

# VISION powered smart cataloging of books

- Aritra Raut

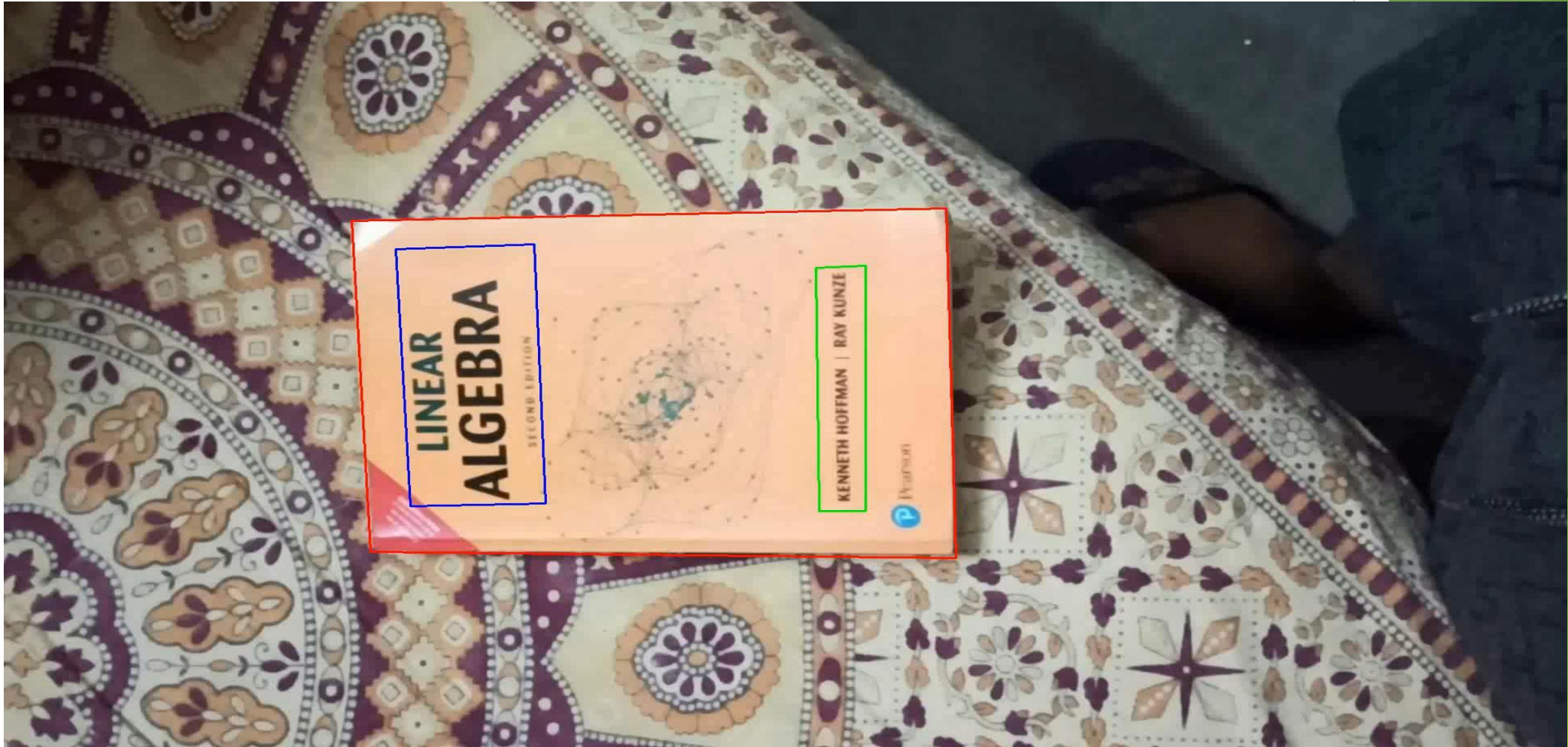
My work has mainly two phases :

1. Book detection (From video inputs)
2. Optical character recognition

# 1. Book detection :

- I have used YOLOv1 approach for object detection
- Original YOLOv1 was trained on PascalVOC dataset, which consists of 20 different class objects.

