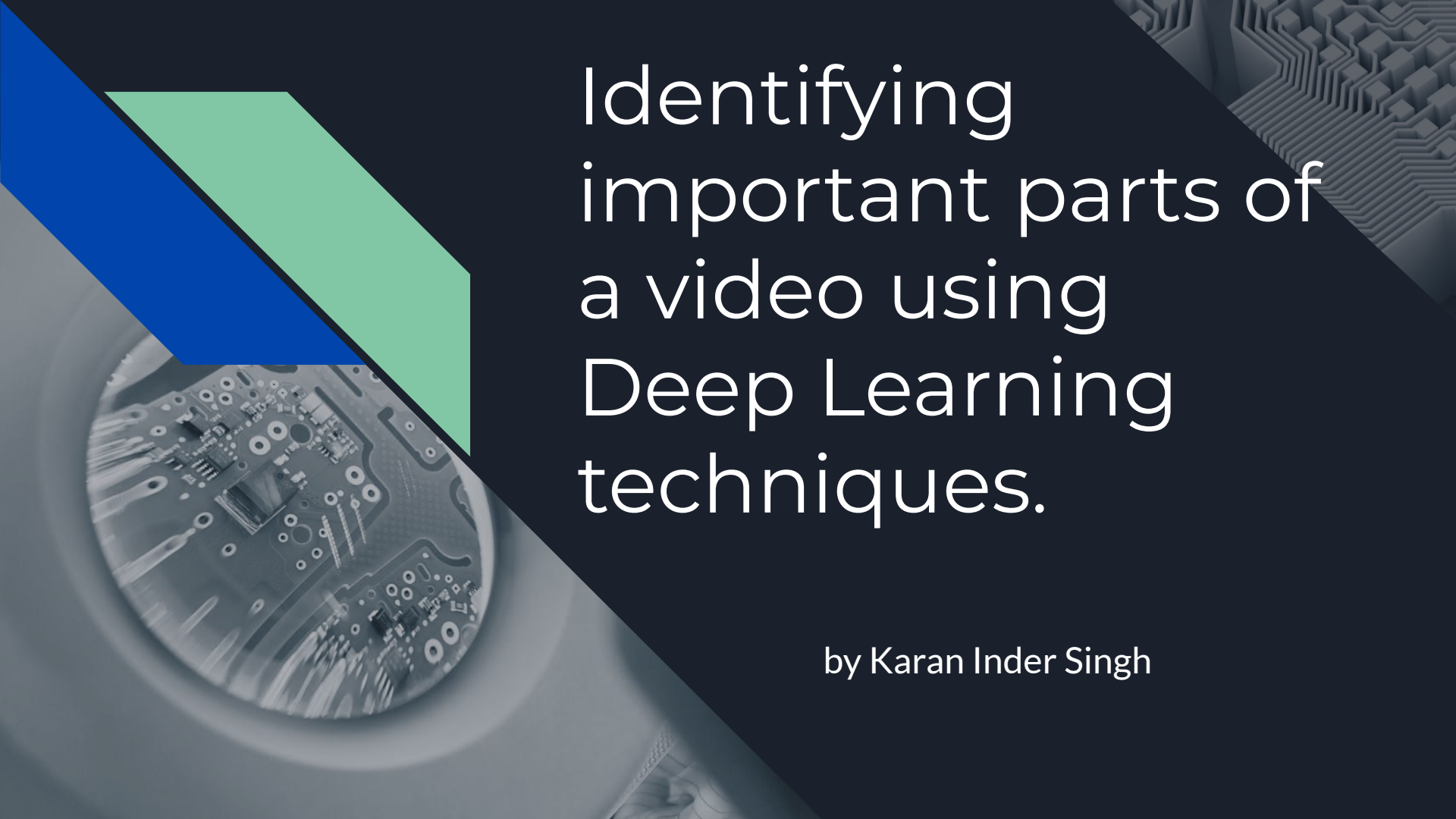


Identifying important parts of a video using Deep Learning techniques.

by Karan Inder Singh

Problem Statement

Make a software recognizes important parts of a video using Deep Learning Techniques.

