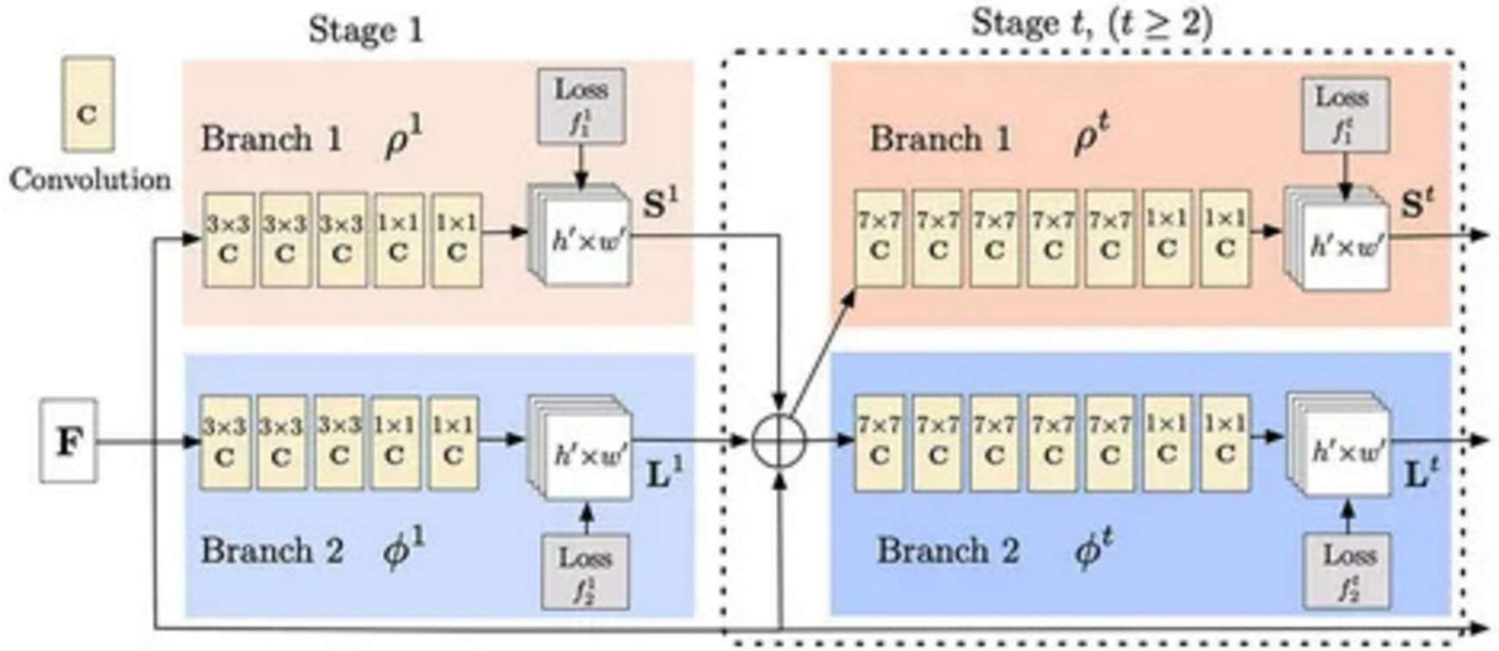
YOLO is a deep learning-based object detection model designed for real-time applications. It processes an entire image in a single pass, making it significantly faster than traditional region-based detection methods.