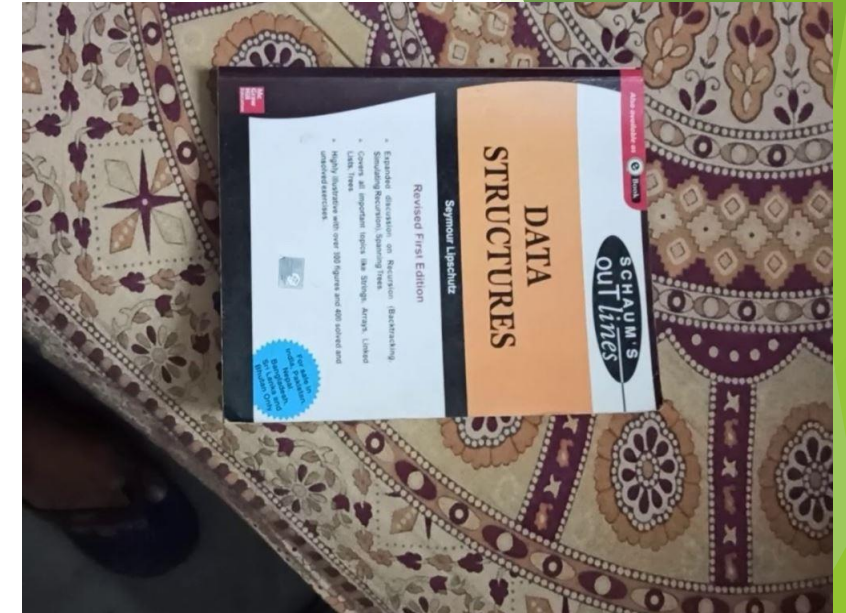
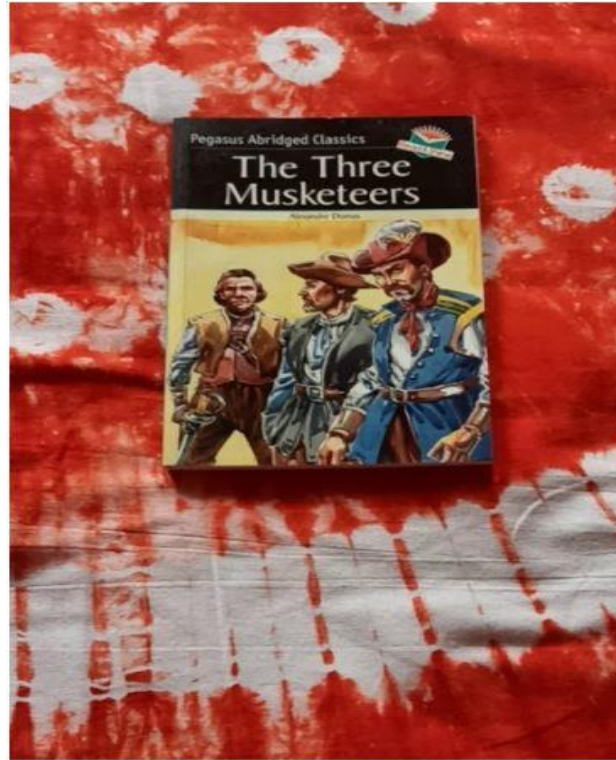
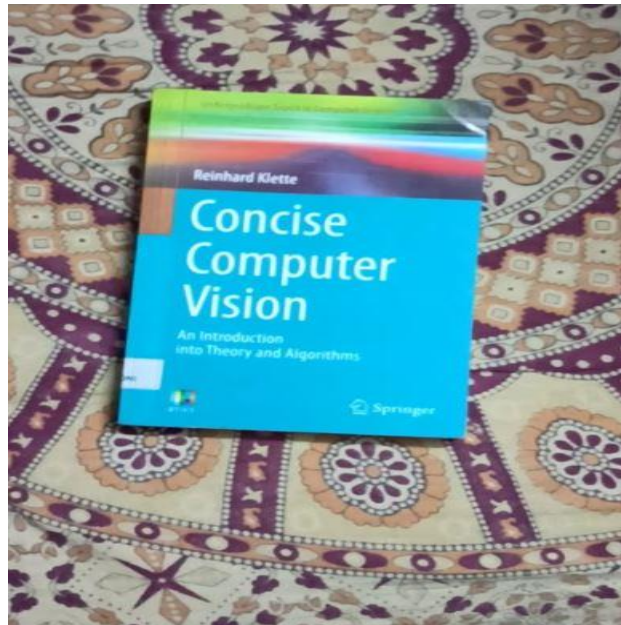
# My Data →





# Details of my data :

Slide-04



# Detail of my data:

## For Training :::

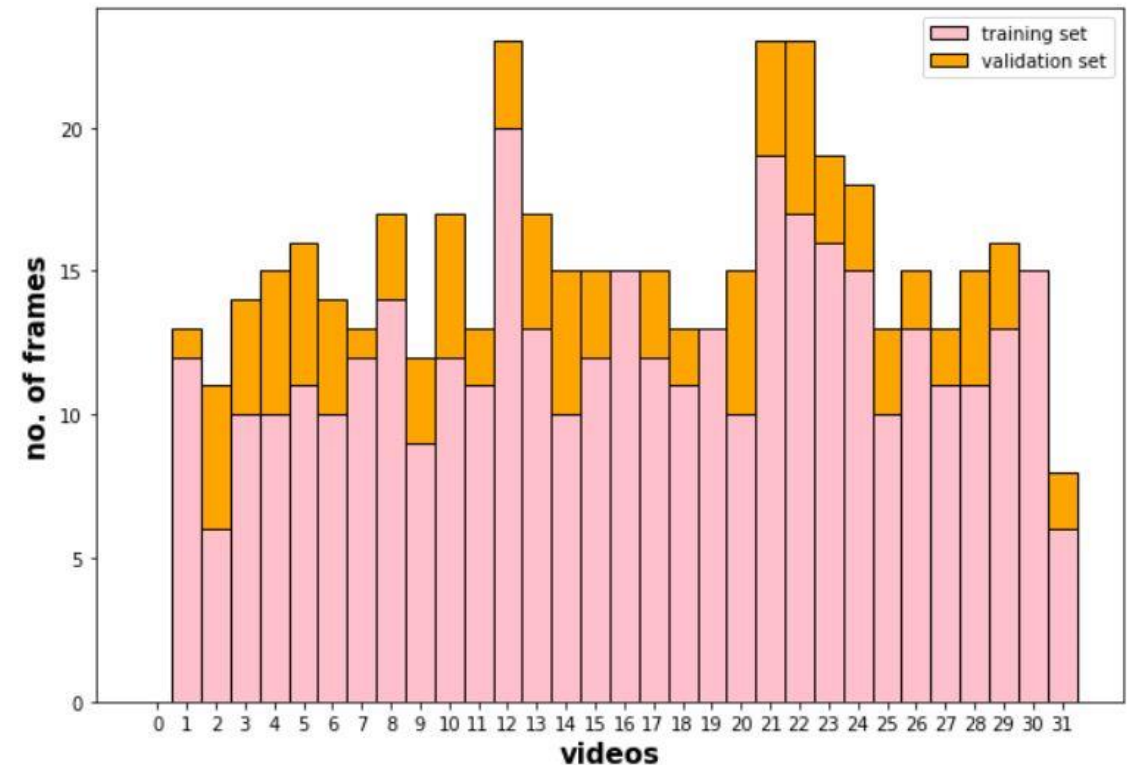
Total no. of videos annotated - 31

Total no. of frames extracted - 4738

I have taken 1 per 10 images .

