L.N. Gumilyov Eurasian National University
Faculty of Information Technology
Department of «Information Systems»

DeepLabV3 & YOLO

IWS №9 by discipline «Computer graphics and pattern recognition»

Presented by: Toleubay D.M.
Checked by: Prof. Dr. Zhukabayeva T.K.

What is DeepLabV3?

One very important benefit that DeepLabV3 has over other semantic segmentation and classification models is that it is extremely accurate when it comes to multi-scale segmentation. Multi-scale segmentation involves analysing the image at different scales to capture objects of different sizes and shapes. DeepLabV3 uses atrous (or dilated) convolutions, a technique that allows it to capture context information at different scales without increasing the complexity of the model.

The Evolution of DeepLab Series

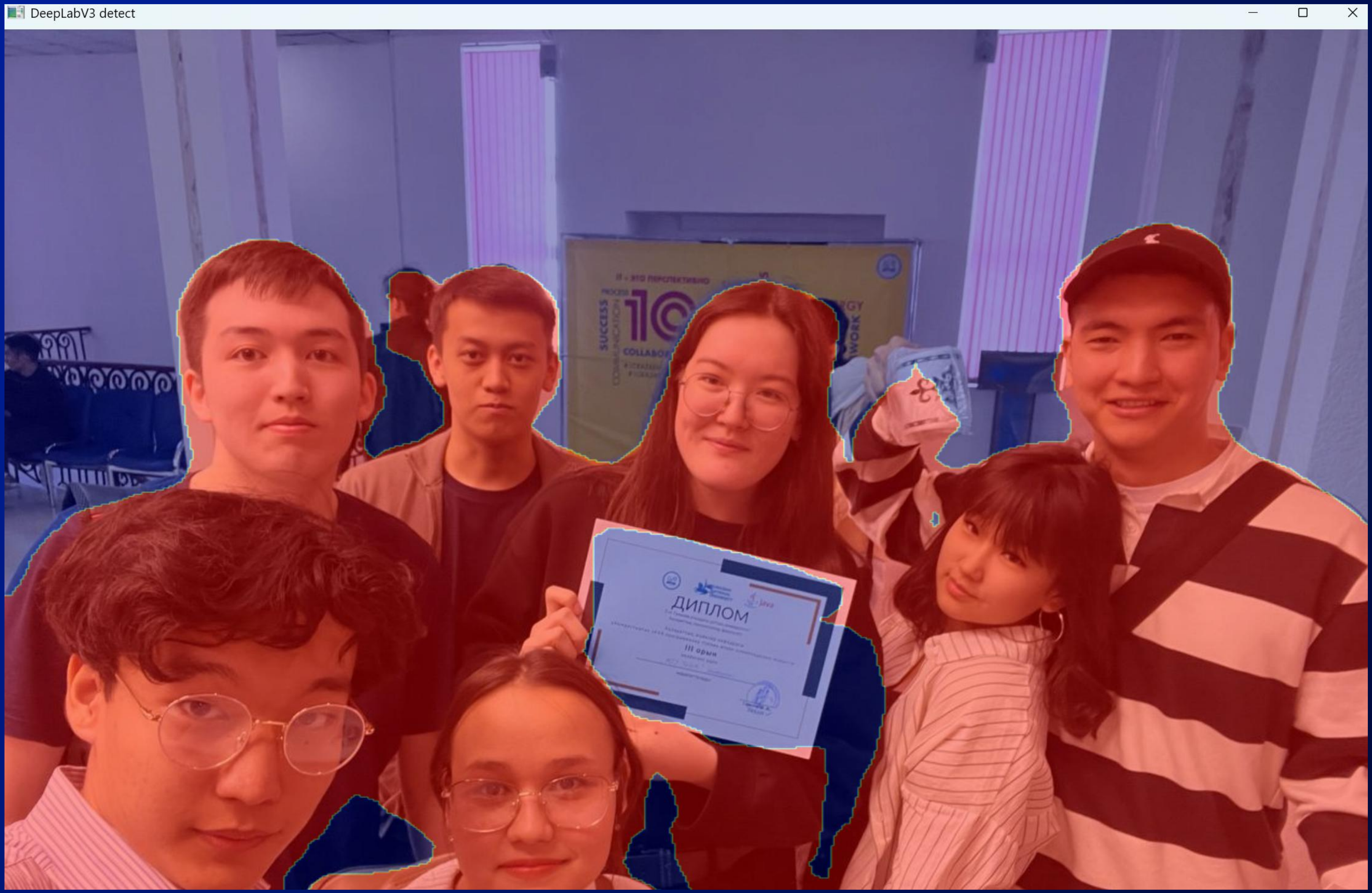
DeepLabV1: Introduced atrous (dilated) convolutions, enhancing context capture without resolution loss.