Whole-body (Body, Foot, Face, and Hands) 2D Pose Estimation

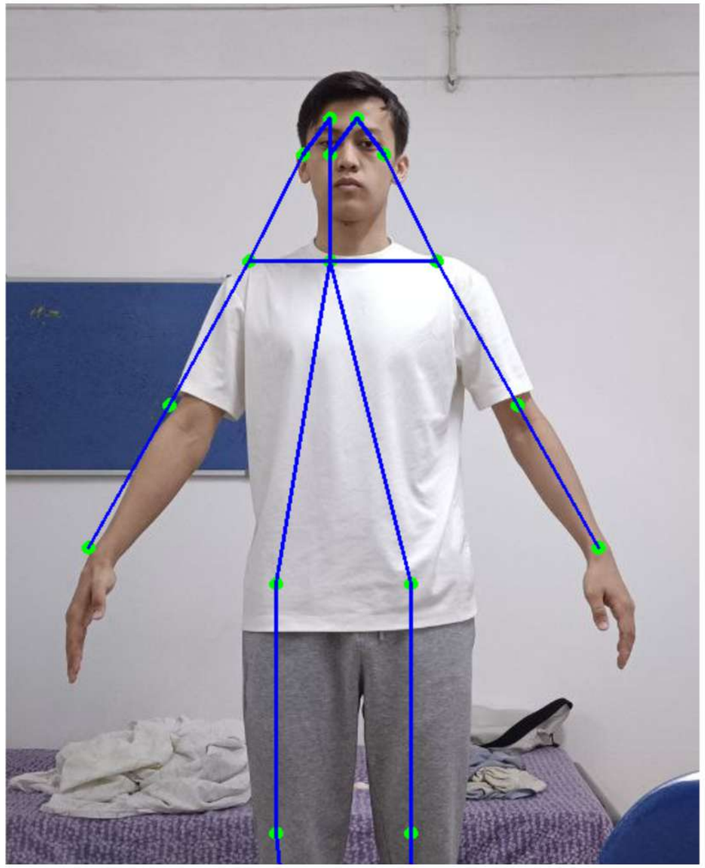


OPENPOSE ARCHITECTURE

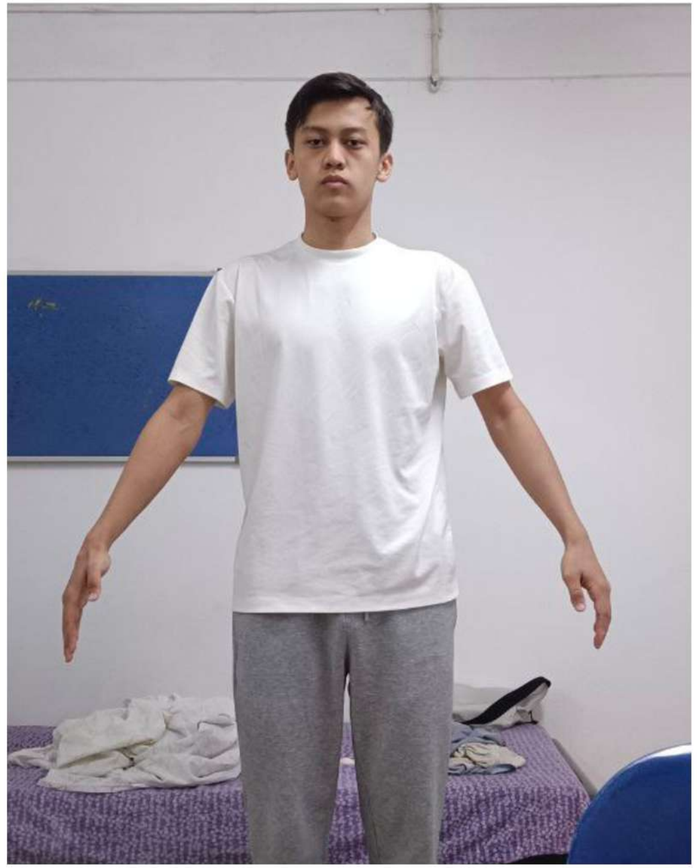


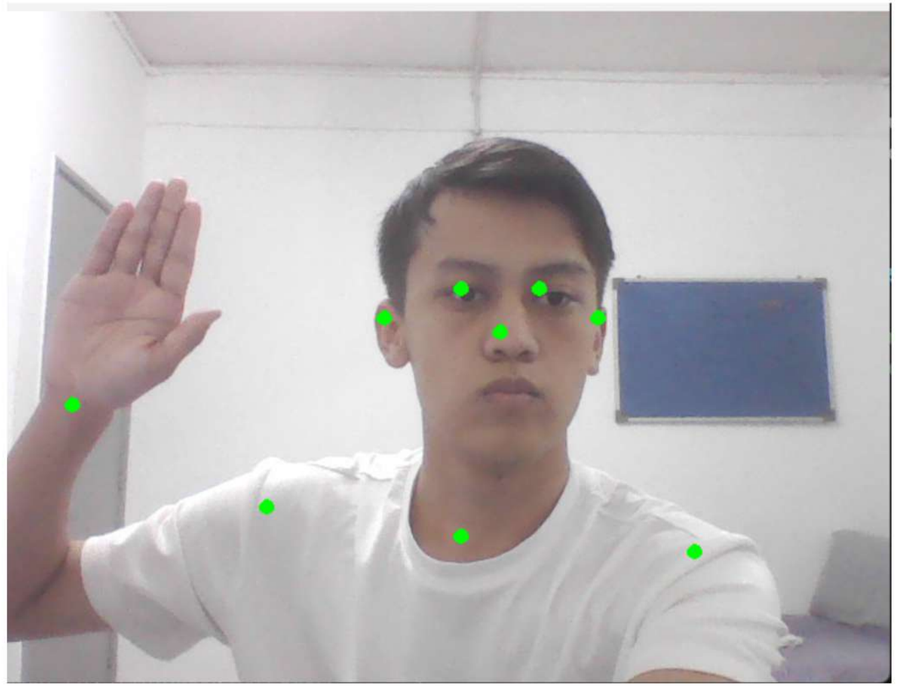
OpenPose analyzes the input image with a Convolutional Neural Network (CNN), a deep learning algorithm suited for processing visual data. This initial step extracts what are known as "feature maps" from the image. Feature maps are essentially detailed layers of the image that highlight various aspects, such as edges, textures, or specific shapes. After the feature maps are extracted, OpenPose uses a specialized, multi-stage CNN pipeline.

