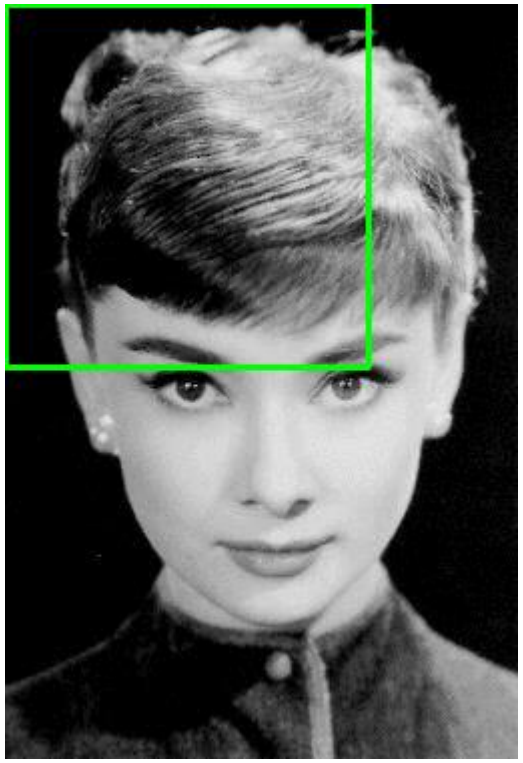
YOLO (You Look Only Once)

May. 2. 2023

DL_2023, PJK

YOLO History

Sliding Window Object Detection



R CNN

Fast R CNN

Faster R CNN

·

·

·

YOLO

YOLO History

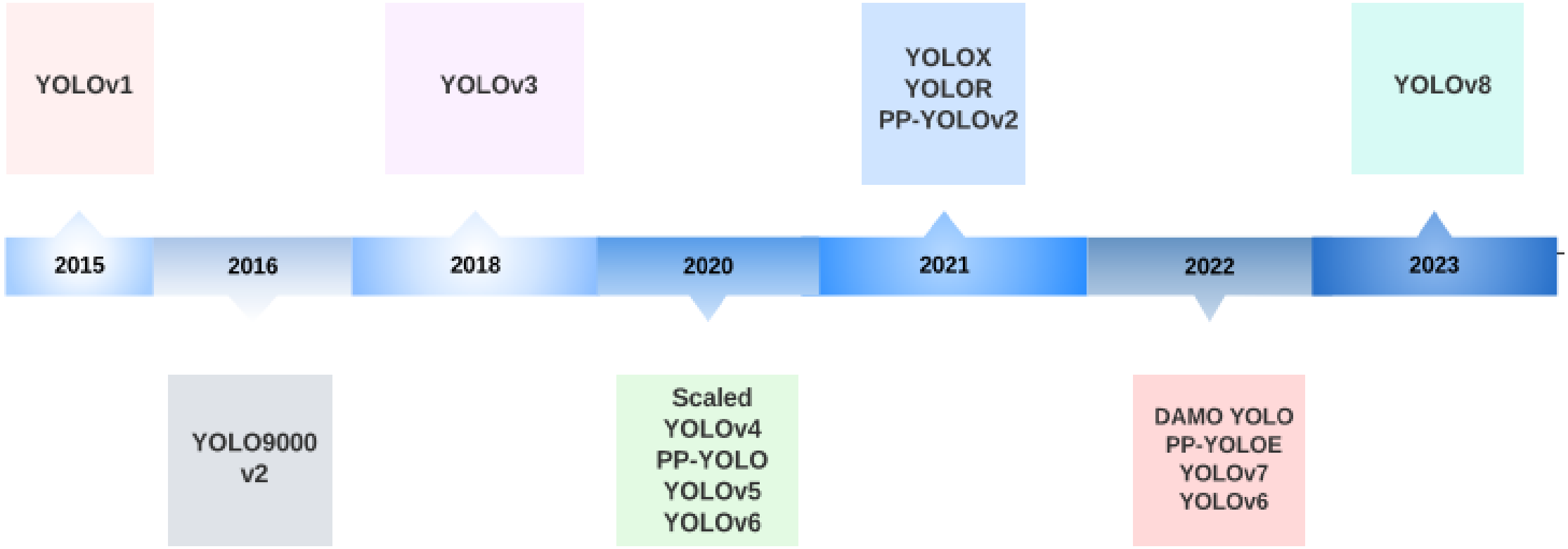


Figure 1: A timeline of YOLO versions.

https://arxiv.org/pdf/2304.00501.pdf

YOLO History

- You Only Look Once (YOLO) proposes using an end-to-end neural network that makes **predictions of bounding boxes and class probabilities all at once.**

- YOLO (You Only Look Once) is a popular object detection algorithm that has revolutionized the field of computer vision. It is fast and efficient, making it **an excellent choice for real-time object detection tasks**. It has achieved **state-of-the-art performance** on various benchmarks and has been widely adopted in various real-world applications.

- 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.

https://arxiv.org/pdf/2304.00501.pdf

YOLO History

YOLO (You Only Look Once), a popular object detection and image segmentation model, was developed by Joseph Redmon and Ali Farhadi at the University of Washington. Launched in **2015**, YOLO quickly gained popularity for its high speed and accuracy.

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/

https://arxiv.org/pdf/2304.00501.pdf

YOLO History

YOLO (You Only Look Once), a popular object detection and image segmentation model, was developed by Joseph Redmon and Ali Farhadi at the University of Washington. Launched in **2015**, YOLO quickly gained popularity for its high speed and accuracy.

•YOLOv2, released in **2016**, improved the original model by incorporating batch normalization, anchor boxes, and dimension clusters.

YOLO9000:
Better, Faster, Stronger

Joseph Redmon*†, Ali Farhadi*†

University of Washington*, Allen Institute for AI†

http://pjreddie.com/yolo9000/

YOLO History

YOLO (You Only Look Once), a popular object detection and image segmentation model, was developed by Joseph Redmon and Ali Farhadi at the University of Washington. Launched in **2015**, YOLO quickly gained popularity for its high speed and accuracy.

•YOLOv2, released in **2016,** improved the original model by incorporating batch normalization, anchor boxes, and dimension clusters.

•YOLOv3, launched in **2018**, is called **Darknet-53** further enhanced the model's performance using a more efficient backbone network, multiple anchors and spatial pyramid pooling.

YOLOv3: An Incremental Improvement

Joseph Redmon Ali Farhadi

University of Washington

YOLO History

YOLO (You Only Look Once), a popular object detection and image segmentation model, was developed by Joseph Redmon and Ali Farhadi at the University of Washington. Launched in **2015**, YOLO quickly gained popularity for its high speed and accuracy.

•YOLOv2, released in **2016**, improved the original model by incorporating batch normalization, anchor boxes, and dimension clusters.

•YOLOv3, launched in **2018**, is called **Darknet-53** further enhanced the model's performance using a more efficient backbone network, multiple anchors and spatial pyramid pooling.

•YOLOv4 was released in **2020**, introducing innovations like Mosaic data augmentation, a new anchor-free detection head, and a new loss function.

YOLOv4: Optimal Speed and Accuracy of Object Detection

Alexey Bochkovskiy*
alexeyab84@gmail.com

Chien-Yao Wang*
Institute of Information Science
Academia Sinica, Taiwan
kinyiu@iis.sinica.edu.tw

Hong-Yuan Mark Liao
Institute of Information Science
Academia Sinica, Taiwan
liao@iis.sinica.edu.tw

YOLO History

- **YOLOv4** was released in **2020**, introducing innovations like Mosaic data augmentation, a new anchor-free detection head, and a new loss function.
- **YOLOv5**, in **2020**, further improved the model's performance and added new features such as hyperparameter optimization, integrated experiment tracking and automatic export to popular export formats. YOLOv5 is open source and actively maintained by **Ultralytics**, with more than 250 contributors and new improvements frequently. YOLOv5 is easy to use, train and deploy. Ultralytics provide a mobile version for iOS and Android and many integrations for labeling, training, and deployment.

A DEEP LEARNING OBJECT DETECTION METHOD FOR AN EFFICIENT CLUSTERS INITIALIZATION

Raphaël Couturier
Univ. Bourgogne Franche-Comté (UBFC),
FEMTO-ST Institute,
France

Hassan N. Noura
Univ. Bourgogne Franche-Comté (UBFC),
FEMTO-ST Institute,
France

Ola Salman
American University of Beirut,
Electrical and Computer Engineering Department,
Beirut 1107 2020, Lebanon

Abderrahmane Sider
Laboratoire LIMED, University of Bejaia, Algeria

YOLO History

•**YOLOv6** was open-sourced by Meituan in **2022** and is in use in many of the company's autonomous delivery robots.

YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications

Chuyi Li* Lulu Li* Hongliang Jiang* Kaiheng Weng* Yifei Geng* Liang Li*
Zaidan Ke* Qingyuan Li* Meng Cheng* Weiqiang Nie* Yiduo Li* Bo Zhang*
Yufei Liang Linyuan Zhou Xiaoming Xu† Xiangxiang Chu Xiaoming Wei Xiaolin Wei
Meituan Inc.
{lichuyi, lilulu05, jianghongliang02, wengkaiheng, gengyifei,liliang58, kezaidan, liqingyuan02, chengmeng05, nieweiqiang, liyiduo, zhangbo97, liangyufei, zhoulinyuan, xuxiaoming04, chuxiangxiang, weixiaoming, weixiaolin02}@meituan.com

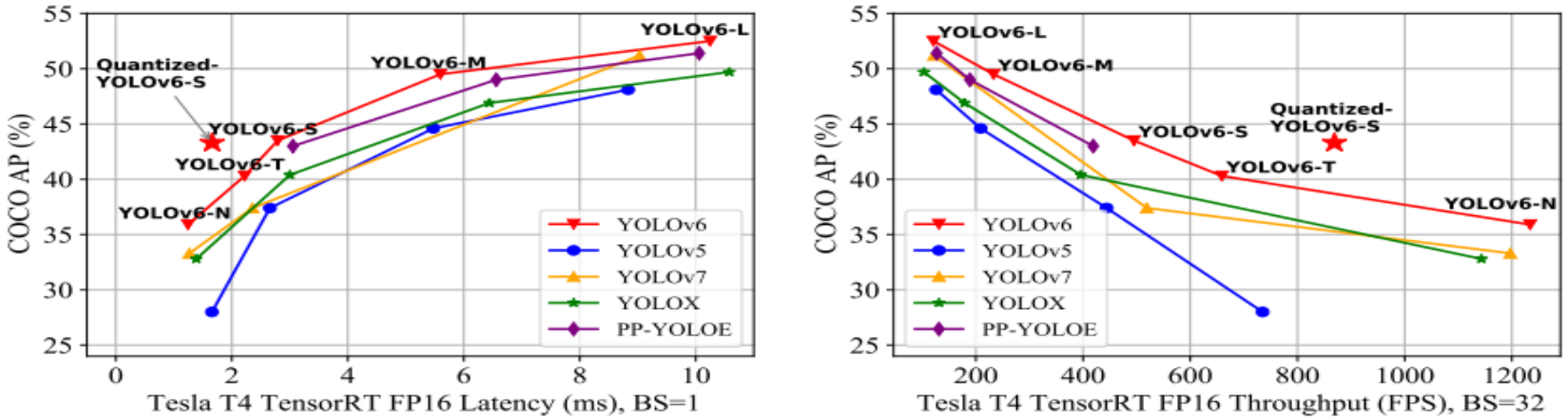


Figure 1: Comparison of state-of-the-art efficient object detectors. Both latency and throughput (at a batch size of 32) are given for a handy reference. All models are test with TensorRT 7 except that the quantized model is with TensorRT 8.

YOLO History

- **YOLOv6** was open-sourced by Meituan in **2022** and is in use in many of the company's autonomous delivery robots.
- **YOLOv7** was published in ArXiv in July **2022** by the same authors of YOLOv4 and YOLOR, added additional tasks such as pose estimation on the COCO keypoints dataset.

YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors

Chien-Yao Wang[1], Alexey Bochkovskiy, and Hong-Yuan Mark Liao[1]
[1]Institute of Information Science, Academia Sinica, Taiwan
kinyiu@iis.sinica.edu.tw, alexeyab84@gmail.com, and liao@iis.sinica.edu.tw

YOLO History

•**YOLOv6** was open-sourced by Meituan in **2022** and is in use in many of the company's autonomous delivery robots.
•**YOLOv7** was published in ArXiv in July **2022** by the same authors of YOLOv4 and YOLOR, added additional tasks such as pose estimation on the COCO keypoints dataset.
•**YOLOv8** was released in January **2023** by Ultralytics, the company that developed YOLOv5, is the latest version of YOLO by Ultralytics. YOLOv8 [100] was released in January 2023 by Ultralytics, the company that developed YOLOv5.As a cutting-edge, state-of-the-art (SOTA) model, YOLOv8 builds on the success of previous versions, introducing new features and improvements for enhanced performance, flexibility, and efficiency. YOLOv8 supports a full range of vision AI tasks, including detection, segmentation, pose estimation, tracking, and classification. This versatility allows users to leverage YOLOv8's capabilities across diverse applications and domains.

➢ All About YOLOs

A Comprehensive Review of YOLO: From YOLOv1 to YOLOv8 and Beyond

Under review in ACM Computing Surveys

Juan R. Terven
CICATA-Qro
Instituto Politecnico Nacional
Mexico
jrtervens@ipn.mx

Diana M. Cordova-Esparaza
Facultad de Informática
Universidad Autónoma de Querétaro
Mexico
diana.cordova@uaq.mx

April 4, 2023

Abstract

YOLO has become a central real-time object detection system for robotics, driverless cars, and video monitoring applications. We present a comprehensive analysis of YOLO's evolution, examining the innovations and contributions in each iteration from the original YOLO to YOLOv8. We start by describing the standard metrics and postprocessing; then, we discuss the major changes in network architecture and training tricks for each model. Finally, we summarize the essential lessons from YOLO's development and provide a perspective on its future, highlighting potential research directions to enhance real-time object detection systems.

https://arxiv.org/pdf/2304.00501.pdf

YOLO Algorithms

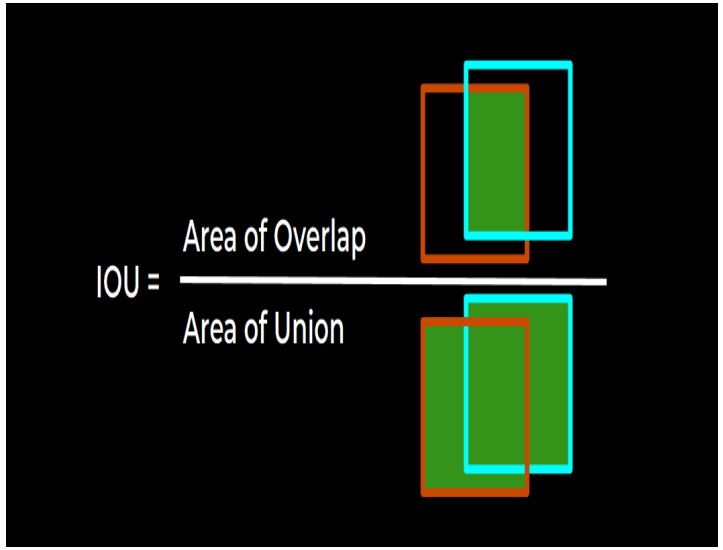


Figure 3: Non-Maximum Suppression (NMS). a) Shows the typical output of an object detection model containing multiple overlapping boxes. b) Shows the output after NMS.

AP@IoU=0.50	AP50	0.50
AP@IoU=0.55	AP55	0.55
AP@IoU=0.60	AP60	0.60
AP@IoU=0.65	AP65	0.65
AP@IoU=0.70	AP70	0.70
AP@IoU=0.75	AP75	0.75
AP@IoU=0.80	AP80	0.80

- IOU : Intersection Over Union
- Microsoft COCO (Common Objects in Context)
- PASCAL VOC (Visual Object Challenge)
- The Average Precision (AP), traditionally called Mean Average Precision (mAP), is the commonly used metric for evaluating the performance of object detection models. It measures the average precision across all categories, providing a single value to compare different models.
- Non-Maximum Suppression (NMS) is a post-processing technique used in object detection algorithms to reduce the number of overlapping bounding boxes and improve the overall detection quality