DeepLabV2: Added Atrous Spatial Pyramid Pooling (ASPP) for better multi-scale object segmentation.

DeepLabV3 (2017): Optimized ASPP and network depth, improving segmentation accuracy.

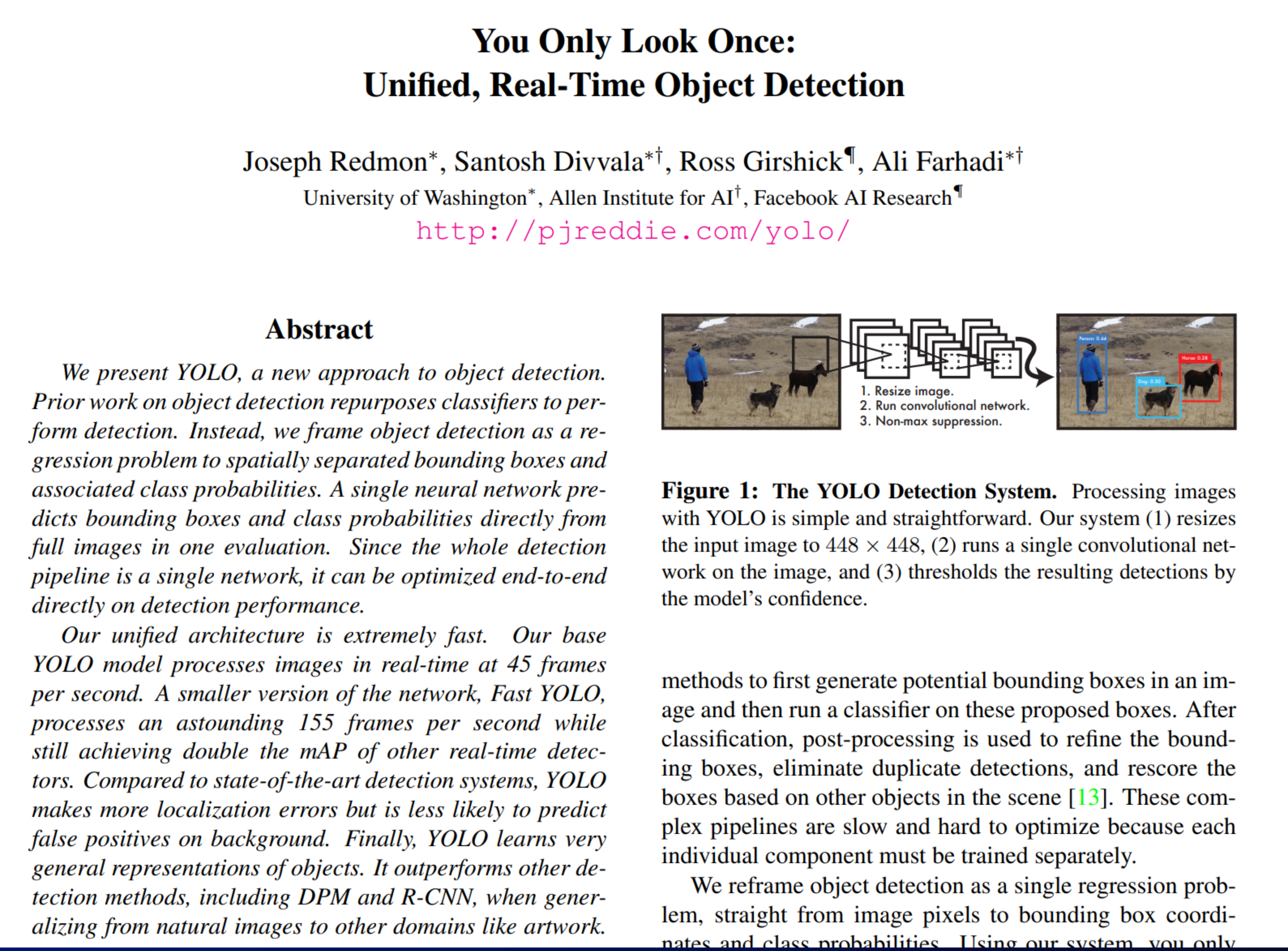
DeepLabV3+ (2018): Enhanced version with SOTA performance (89% mIOU on PASCAL VOC 2012, 82.1% on Cityscapes).

Results of my experience



What is YOLO?

YOLO (You Only Look Once) is a family of computer vision models that gained widespread popularity after Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi presented a new architecture at the 2016 CVPR conference. It even won the OpenCV People Choice Awards.



