Topics

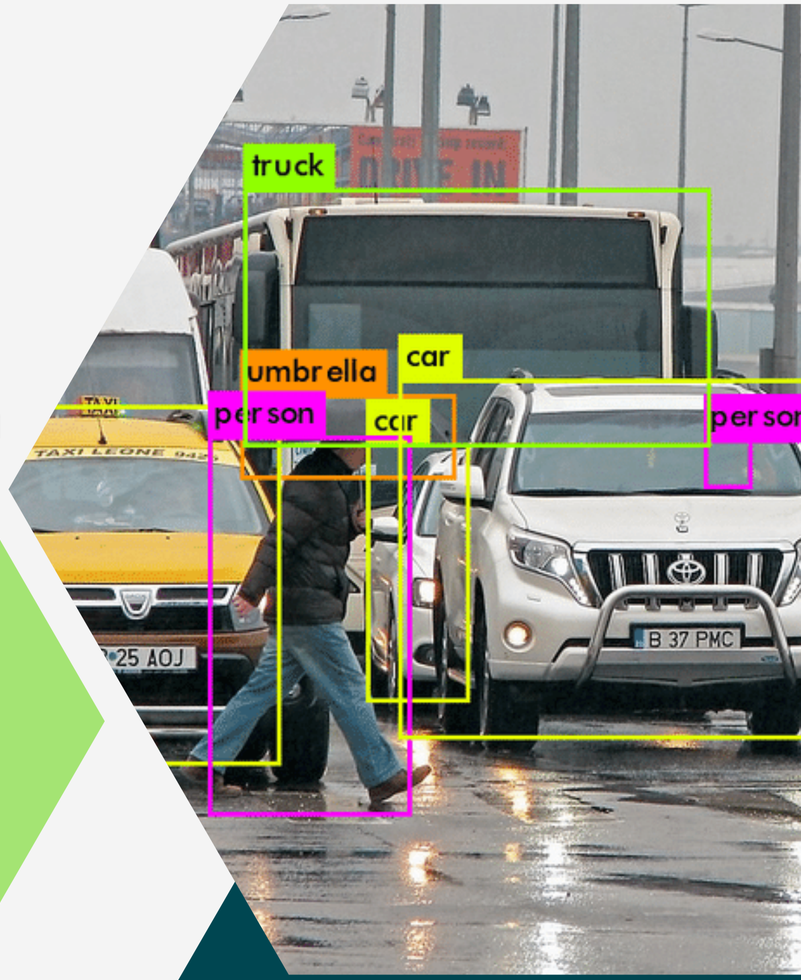
- What is Object Detection?
- Applications of object detection
- Different techniques of OD
- Difficulties faced in OD
- R-CNN
- YOLO
- Code Example using OpenCV

What is Object Detection?

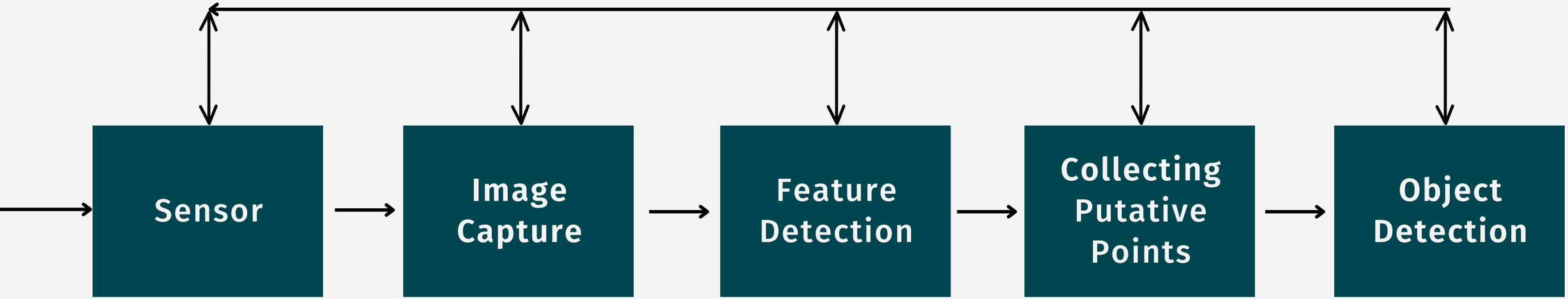
Object detection is a computer vision task that involves identifying and locating objects in images or videos.
Leverage machine learning or deep learning to produce meaningful results. When humans look at images or video, we can recognize and locate objects of interest within a matter of moments. Goal is to replicate this intelligence using a computer.

Applications of Object Detection

- Autonomous vehicles: Identify and track objects such as pedestrians, vehicles, and traffic signs in the environment, which is important for the safe navigation of self-driving cars.
- Surveillance: Identify and track individuals or objects of interest in surveillance videos, helping to improve security and public safety.
- Robotics: Help robots navigate and manipulate objects in their environment.
- Image/video analysis: Used to automatically classify and tag objects in images or videos, which can be useful for organizing and searching large collections of media.



Object Detection Flowchart



https://www.mathworks.com/help/vision/ug/object-detection-in-a-cluttered-scene-using-point-feature-matching.html

Different Object Detection Techniques

Single Shot Multibox Detector (SSD)

SSD is a real-time object detection technique that uses a single CNN to predict bounding boxes and class probabilities for objects in an image. SSD is fast and accurate but may not be as precise as some other techniques.

