CONVOLUTIONAL NEURAL NETWORKS

and Comparison of Object Detection Algorithms

Mustafa Ercengiz 070210255
Fatma Sultan Teke 070220348
Abdullah Bilal Kibar 0700220349
Hüseyin Sezen 070230334

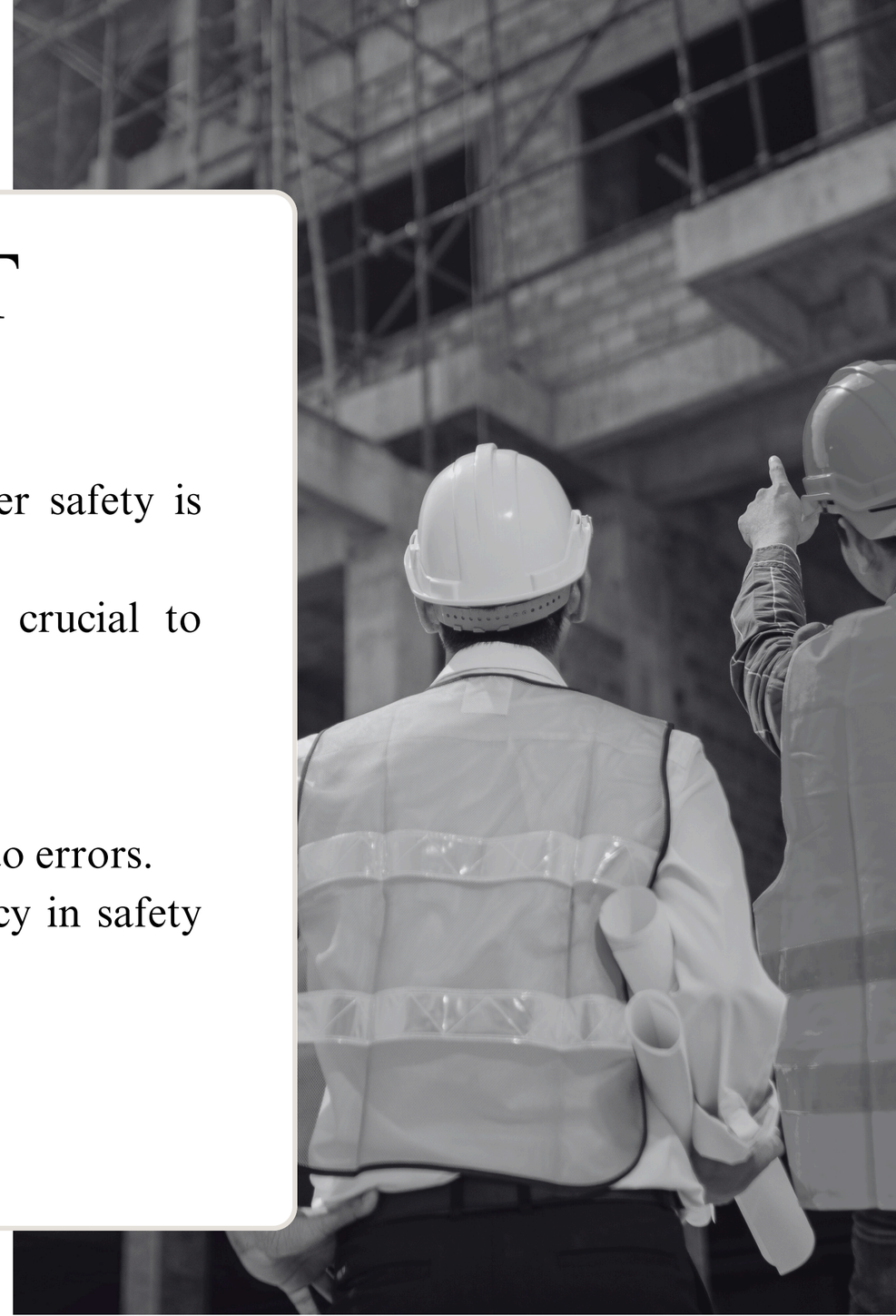
PROBLEM STATEMENT

Background:

- Construction zones are high-risk areas where ensuring worker safety is essential.
- Compliance with safety protocols, like wearing helmets, is crucial to reduce accidents.