You Only Look Once: Unified, Real-Time Object Detection

Joseph Redmon*, Santosh Divvala*†, Ross Girshick¶, Ali Farhadi*†
University of Washington*, Allen Institute for AI†, Facebook AI Research¶
http://pjreddie.com/yolo/

Abstract

We present YOLO, a new approach to object detection. Prior work on object detection repurposes classifiers to perform detection. Instead, we frame object detection as a regression problem to spatially separated bounding boxes and associated class probabilities. A single neural network predicts bounding boxes and class probabilities directly from full images in one evaluation. Since the whole detection pipeline is a single network, it can be optimized end-to-end directly on detection performance.

Our unified architecture is extremely fast. Our base YOLO model processes images in real-time at 45 frames per second. A smaller version of the network, Fast YOLO, processes an astounding 155 frames per second while still achieving double the mAP of other real-time detectors. Compared to state-of-the-art detection systems, YOLO makes more localization errors but is less likely to predict false positives on background. Finally, YOLO learns very general representations of objects. It outperforms other detection methods, including DPM and R-CNN, when generalizing from natural images to other domains like artwork.

1. Resize image.
2. Run convolutional network.
3. Non-max suppression.

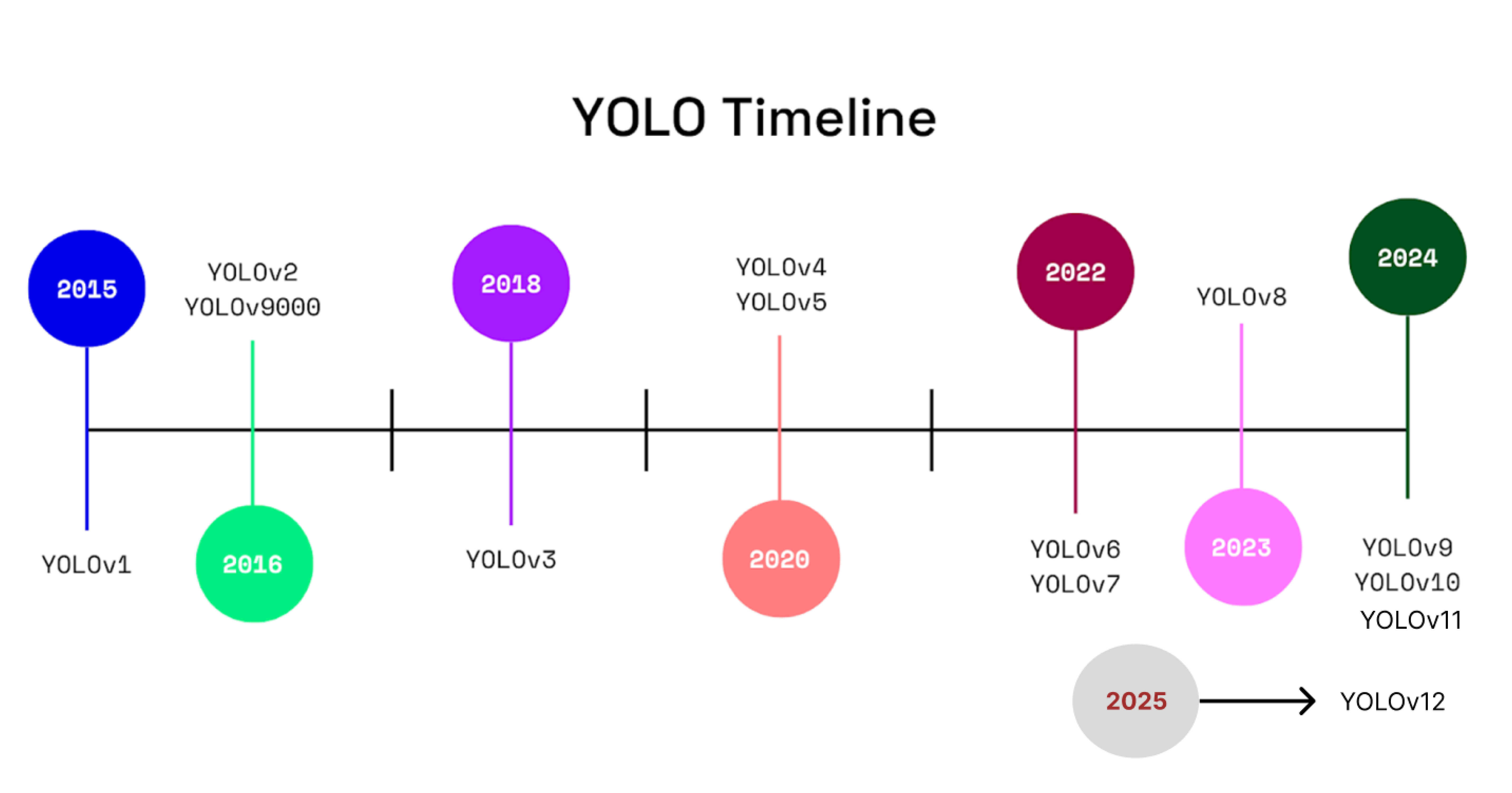
Figure 1: The YOLO Detection System. Processing images with YOLO is simple and straightforward. Our system (1) resizes the input image to 448 × 448, (2) runs a single convolutional network on the image, and (3) thresholds the resulting detections by the model's confidence.