You Only Look Once (YOLO)

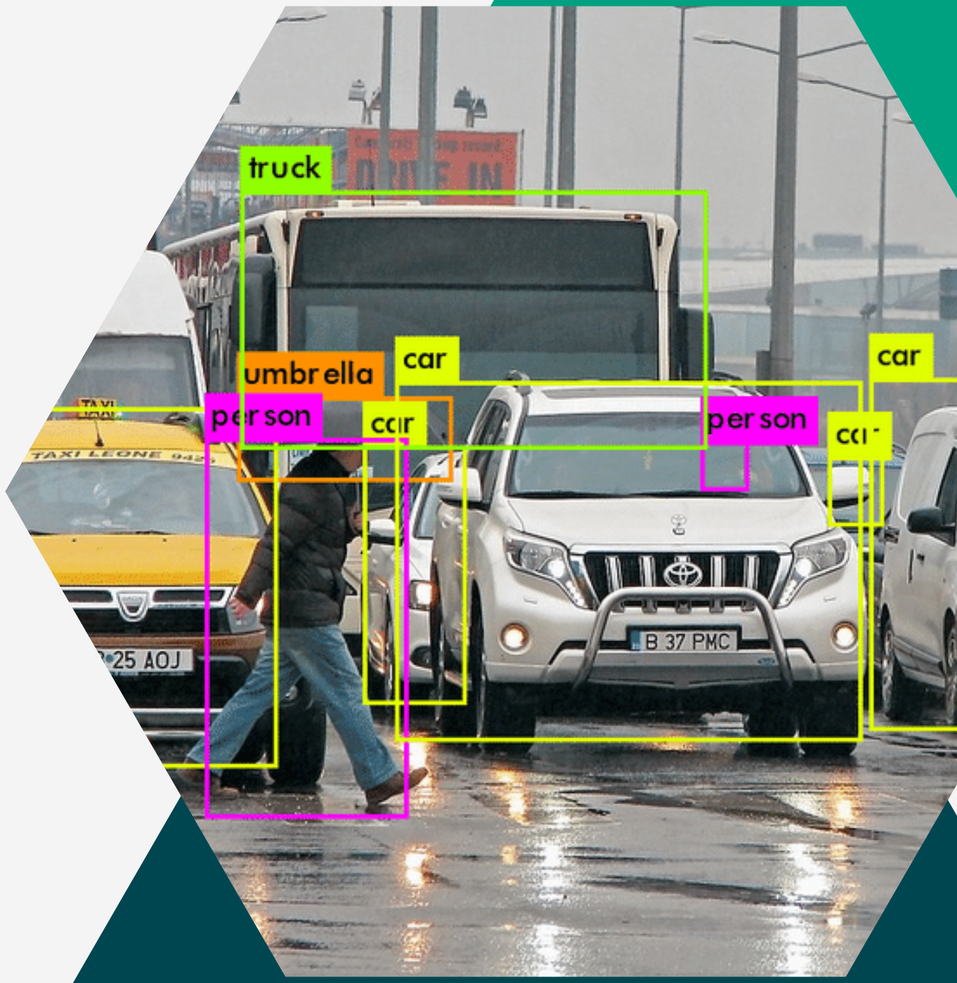
YOLO is a real-time object detection technique that uses a single CNN to predict bounding boxes and class probabilities for objects in an image. YOLO is fast and accurate but may not be as precise as some other techniques.

Region-based Convolutional Neural Networks (R-CNN)

R-CNN is a deep learning-based object detection technique that uses a convolutional neural network (CNN) to identify and classify objects in images. R-CNN is accurate but can be slow due to the need to perform region proposal and separate classification steps.

Difficulties in Object Detection

- Illumination: The lightning conditions may differ during the course of the day.
- Positioning: The change in position must not affect the recognition system.
- Rotation: The image can be in rotated form. The system must be capable to handle such difficulties.
- Occlusion: The condition when object in an image is not completely visible is reffered as occlusion.
- Mirroring: The mirrored image must be recognised by the system.
- Scaling: Changes in the size must not affect the recognition system



What is R-CNN?

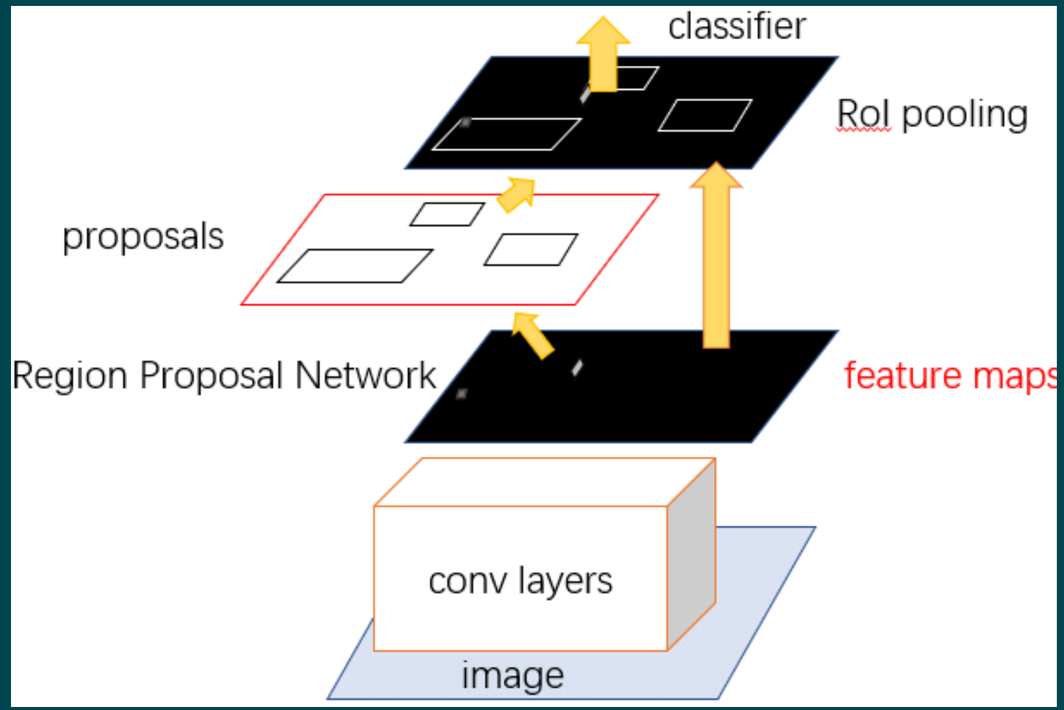
Region-based Convolutional Neural Networks (R-CNN) is a deep learning-based object detection technique that uses a convolutional neural network (CNN) to identify and classify objects in images. R-CNN consists of three main steps: region proposal, feature extraction, and classification.