YOLO Algorithms

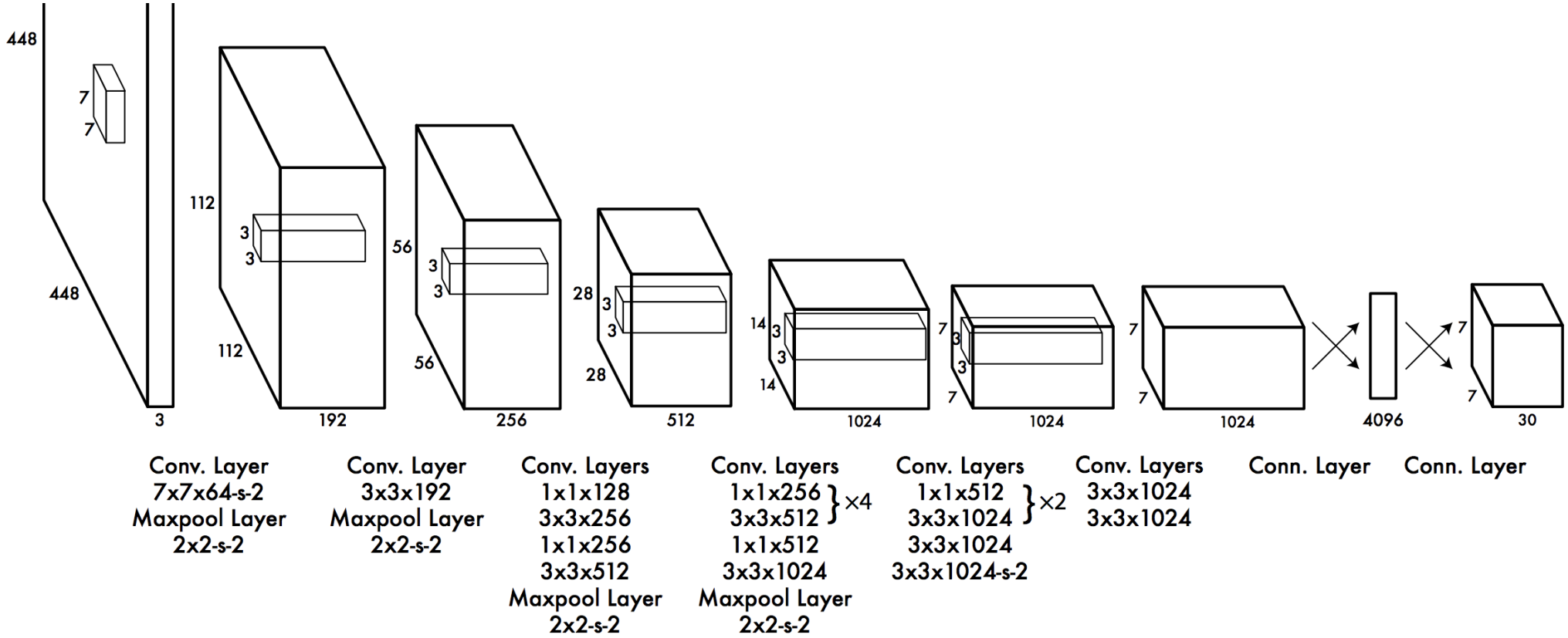
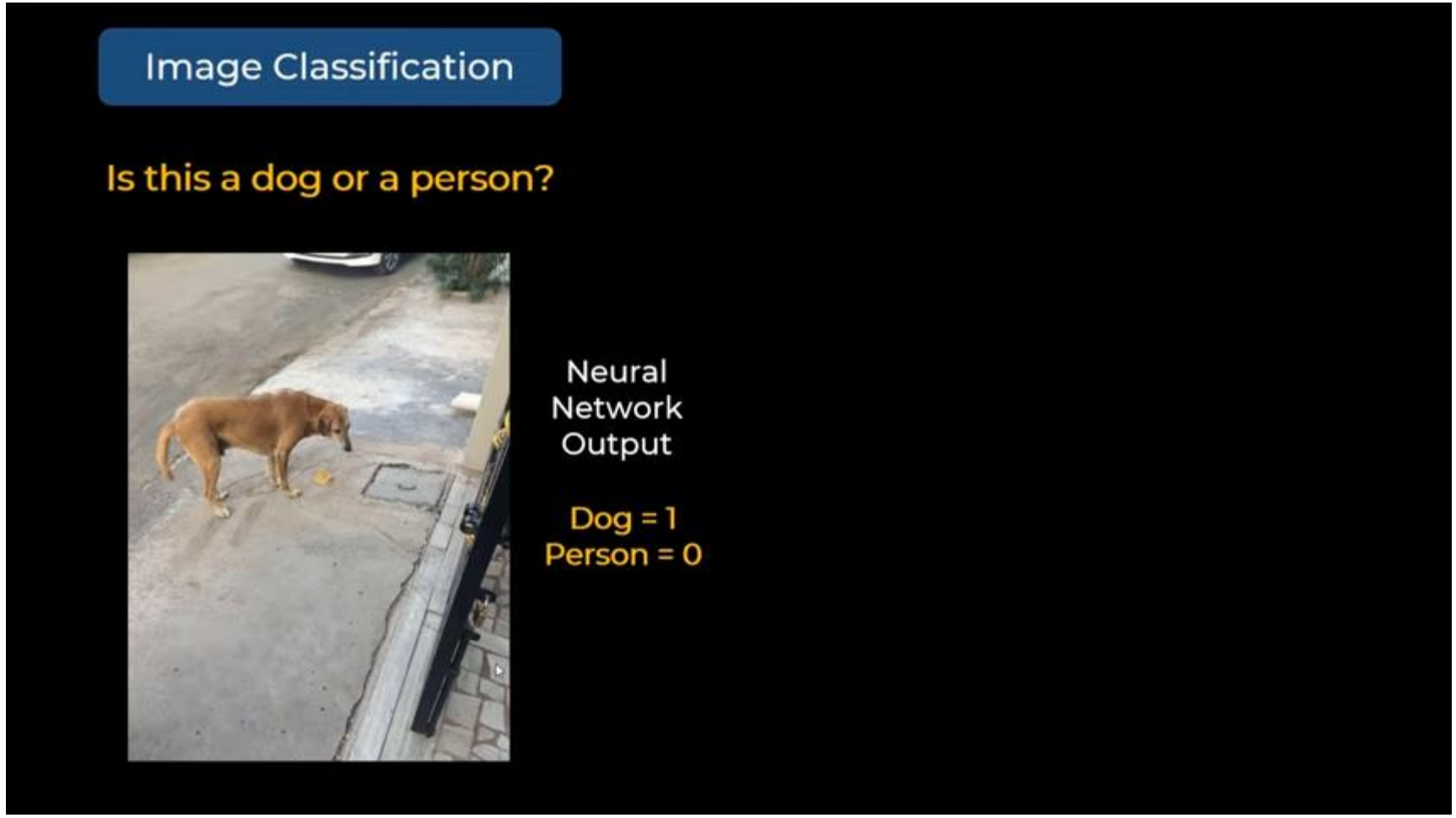
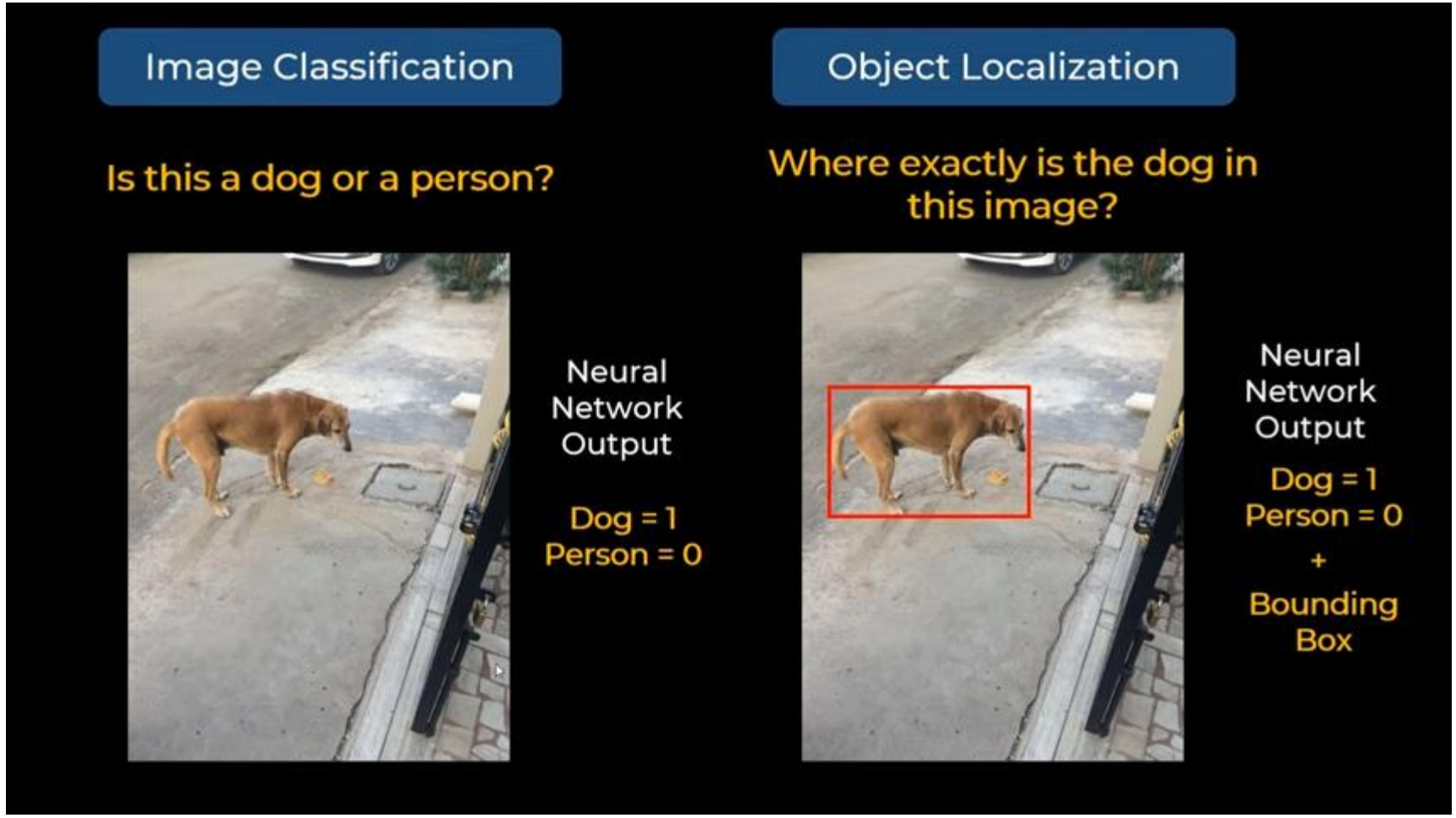


Figure 3: The Architecture. Our detection network has 24 convolutional layers followed by 2 fully connected layers. Alternating 1×1 convolutional layers reduce the features space from preceding layers. We pretrain the convolutional layers on the ImageNet classification task at half the resolution (224×224 input image) and then double the resolution for detection.