methods to first generate potential bounding boxes in an image and then run a classifier on these proposed boxes. After classification, post-processing is used to refine the bounding boxes, eliminate duplicate detections, and rescore the boxes based on other objects in the scene [13]. These complex pipelines are slow and hard to optimize because each individual component must be trained separately.

We reframe object detection as a single regression problem, straight from image pixels to bounding box coordinates and class probabilities. Using our system, you only

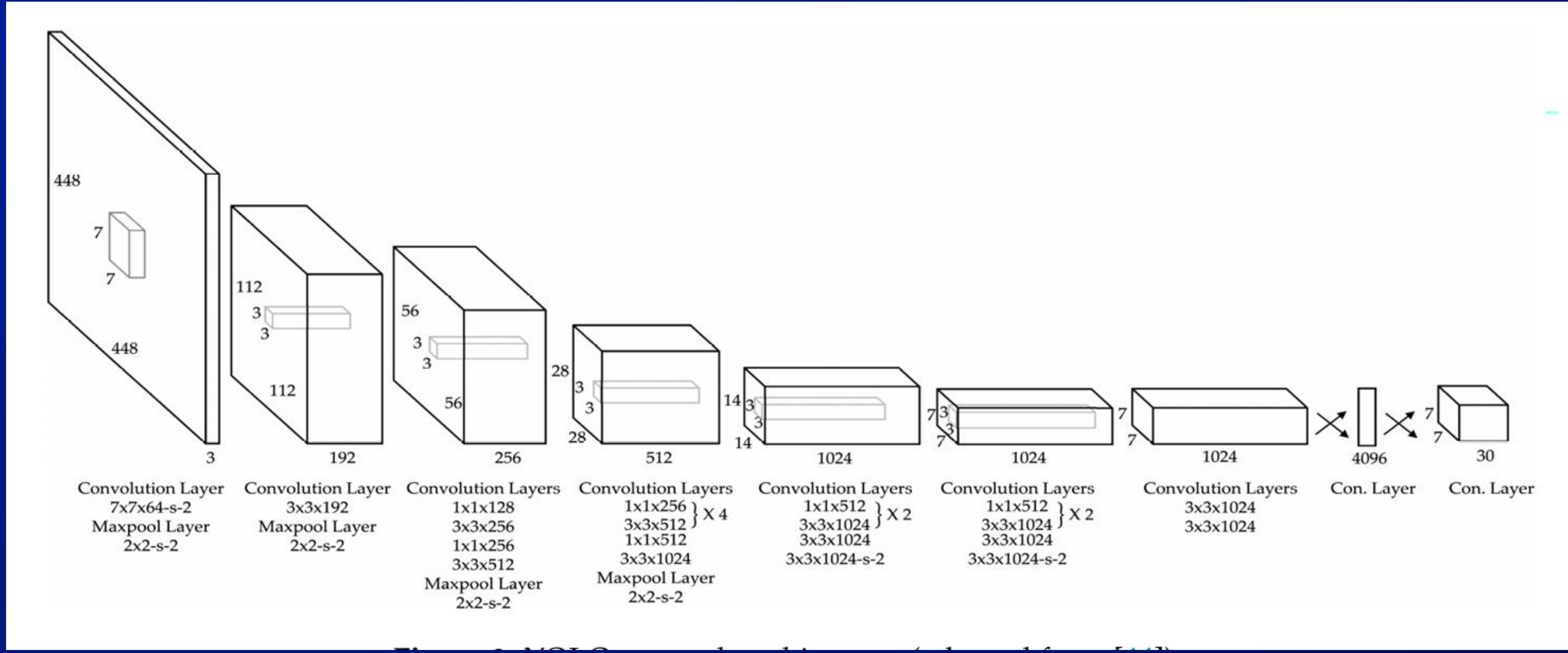
The authors frame the object detection problem as a regression rather than a classification task by spatially separating bounding boxes and associating probabilities to each detected image using a single convolutional neural network (CNN).