How R-CNN Works?

- Region proposal: In the region proposal step, R-CNN generates a set of candidate regions or "proposals" in the image that are likely to contain objects. This is typically done using a sliding window approach or by using a pre-trained object proposal algorithm such as Selective Search.
- Feature extraction: In the feature extraction step, R-CNN uses a CNN to extract features from each of the proposed regions. These features are used to represent the content of the region and are passed to the classifier for further processing.
- Classification: In the classification step, R-CNN uses a linear Support Vector Machine (SVM) classifier to predict the class label (e.g. pedestrian, car, etc.) for each proposed region. The classifier is trained on a large dataset of annotated images to learn how to classify objects based on their features.

Advantages of R-CNN

- High accuracy: R-CNN has been shown to achieve very high accuracy on a variety of object detection tasks.
- Robust to changes in scale and orientation: R-CNN is able to handle a wide range of object scales and orientations, making it relatively robust to these types of variations.
- Can handle multiple object classes: R-CNN can be trained to detect multiple object classes, making it a versatile technique for many applications.

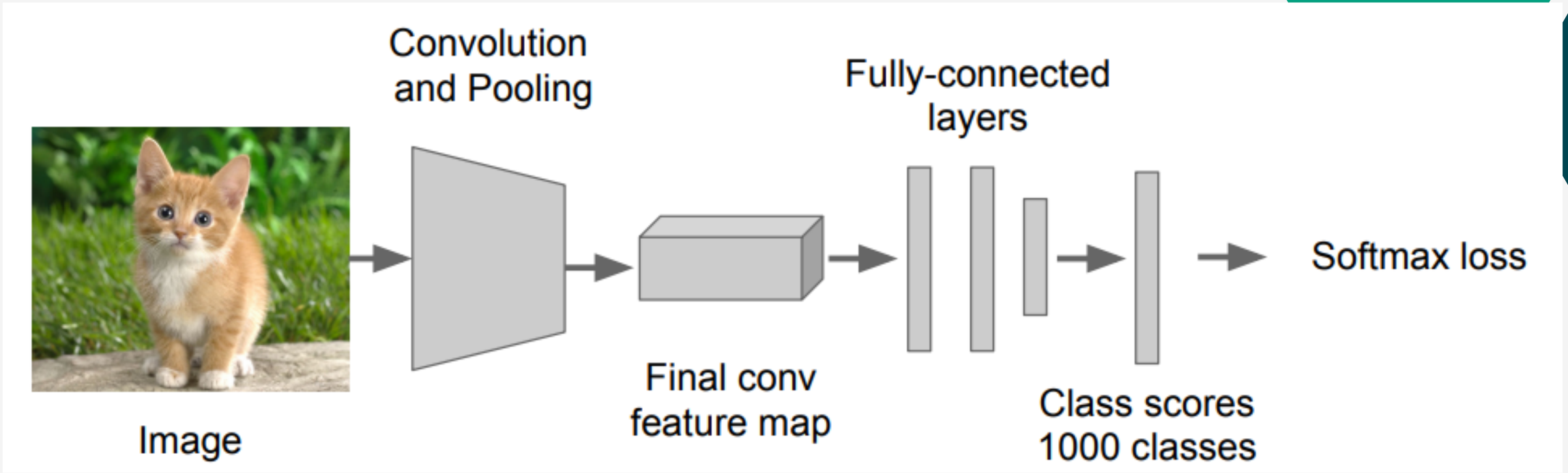


Disadvantages of R-CNN

- Slow: R-CNN can be slow due to the need to perform region proposals and separate classification steps, which can make it less suitable for real-time applications.
- Complex to implement: R-CNN is a complex technique that requires multiple steps and components, which can make it challenging to implement and maintain.
- Sensitive to initialization: R-CNN can be sensitive to the initialization of the CNN and SVM classifiers, which can affect its performance.