OPENPOSE EXAMPLE



Pose tracking

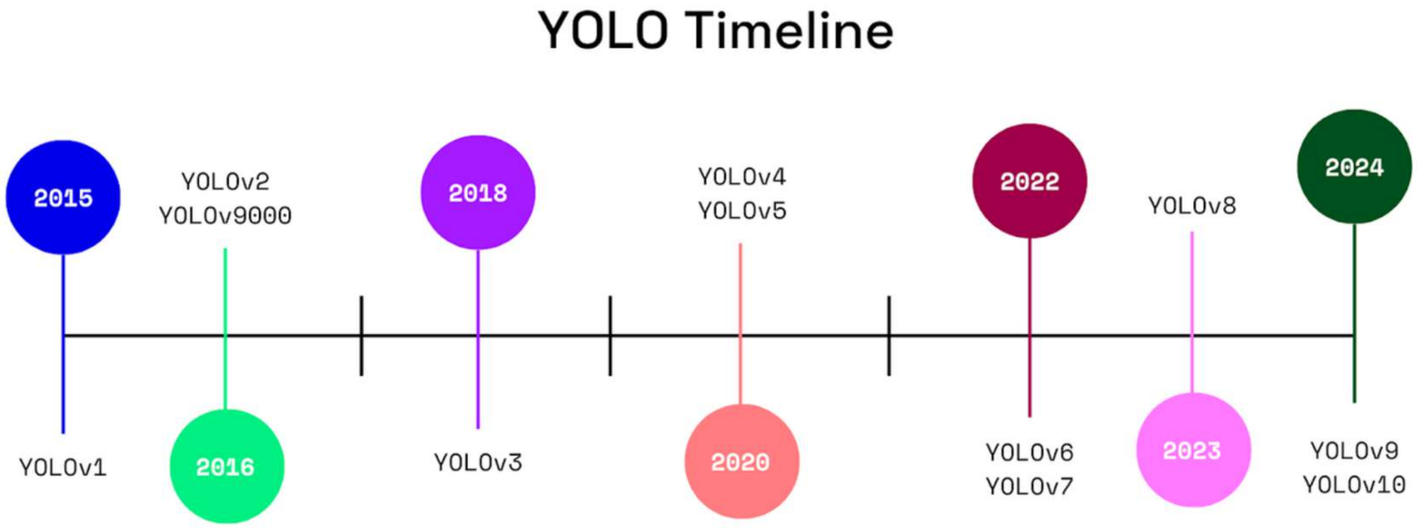




Real-Time tracking

WHAT IS YOLO?