The Evolution of YOLO: From 2015 to 2025



Architecture YOLO

In the first layer, while the image is a tensor with only three channels, a 7x7 convolution with a stride of 2 is applied to immediately reduce it and to perform subsequent calculations more efficiently at a reduced resolution. At the end, there are two fully connected layers so that the final detections can take into account the context of the entire image.



The architecture works as follows:

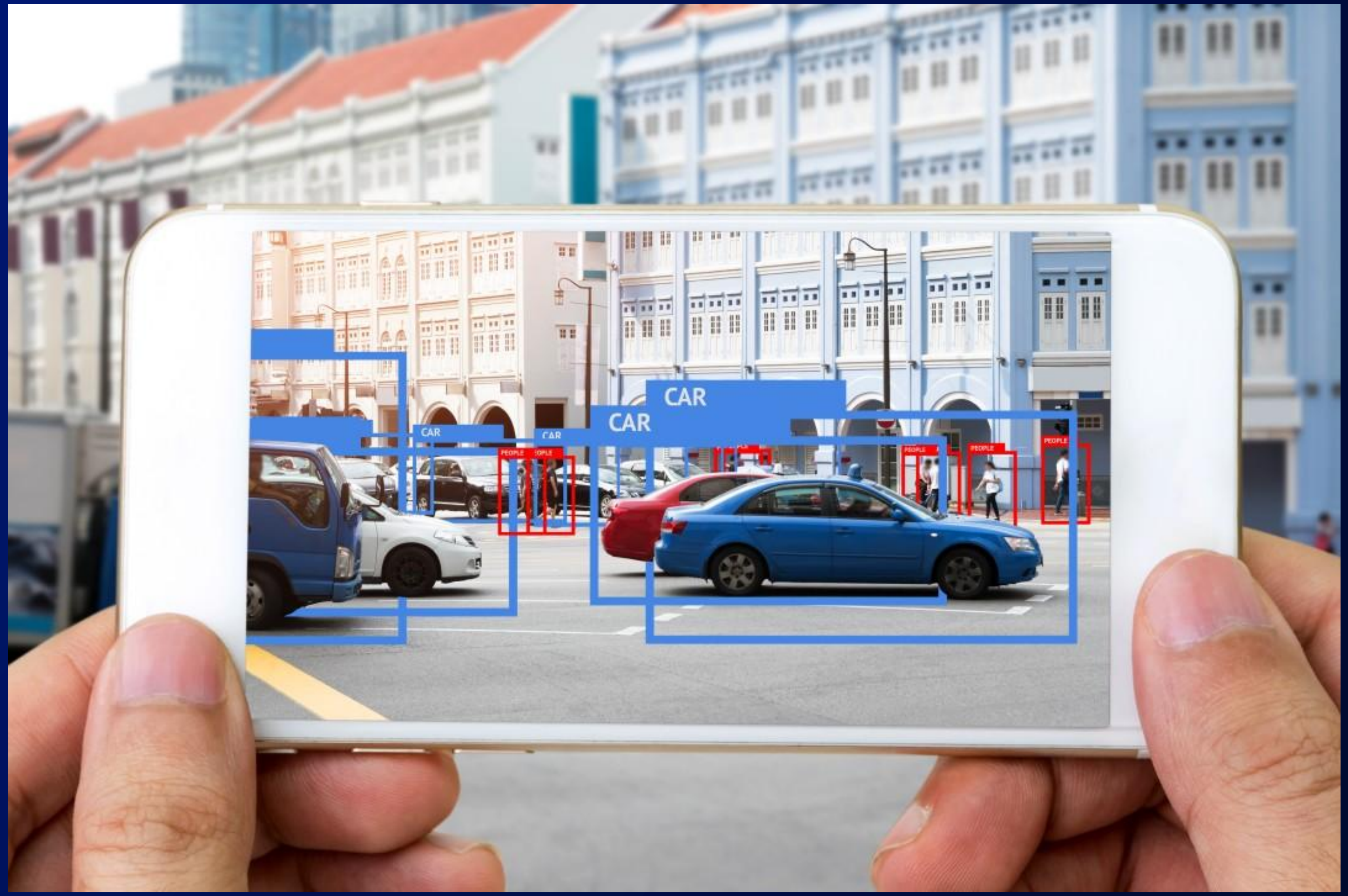
- Resizes the input image into 448x448 before going through the convolutional network.
- A 1x1 convolution is first applied to reduce the number of channels, followed by a 3x3 convolution to generate a cuboidal output.
- The activation function under the hood is ReLU, except for the final layer, which uses a linear activation function.
- Some additional techniques, such as batch normalization and dropout, regularize the model and prevent it from overfitting.