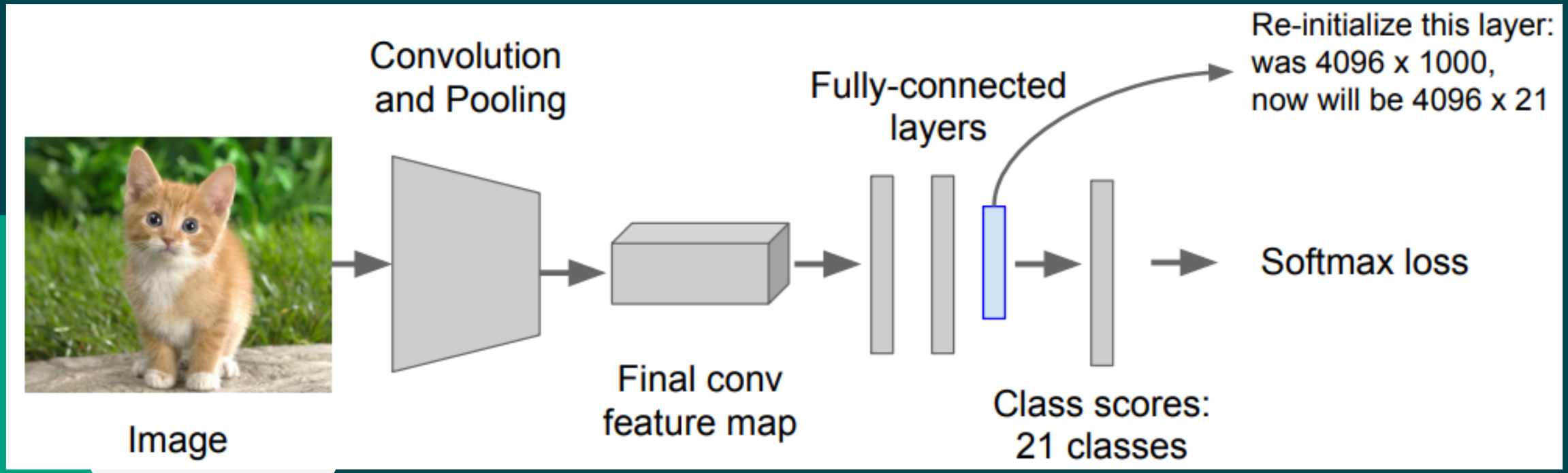
R-CNN Working

Step 1 : Train (or download) a classification model for ImageNet (AlexNet)



Step 2 : Fine-tune model for detection

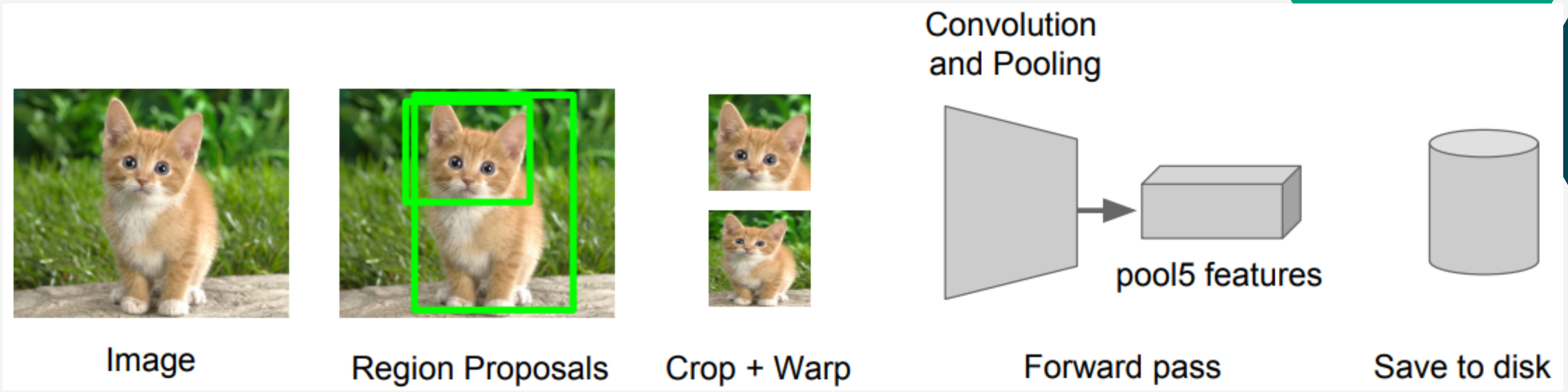
- Instead of 1000 ImageNet classes, want 20 object classes + background
- Throw away final fully-connected layer, reinitialize from scratch
- Keep training model using positive / negative regions from detection images