Slide 10: Reasons for using YOLO

Further details: https://arxiv.org/pdf/1804.02767.pdf

Introduction

In deep learning-based object detection, there are 2 primary object detection methods:

- R-CNN and their variants (2 stage detector, high accuracy and extremely slow)
- YOLO (You Look Only Once)

RCNNs

R-CNNs are one of the first deep learning-based object detectors and are an example of a two-stage detector.

1. Proposing candidate bounding boxes (probable locations) that could contain objects.
2. CNN for classification, to classify what the object in the box actually is (car, kite, etc.)

It is highly accurate & extremely slow.

YOLO

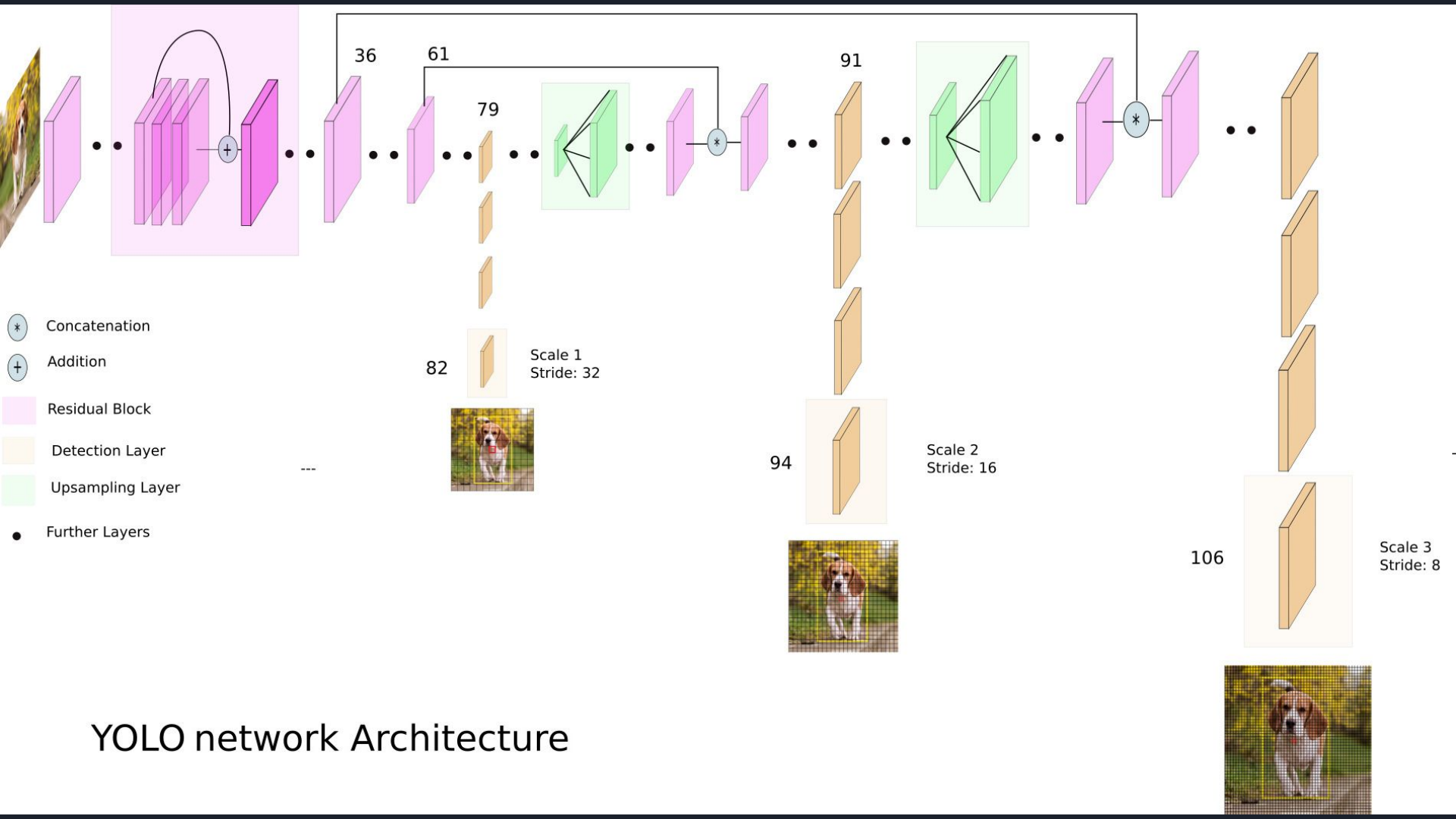
This method is extremely fast and due to limited hardware (laptop CPU) this is the ideal choice for implementation.