What are YOLO models used for?

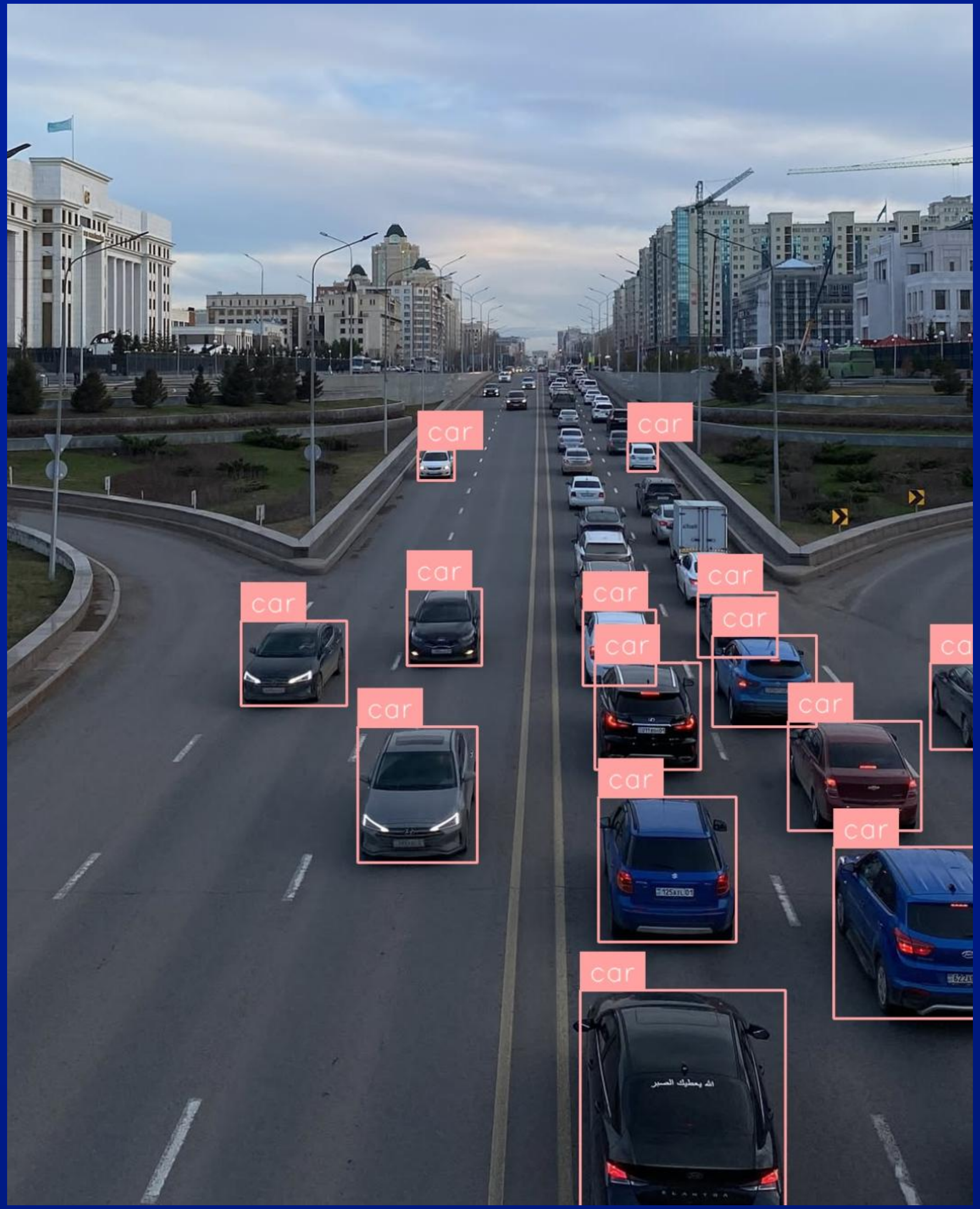
The YOLO framework has transformed computer vision by enabling rapid and accurate object detection. Its speed and precision are ideal for various real-time applications, including:

- Vehicle detection in autonomous driving
- Animal identification in wildlife conservation
- Safety monitoring in industrial settings
- Retail inventory management