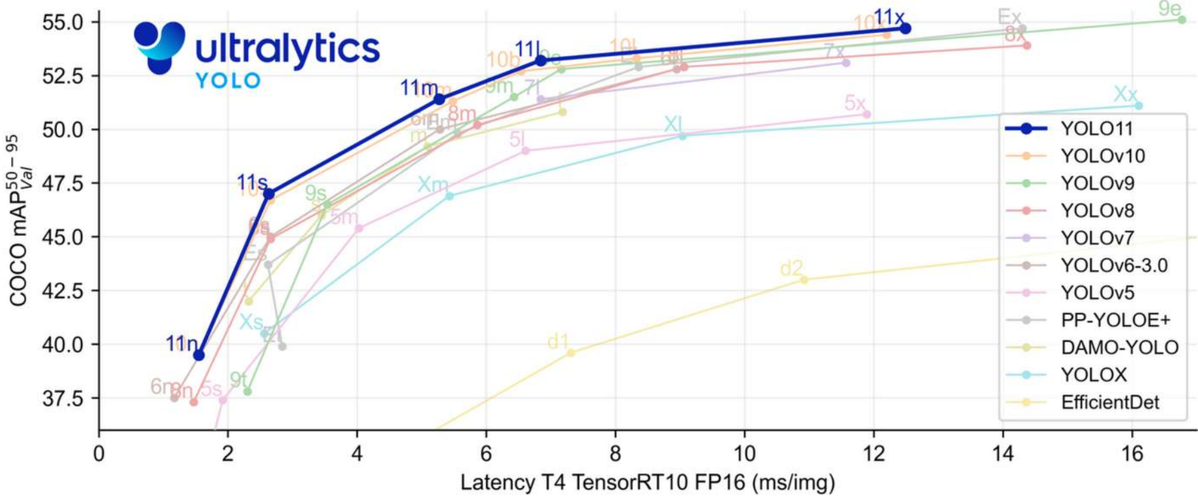
YOLO is a deep learning-based object detection model designed for real-time applications. It processes an entire image in a single pass, making it significantly faster than traditional region-based detection methods.



WHAT IS YOLOV8?

YOLOv8 is the latest generation of object detection models from Ultralytics, offering state-of-the-art performance.
Compared to previous YOLO versions, YOLOv8 is faster and more accurate while providing a unified framework for training models to perform:

- Object detection
- Instance segmentation
- Image classification



WHAT'S NEW IN YOLOV8+?

Ultralytics has released an entirely new repository for YOLO models. It is designed as a unified platform for training models in object detection, instance segmentation, and image classification.
Here are some key features of the new version:

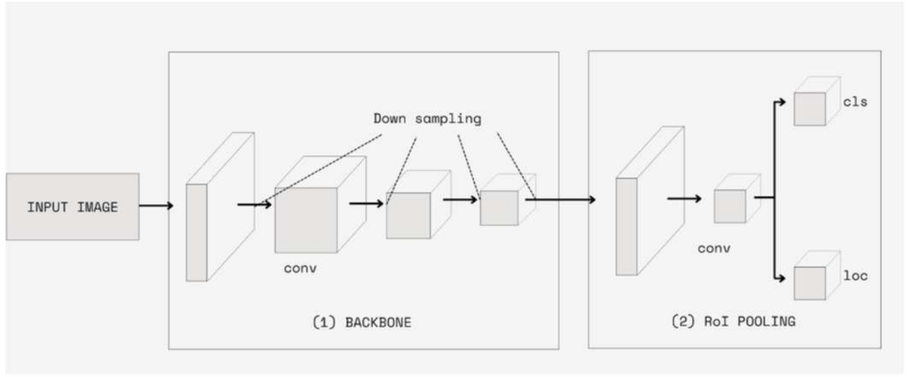
- User-friendly API (command-line + Python).
- Faster and more accurate.
  - Object detection
  - Instance segmentation
  - Image classification
- Extensible to all previous versions.
- New backbone network.
- New anchor-free head.
- New loss function.

