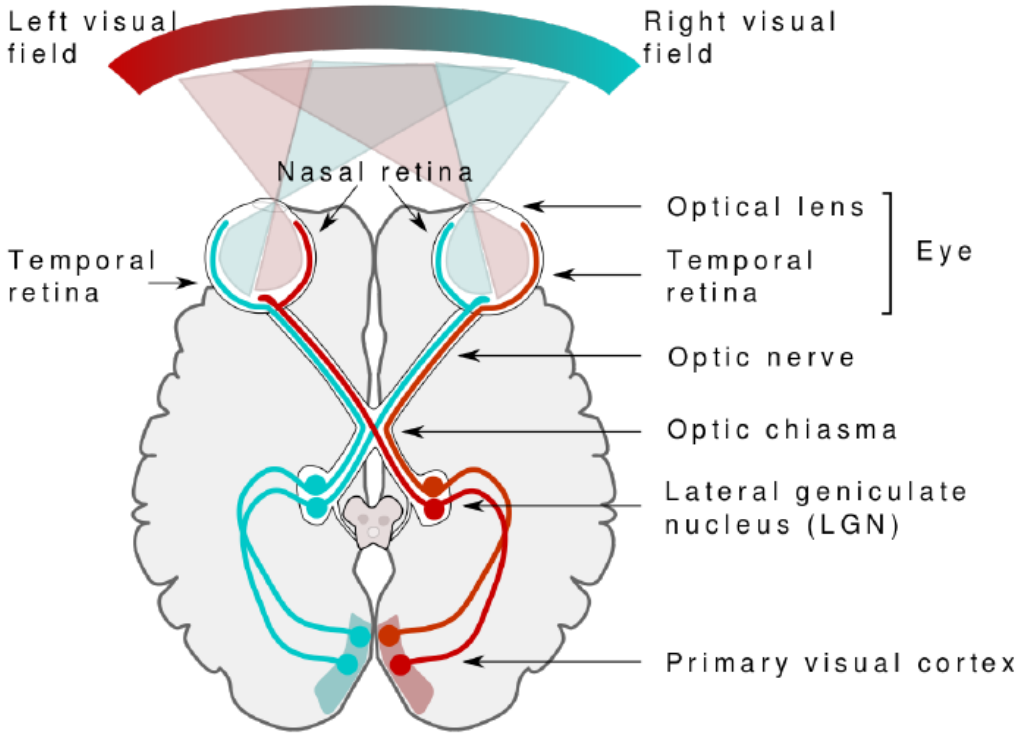
IOWA STATE UNIVERSITY

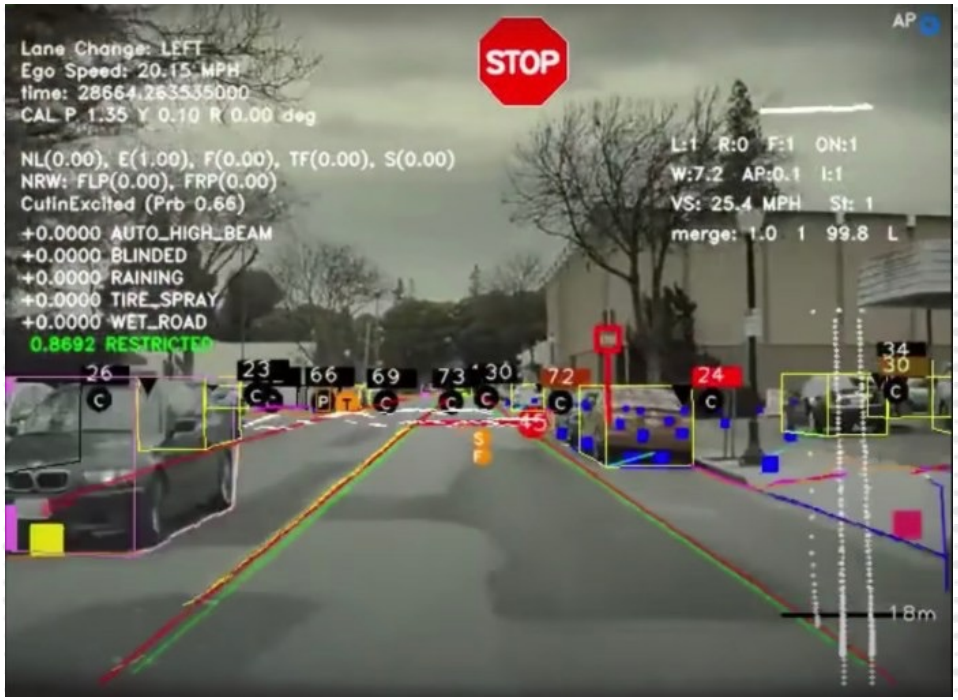
Translational AI Center

Computer Vision: What and Why

- Humans can see, interpret, and understand with our eyes and mind
- Teaching computers to do the same is computer vision
- Computer vision is a subset of artificial intelligence (AI)
- Produces meaningful outputs from:
  - Images
  - videos
  - and other visual datasets.