My dataset size - 474 images

train-validation split :-  
(80:20) → 379:95



# Detail of my data :

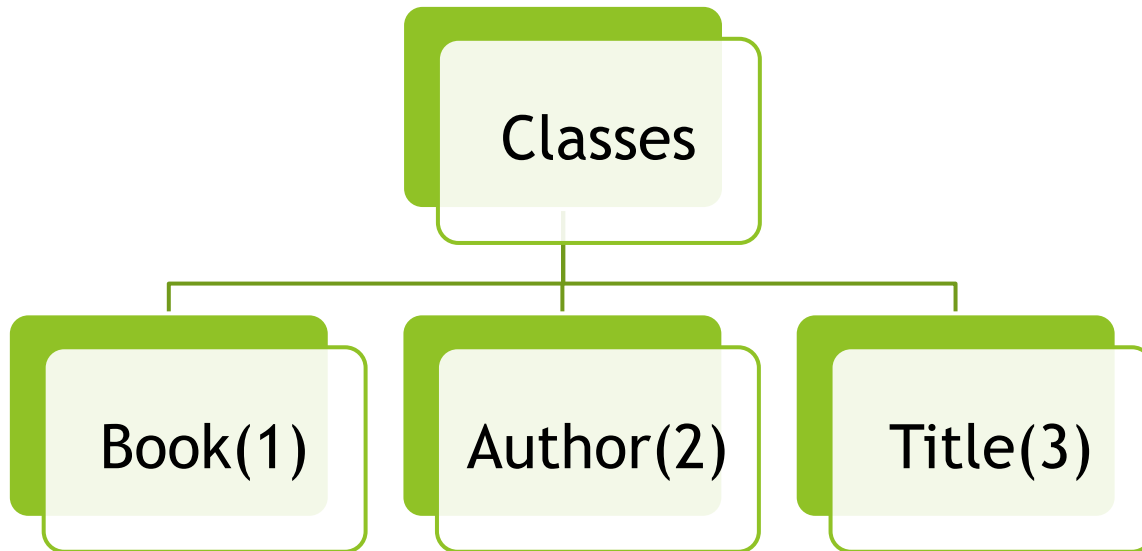
For Testing :::

Total no. of videos annotated : 10

Total no. of frames : 1376

After reduction(by the same manner) :  
137

# Detail of my data :



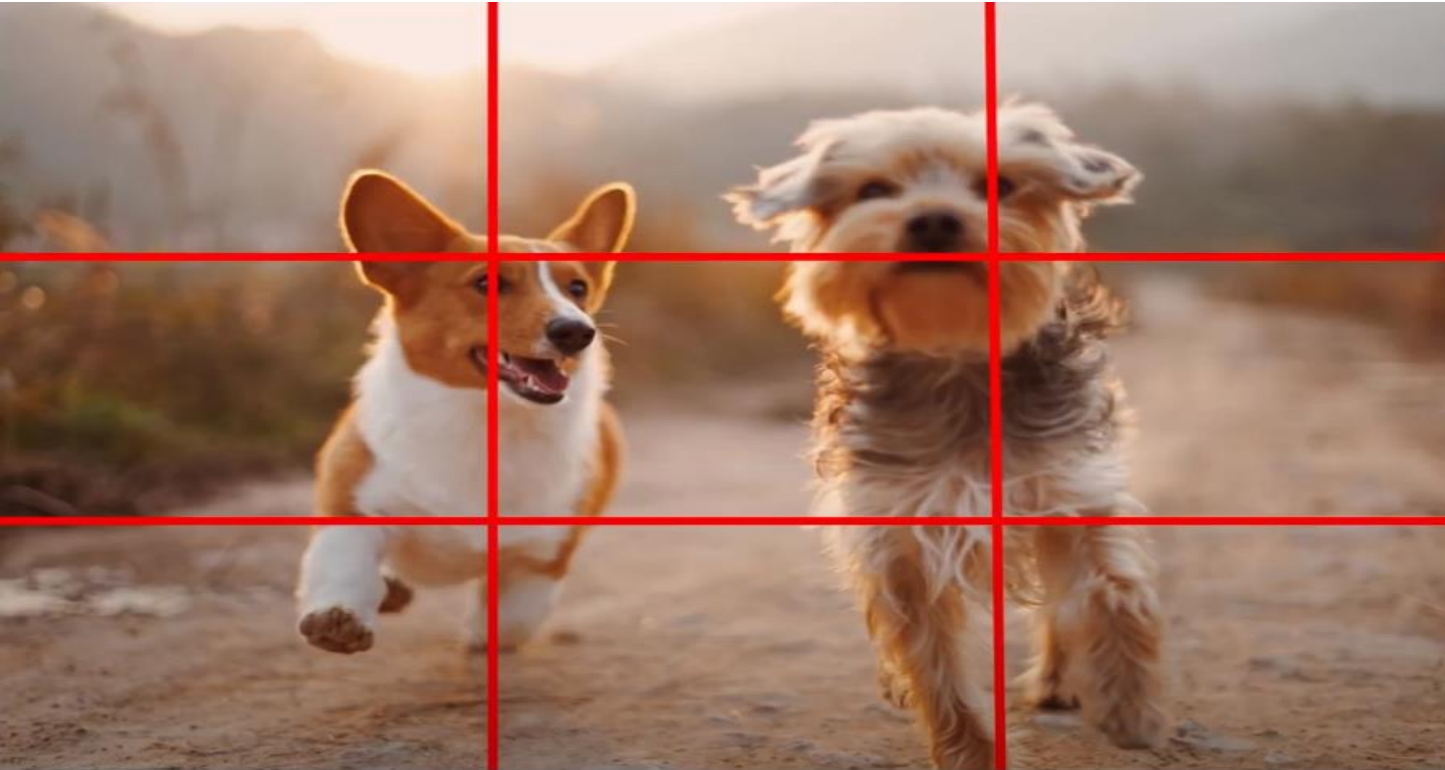
## Initial inputs :::

(class\_label , center\_x , center\_y , width , height)

```
1 0.35 0.32708333333333334 0.253125 0.2916666666666667
3 0.2916666666666667 0.2515625 0.05416666666666667 0.1527777777777778
2 0.40185185185185185 0.3697916666666667 0.0828125 0.16388888888888889
```