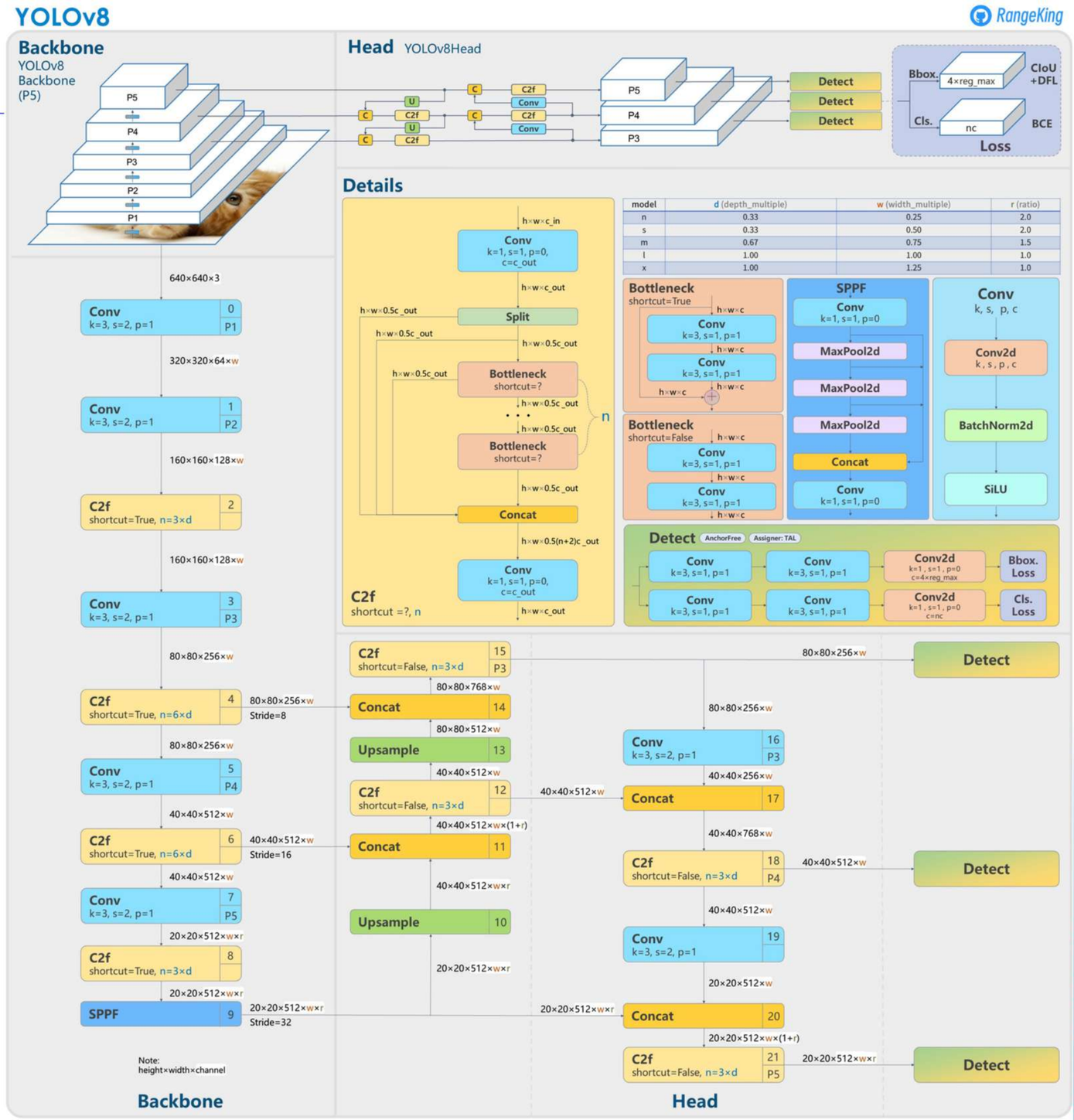
YOLO ARCHITECTURE

The architecture of YOLOv8 builds on improvements made in previous YOLO versions:

- Backbone: A deep convolutional network that extracts important image features.
- Neck: A series of layers that refine the extracted features before passing them to the detection head.
- Head: Responsible for final object detection, bounding box regression, and classification.

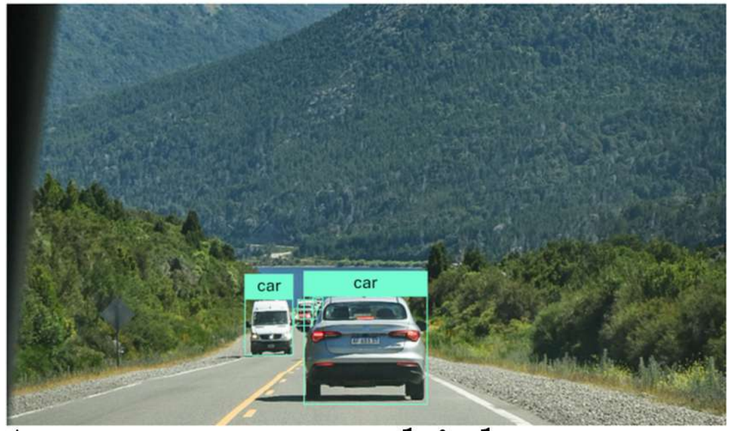
The new anchor-free detection head improves speed and accuracy by simplifying the object localization process.