Current Situation:

- Monitoring safety gear manually is time-consuming and prone to errors.
- Automating this process can enhance efficiency and consistency in safety compliance.



Literature Review about Object Detection in Safety Applications

- Object detection can be used to increase safety at construction sites
- Key applications include ensuring compliance with safety protocols, particularly helmet usage, by detecting non-compliance in real time.Machine learning models, such as
- Convolutional Neural Networks (CNNs), provide reliable tools for implementing automated safety monitoring.

Key Studies and Findings

Object detection for helmet usage has proven effective in identifying compliance using various machine learning models.

Studies emphasize using lightweight and real-time detection models, such as YOLO, to streamline monitoring in dynamic environments.

A focus on simple applications, like helmet detection, demonstrates how tailored solutions can significantly improve workplace safety.

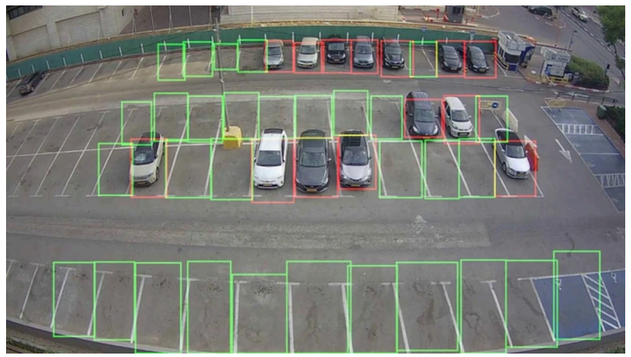
Where and Why Is Object Detection Used?

Construction Sites

Why? To ensure worker safety and monitor the use of protective equipment like helmets and vests.
How? By using real-time image processing to quickly detect non-compliance and prevent accidents.

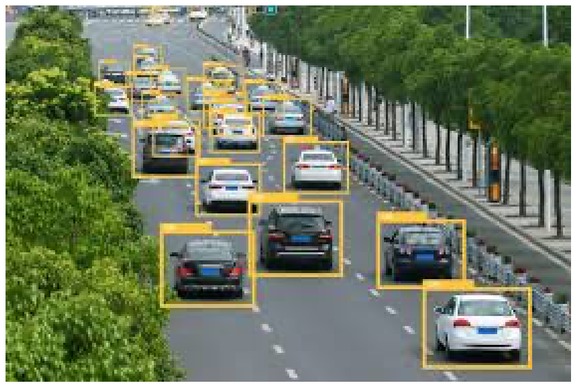
Security and Surveillance

Why? To identify threats, unauthorized access, and hazardous situations.
How? By analyzing camera footage to recognize objects and respond to risks promptly.



Traffic Management

Why? To detect license plates, analyze traffic flow, and identify violations.
How? By using traffic cameras to automatically recognize and monitor vehicles.



What is Convolutional Neural Networks(CNNs)?

Convolutional Networks and Applications in Vision

Yann LeCun, Koray Kavukcuoglu and Clément Farabet
Computer Science Department, Courant Institute of Mathematical Sciences, New York University
{yann,koray,cfarabet}@cs.nyu.edu

***Abstract*— Intelligent tasks, such as visual perception, auditory perception, and language understanding require the construction of good internal representations of the world (or "features"), which must be invariant to irrelevant variations of the input while, preserving relevant information. A major question for Machine Learning is how to *learn* such good features automatically. Convolutional Networks (ConvNets) are a biologically-inspired trainable architecture that can learn invariant features. Each stage in a ConvNets is composed of a filter bank, some non-linearities, and feature pooling layers. With multiple stages, a ConvNet can learn multi-level hierarchies of features. While ConvNets have been successfully deployed in many commercial applications from OCR to video surveillance, they require large amounts of labeled training samples. We describe new unsupervised learning algorithms, and new non-linear stages that allow ConvNets to be trained with very few labeled samples. Applications to visual object recognition and vision navigation for off-road mobile robots are described.**