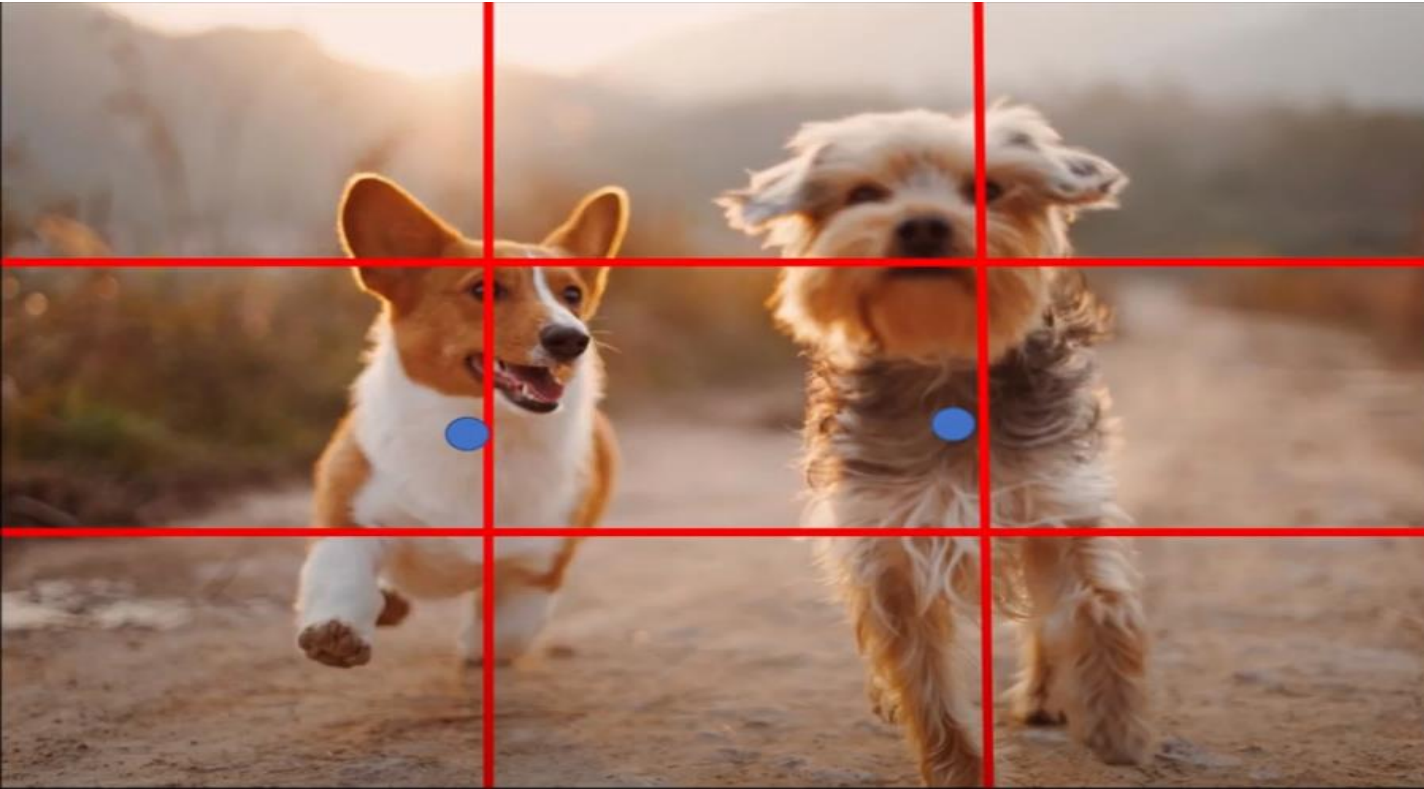


# Basic idea :



- Each picture is divided into  $S \times S$  grids.
- Each cell will output a prediction with a corresponding bounding box

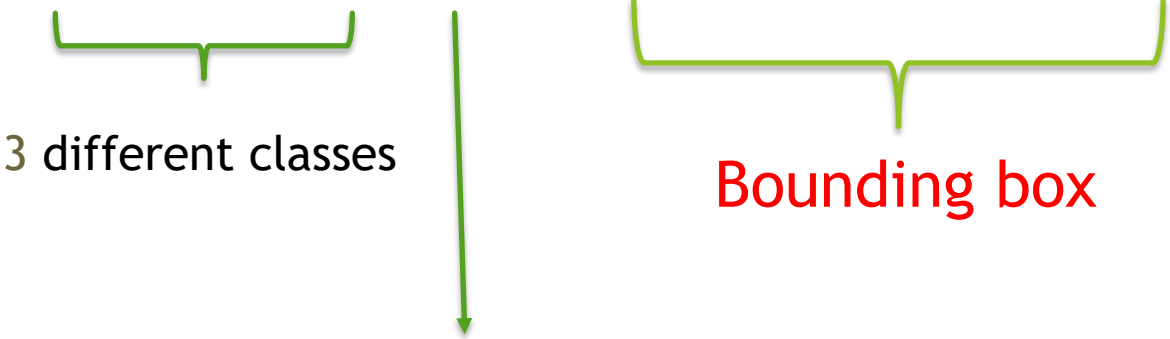
# Basic idea :



We have to identify those cells which contains objects' mid points

How the ground truth vectors for each cell actually look like?