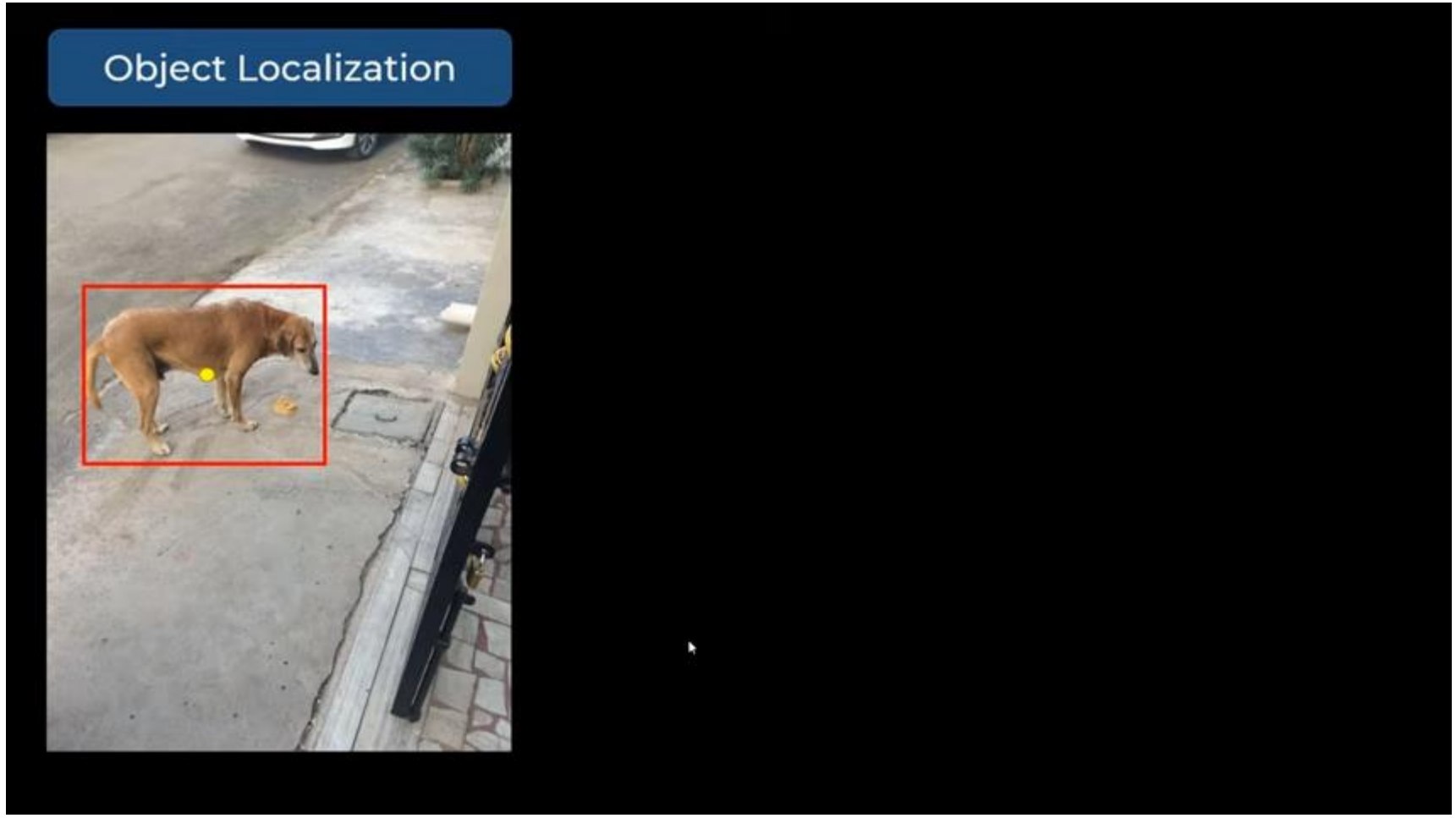
YOLO Algorithms



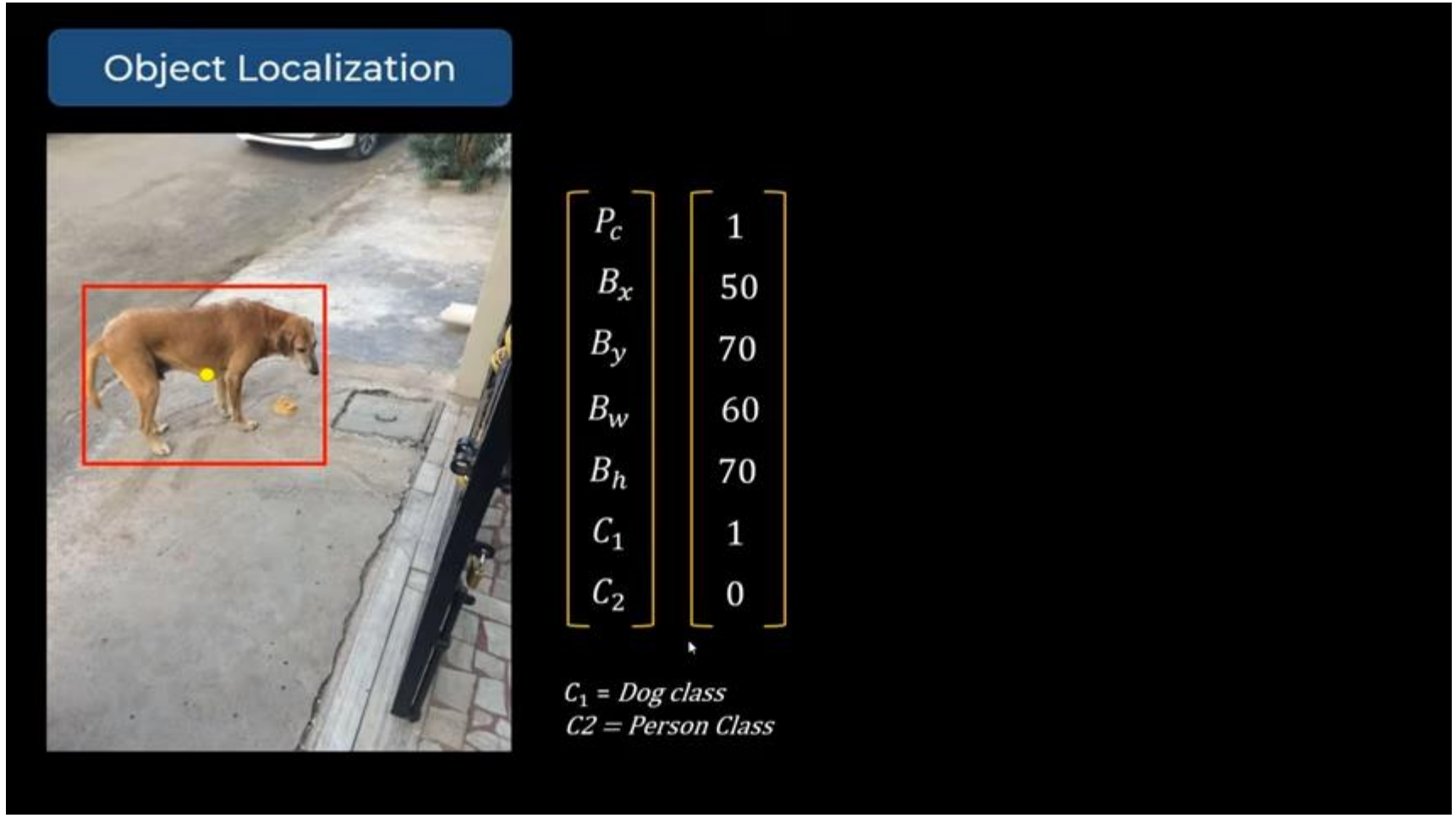
YOLO Algorithms



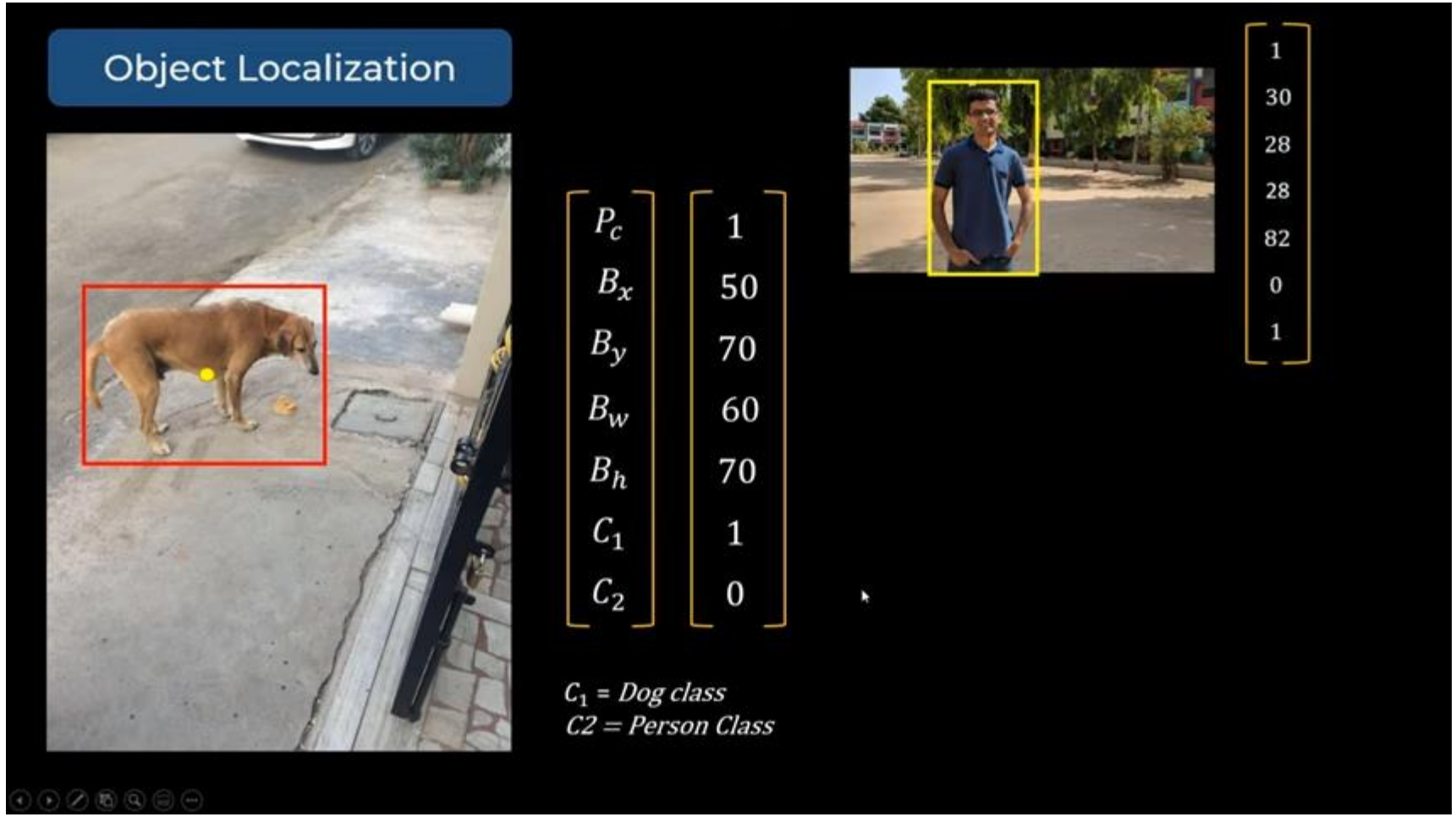
YOLO Algorithms



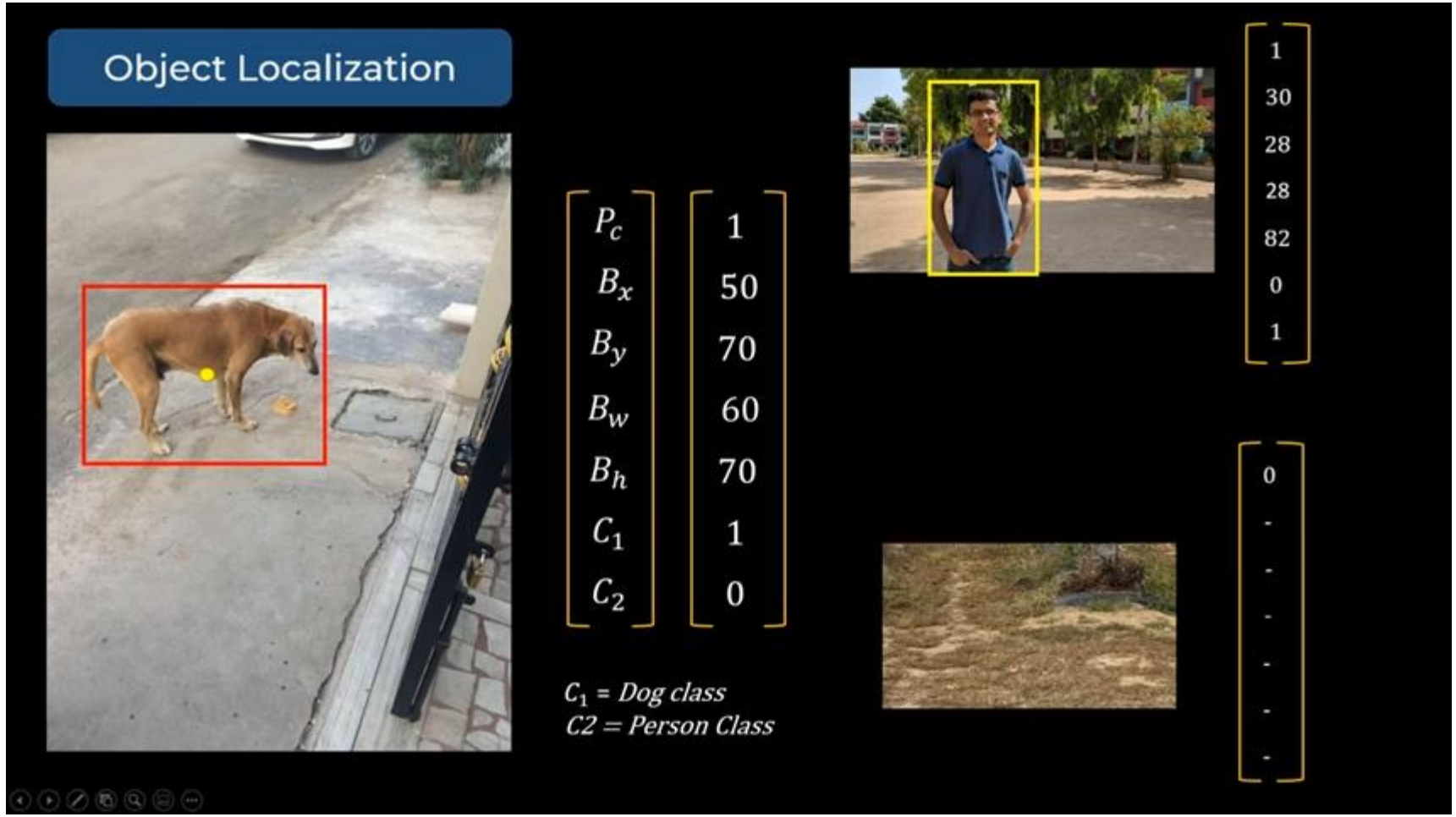
YOLO Algorithms



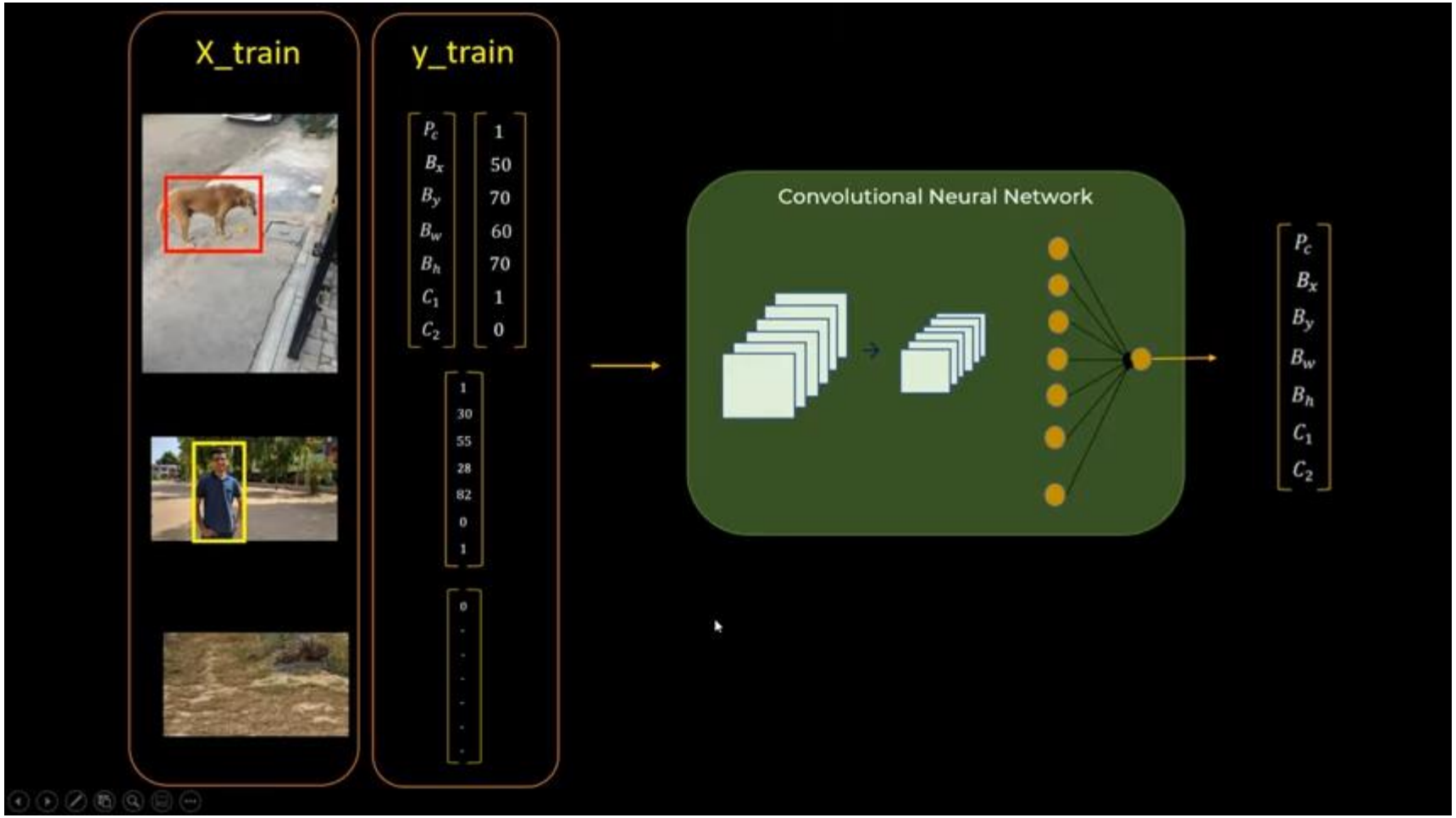
YOLO Algorithms



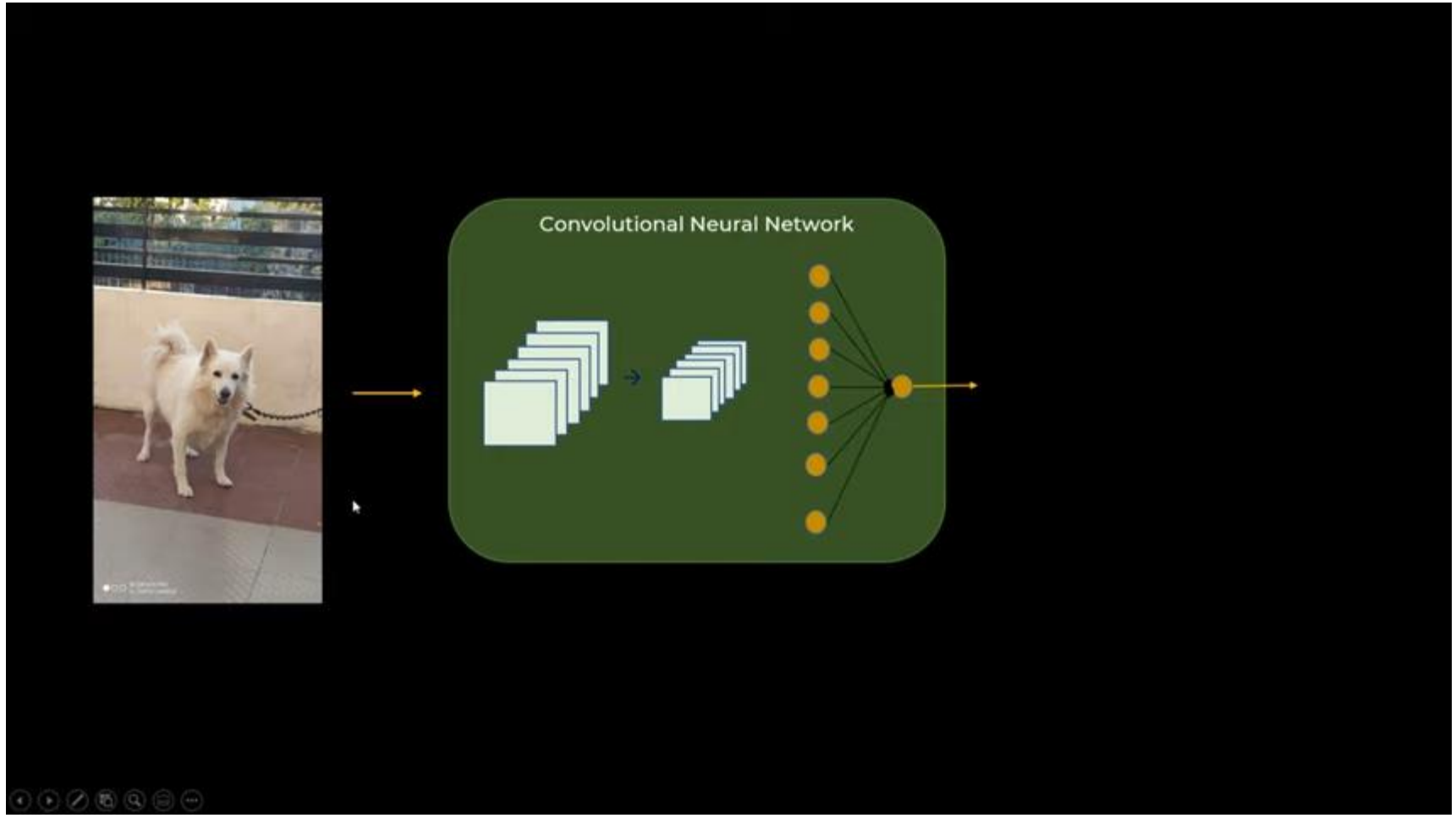
YOLO Algorithms