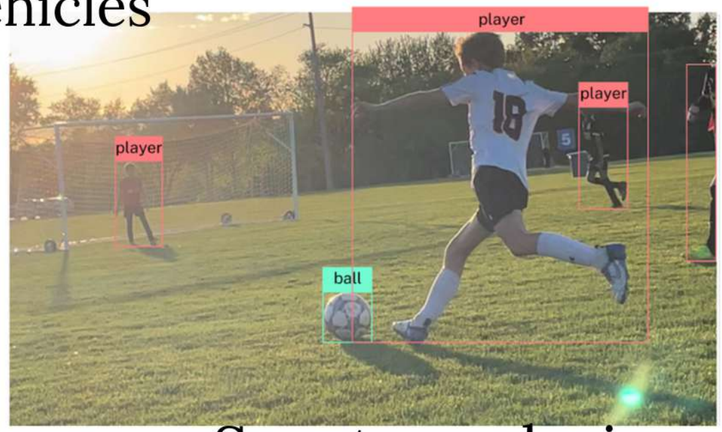


Use Cases of YOLOv8

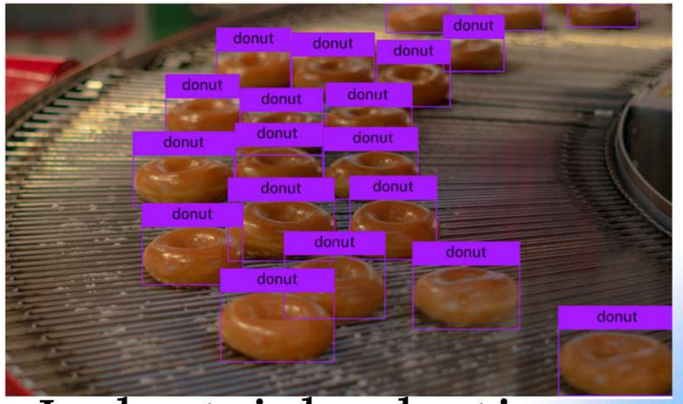
- Edge devices (Raspberry Pi, Jetson Nano) for real-time applications.
- Cloud services (Google Cloud, AWS) for large-scale inference.
- Mobile devices for AI-powered applications.



Autonomous vehicles



Sports analysis



Industrial robotics

YOLOV8 VS YOLOV5

Performance Comparison of YOLOv8 vs YOLOv5

Model Size	Detection#	Segmentation#	Classification*
Nano	+33.21%	+32.97%	+3.10%
Small	+20.05%	+18.62%	+1.12%
Medium	+10.57%	+10.89%	+0.66%
Large	+7.96%	+6.73%	0.00%
Xtra Large	+6.31%	+5.33%	-0.76%

#Image Size = 640 *Image Size = 224

WHAT ARE THE DISADVANTAGES OF YOLO?

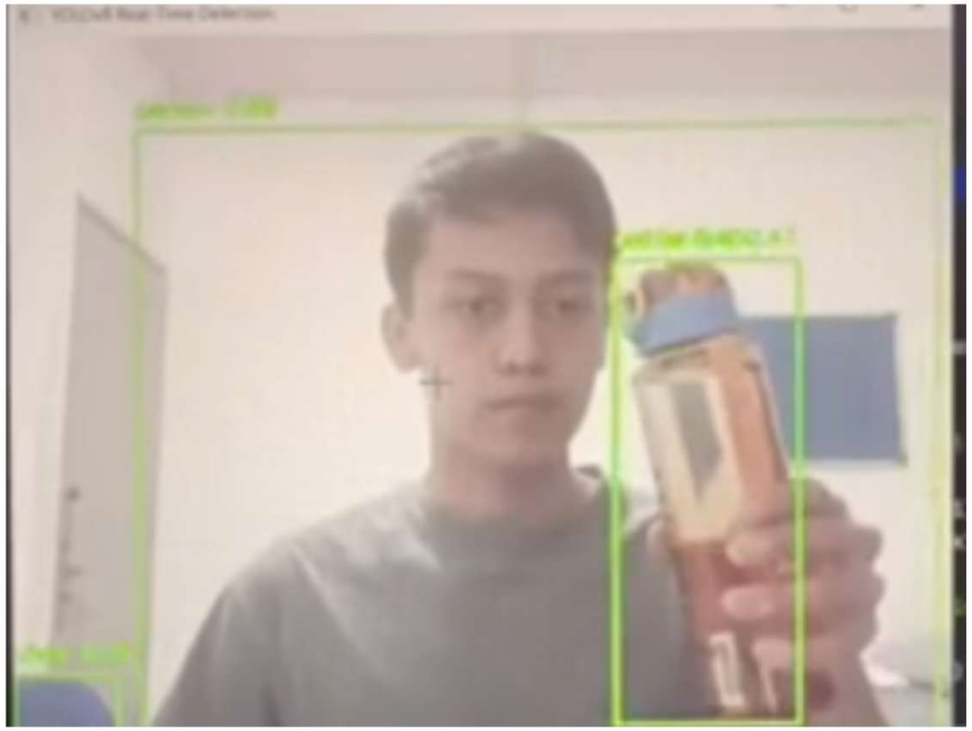
While YOLO is fast, this speed can come at the cost of accuracy, particularly in terms of localization.
The bounding boxes might not always be perfectly tight, sometimes missing part of the face or including extra space.