$$\text{Label}_{\text{cell}} = [C_1, C_2, C_3, P_c, X, Y, W, H]$$

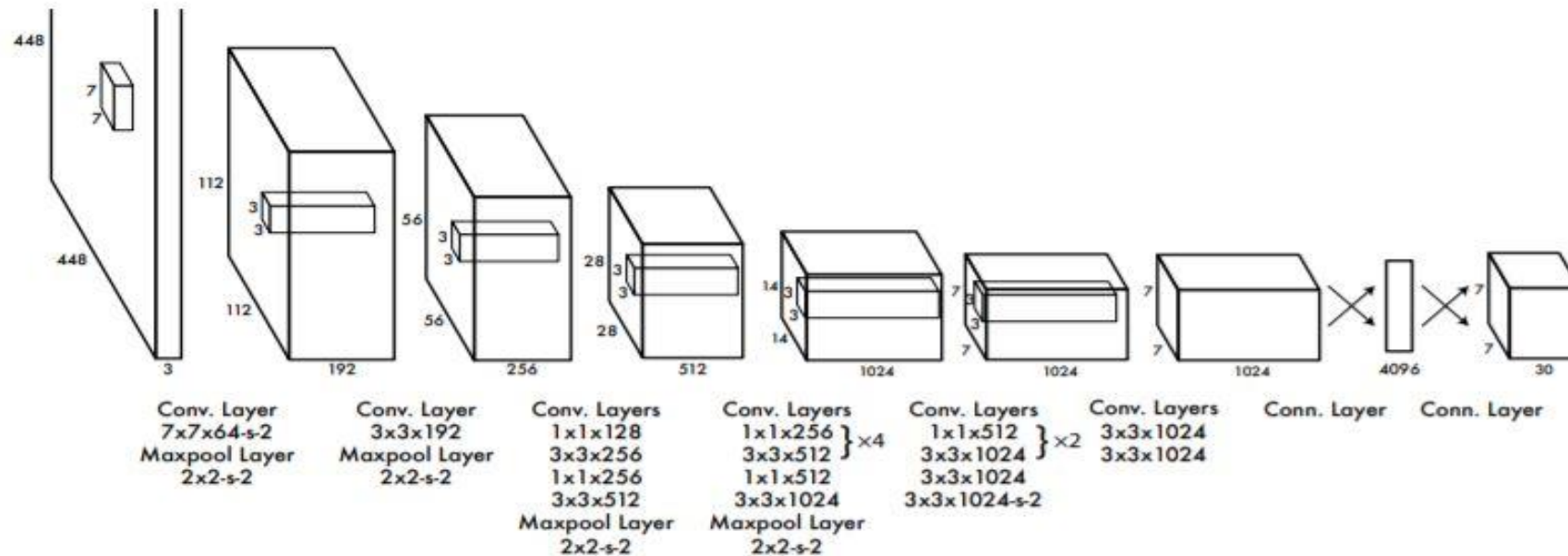


3 different classes

Bounding box

Probability that there is an object

# The network →



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

# How the prediction vectors will look like ?

$$\text{Pred}_{\text{cell}} = [ \underbrace{C_1, C_2, C_3}_{\text{3 different classes}}, \underbrace{P_{C_1}, X, Y, W, H}_{\text{Probability that there is an object}}, \underbrace{P_{C_2}, X, Y, W, H}_{\text{Bounding boxes}} ]$$