I. Learning Internal Representations

One of the key questions of Vision Science (natural and artificial) is how to produce good internal representations of the visual world. What sort of internal representation would

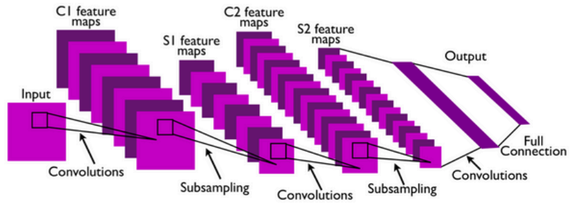


Fig. 1. A typical ConvNet architecture with two feature stages

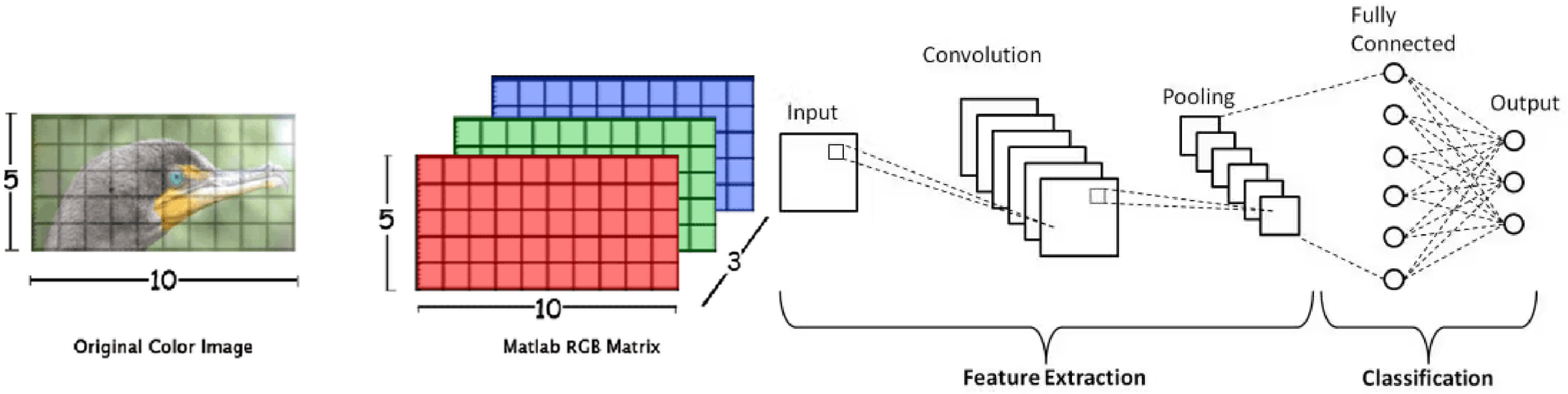
extracted at all locations on the input. Each stage is composed of three layers: a *filter bank layer*, a *non-linearity layer*, and a *feature pooling layer*. A typical ConvNet is composed of one, two or three such 3-layer stages, followed by a classification module. Each layer type is now described for the case of image recognition.

Filter Bank Layer - F: the input is a 3D array with n_1 2D *feature maps* of size $n_2 \times n_3$. Each component is denoted x_{ijk}, and each feature map is denoted x_i. The output is also a 3D array, y composed of m_1 feature maps of size $m_2 \times m_3$. A