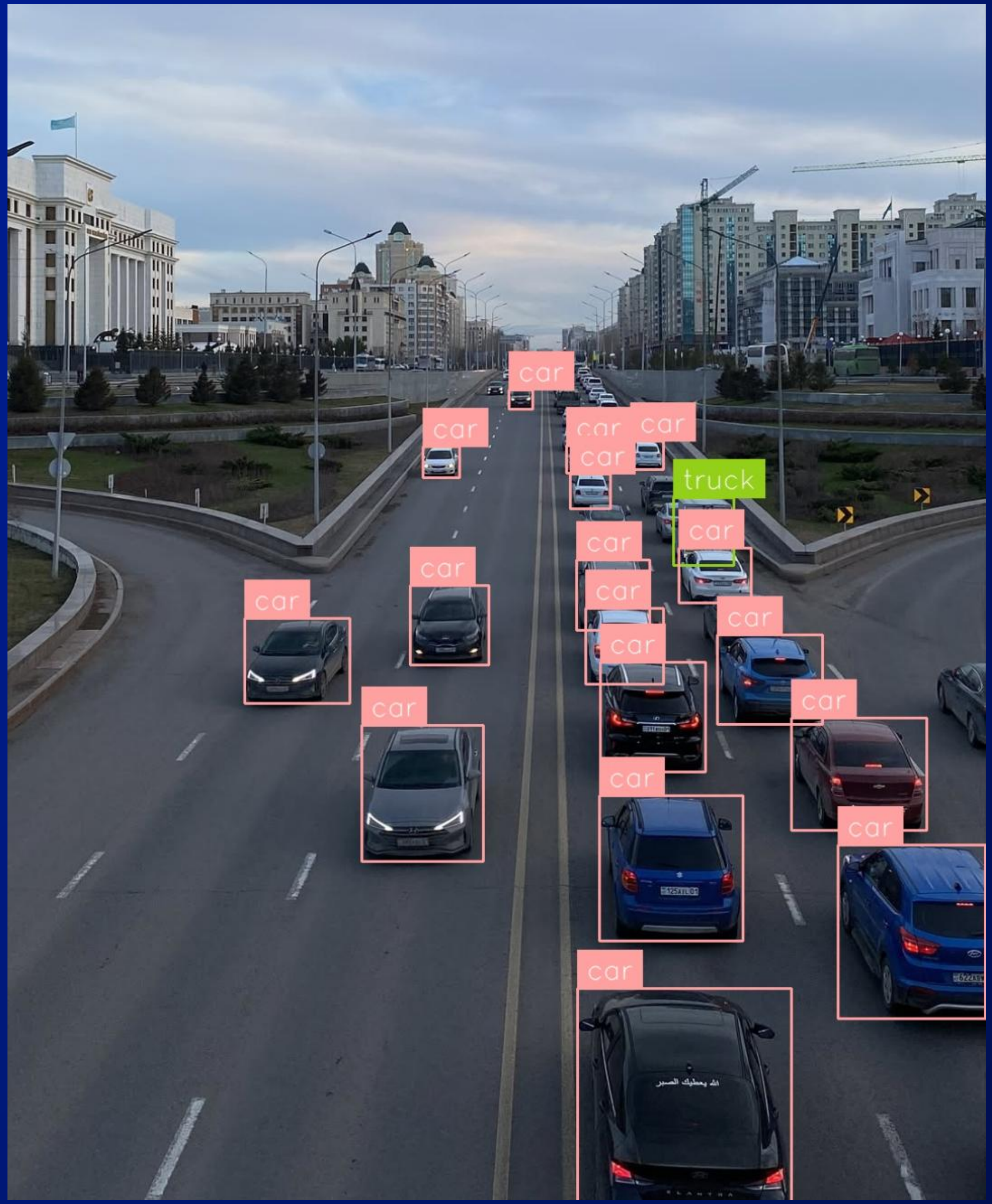
The YOLO algorithm is used for real-time object detection. Before YOLO, R-CNNs were one of the most common methods for object detection, but they were slow and not as useful for real-time applications. YOLO provides the speed needed for use cases that require fast inference, such as car detection, animal identification, and security breach monitoring.