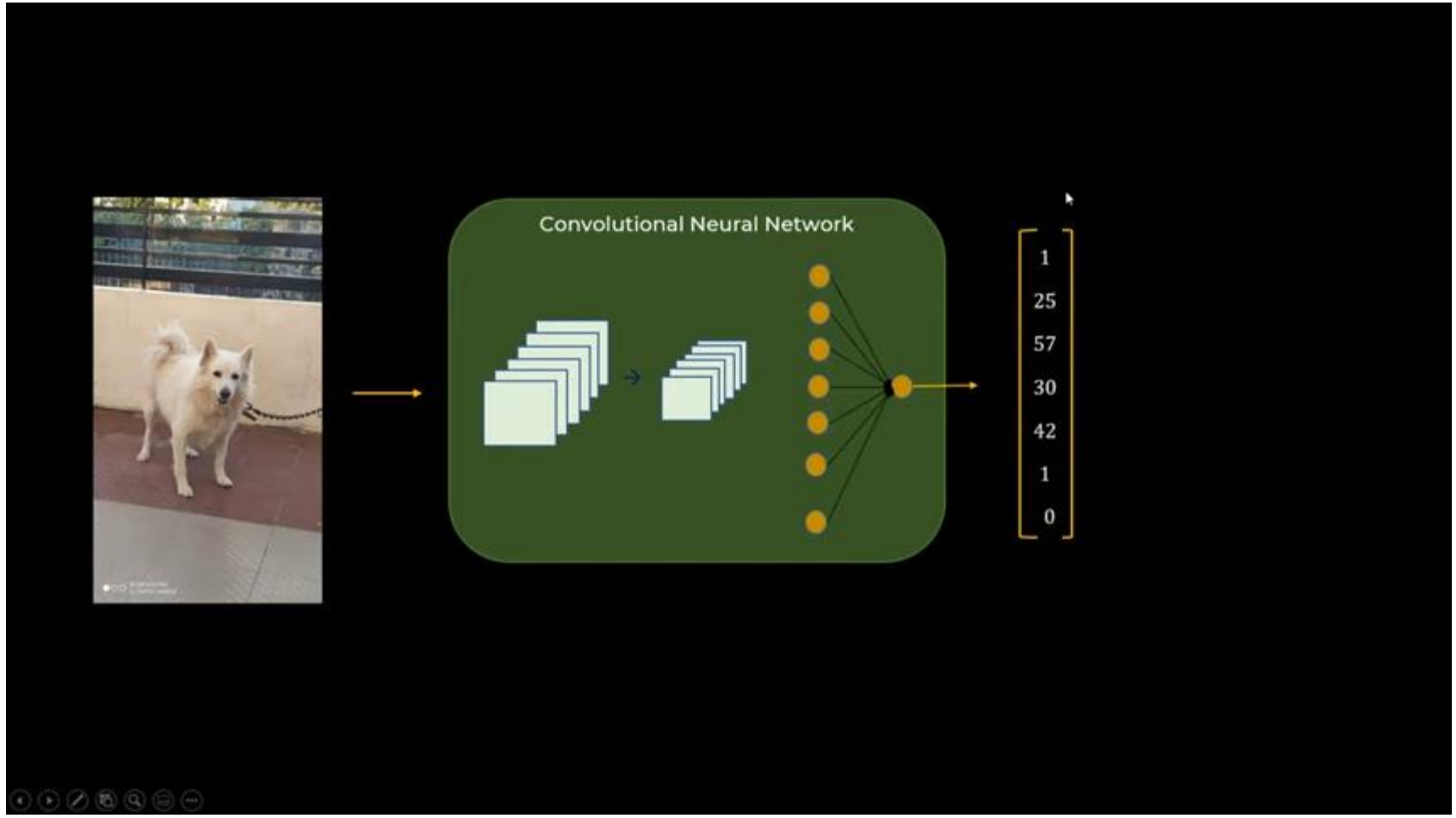
YOLO Algorithms



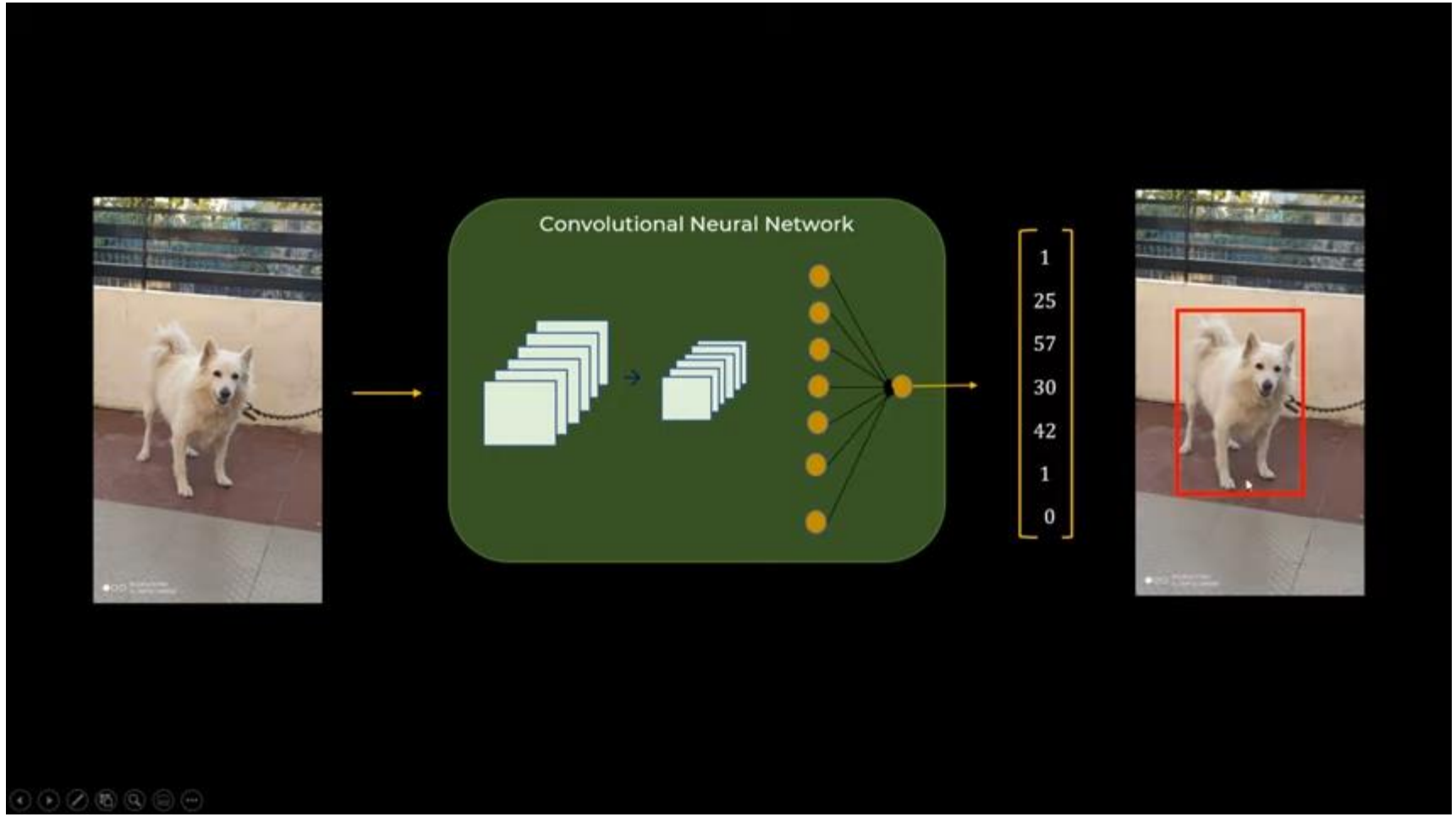
YOLO Algorithms



YOLO Algorithms



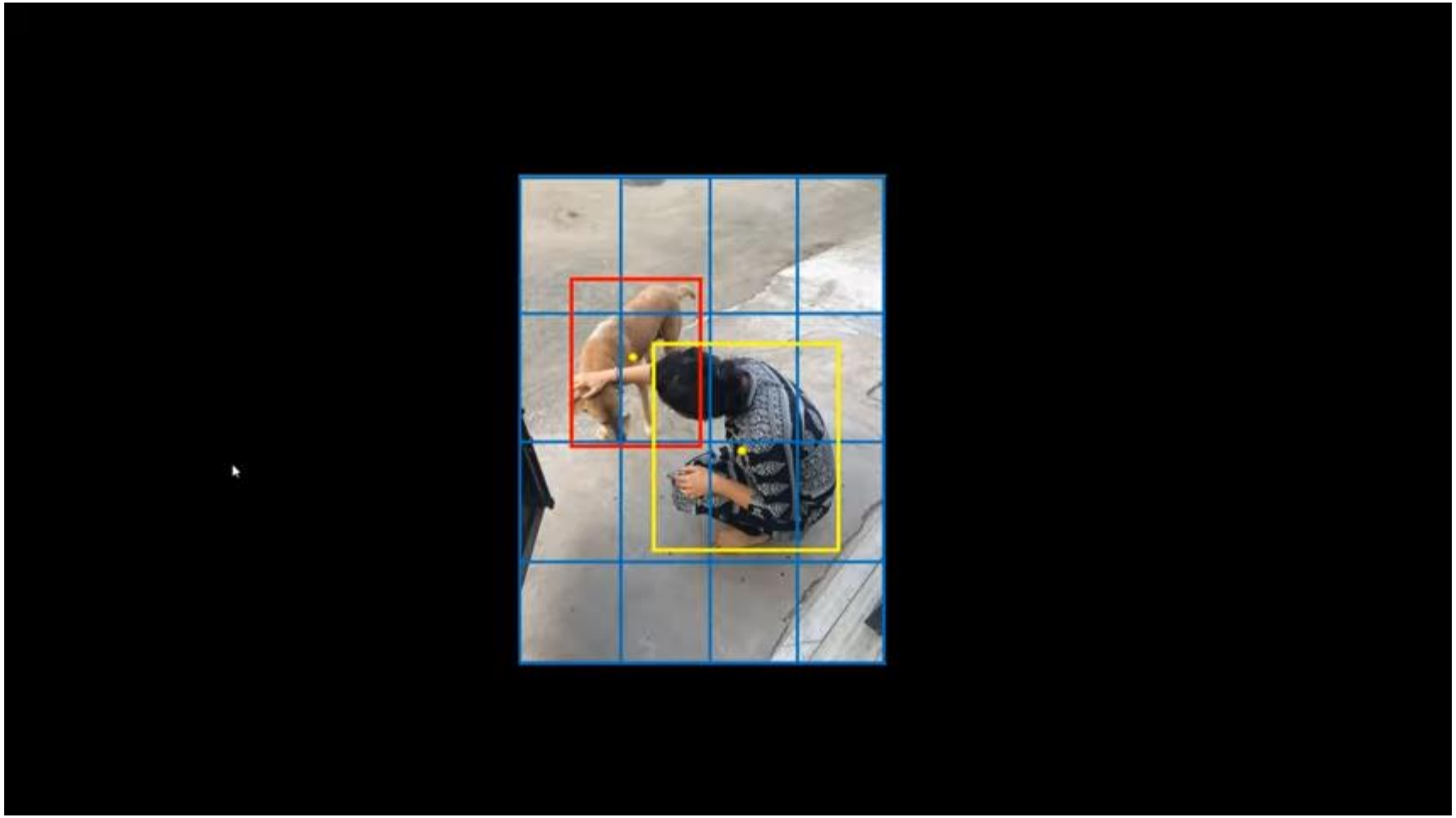
YOLO Algorithms



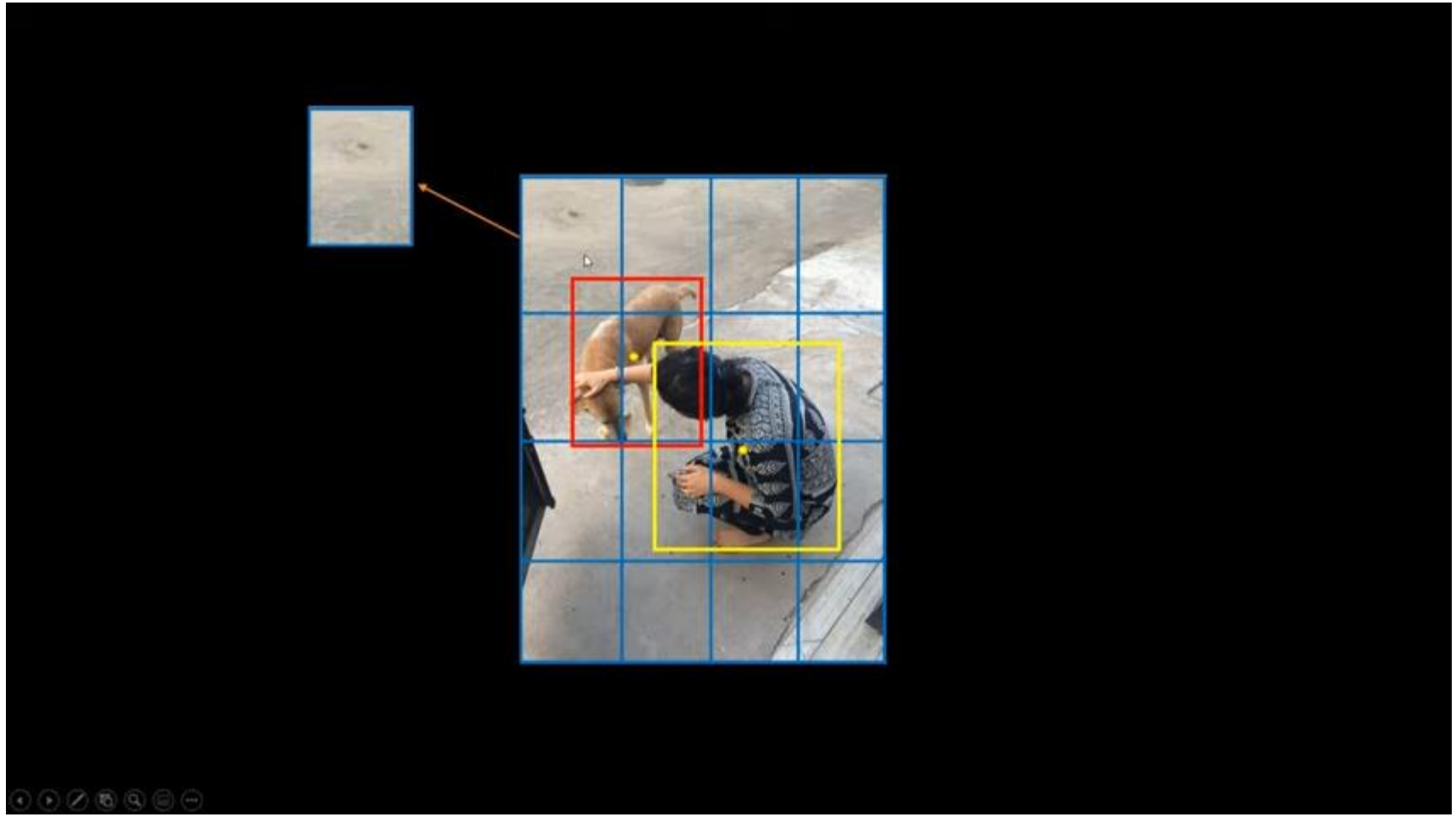
YOLO Algorithms

YOLO Algorithms

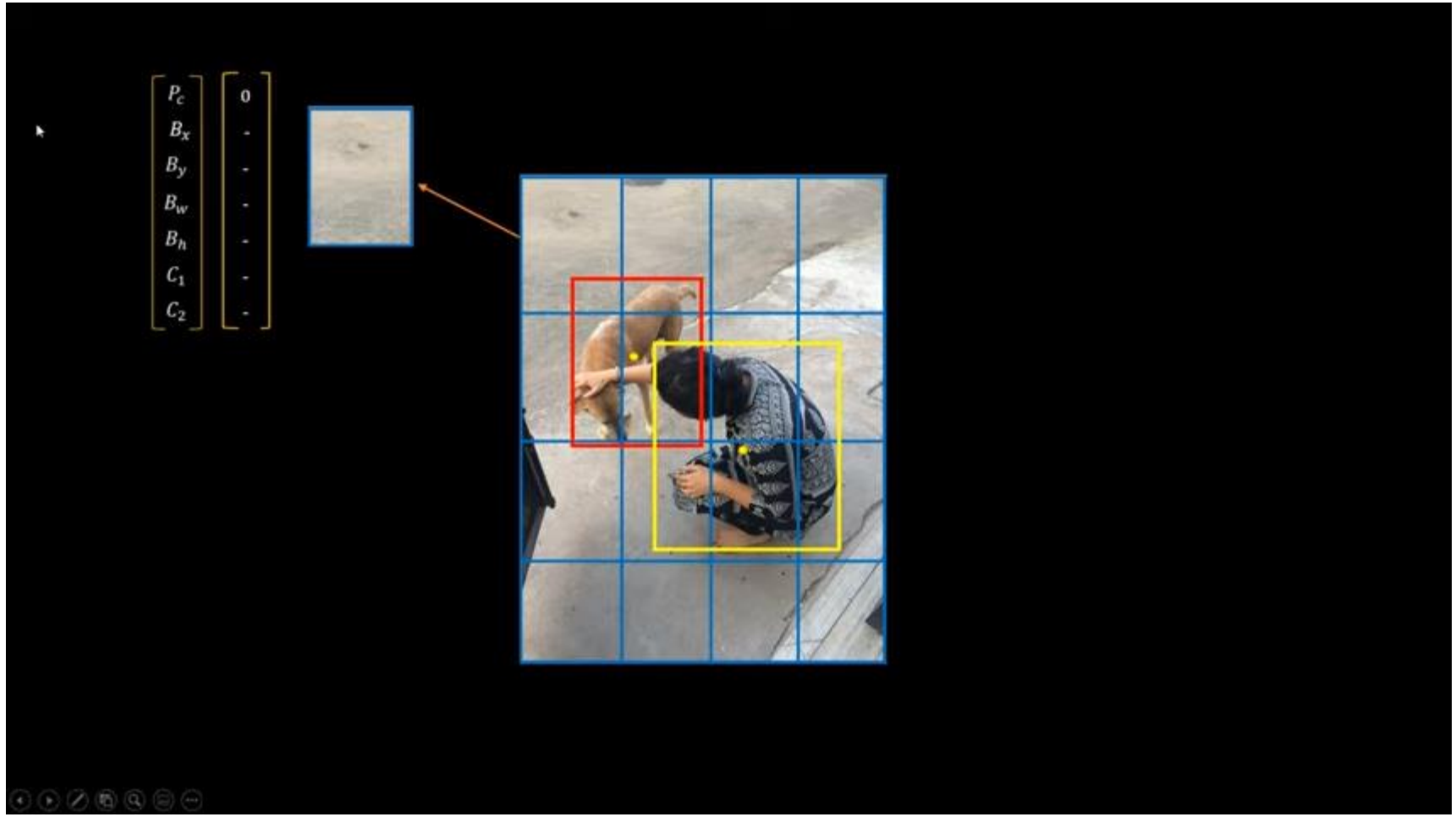
YOLO Algorithms



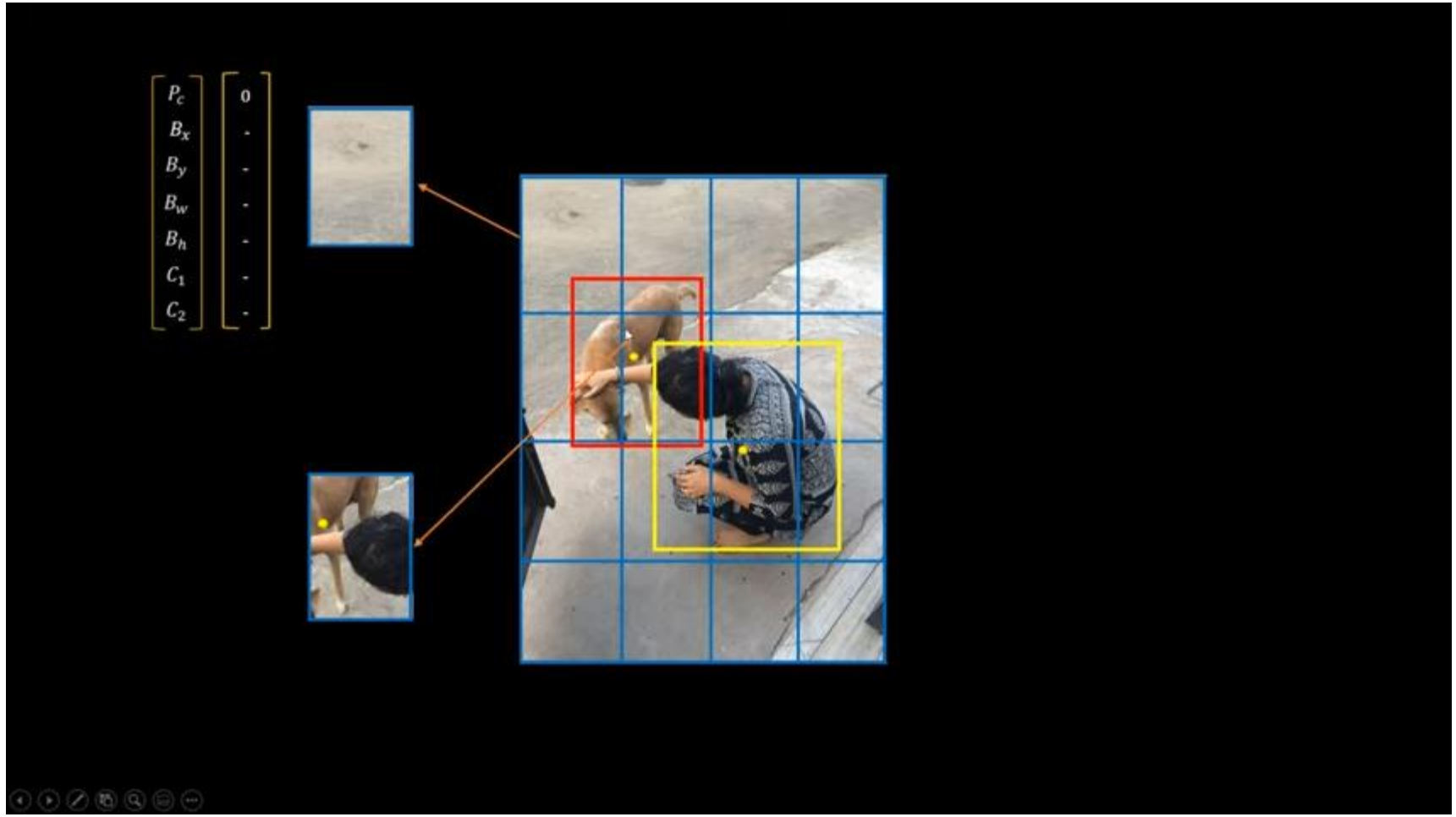
YOLO Algorithms