3 different classes

Probability that there is an object

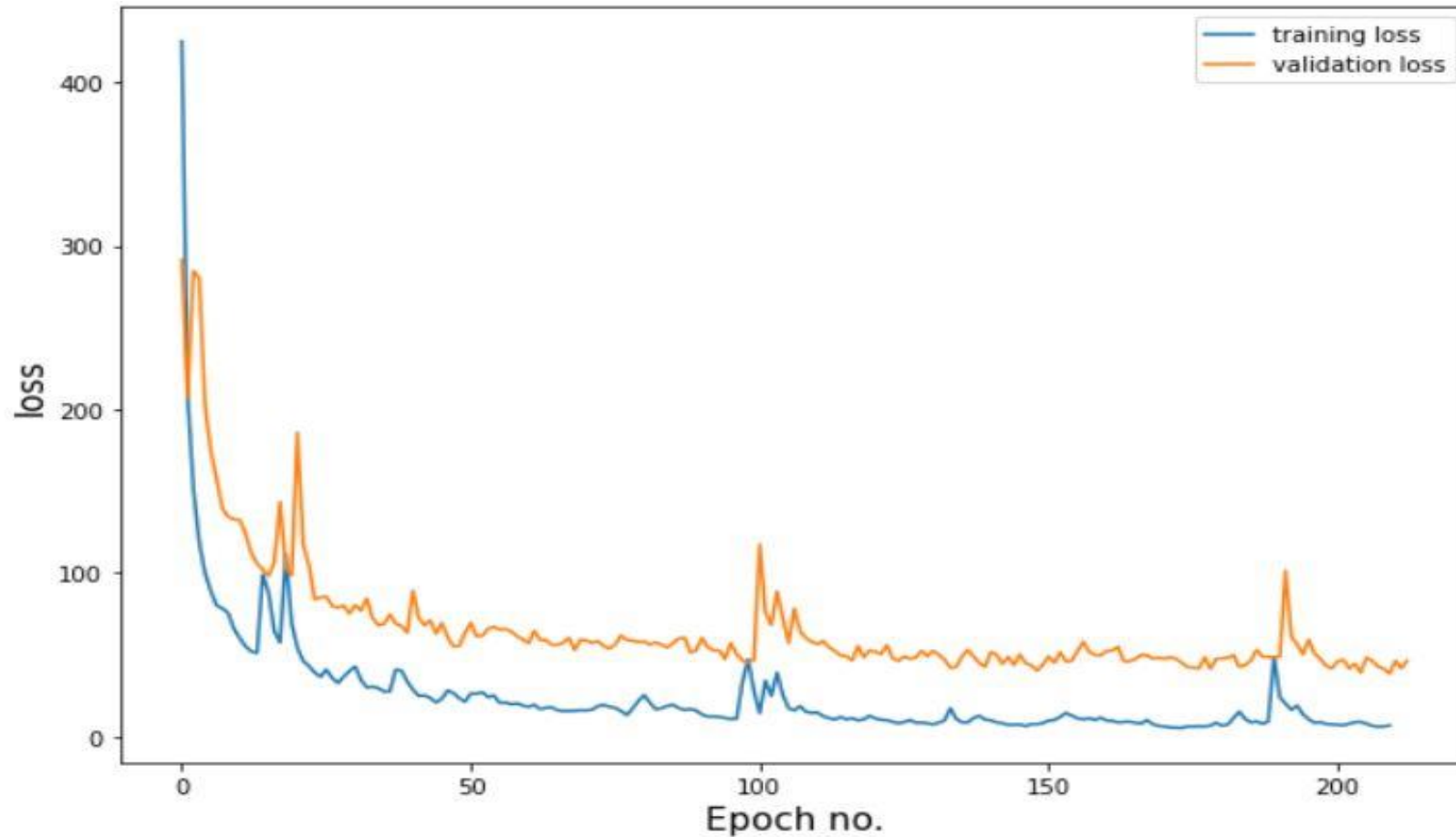
Bounding boxes



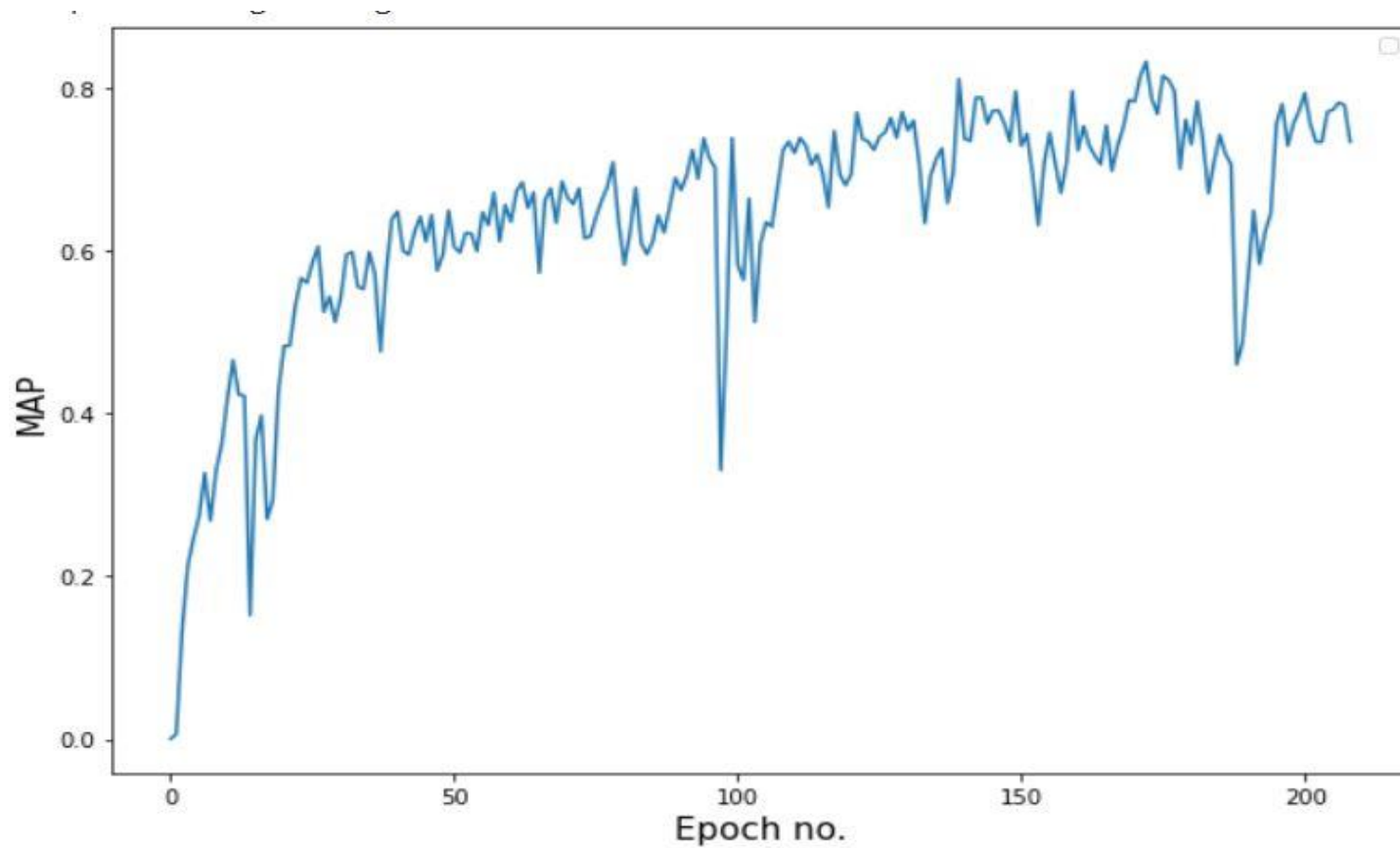
# Loss function :

$$\begin{aligned}
 & \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} \left[ (x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] \\
 & + \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} \left[ \left( \sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left( \sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right] \\
 & + \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} (C_i - \hat{C}_i)^2 \\
 & + \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{noobj}} (C_i - \hat{C}_i)^2 \\
 & + \sum_{i=0}^{S^2} \mathbb{1}_i^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2
 \end{aligned}$$

# Visualization of training and validation loss :

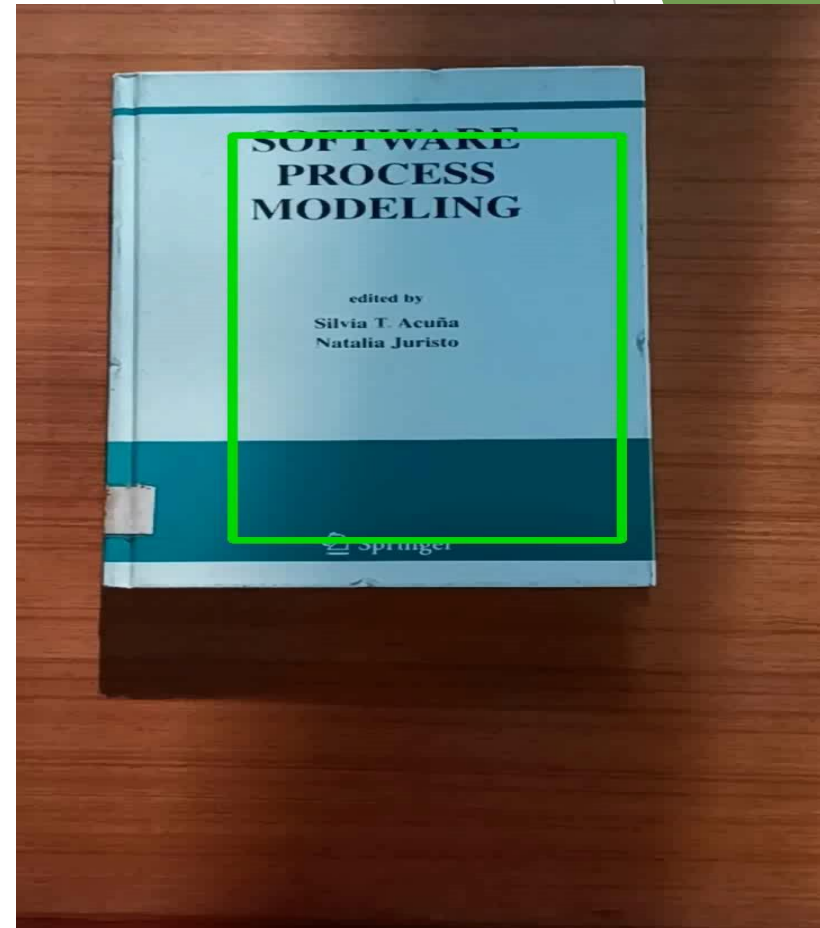
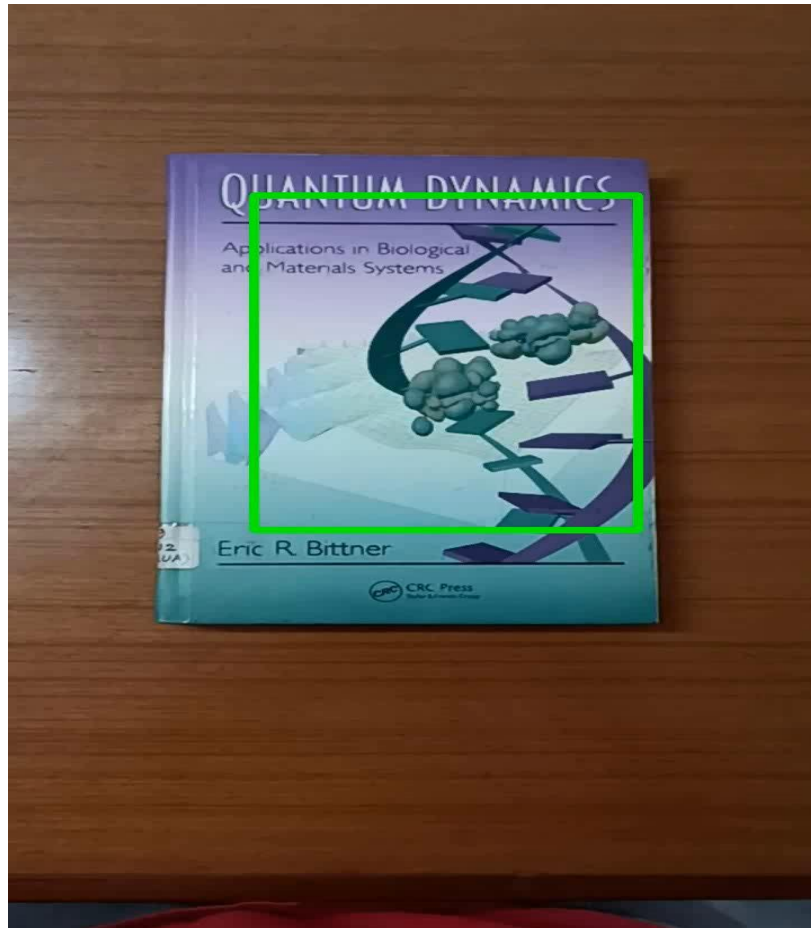