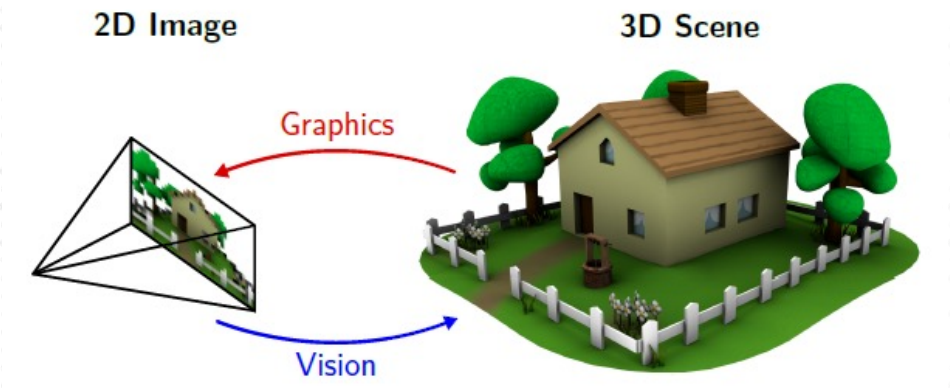


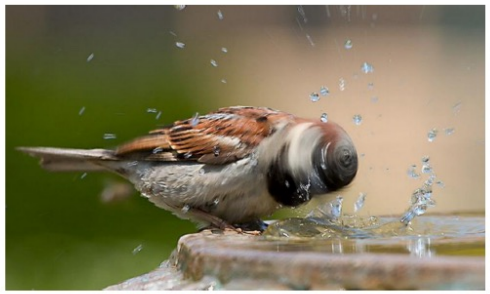
Computer Vision : What and Why

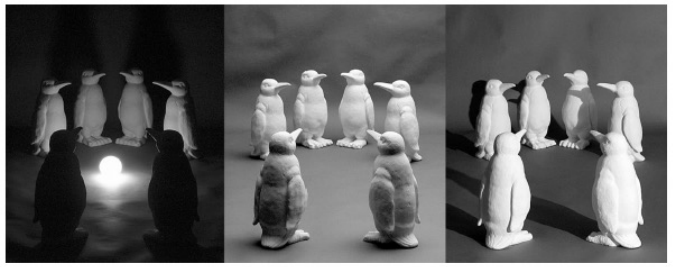


https://plainsight.ai/blog/autonomous-vehicles-computer-vision/

Computer Vision : What and Why







From CV Lecture by Andreas Geiger

Computer Vision : What and Why

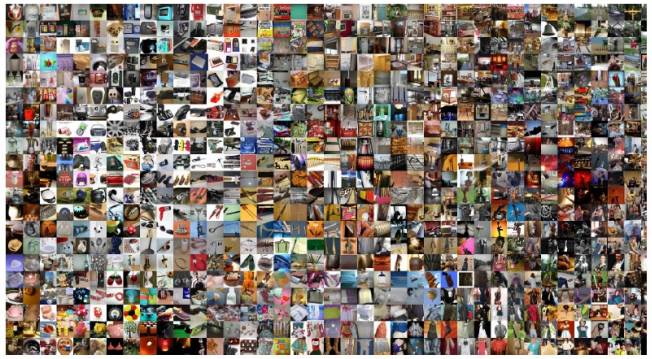
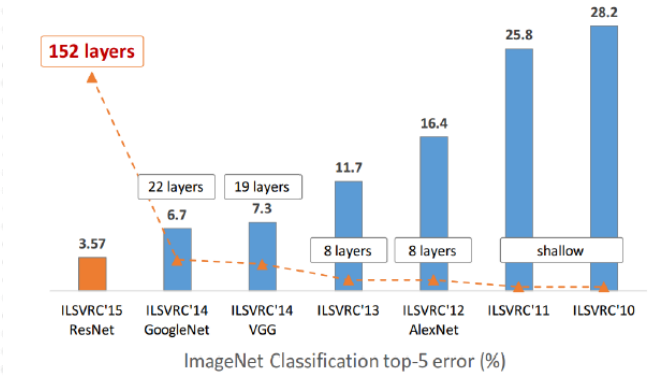


Figure 6: ImageNet



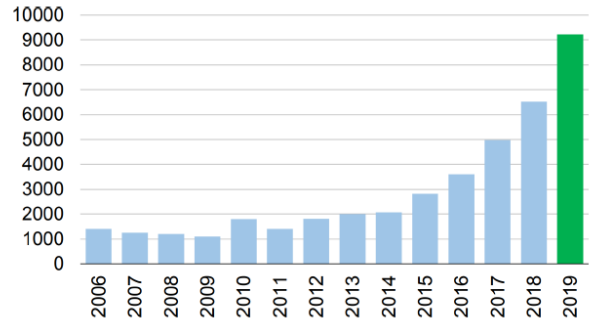


Figure 8: CVPR attendance

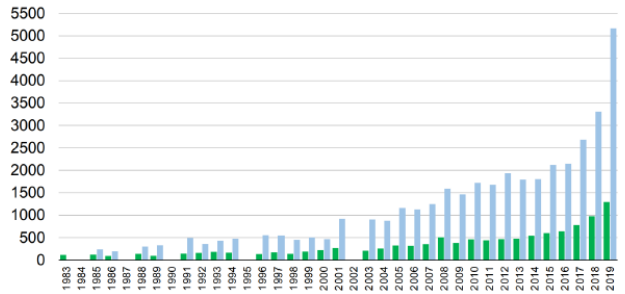
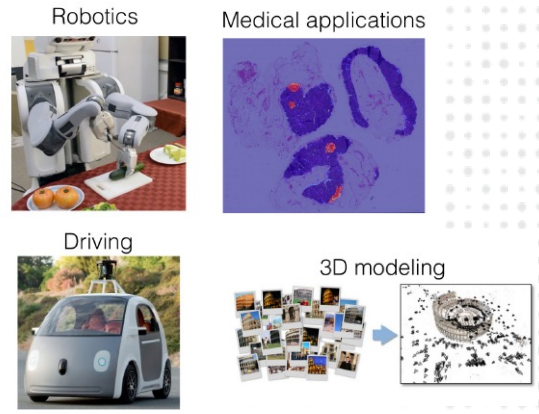