YOLO makes use of 75 CNN-layers (convolutional layers) + 31 other layers (shortcut(23), route(4), upsample(2), yolo(2)) = 106 layers in total.

YOLO

1. We made use of a dataset to create a convolution weight model which can be used to distinguish between different objects.
2. We then use this convolution weight model and input image/video as input to the neural network.

YOLO makes use of 3 different detection scales each at a different place in the network. This is achieved by applying a detection kernel on the features maps of three different sizes. These 3 detection layers are used to make predictions about the object.



YOLO network Architecture

Dataset

COCO is a large-scale object detection, segmentation, and captioning dataset. It consist of images with Object segmentation, Recognition in context, etc. It is one of the most commonly used datasets for any object recognition algorithm. Thousands of research papers have made use of this dataset for training, testing and validation of their machine learning & deep learning models.

It is used as a standard/benchmark for making comparisons & for speed and accuracy calculations.

It consists of 330K images (>200K labeled) & 1.5 million object instances.

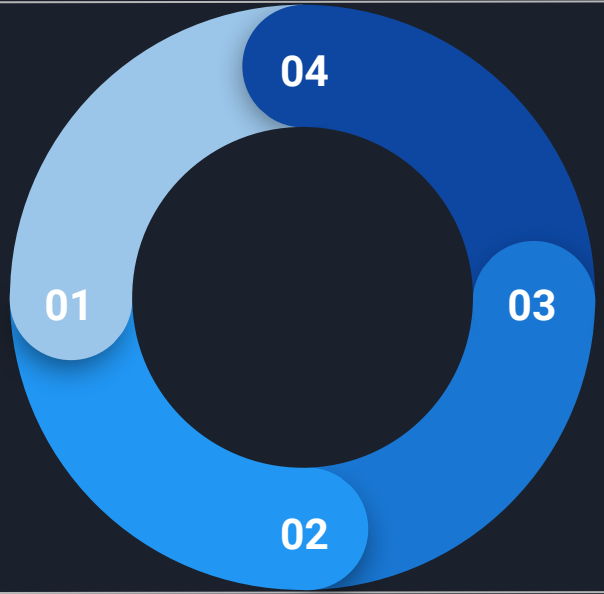
Tools/ Technologies Used

Anaconda / Python 3.6

Open-source distribution of the Python scientific computing, that aims to simplify package management and deployment.

Darknet

Darknet is an open source neural network framework written in C and CUDA. It is fast & easy to use.



OpenCV 4.0

Open Source Computer Vision Library enables computational efficiency and with a strong focus on real-time applications.

NumPy

Library for extremely fast and scientific computing for python.

Reasons for using YOLO

1. This method is fast and can run on basic hardware i.e. only an X64 CPU required.
2. This method does not employ the use of GPUs making it much more portable to other platforms & easy to deploy.
3. It does not depend on the use of libraries specifically compiled for use with a GPU like CUDNN which makes it hardware independent and portable.
4. Other methods **require** the the use of high end GPUs to offer feasible framerates, whereas this method performs very well on a CPU only.

References used:

- http://cocodataset.org/

  (link of dataset used COCO dataset)

- https://arxiv.org/pdf/1804.02767.pdf

  (link of paper implemented i.e. YOLOv3: An Incremental Improvement)

Thank You!