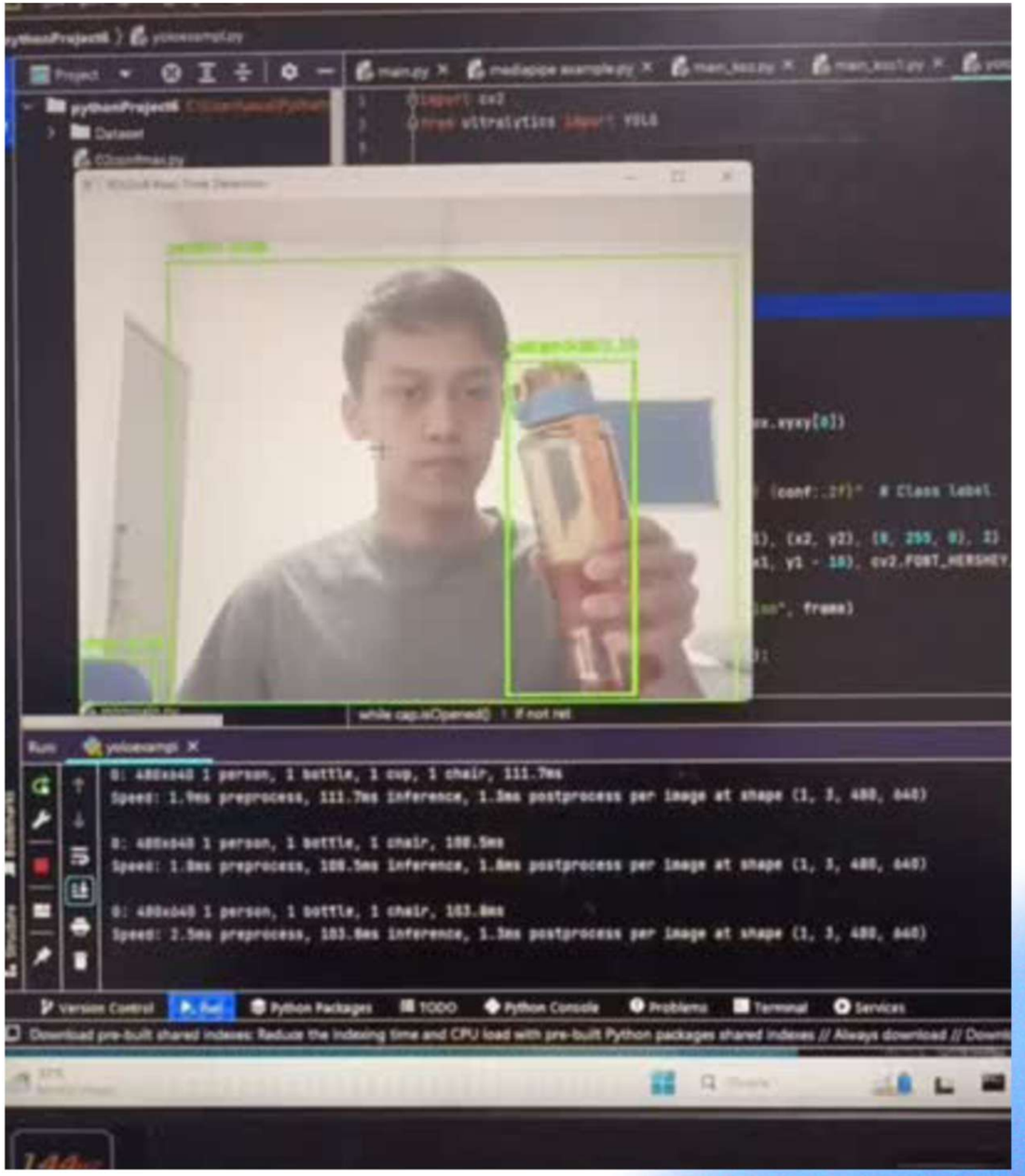
This happens because YOLO doesn't go through multiple stages of refining and comparing similar boxes like a region proposal network would, potentially leading to minor localization inaccuracies.
Skipping multiple passes also leads YOLO models to struggle with:

- Small objects: Its grid-based detection approach might not offer enough resolution for smaller objects, making them more difficult to detect.
- Detailed scenes: Complex environments or scenes with overlapping objects make it harder for YOLO to pinpoint objects accurately.