# Visualization of mean avg. precision :



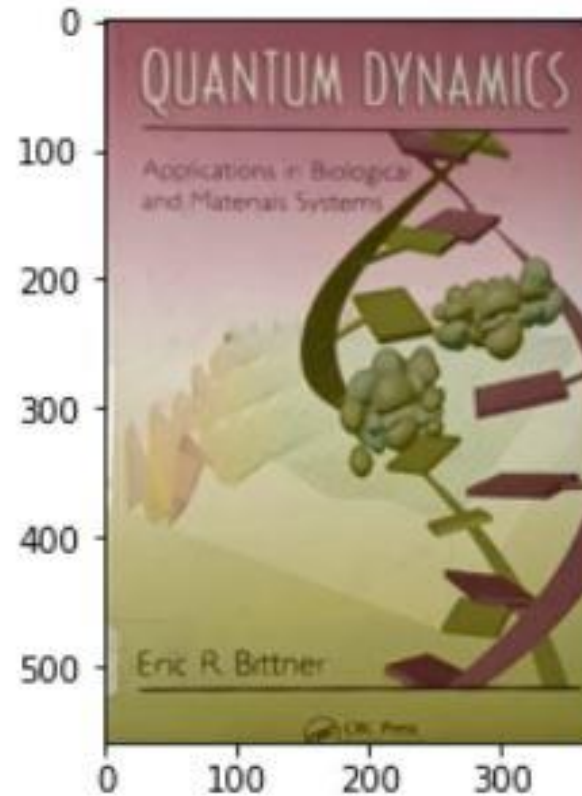
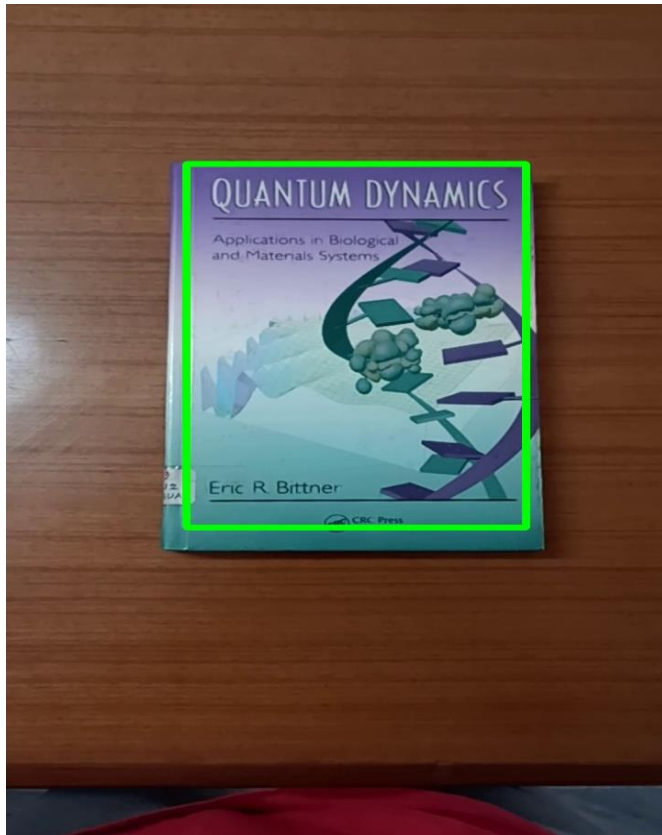
# Result →