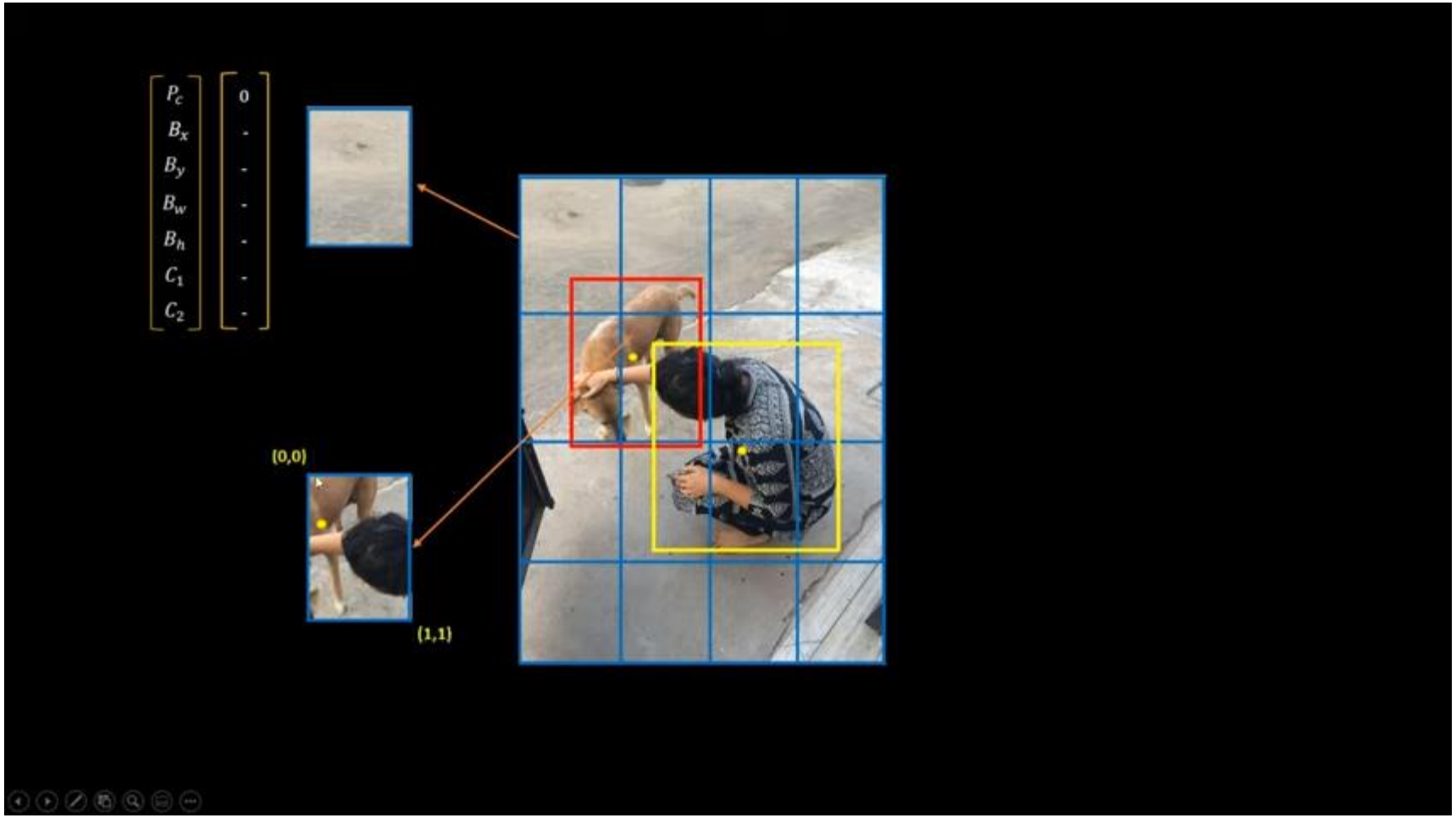
YOLO Algorithms



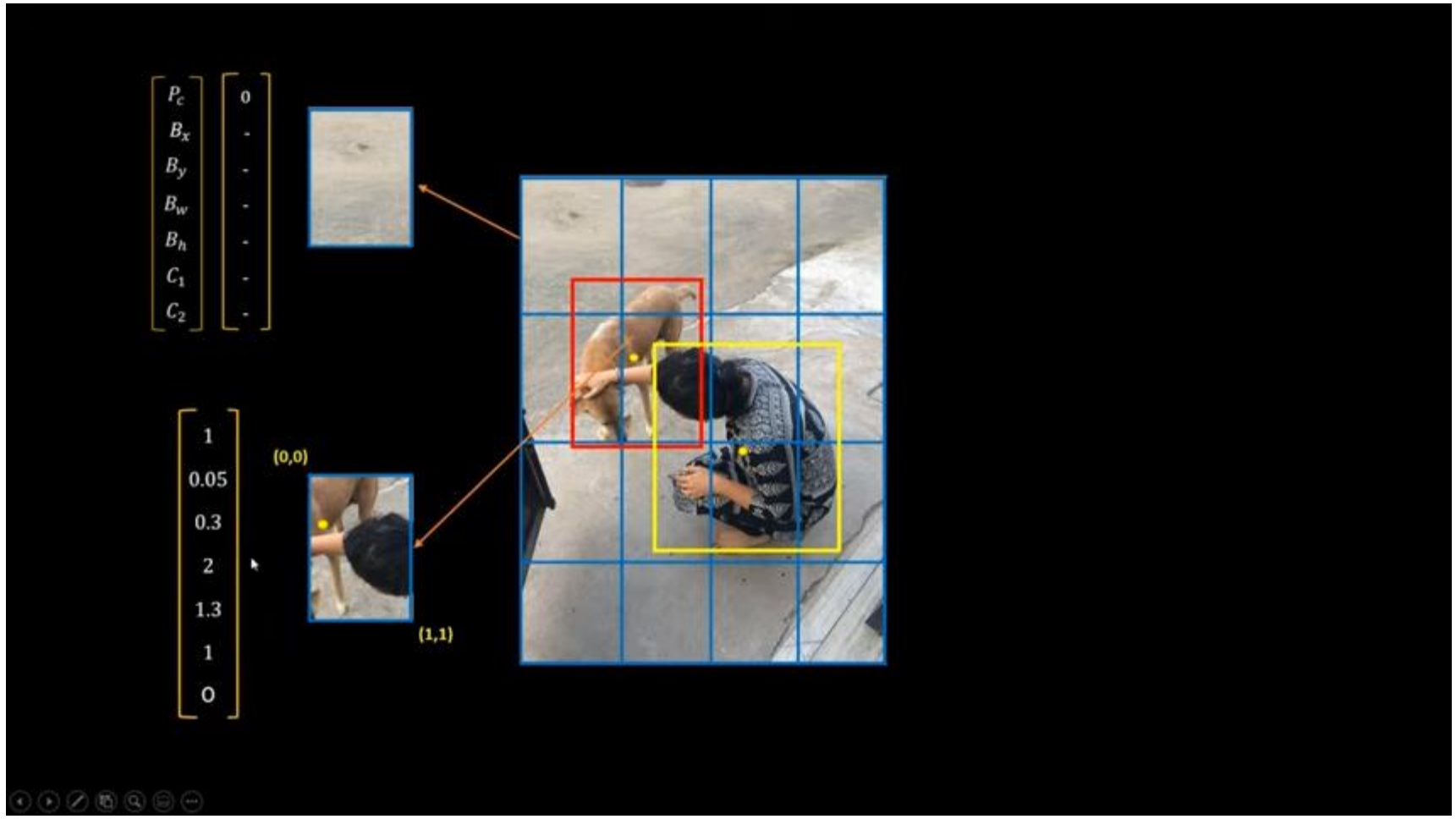
YOLO Algorithms



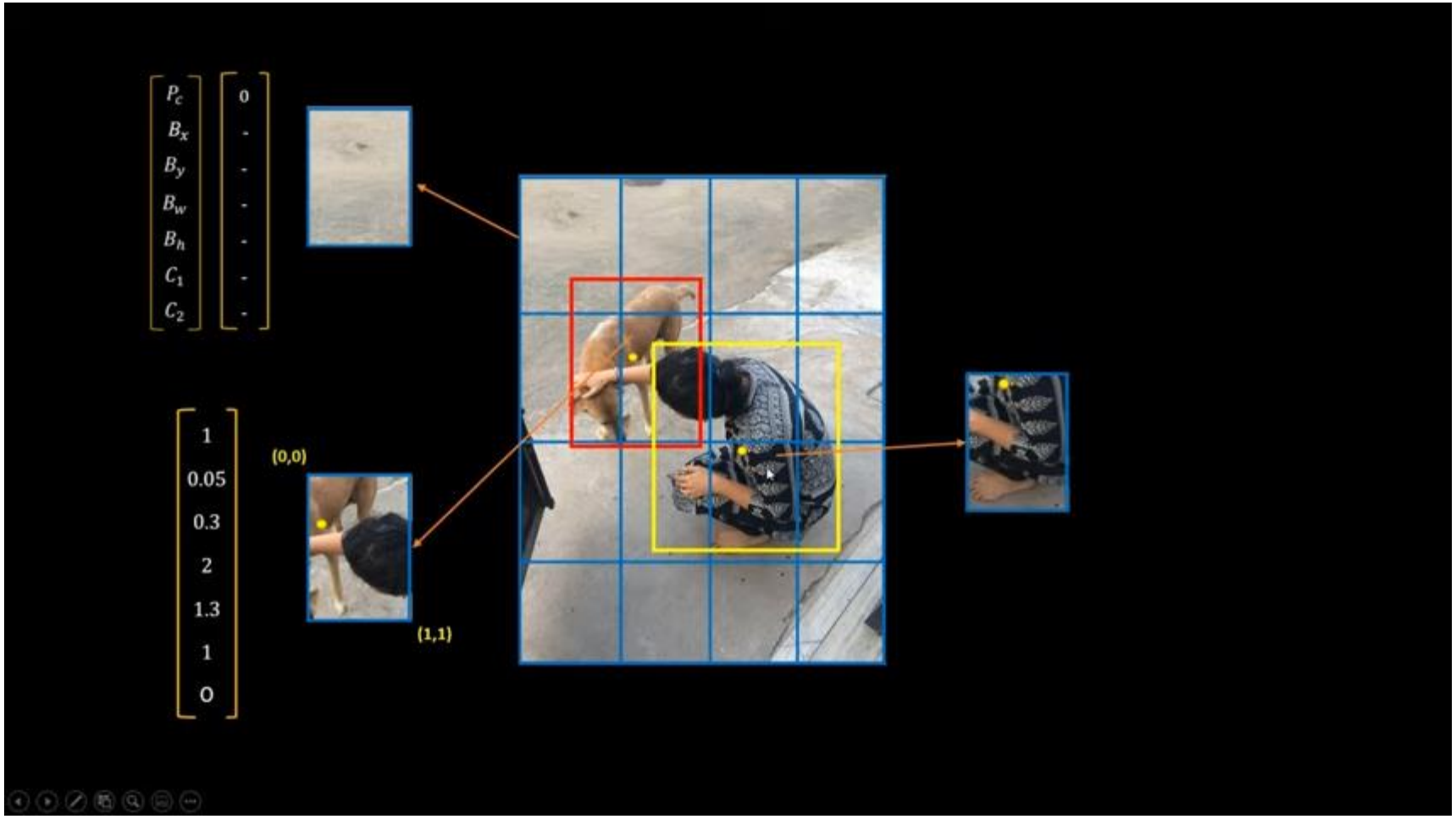
YOLO Algorithms



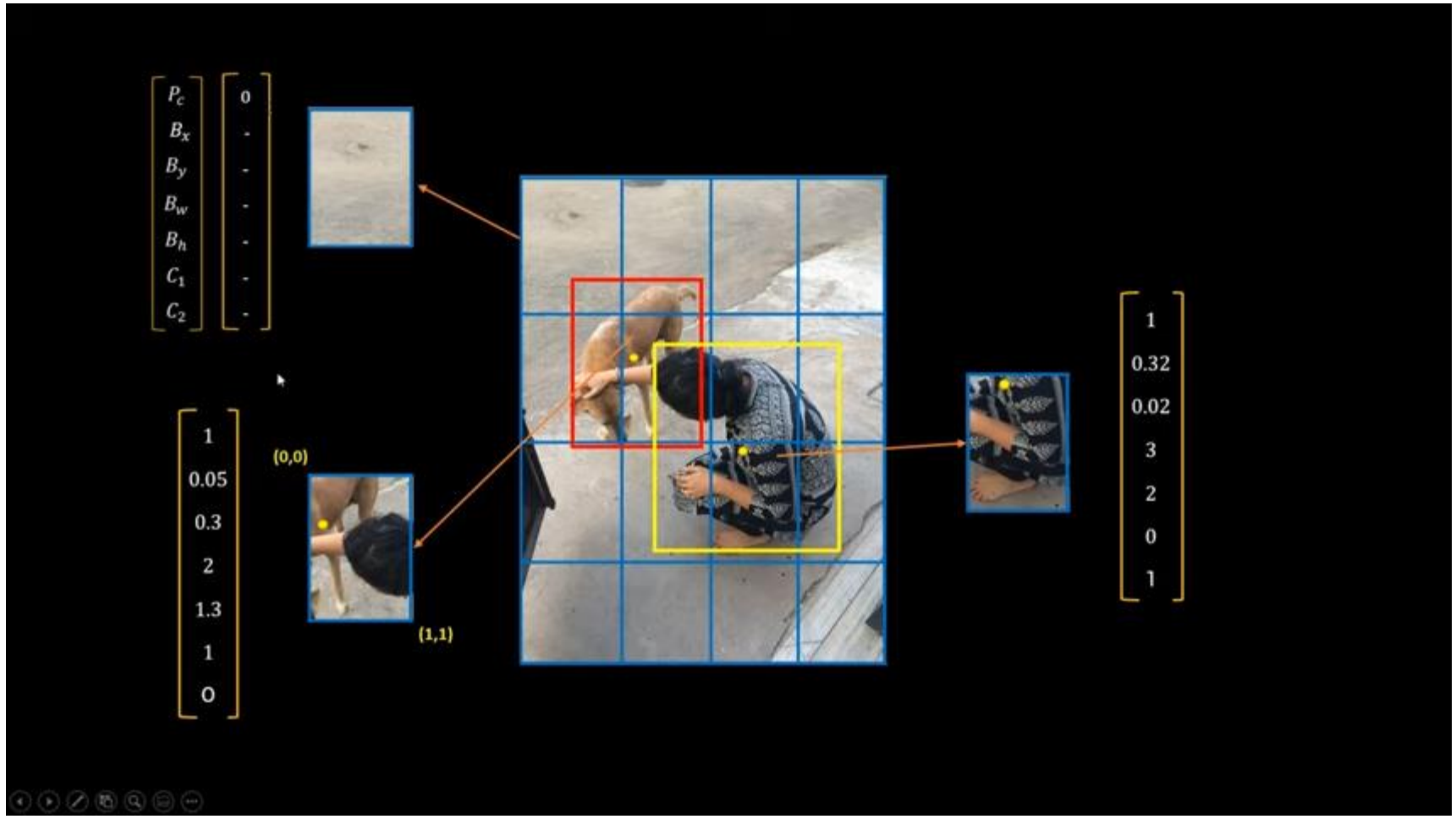
YOLO Algorithms



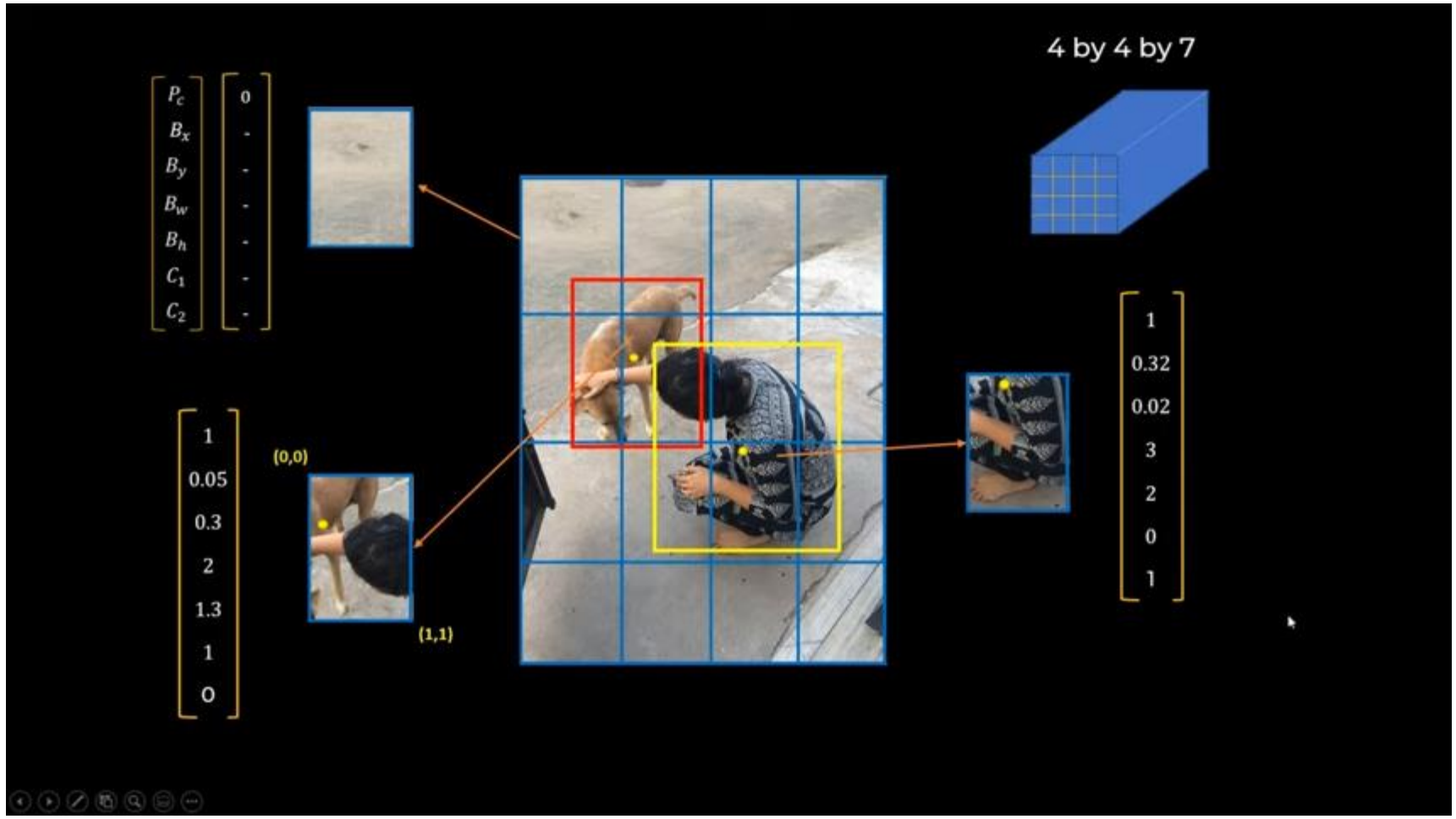
YOLO Algorithms



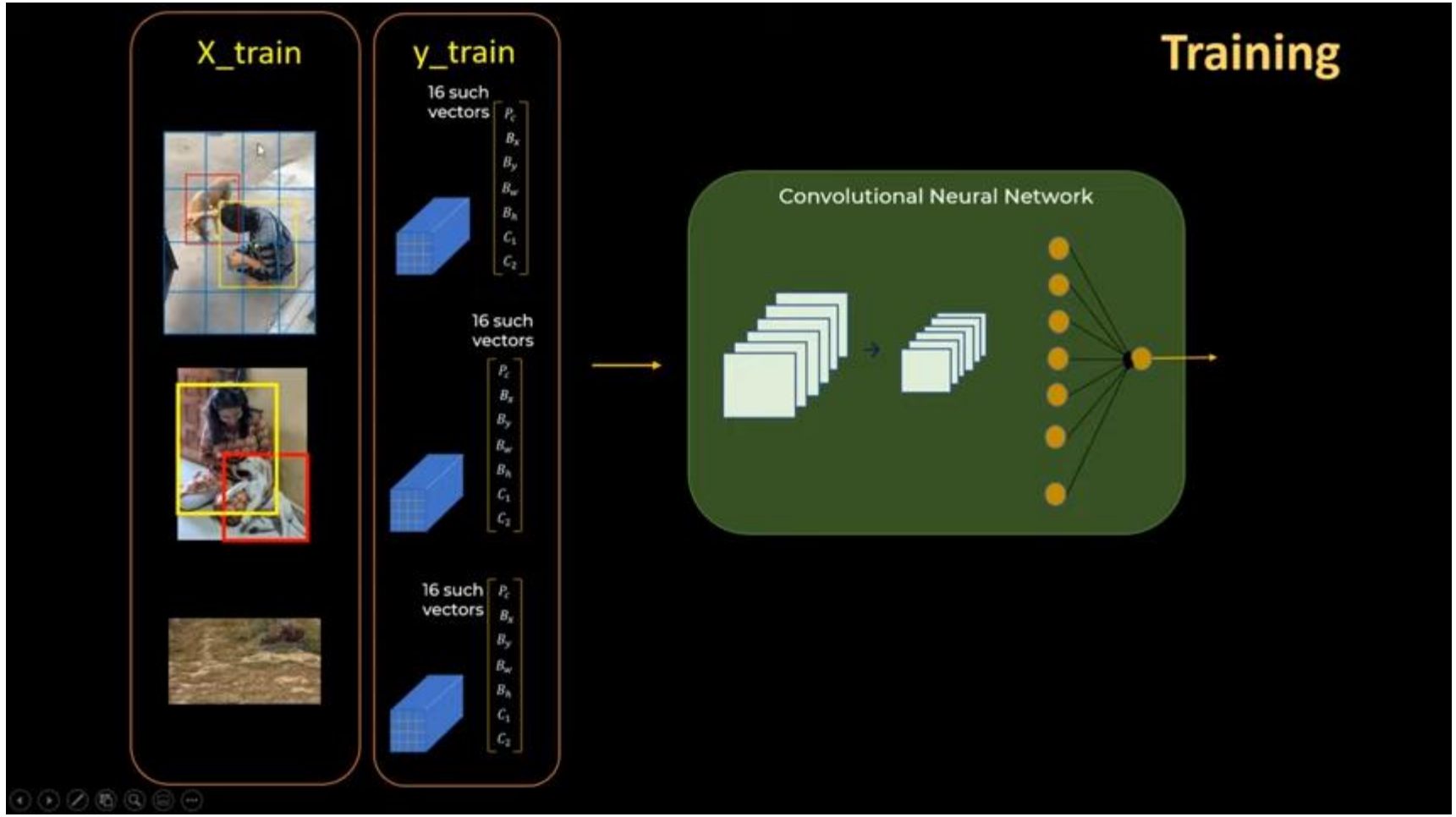
YOLO Algorithms