Invention of CNN's 1992

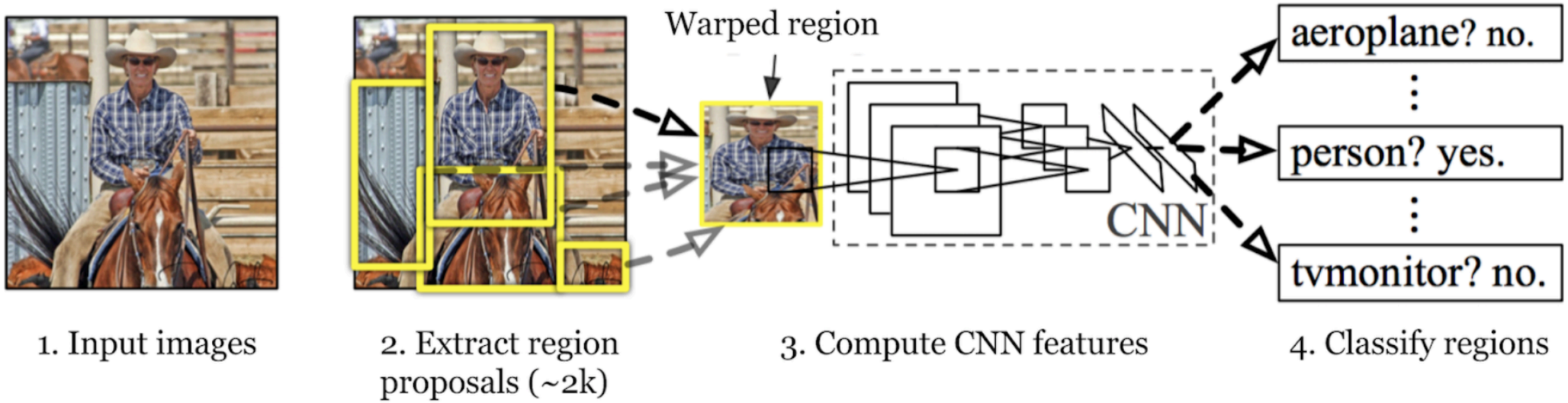
The Scene From Yann Lecun Invention at 1993

How CNNs work and what is differences between NN's?



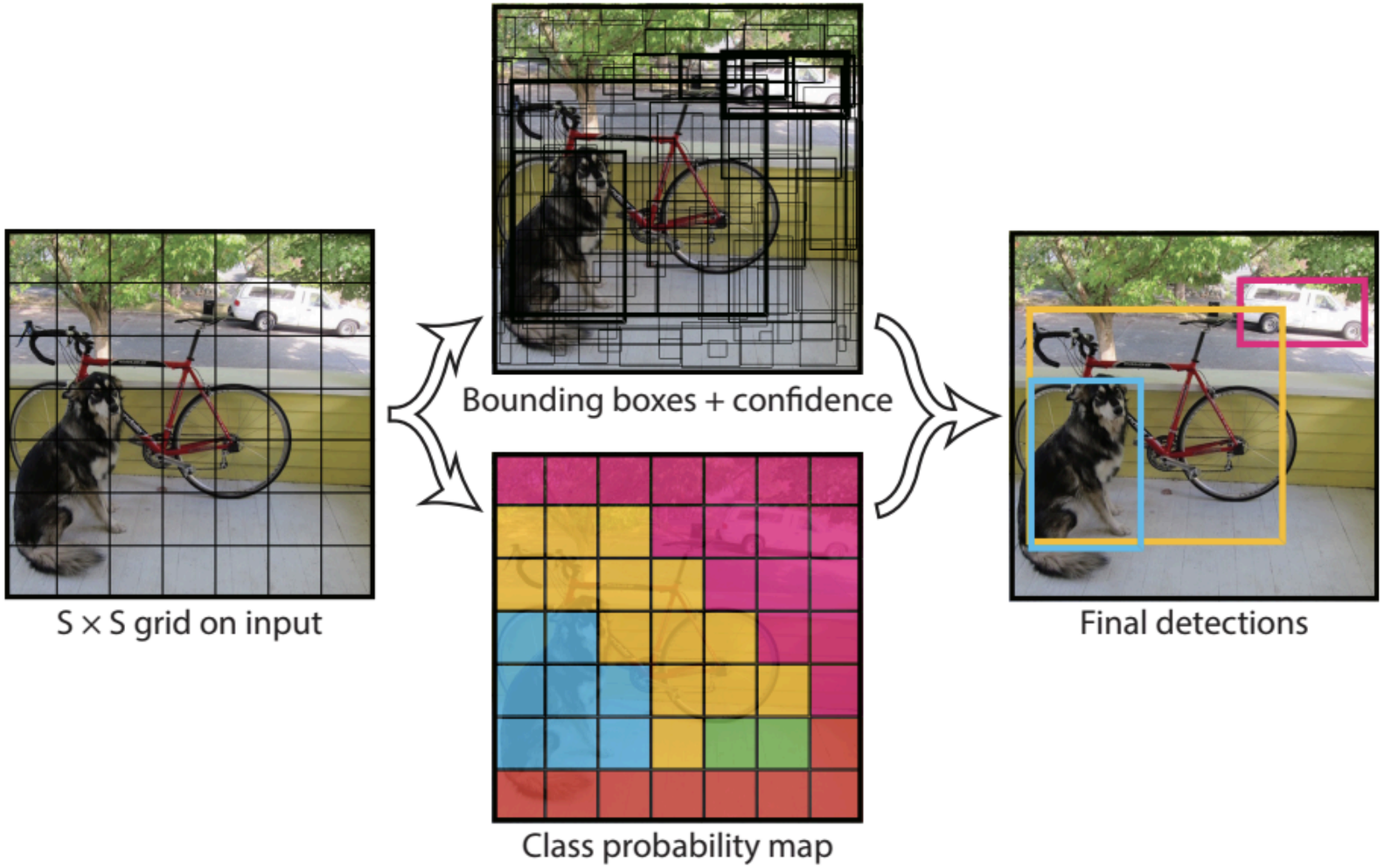
How Object Detection Algorithms works?