Slide-16



# Result →

Slide-17



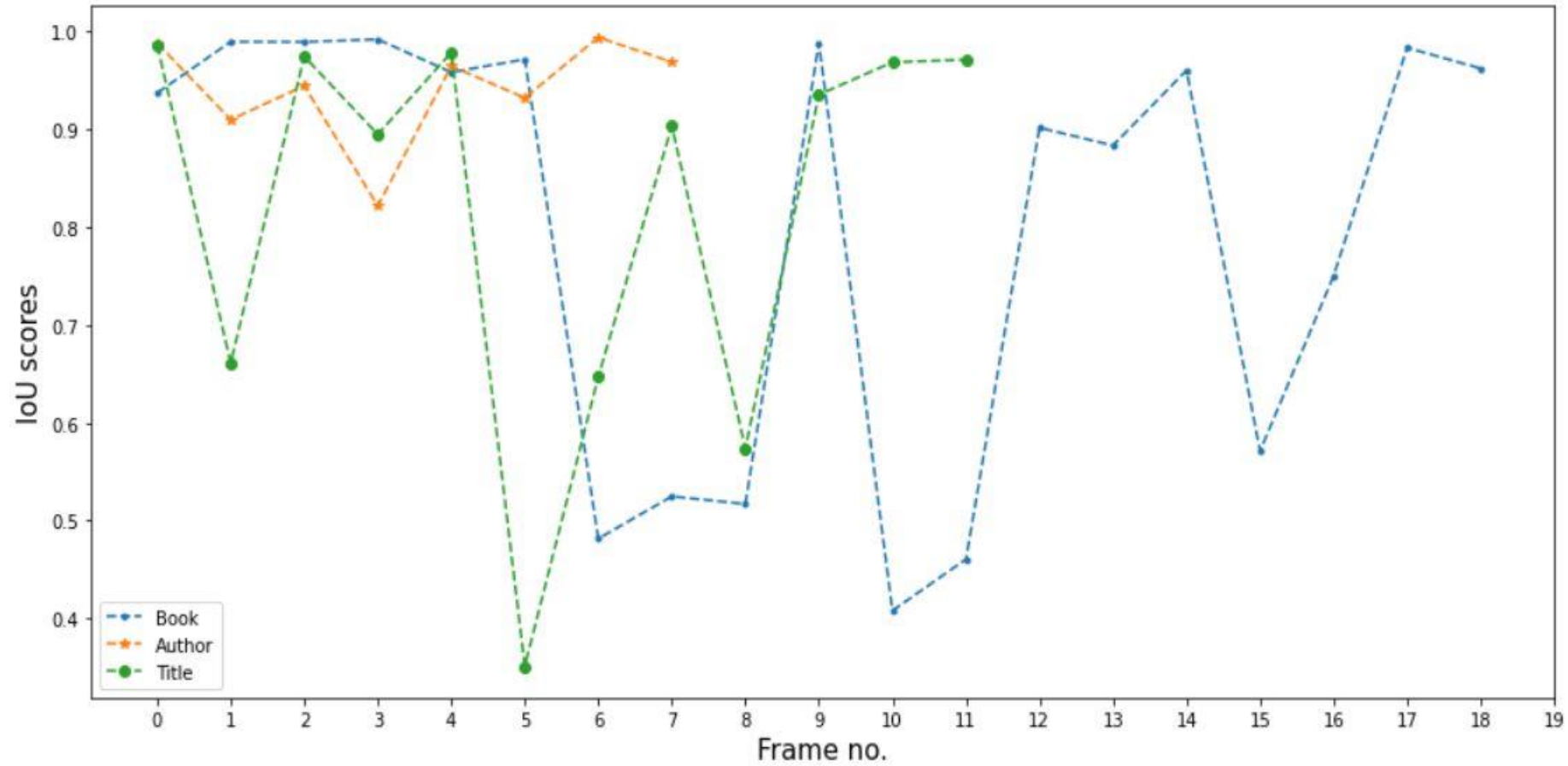
CUENTA Gs

Applications in Biological  
and Materials Systems

For this portion I have used  
python's **pytesseract**  
package



# Evaluation (Object detection):



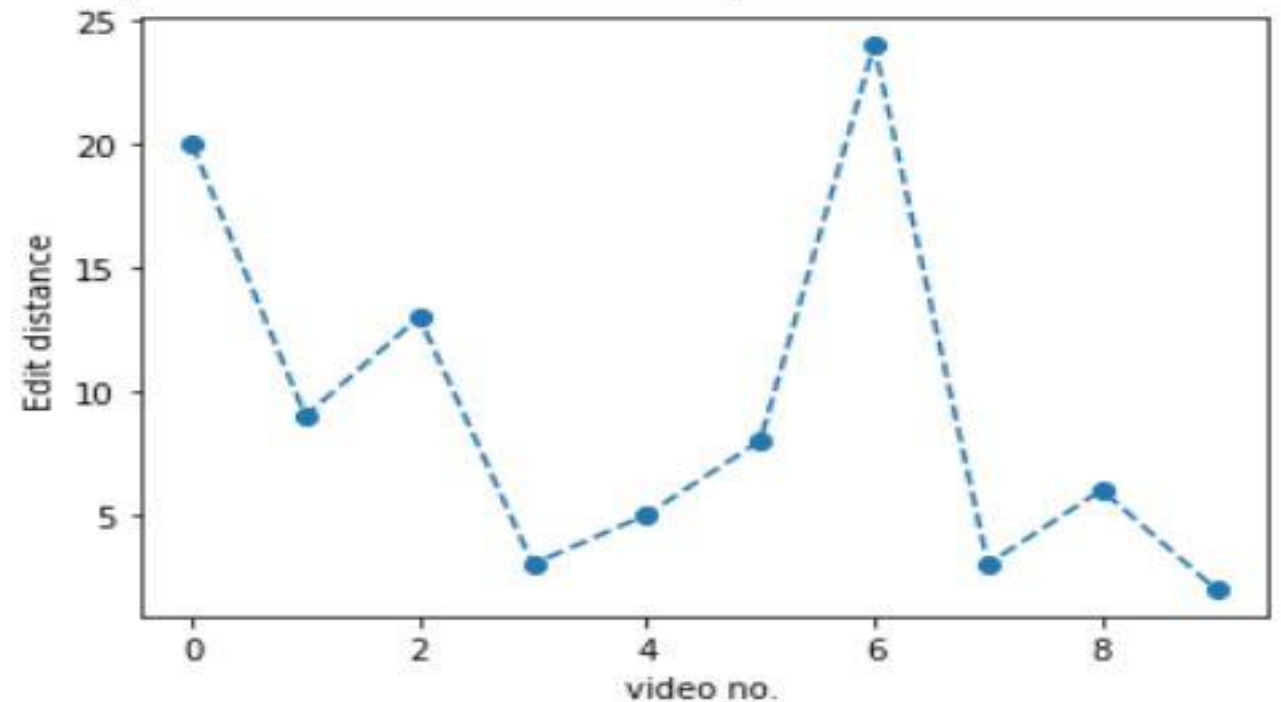
# Evaluation (optical character recognition):

## What is edit distance?

Given two strings str1 and str2 and below operations that can be performed on str1.

- 1.Insert
- 2.Remove
- 3.Replace

Find minimum number of edits (operations) required to convert 'str1' into 'str2'. That is basically called Edit distance



The background features abstract, overlapping geometric shapes in various shades of green, ranging from light lime to dark forest green. These shapes are primarily located on the left and right sides of the frame, creating a modern, layered effect. The central area is a plain white background.

Thank You