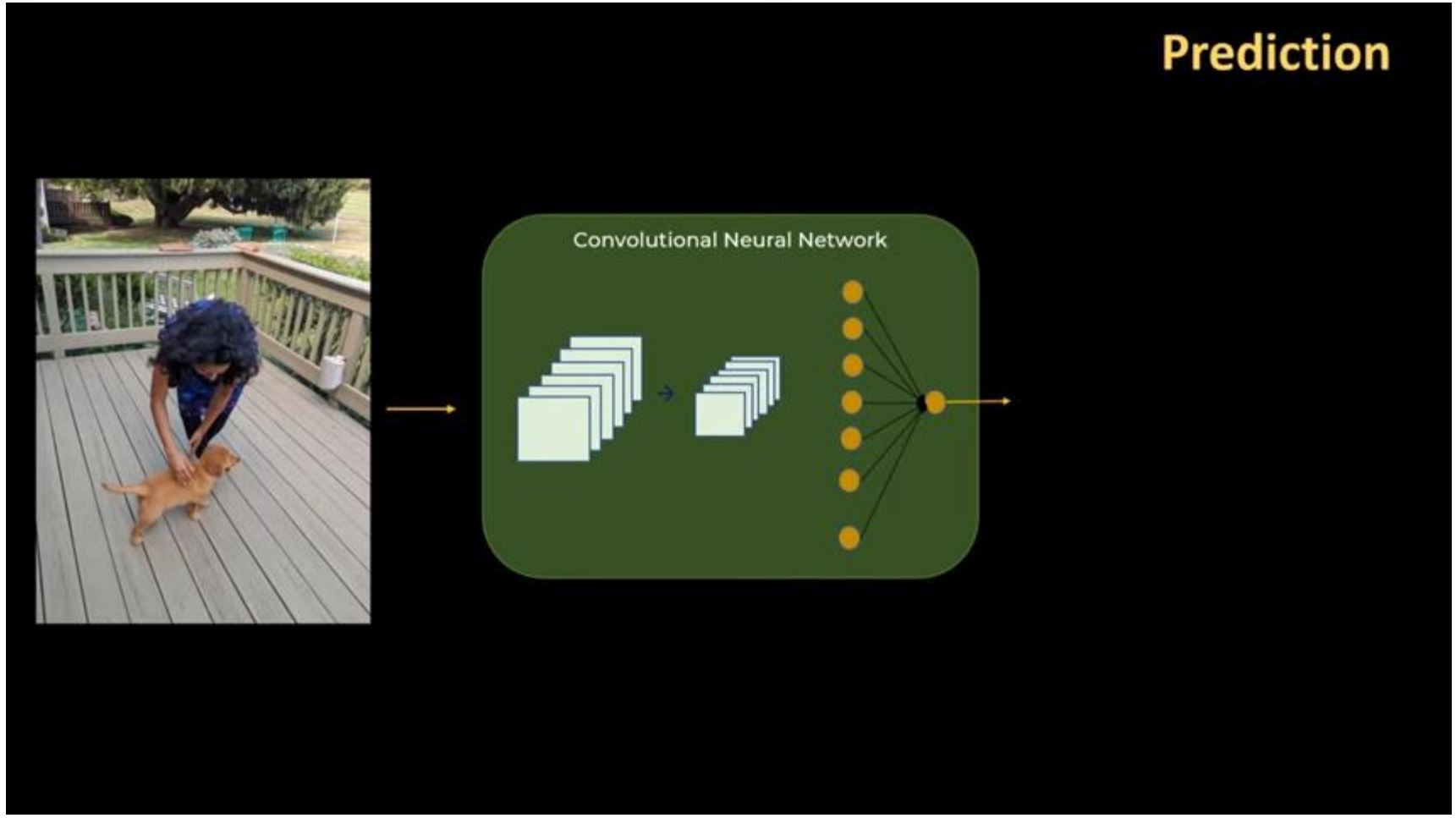
YOLO Algorithms



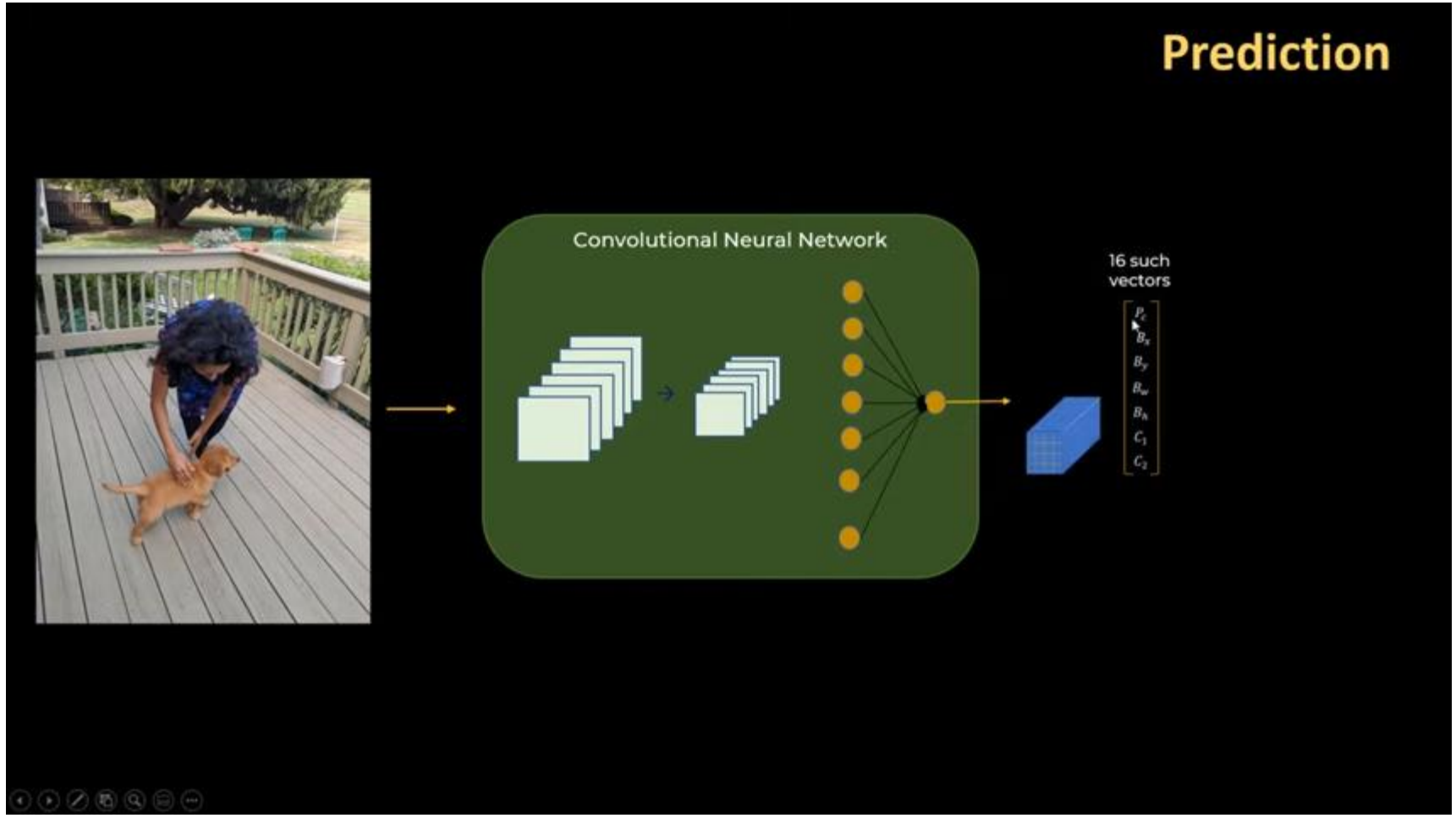
YOLO Algorithms



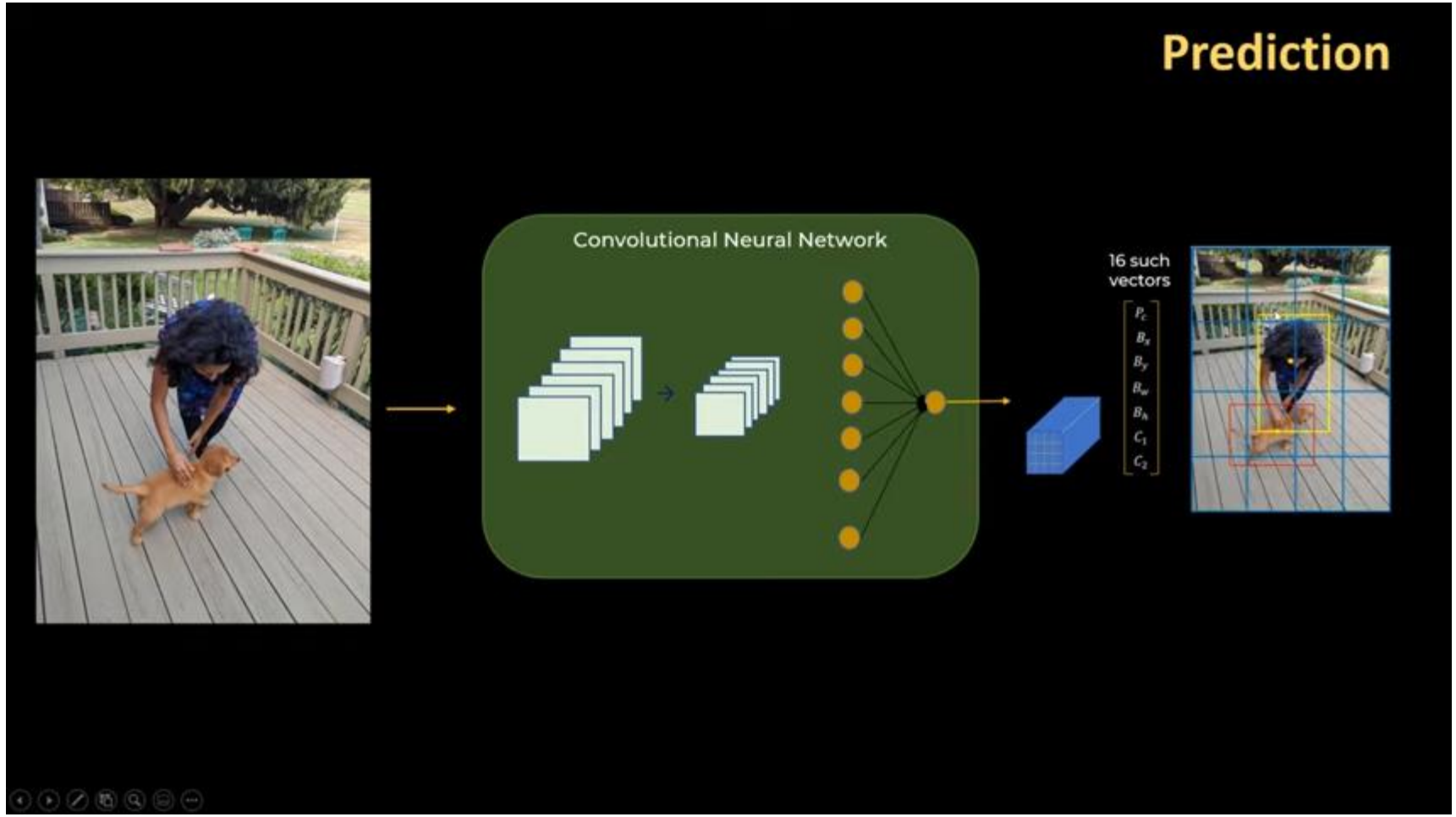
YOLO Algorithms



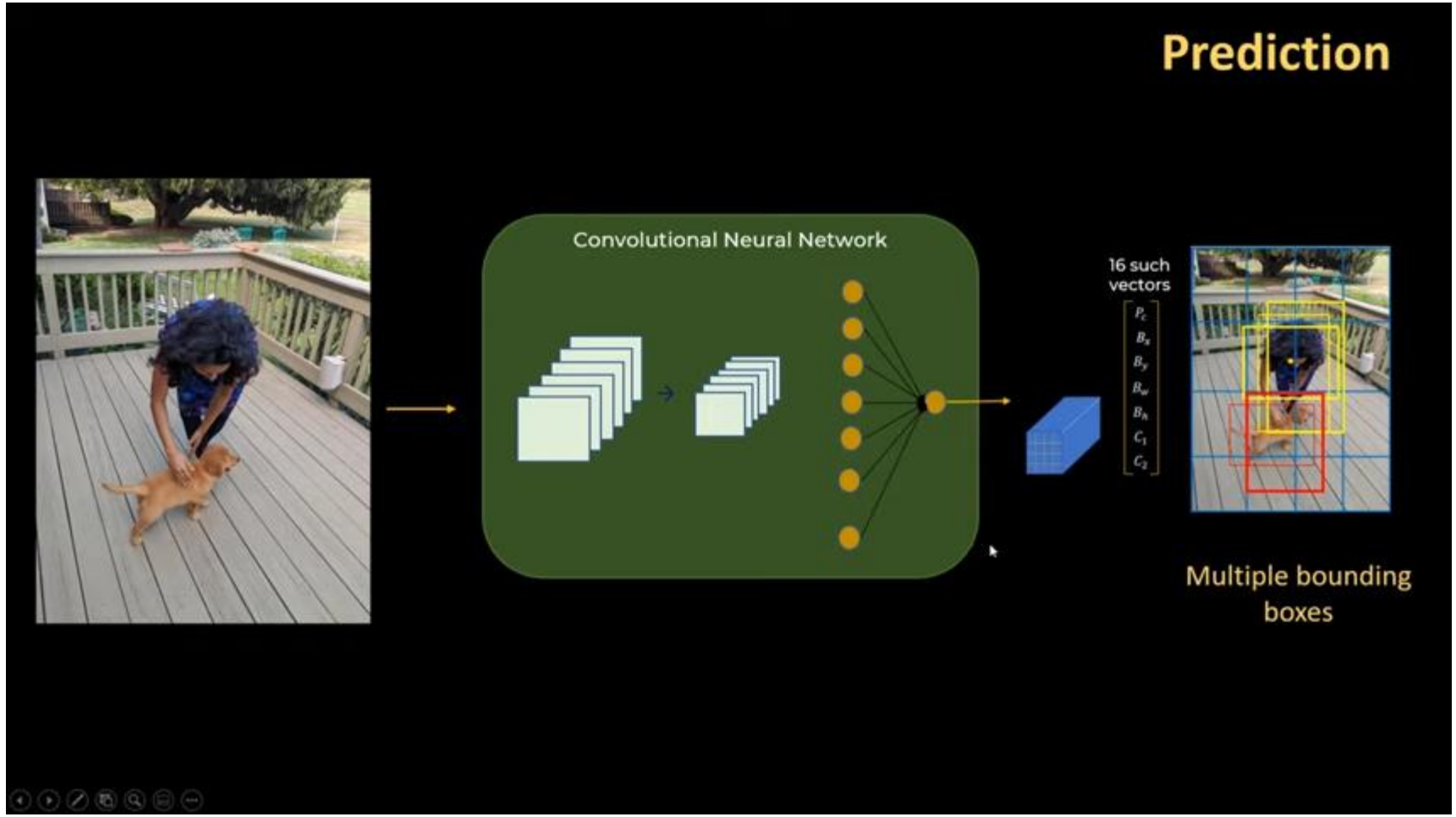
YOLO Algorithms



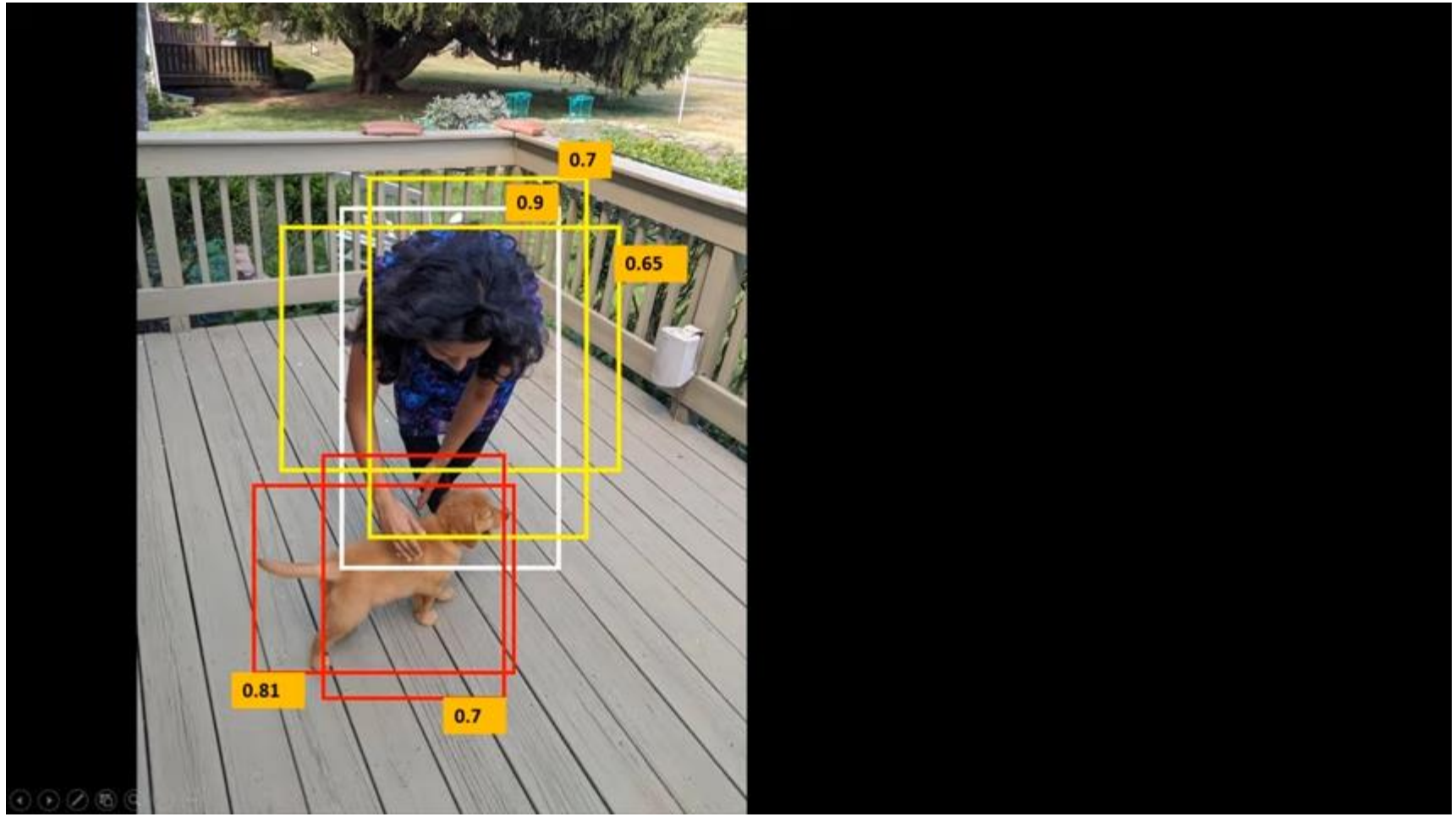
YOLO Algorithms



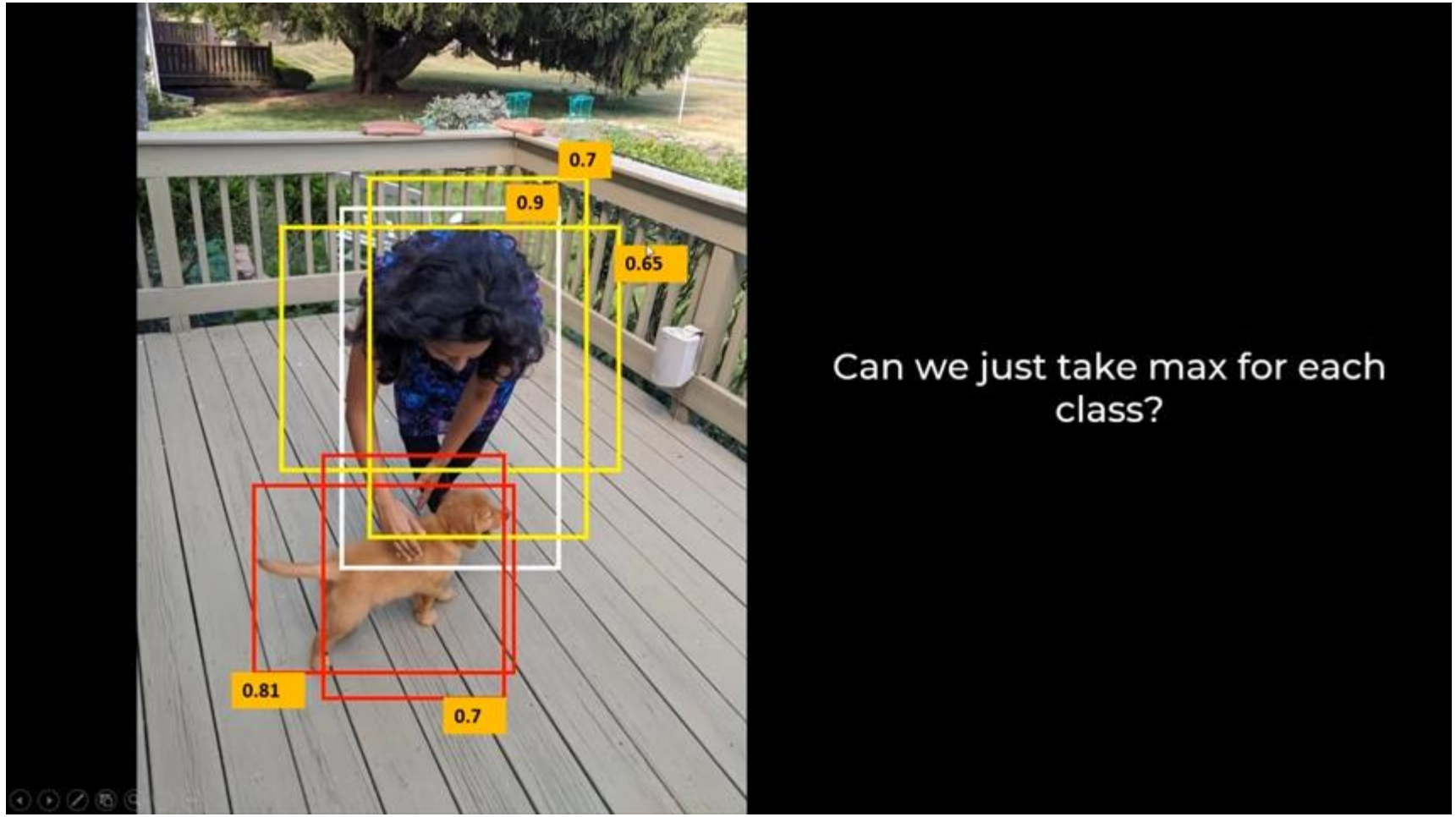
YOLO Algorithms



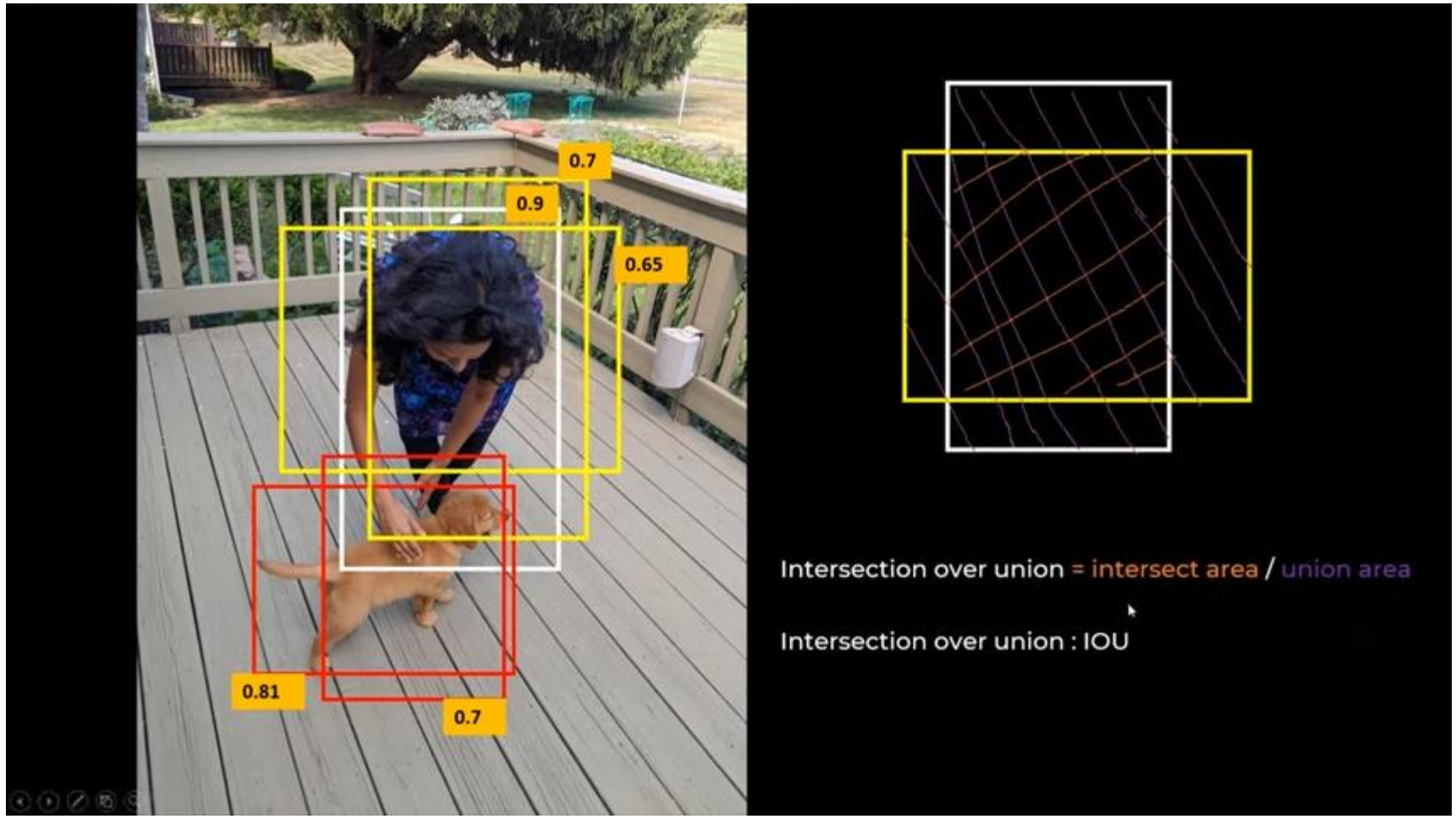