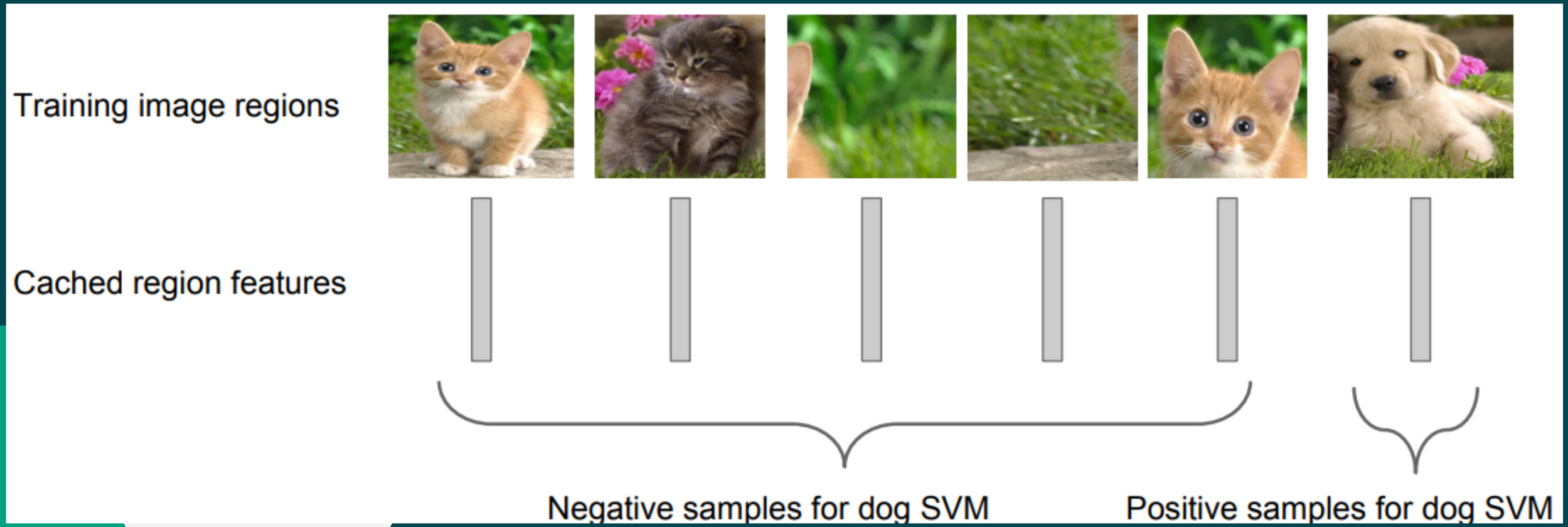
Step 3 : Extract features

- Extract region proposals for all images

- For each region: warp to CNN input size, run forward through CNN, save pool5 features to disk

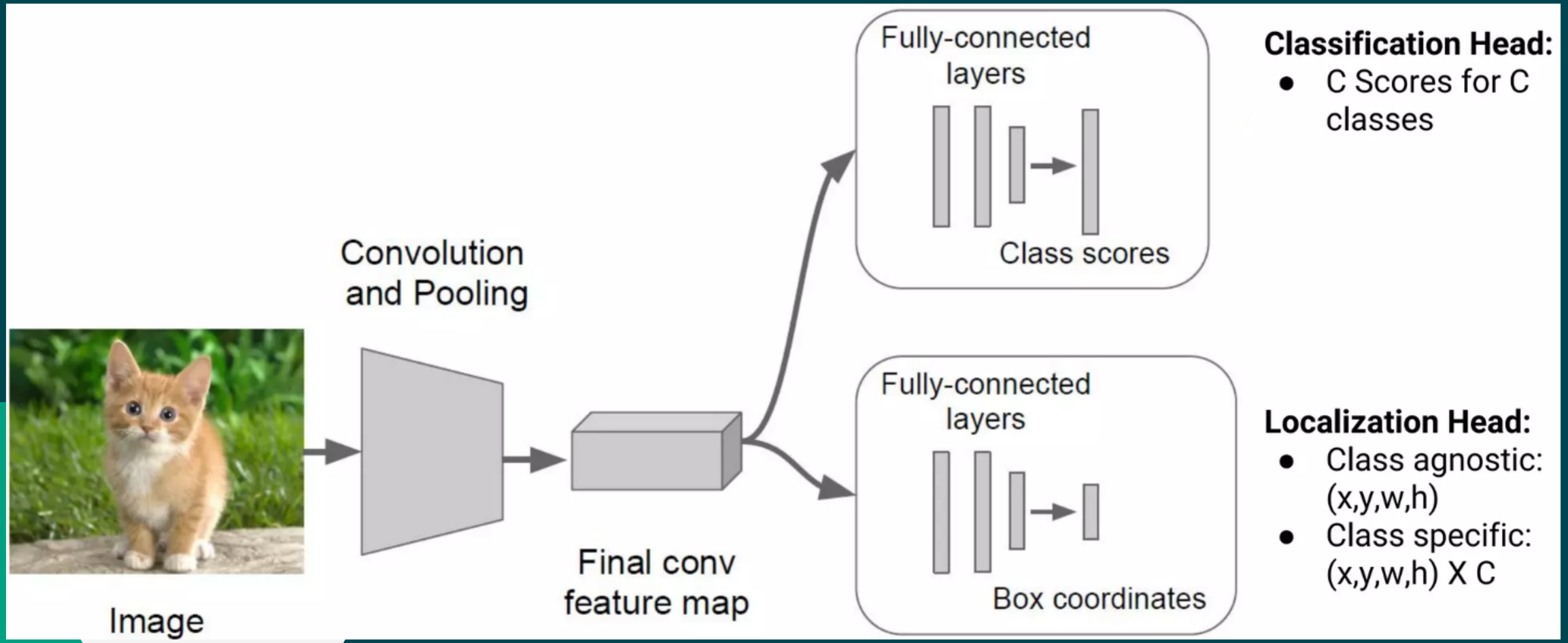


Step 4 : Train one binary SVM per class to classify region features



Step 5 (bbox regression) : For each class, train a linear regression model to map from cached features to offsets to GT boxes to make up for "slightly wrong" proposals

Clasification + Localisation Model



What is YOLO?

You Only Look Once (YOLO) is a real-time object detection technique that uses a single convolutional neural network (CNN) to predict bounding boxes and class probabilities for objects in an image. YOLO divides the input image into a grid of cells and assigns each cell a set of bounding boxes and class probabilities. During training, the CNN is trained to predict the class probabilities and bounding box coordinates for each cell.

How YOLO Works?

- Input image: The input to the YOLO model is an image of any size.
- Grid division: The input image is divided into a grid of cells, with each cell responsible for predicting the bounding boxes and class probabilities for a small region of the image.
- Bounding box prediction: For each cell, YOLO predicts a set of bounding boxes and class probabilities. The bounding boxes are represented as coordinates (x, y, w, h) where (x, y) is the center of the box and (w, h) are the width and height of the box.
- Non-max suppression: To remove overlapping bounding boxes and improve the overall accuracy of the object detection, YOLO uses non-max suppression to select the bounding box with the highest class probability for each object.
- Output: The final output of YOLO is a set of bounding boxes and class probabilities for the objects in the input image.