Alright, How does RCNN Object Detection Algorithms works?



Instead of using sliding window tecnique it uses anchor based predictions with Region Proposel Networks

How does Yolo Object Detection Algorithms works?



Our Data and Annotation Sample

Our Model Architectures

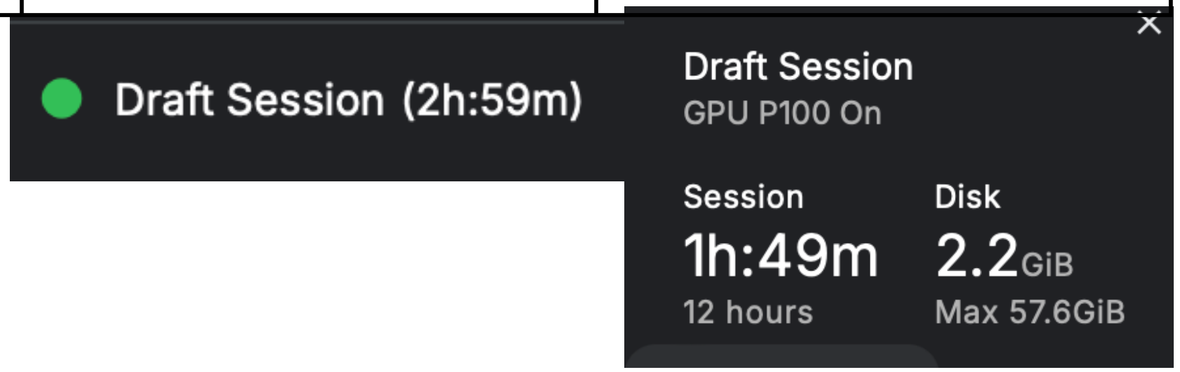
Faster RCNN

- Backbone: ResNet50
- Optimizer: SGD with Momentum
- Learning Rate (η) = 0.001
- Momentum (β) = 0.9
- Weight Decay (λ) = 0.0005
- Learning Rate Scheduler: Step LR
- Step Size = 5
- Gamma (γ) = 0.5
- Epochs: 16 planned, but cut off at 8 due to computational constraints
- Batch Size: 4
- Detection Hyperparameters:
- IoU Threshold = 0.5
- Confidence Score Threshold = 0.8