YOLO Algorithms



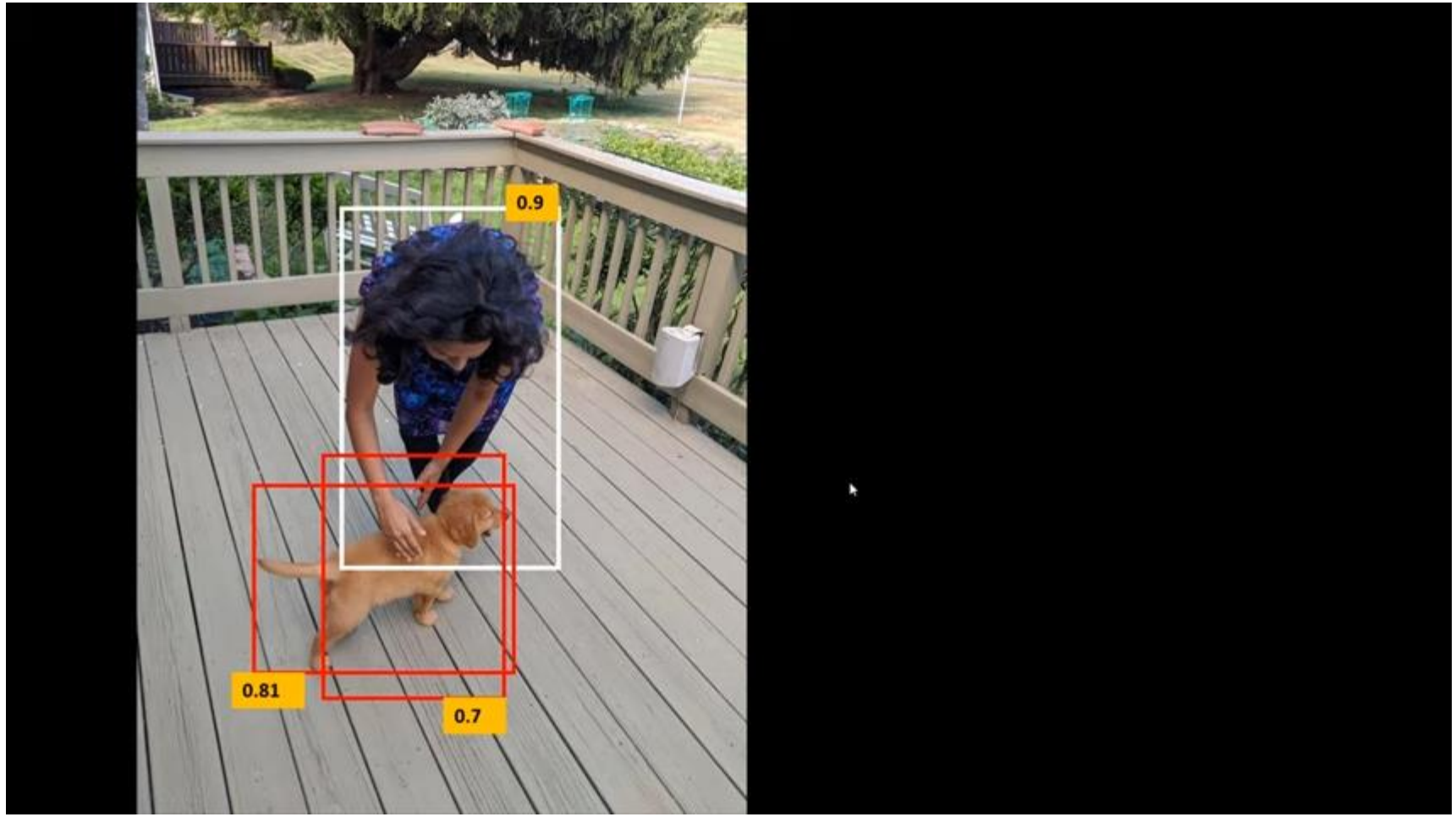
YOLO Algorithms



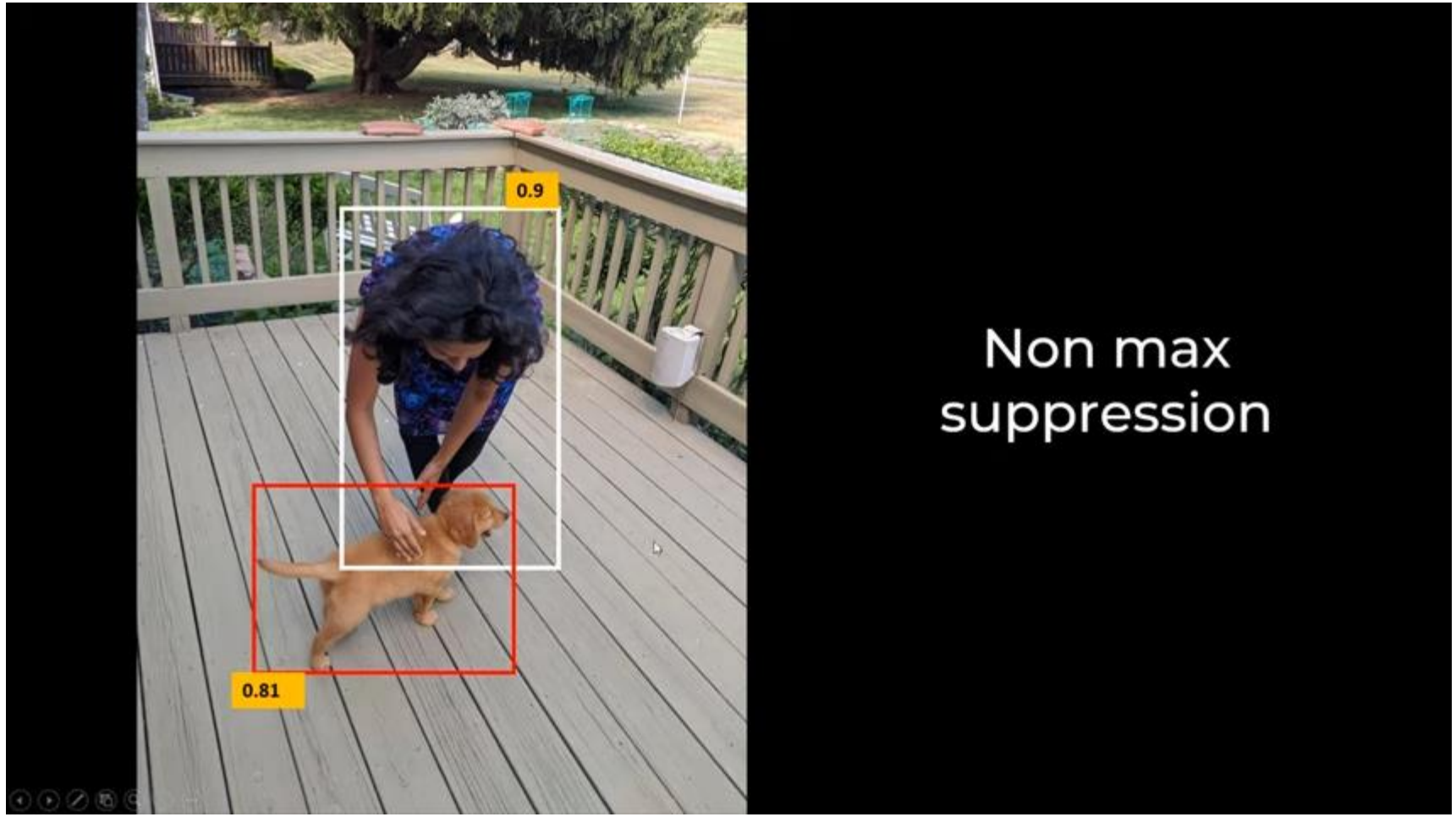
YOLO Algorithms



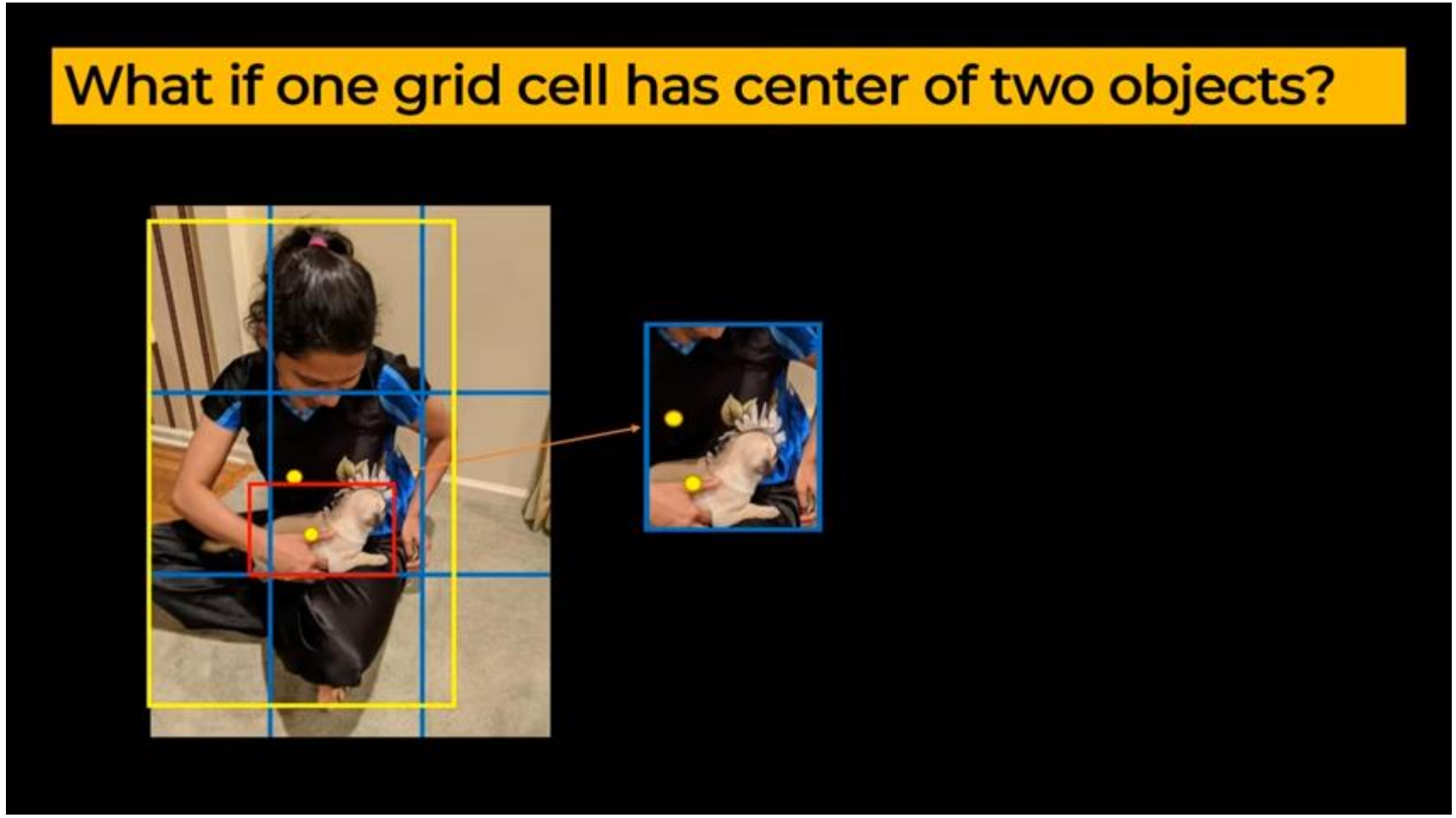
YOLO Algorithms



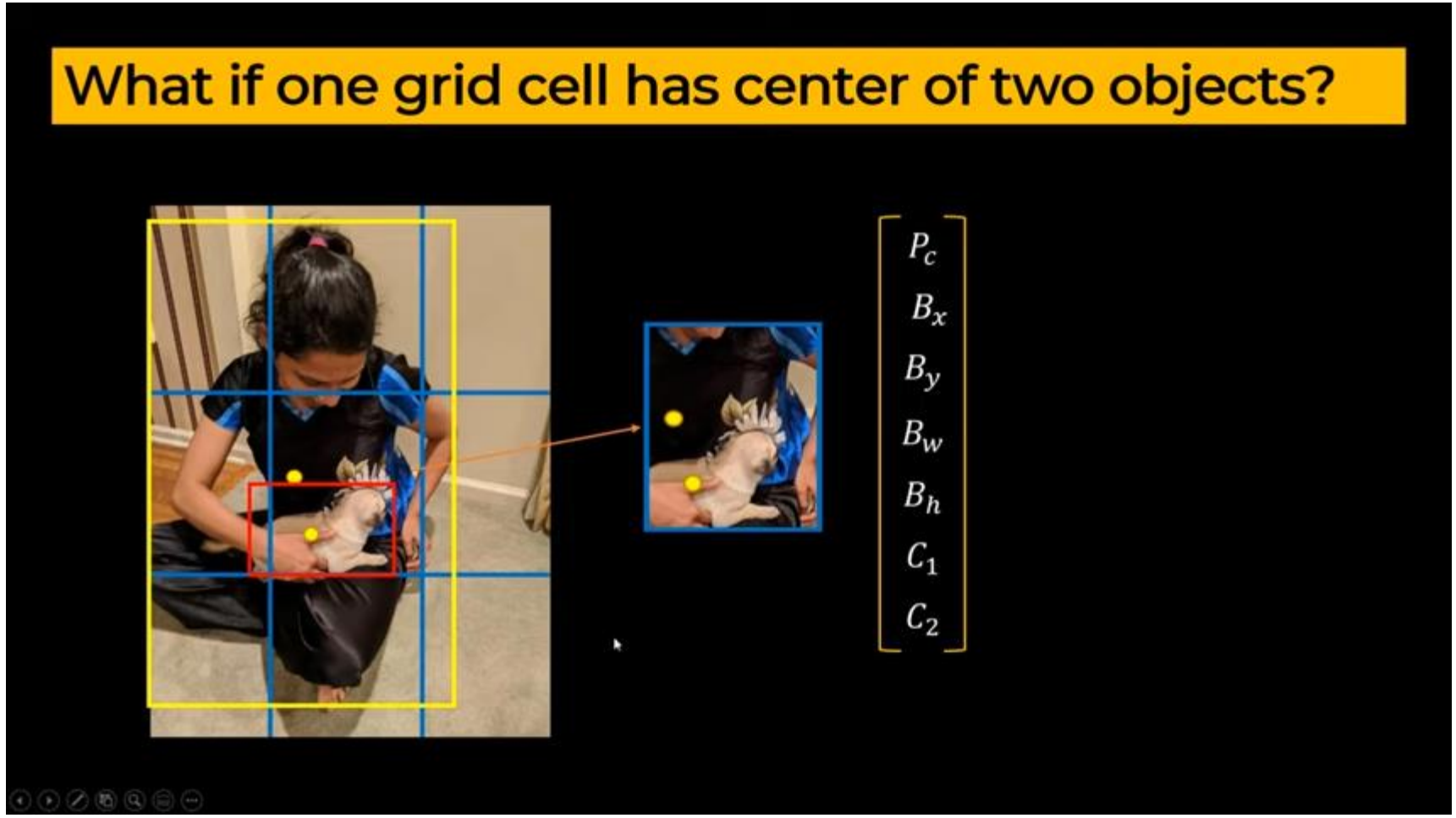
YOLO Algorithms



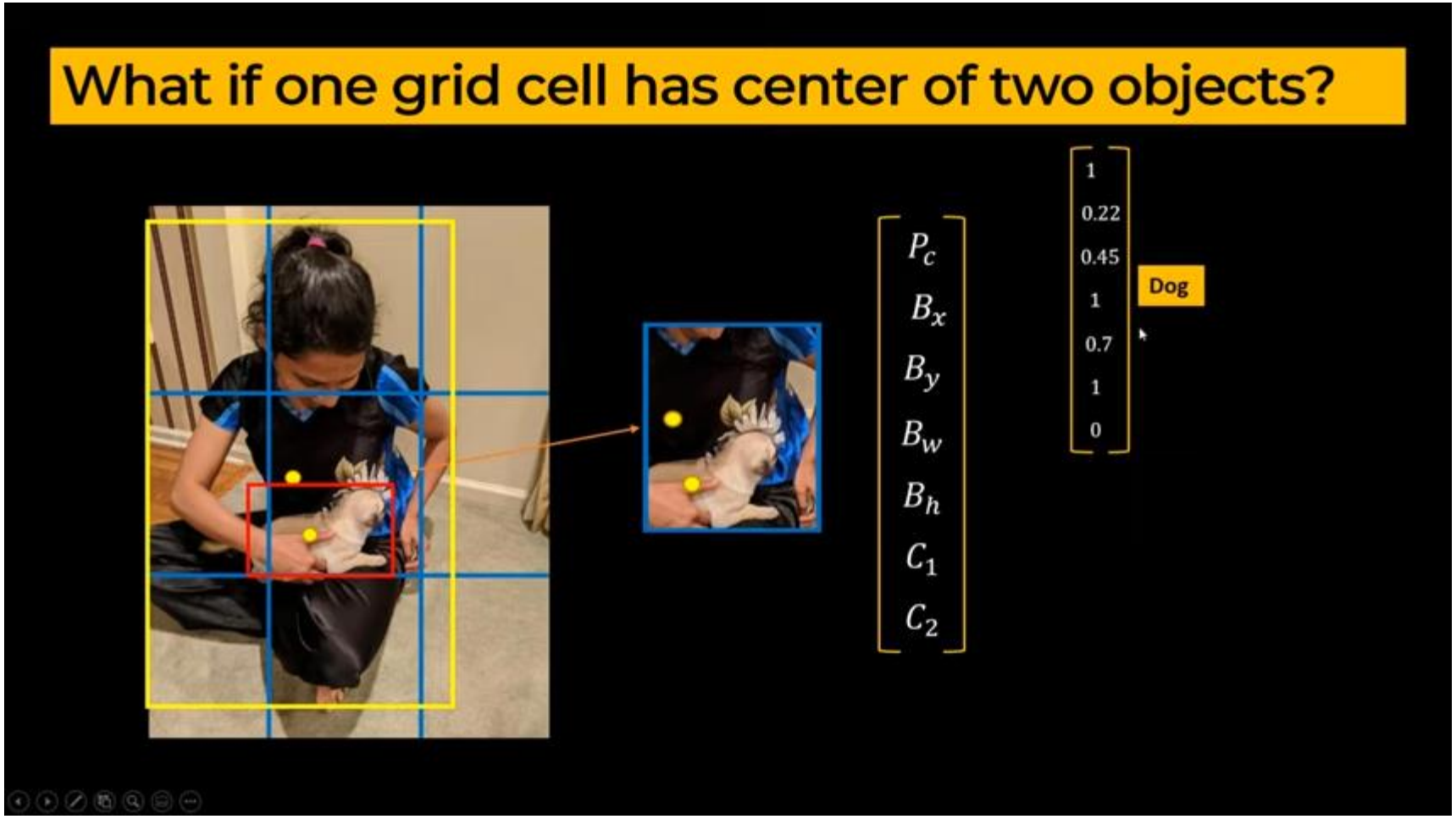
YOLO Algorithms



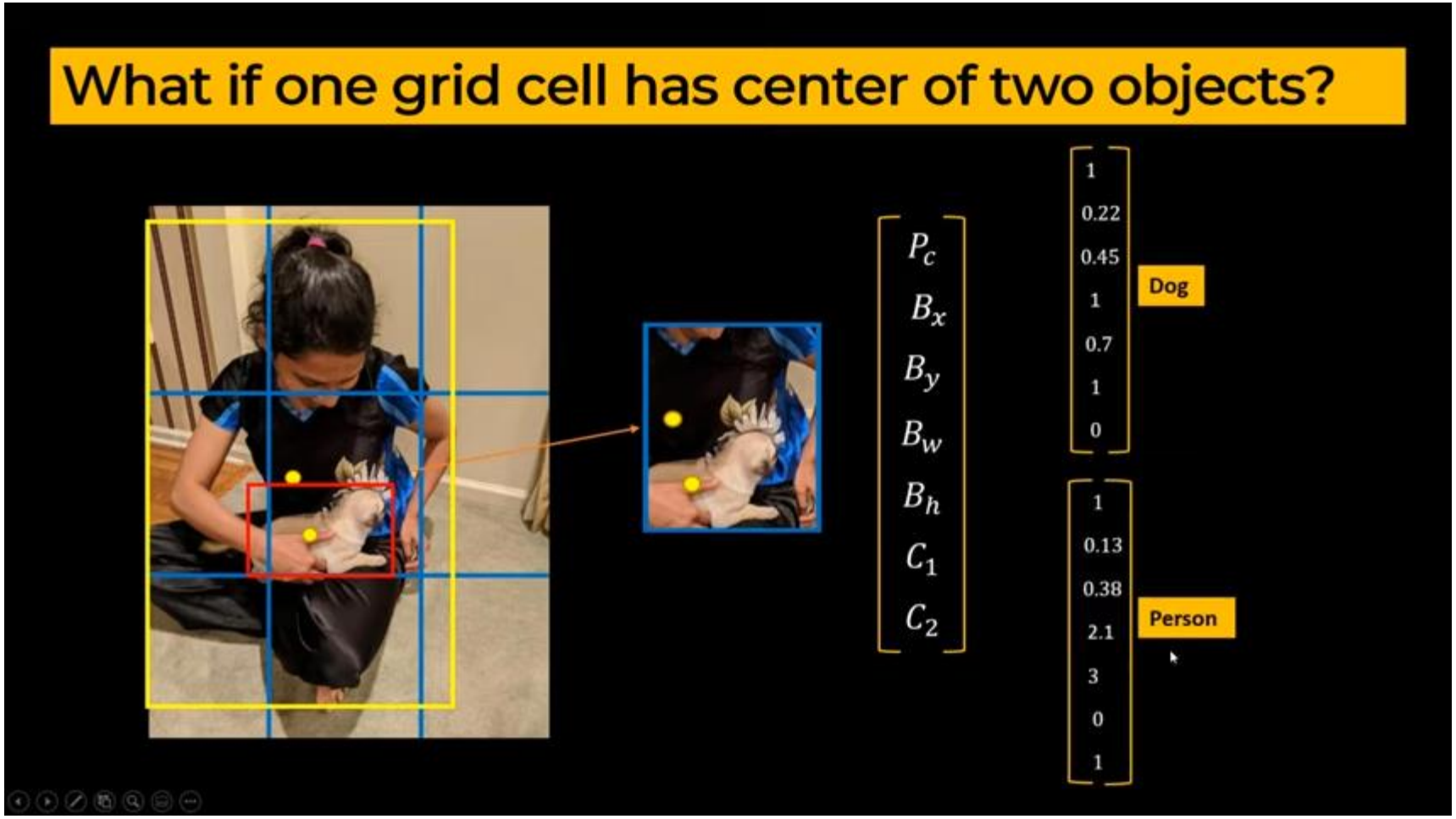