Advantages of YOLO

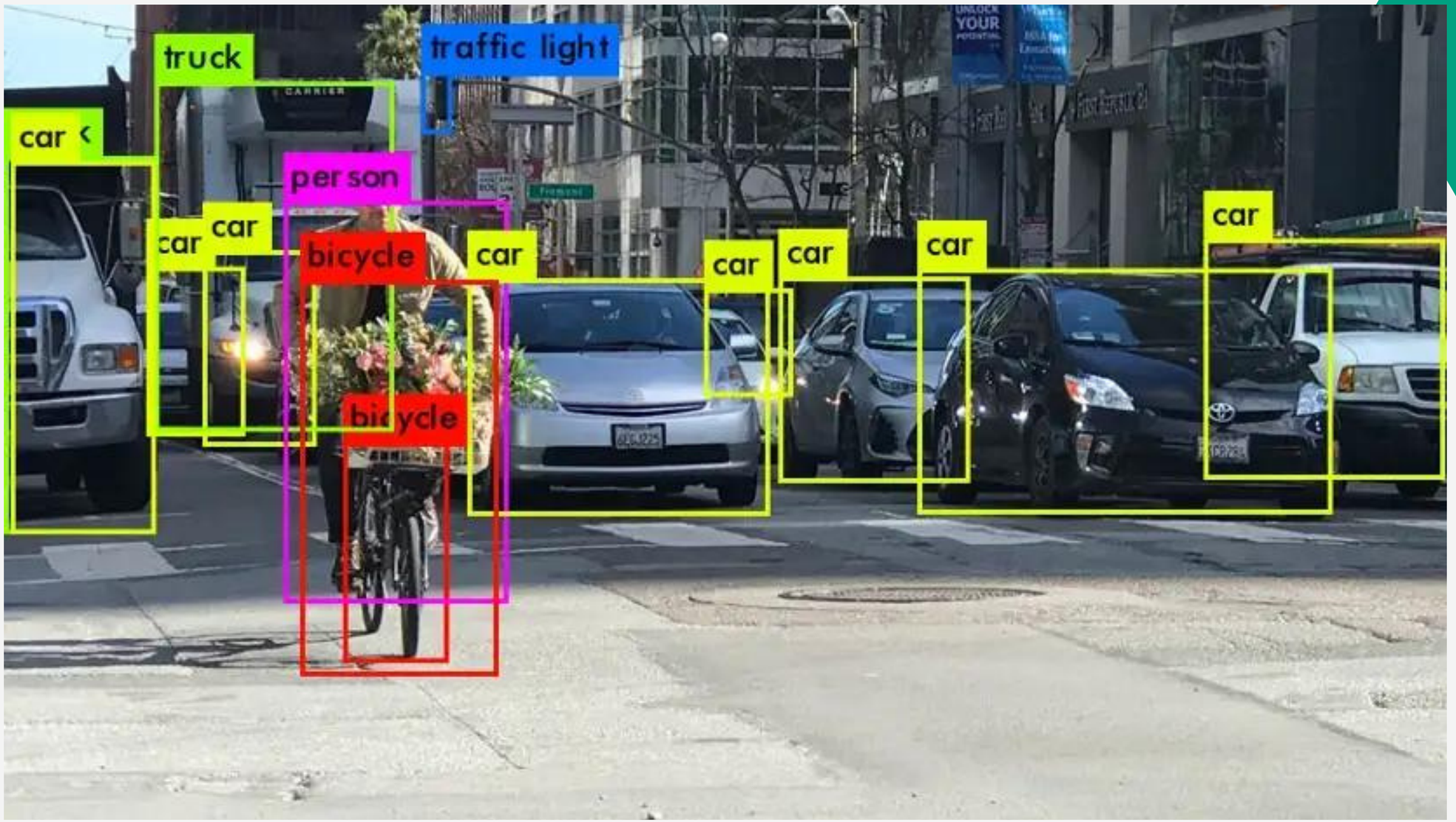
- Fast: YOLO is a real-time object detection technique that can process images quickly, making it well-suited for applications that require fast object detection.
- Simple to implement: YOLO is a relatively simple technique that is easy to implement and maintain, making it a good choice for many applications.
- Can handle multiple object classes: YOLO can be trained to detect multiple object classes, making it a versatile technique for many applications.