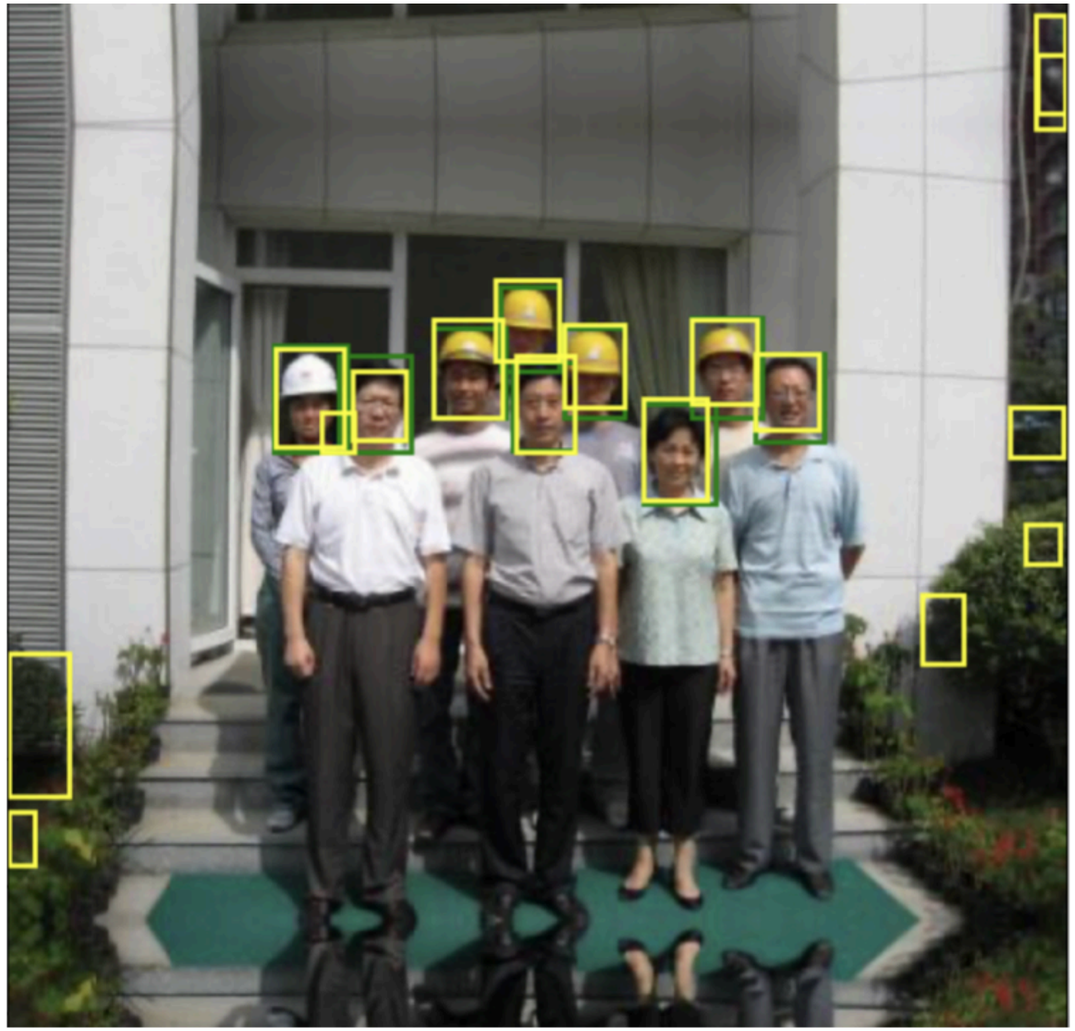
YoloV8

- Epochs: 20
- Batch Size: 16
- Workers: 8 (number of threads for data loading)
- Learning Rate:
- Initial (η_i) = 0.01
- Final (η_f) = 0.0001 (linear decay from 0.01 to 0.0001 over 20 epochs)
- Optimizer: auto
- Momentum (β) = 0.937
- Weight Decay (λ) = 0.0005
- LR Scheduler: Linear decay from 0.01 to 0.0001 by the last epoch.

	Faster RCNN	YoloV8
Architecture	Two-stage (RPN + detector head)	One-stage (end-to-end predictions)
Training Setup	16	20 epochs, batch size = 16
Optimizer	SGD (lr=0.001)	Auto (initial lr=0.01, decays to 0.0001)
mAP@0.5	~25.34% (helmets)	~96.2% (helmets)
Precision (Helmet)	0.7664	0.8984
Recall (Helmet)	0.6235	0.9385



Faster RCNN predictions



YoloV8
Predictions:

Results on Predicted Videos

THANK YOU

for your time and attention