YOLO Algorithms



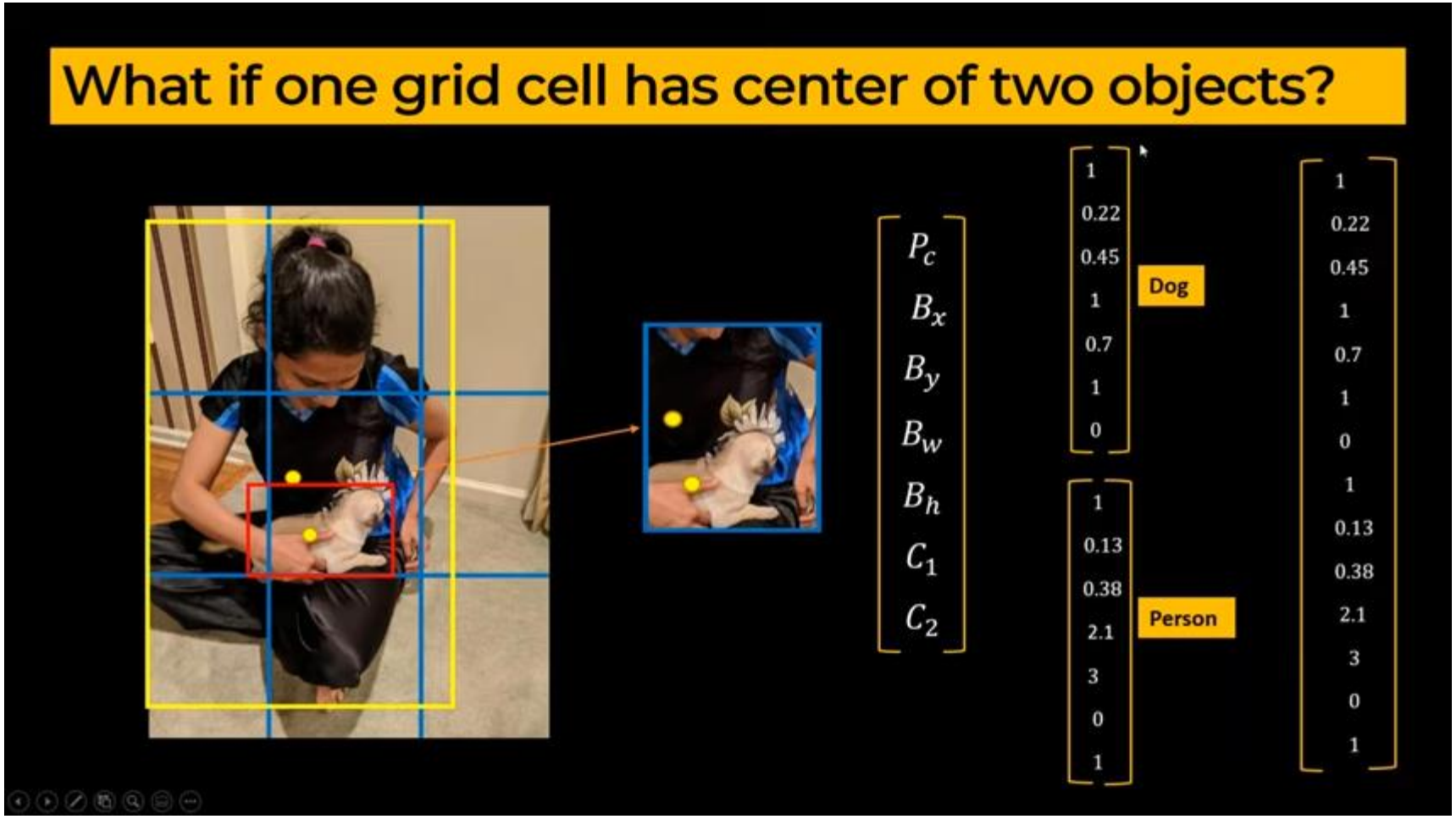
YOLO Algorithms



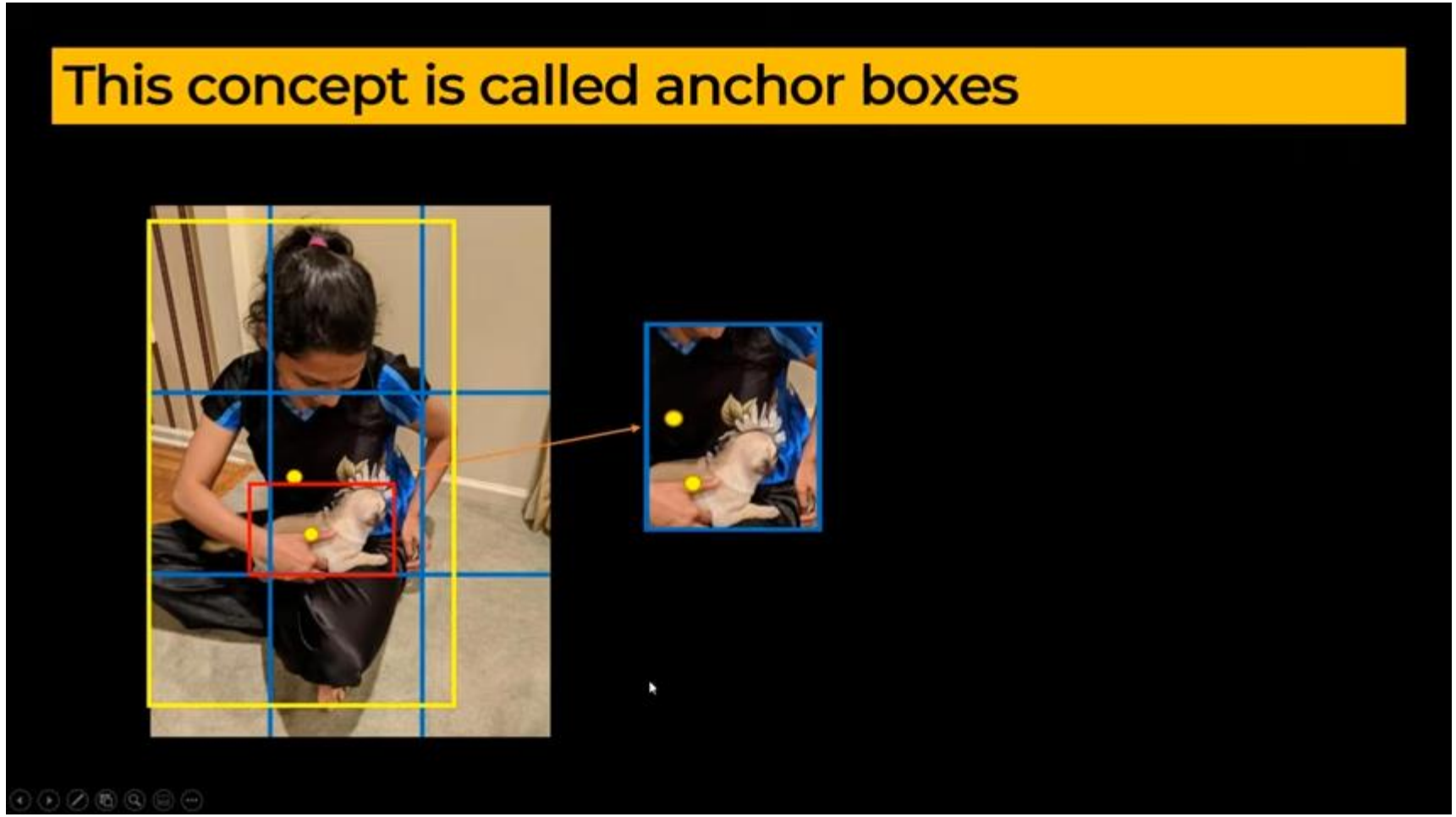
YOLO Algorithms



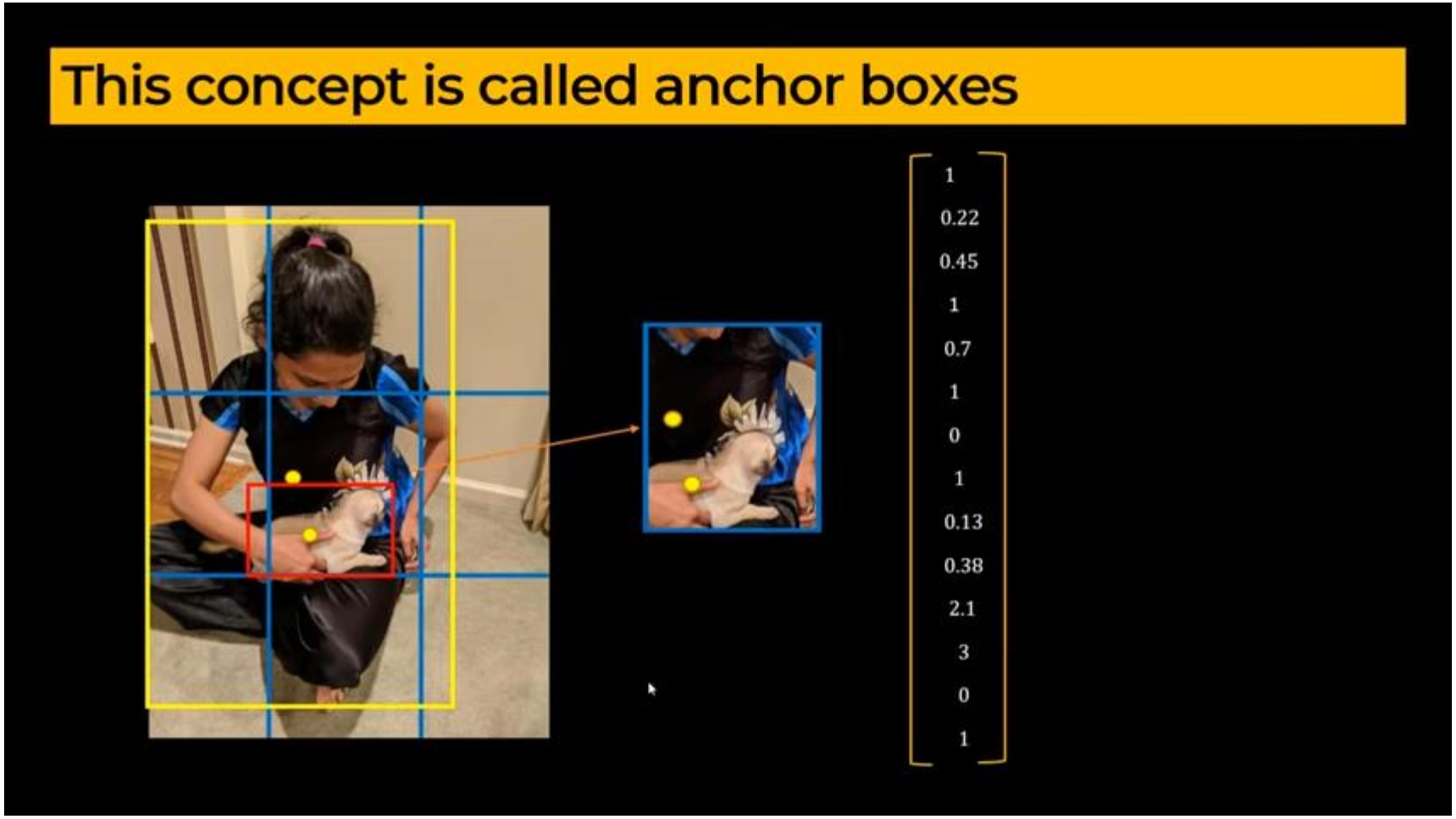
YOLO Algorithms



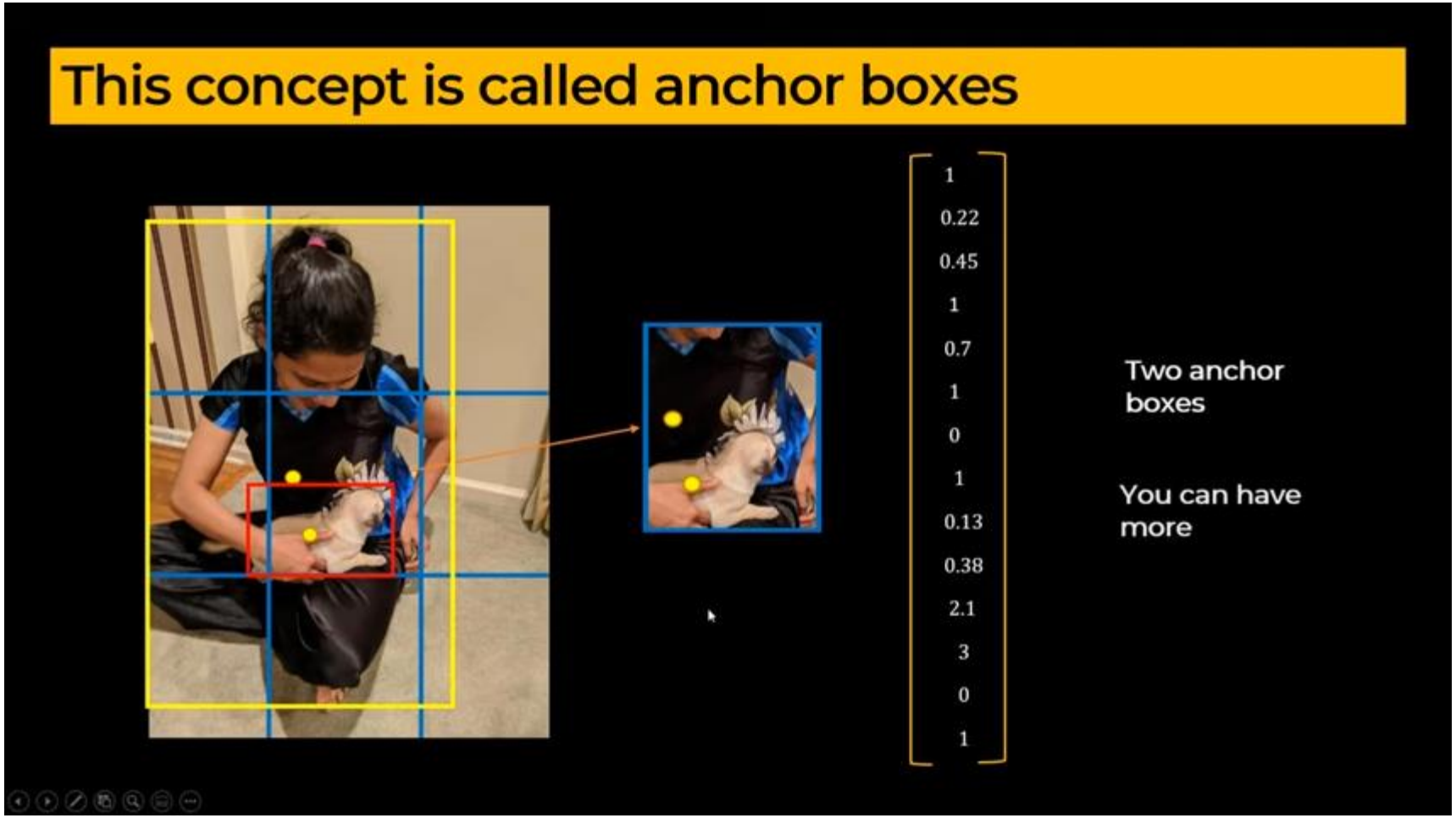
YOLO Algorithms



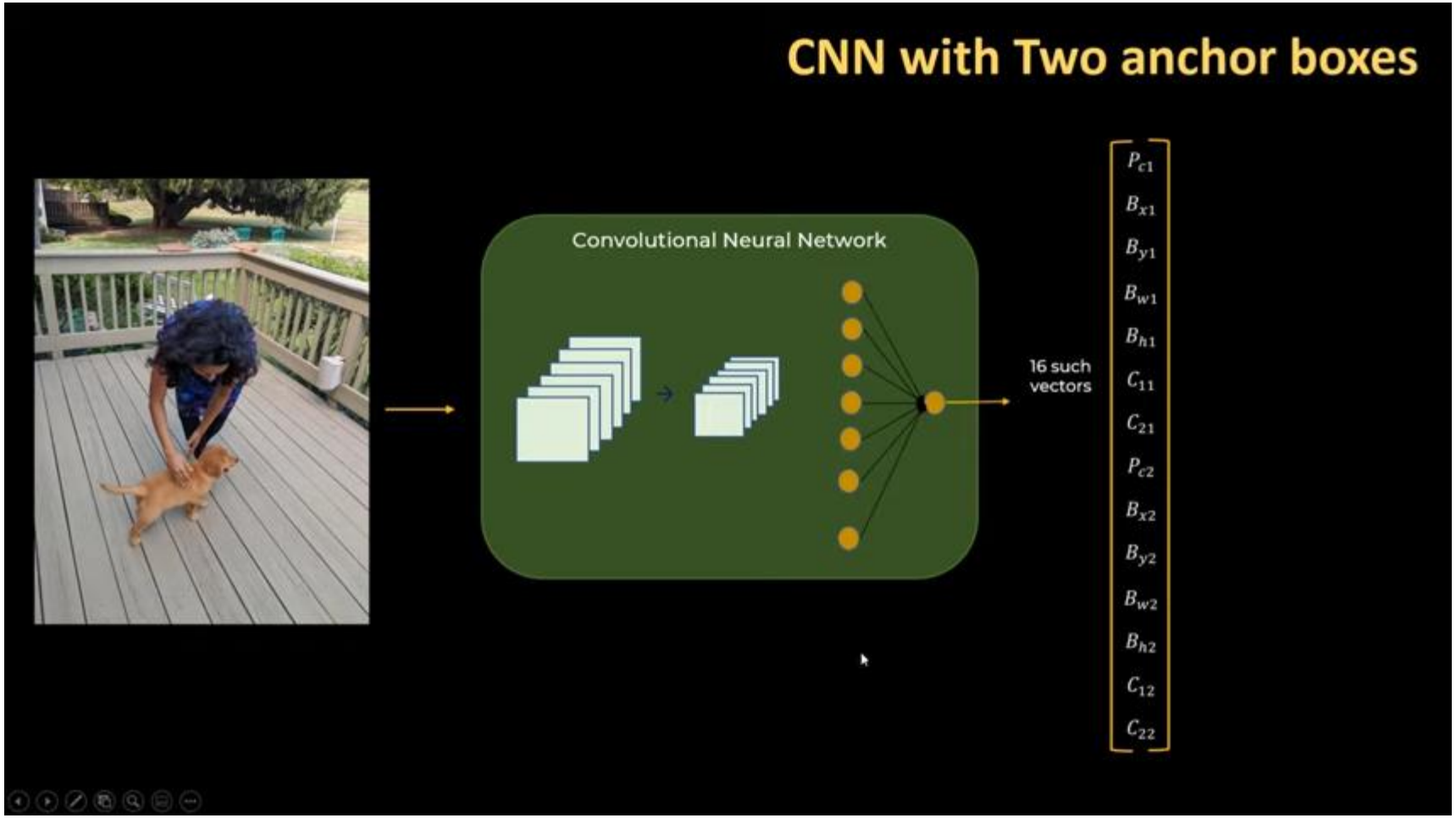
YOLO Algorithms



YOLO Algorithms



YOLO Algorithms



YOLO Algorithms