Disadvantages of YOLO

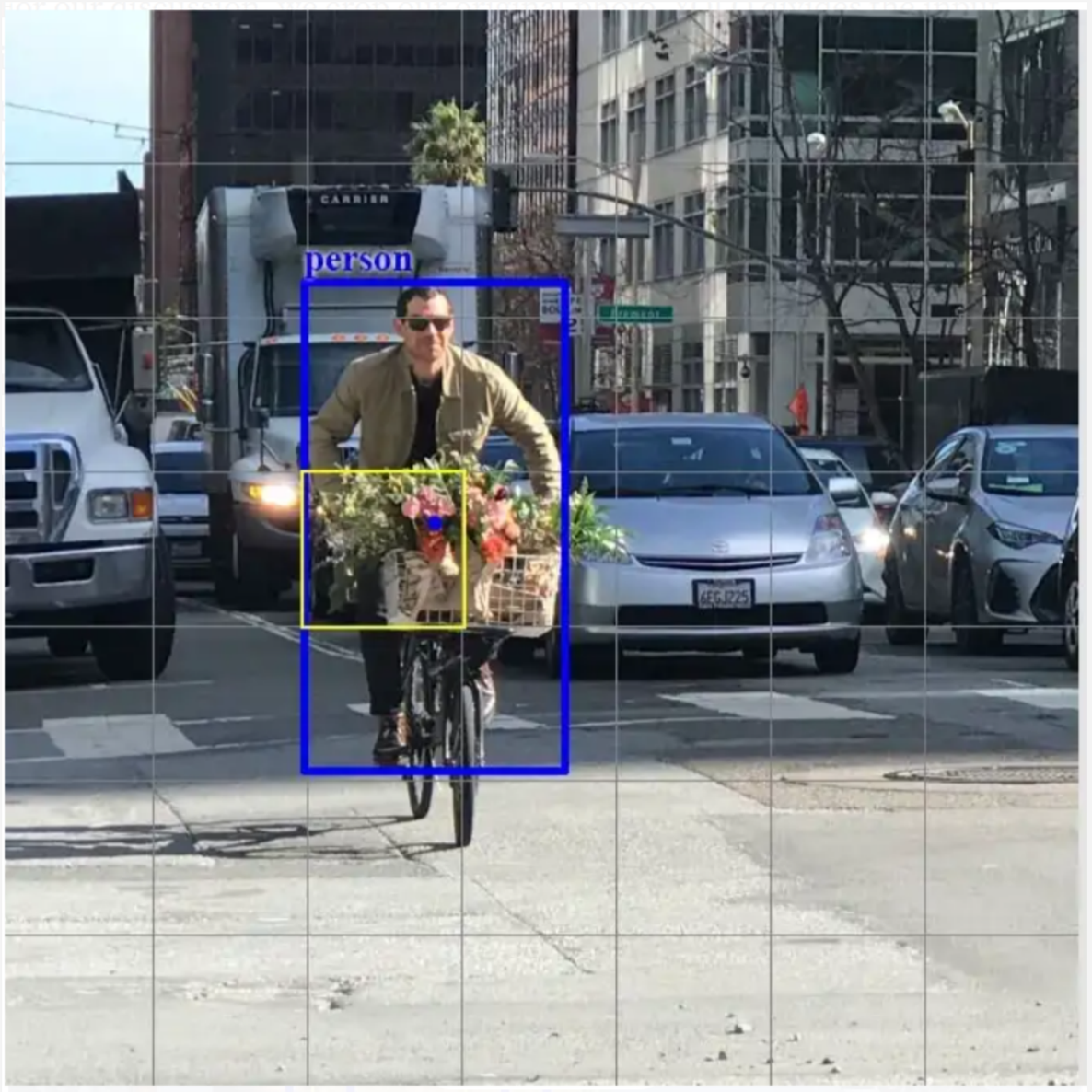
- May not be as precise as some other techniques: YOLO may not be as precise as some other object detection techniques, which can affect its accuracy.
- May not work well with small or complex objects: YOLO may not work as well with small or complex objects due to its reliance on a fixed grid of cells to divide the input image.
- Requires a large dataset for training: YOLO requires a large dataset of annotated images for training, which can be time-consuming and resource-intensive to obtain.