Figure 9: CVPR papers

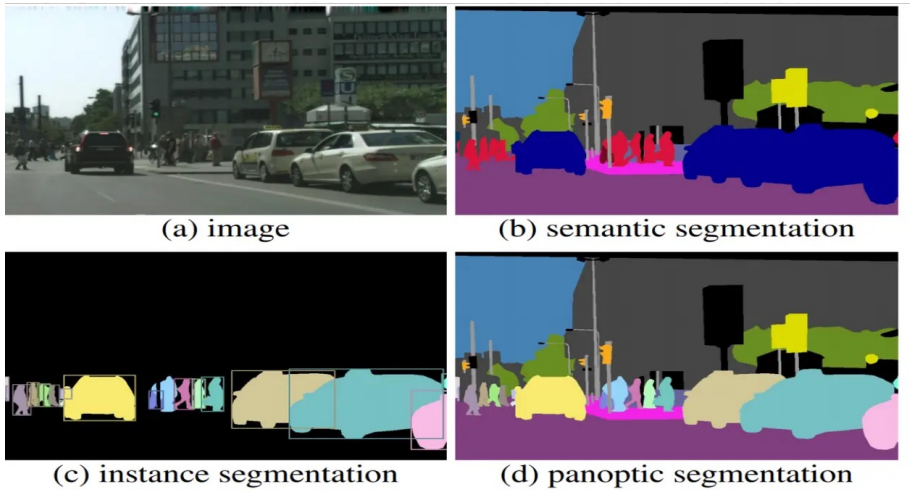


From CV Lecture by Andreas Geiger

CV Tasks : Recognition

- Object Recognition (also called image classification)
- Object Detection
- Keypoint Detection
- Semantic segmentation
- Semantic scene parsing

- Specialized tasks
  - Content-based image retrieval
  - Pose estimation
  - Optical character recognition
  - 2D code reading
  - Facial recognition



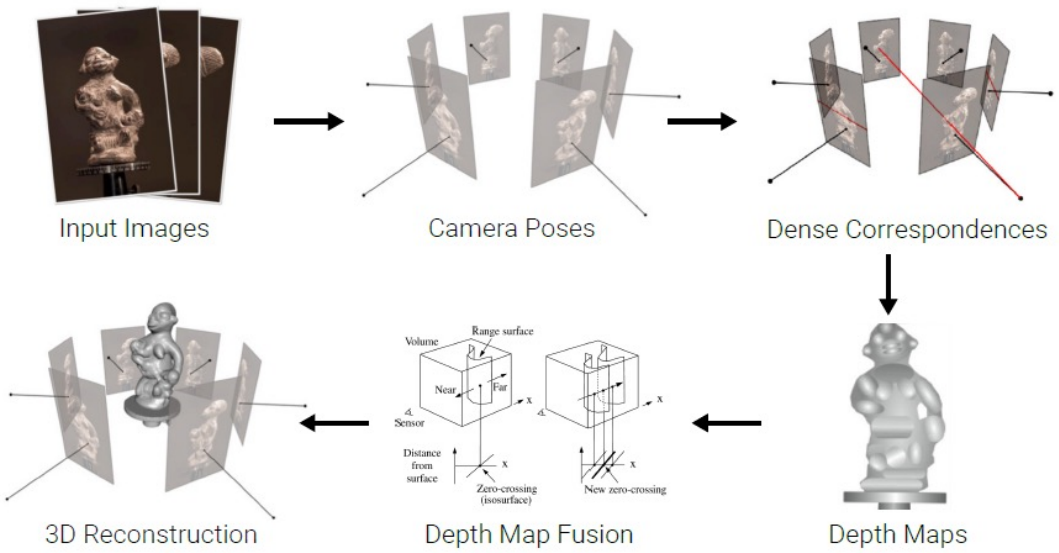
b) Pixelwise class label
c) Pixelwise instance label and bounding box of class
d) Pixelwise instance and class label

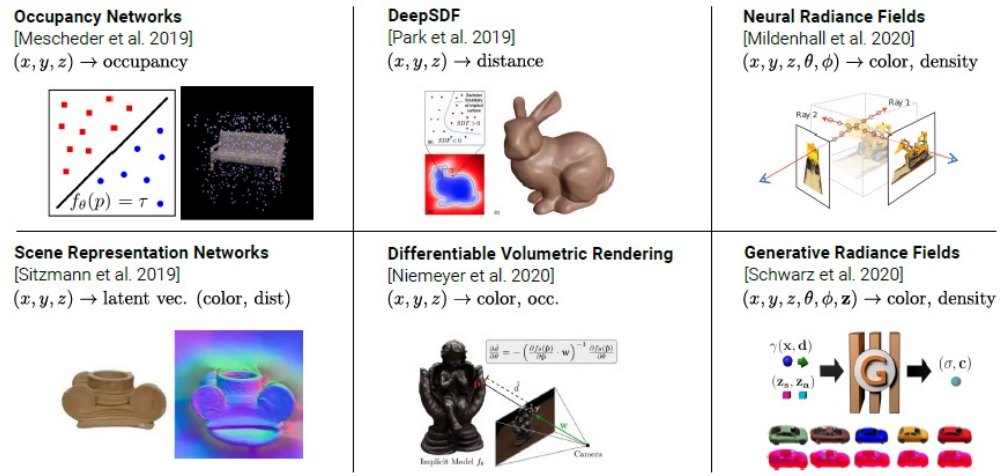
CV Tasks : Motion Analysis

- Egomotion
- Tracking
- Optical Flow

CV Tasks : More Task

- Scene reconstruction
  - 3D scene reconstruction
  - Semantic scene parsing
- Image restoration/Image filteri