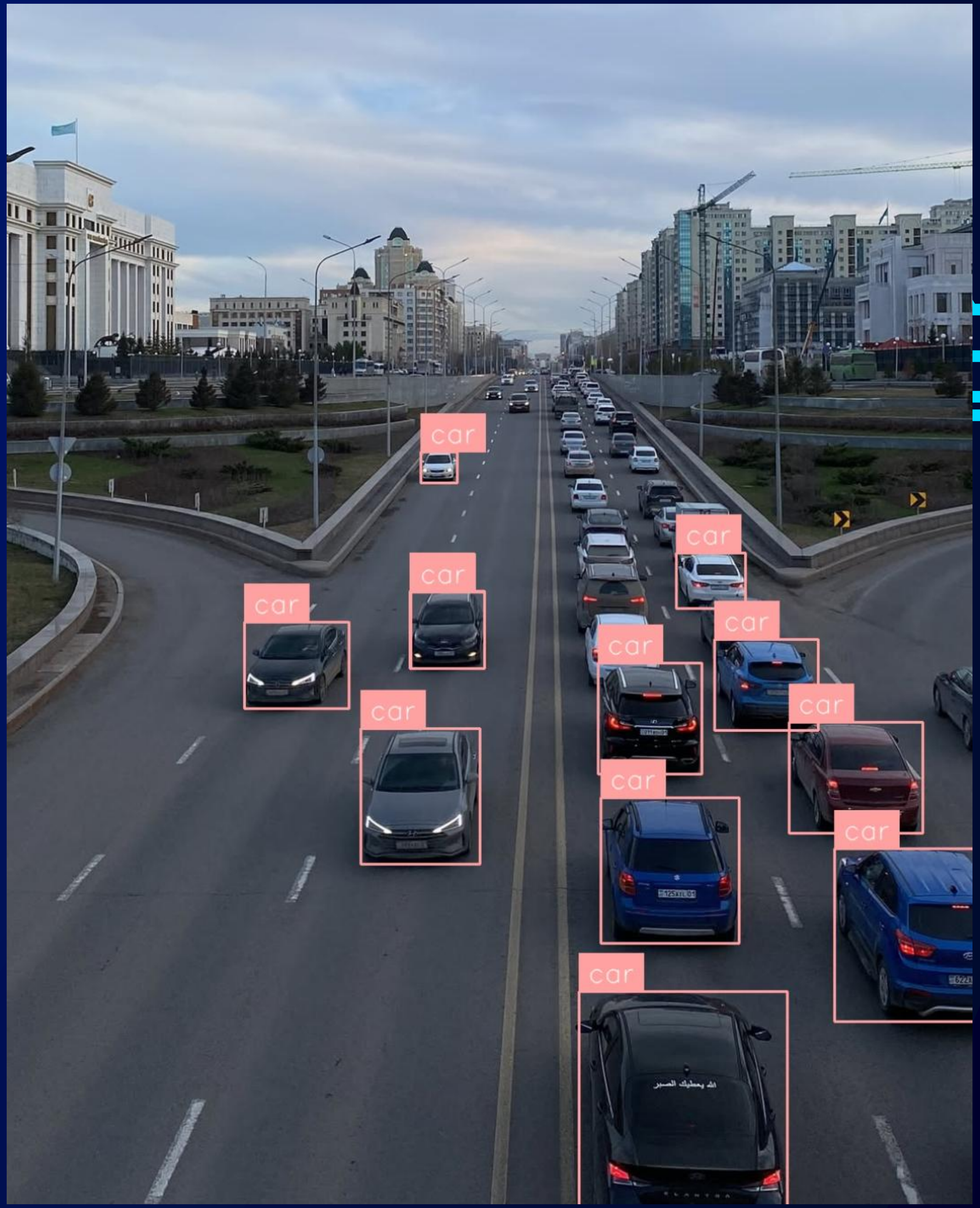
Comparison for YOLO



YOLOv8



YOLOv9



YOLOv10

Conclusion

One of the main advantages of YOLO is its fast inference speed, which allows it to process images in real time. It's well-suited for applications such as video surveillance, self-driving cars, and augmented reality. Additionally, YOLO has a simple architecture and requires minimal training data, making it easy to implement and adapt to new tasks.

Despite limitations such as struggling with small objects and the inability to perform fine-grained object classification, YOLO has proven to be a valuable tool for object detection and has opened up many new possibilities for researchers and practitioners. As the field of Computer Vision continues to advance, it will be interesting to see how YOLO and other object detection algorithms evolve and improve.

References

1. *Chokshi, A. (2023, May 5). A guide to using DeepLabV3 for semantic segmentation. Datature. https://www.datature.io/blog/a-guide-to-using-deeplabv3-for-semantic-segmentation*
2. *Kouidri, A. (2023, December 12). Understanding DeepLabV3 in Image Segmentation. Ikomia. https://www.ikomia.ai/blog/understanding-deeplabv3-image-segmentation*
3. *Nelson, J. (2025, January 9). What is YOLO? The ultimate guide [2025]. Roboflow. https://blog.roboflow.com/guide-to-yolo-models/*
4. *YOLO Model. (n.d.). Machine and Deep Learning. DeepMachineLearning. Retrieved March 12, 2025, from https://deepmachinelearning.ru/docs/Neural-networks/Object-detection/YOLO*
5. aaы

THANK YOU