YOLO Working

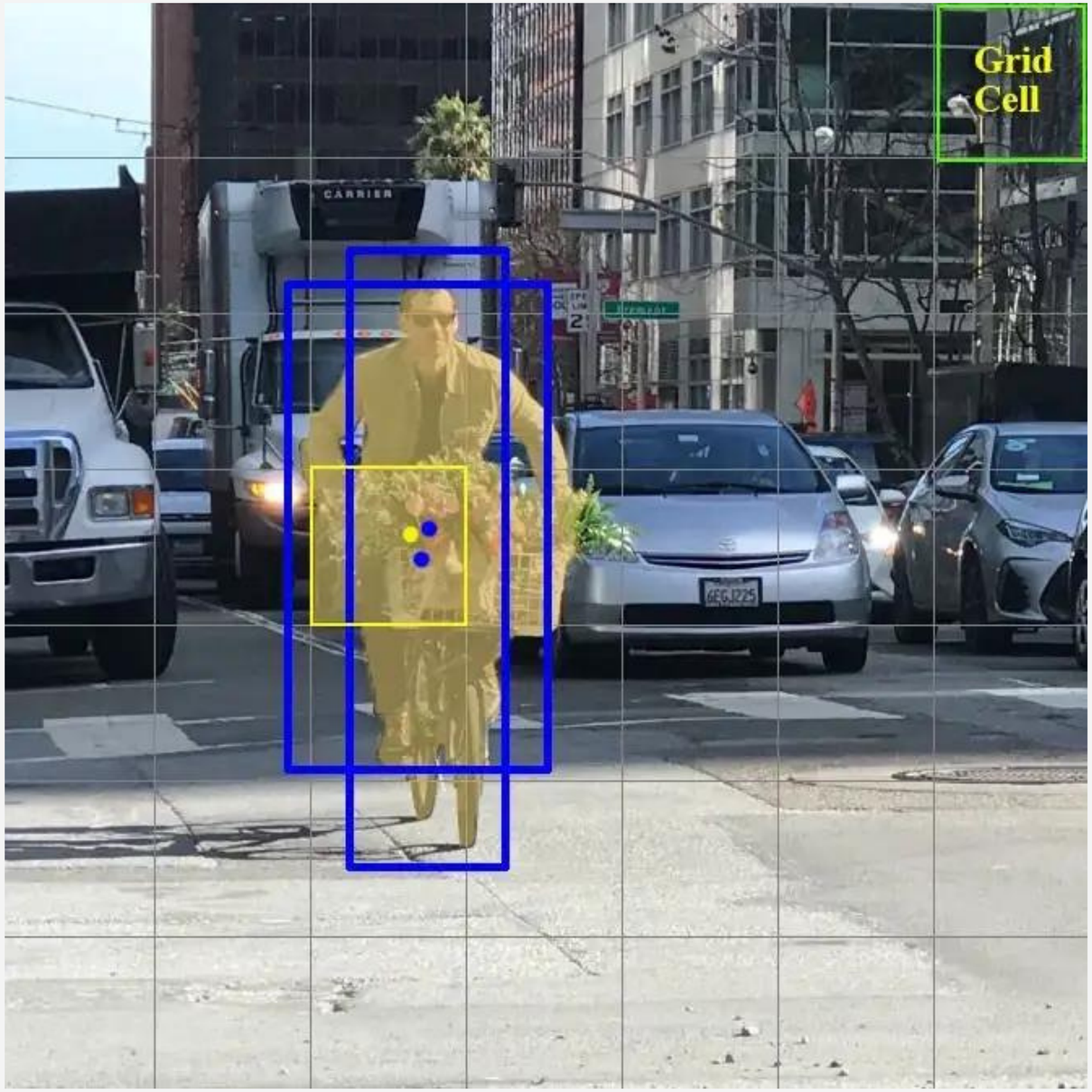
YOLO Working



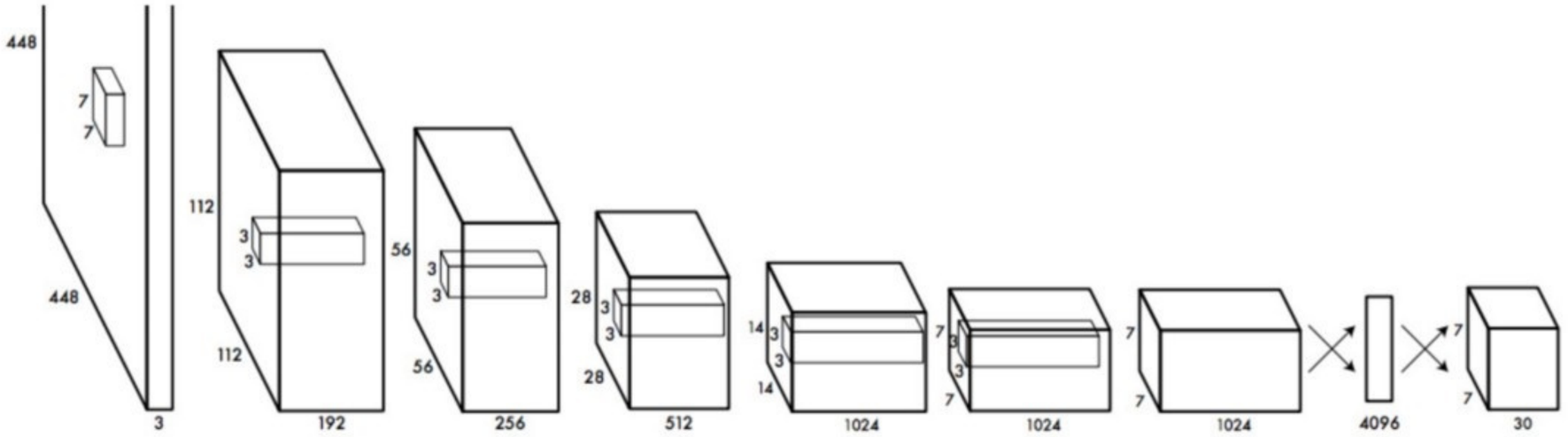
YOLO Working



YOLO Working



YOLO architecture



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.

Future Potential Development

Improved Accuracy

There is always a demand for more accurate object detection techniques, and researchers are working on developing new methods and approaches that can achieve higher levels of accuracy.

Real-time Performance

Many applications of object detection require fast processing of images and videos, and there is ongoing research into developing real-time object detection techniques that can meet these demands.

Improved Handling of small and complex objects

Object detection can be challenging for small or complex objects, and researchers are working on developing techniques that can better handle these types of objects.

Future Potential Development

Increased Robustness

Object detection algorithms need to be robust to a variety of factors such as changes in lighting, background, and object appearance, and researchers are developing methods to improve the robustness of object detection algorithms.

Increased Flexibility

There is a need for object detection techniques that are flexible and can be easily adapted to different domains and applications. Researchers are working on developing techniques that are more modular and can be easily customized for different scenarios.

Conclusion

Object detection is a key ability for most computer and robot vision system. Although great progress has been observed in the last years, and some existing techniques are now part of many consumer electronics (e.g., face detection for auto-focus in smartphones) or have been integrated in assistant driving technolo-gies, we are still far from achieving human-level performance, inparticular in terms of open-world learning.

References

https://www.kaggle.com/getting-started/169984

https://arxiv.org/pdf/1905.05055.pdf

https://arxiv.org/pdf/1611.07791.pdf

https://www.youtube.com/watch?v=RFqvTmEFtOE&ab_channel=DeepLearning_by_PhDScholar

https://www.youtube.com/watch?v=HXDD7-EnGBY&ab_channel=Murtaza%27sWorkshop-RoboticsandAI

https://jonathan-hui.medium.com/real-time-object-detection-with-yolo-yolov2-28b1b93e2088

Thank You!