Applications : Object Recognition

Traffic Sign Recognition (GTSRB)
- German Traffic Sign Reco Bench
- 99.46% accuracy (IDSIA, 2011)

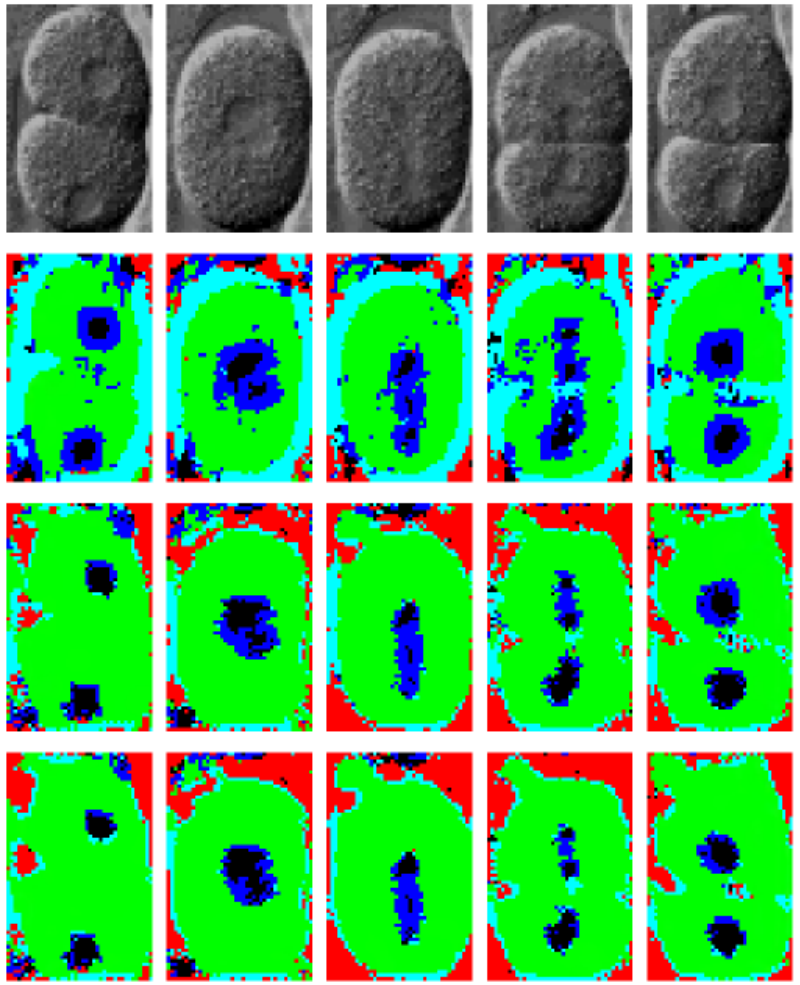
House Number Recognition (SVHN)
- Street View House Numbers
- 96 % accuracy (Google, 2014)