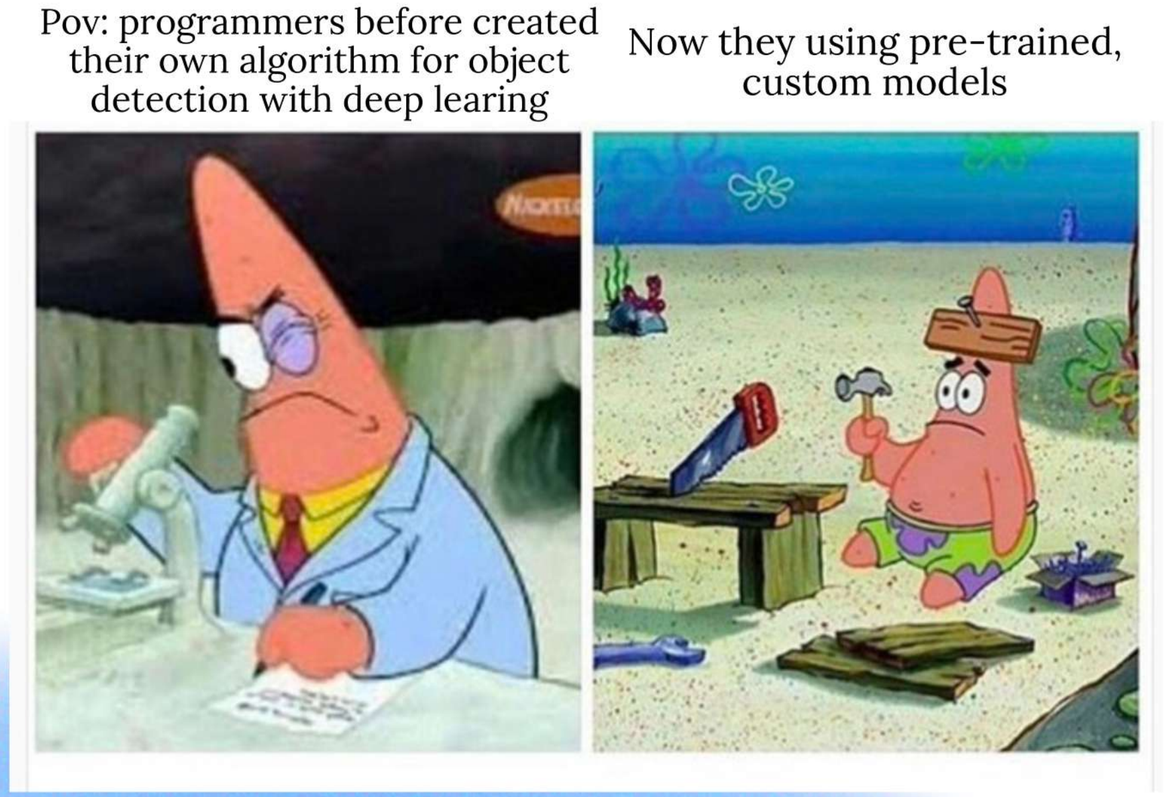
YOLO (OBJECT DETECTION) EXAMPLE

I have developed real-time object detection using YOLOv8 (without a custom dataset)





CONCLUSION



YOLOv8 stands as a testament to the continuous evolution of object detection algorithms. Its innovative architecture, training strategy, and performance metrics position it as a leading solution for real-time object detection tasks.

As the field of computer vision advances, YOLOv8 serves as a benchmark for future developments, pushing the boundaries of what is achievable in object detection.

Whether deployed in autonomous vehicles, surveillance systems, or other applications, YOLOv8's versatility and accuracy make it a powerful tool in the computer vision landscape.

REFERENCES

1. https://habr.com/ru/articles/710016/
2. https://www.viam.com/post/guide-yolo-model-real-time-object-detection-with-examples#:~:text=Discover%20how%20YOLO%20models%20excel%20in%20real-time%20object,and%20examples%20to%20help%20you%20use%20its%20capabilities.
3. https://github.com/ultralytics/ultralytics
4. https://yolov8.org/yolov8-architecture/

THANKS FOR ATTENTION!