ConvNets work pretty well with small images and characters

From NVIDIA teaching kit

Applications: Image Segmentation

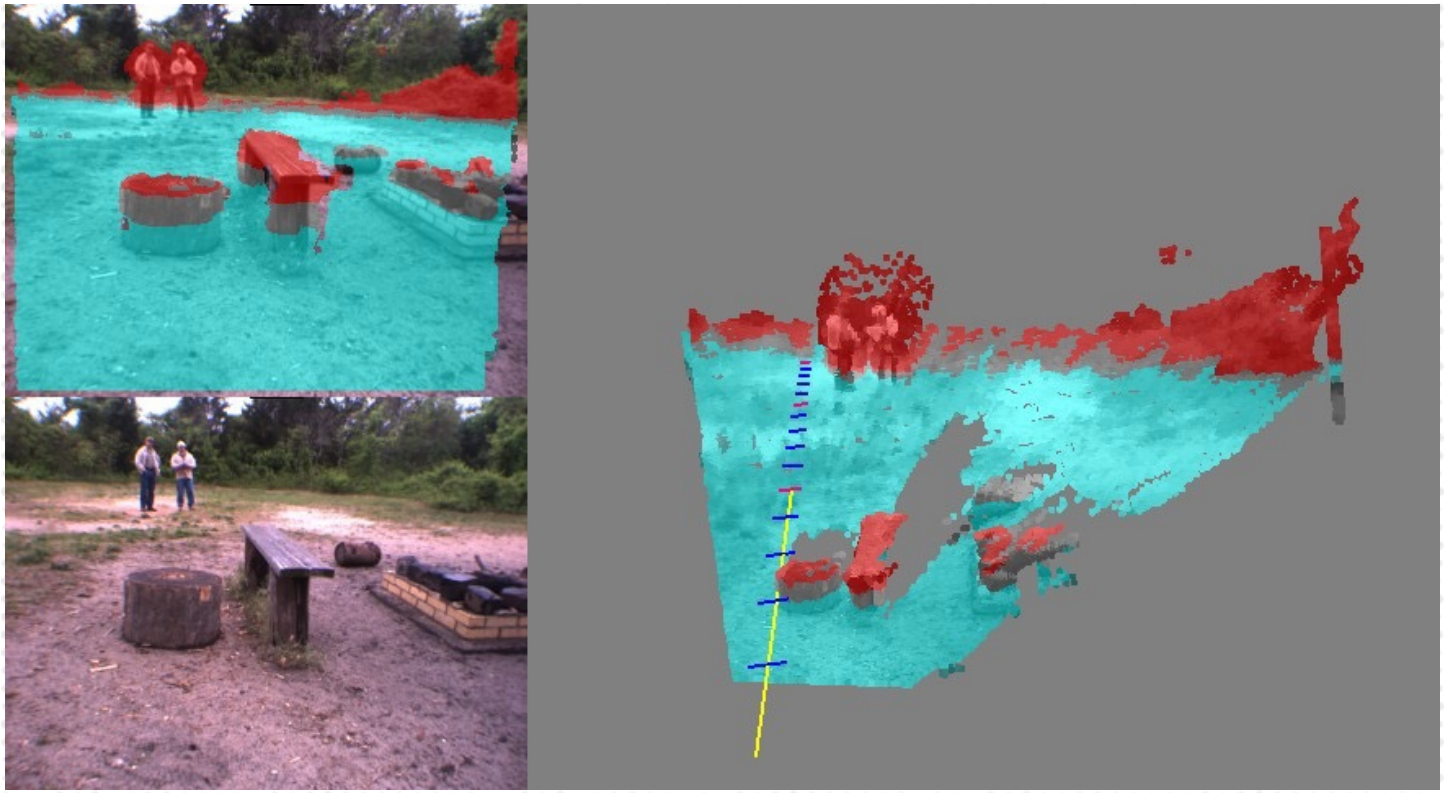
- Biological Image Segmentation
  - [Ning et al. IEEE-TIP 2005]



From NVIDIA teaching kit

CNN Applications: Vision-based navigation

Obstacles overlaid with camera image

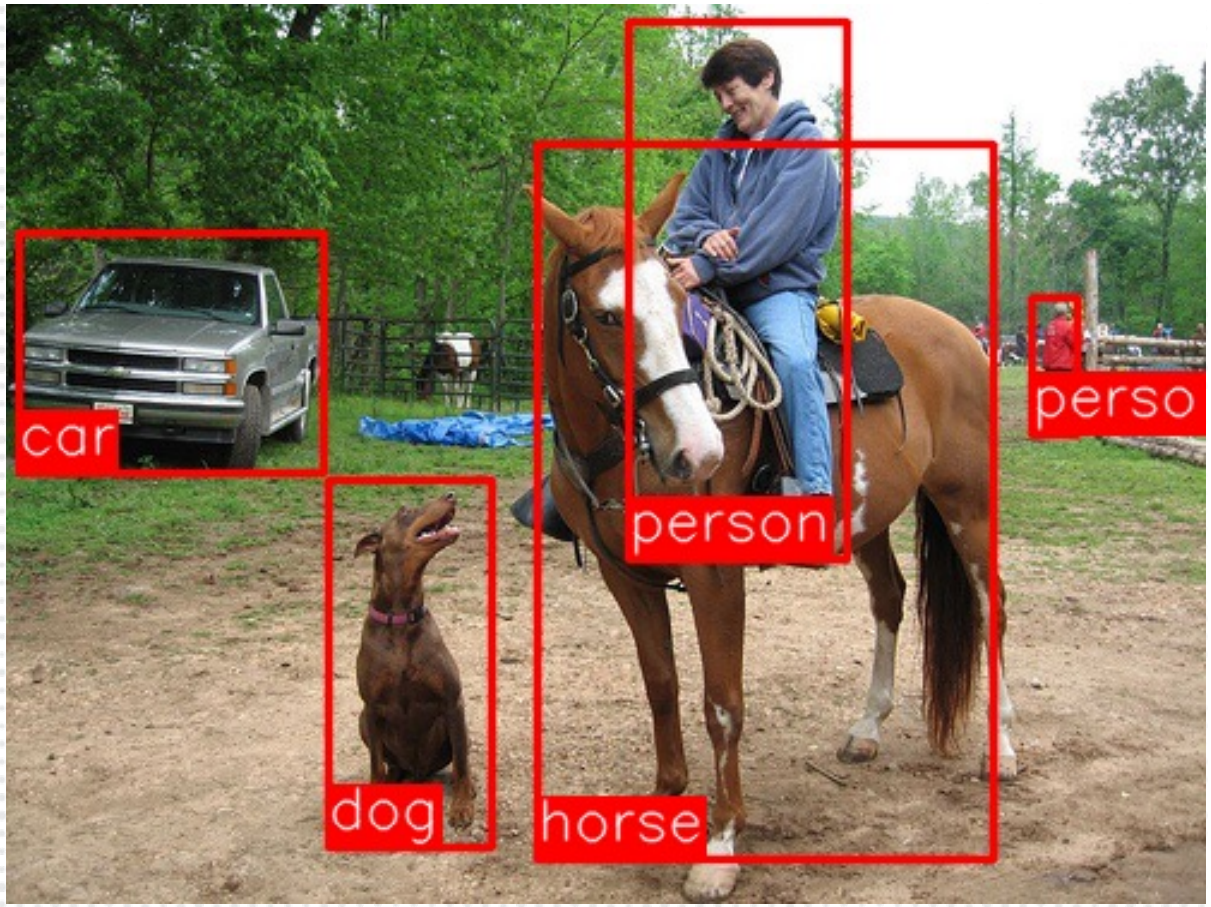


Camera image

Detected obstacles (red)

From NVIDIA teaching kit

CNN Applications: Classification + Localization



YOLO

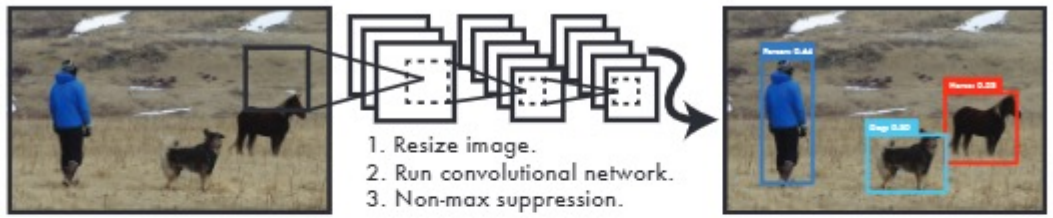


Figure 1: The YOLO Detection System. Processing images with YOLO is simple and straightforward. Our system (1) resizes the input image to 448 × 448, (2) runs a single convolutional network on the image, and (3) thresholds the resulting detections by the model's confidence.

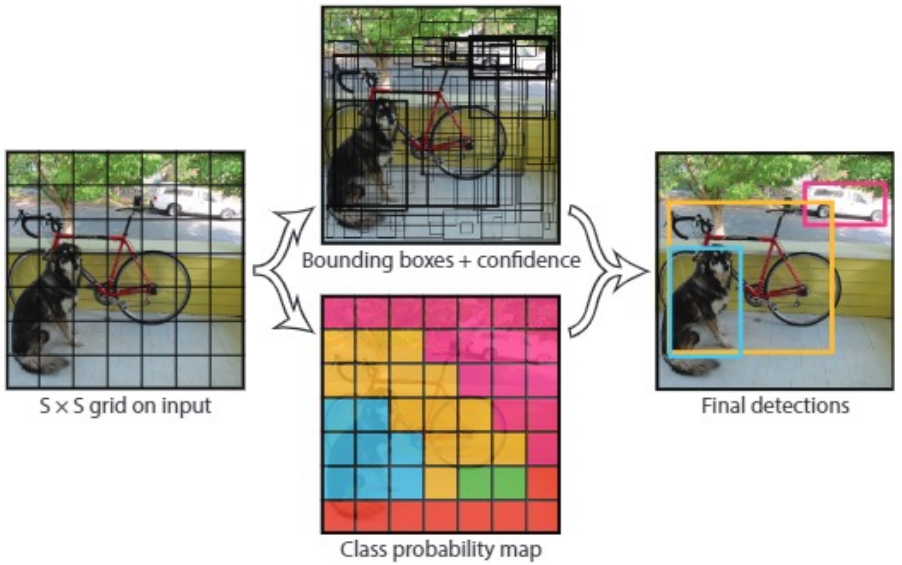
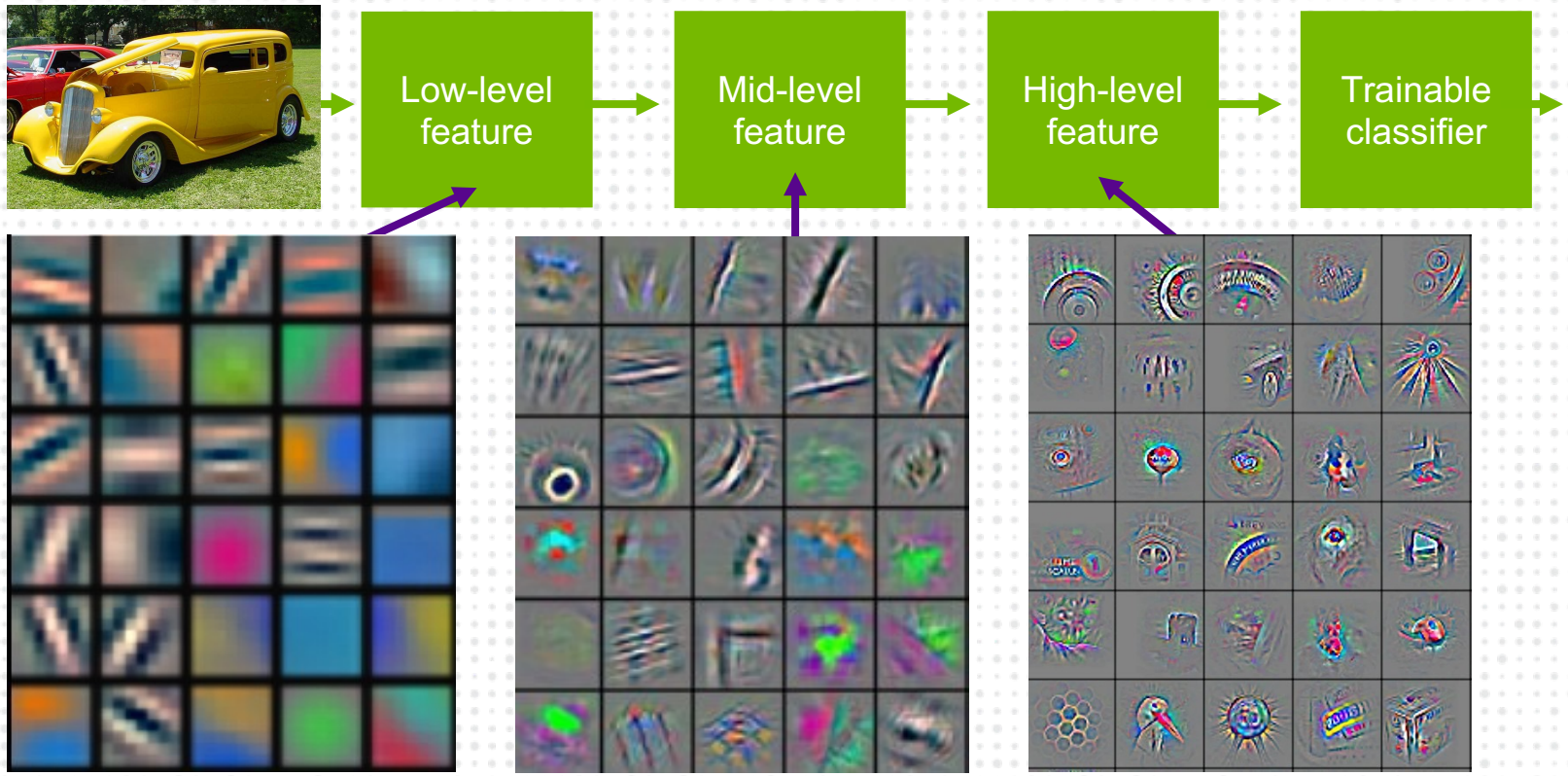


Figure 2: The Model. Our system models detection as a regression problem. It divides the image into an $S \times S$ grid and for each grid cell predicts B bounding boxes, confidence for those boxes, and C class probabilities. These predictions are encoded as an $S \times S \times (B * 5 + C)$ tensor.

https://arxiv.org/abs/1506.02640

CNN = Learning hierarchical features

It's deep if it has more than one stage of non-linear feature transformation



Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]

From NVIDIA teaching kit

Vision Transformers

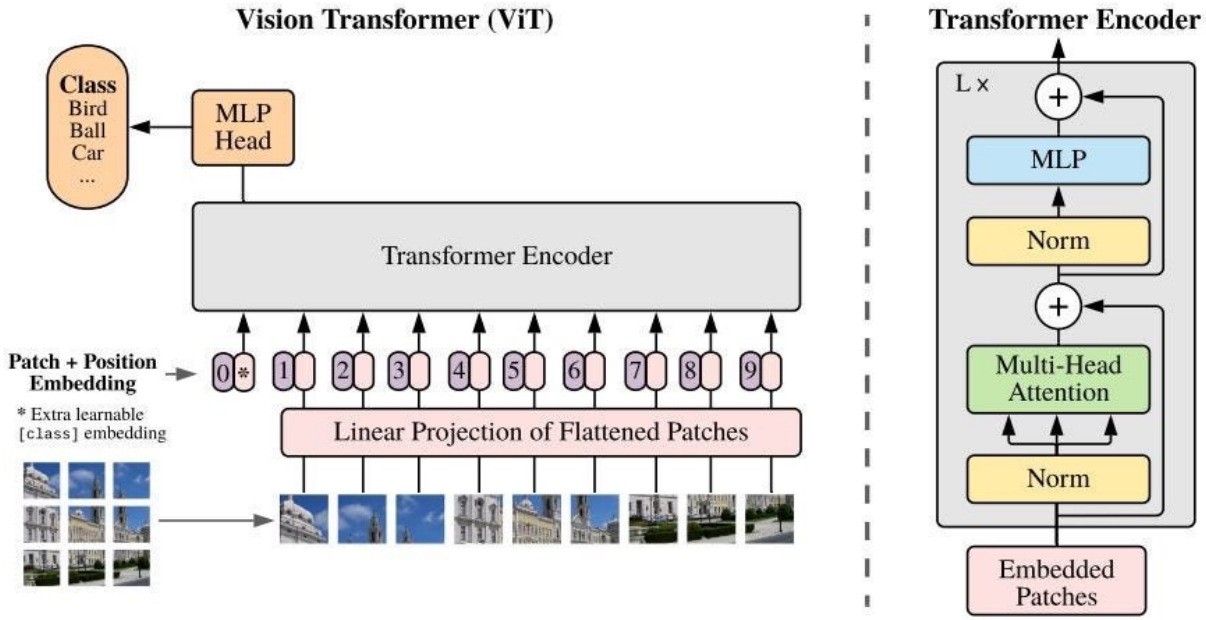


Figure 1: Model overview. We split an image into fixed-size patches, linearly embed each of them, add position embeddings, and feed the resulting sequence of vectors to a standard Transformer encoder. In order to perform classification, we use the standard approach of adding an extra learnable "classification token" to the sequence. The illustration of the Transformer encoder was inspired by Vaswani et al. (2017).

https://arxiv.org/abs/2010.11929v2

CLIP (Contrastive Language–Image Pre-training)

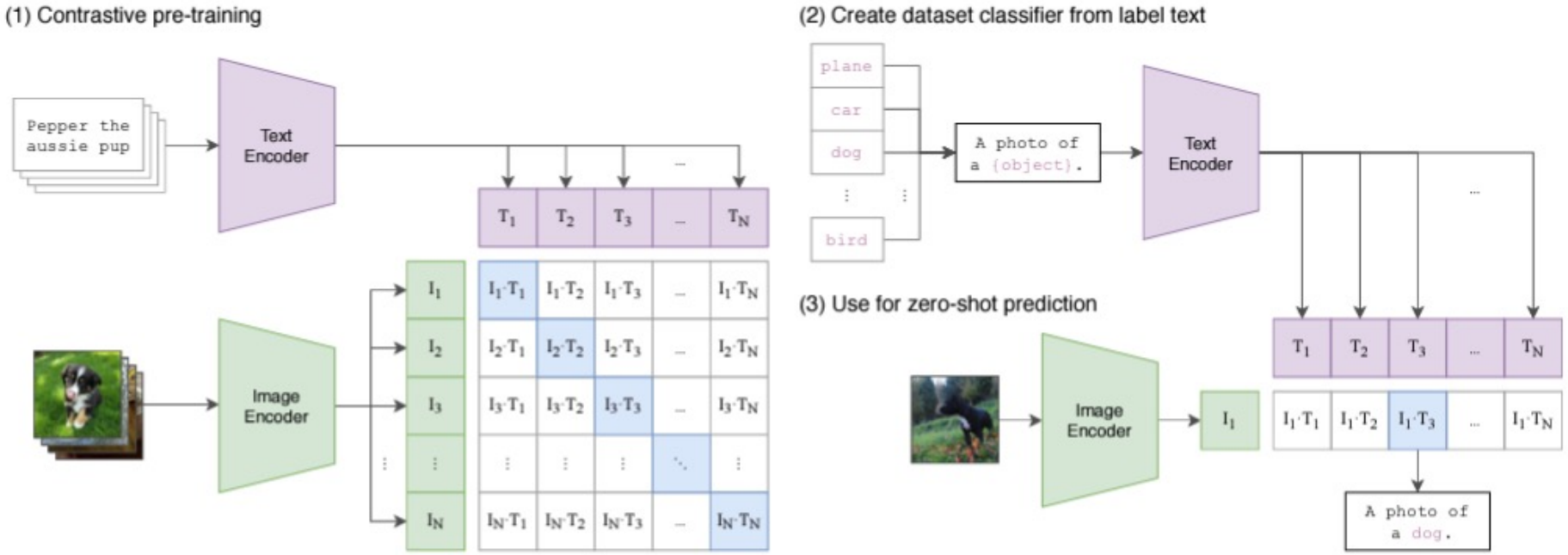
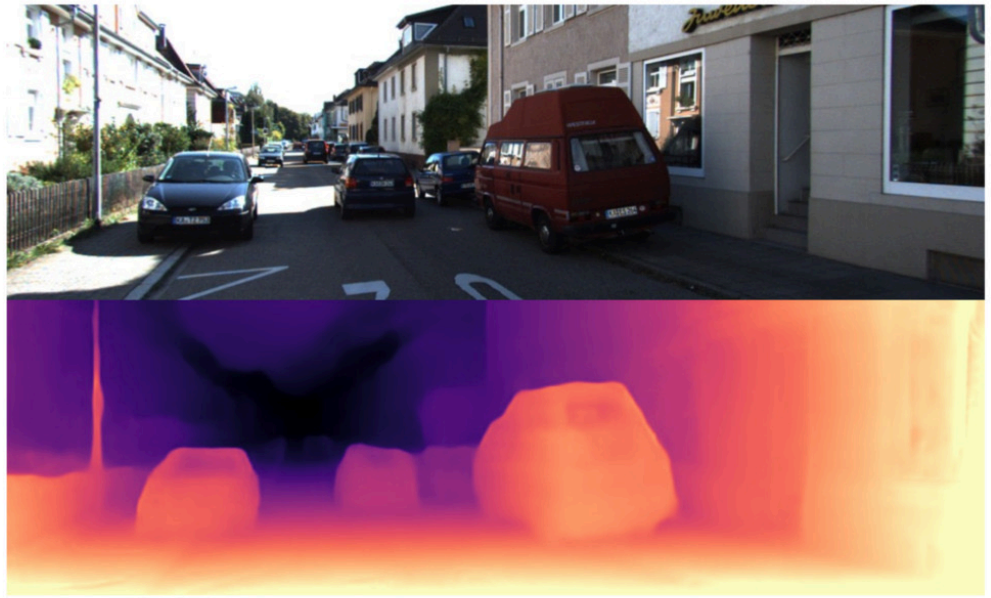


Figure 1. Summary of our approach. While standard image models jointly train an image feature extractor and a linear classifier to predict some label, CLIP jointly trains an image encoder and a text encoder to predict the correct pairings of a batch of (image, text) training examples. At test time the learned text encoder synthesizes a zero-shot linear classifier by embedding the names or descriptions of the target dataset's classes.

https://arxiv.org/abs/2103.00020

Recent Models

- Mono depth estimation



Recent Models

- SAM (segment anything)

FastSAM

https://segment-anything.com/
https://github.com/CASIA-IVA-Lab/FastSAM

Recent Models

- Grounding Dino

Key benchmark architectures

- ResNets, VGG, ViTs etc.
- YoloV5, V7, V8
- MaskedRCNNs
- Unets, Autoencoders etc.

Tutorials

- Yolov8
- Yolo tracking
- Clip
- FAST SAM
- Mono depth estimation
- Grounding Dino