Lecture 4: Convolutional Neural Networks

Wei Qi Yan

Auckland University of Technology

March 22, 2023

Table of Contents

CNNs or ConvNets

Deep Learning

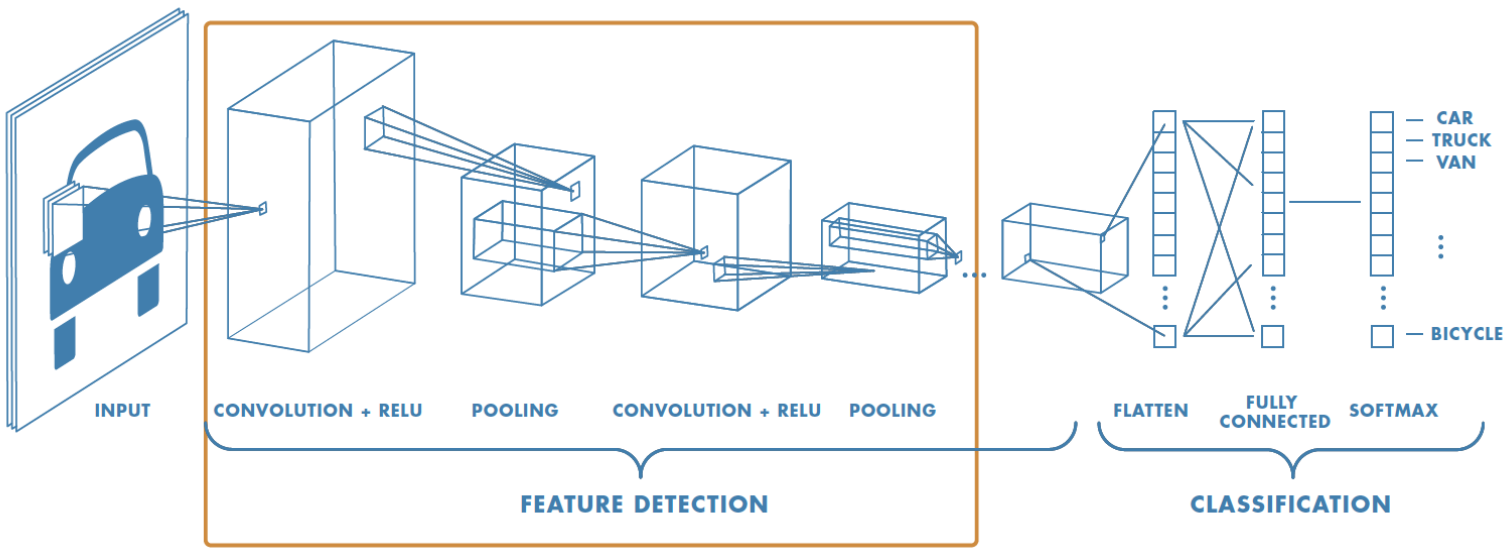
- Deep learning is a type of machine learning methods in which a model is trained to perform classification tasks.
- Deep learning is usually implemented using a neural network architecture.
- The term "deep" refers to the number of layers in the network.
- Conventional neural networks contain only two or three layers, while deep nets can have more.

The State-of-the-Art Technology

- Easy ways to access massive sets of labeled data
- Increased computing power (e.g., GPU, FPGA, etc.)
- Pretrained models created by experts

Web: https://au.mathworks.com/help/pdf_doc/deeplearning/nnet_ug.pdf

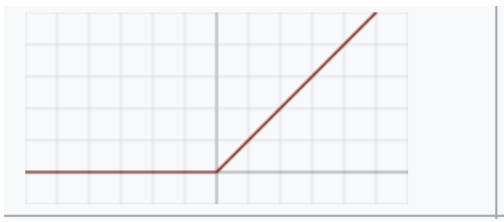
CNNs or ConvNets



Web: https://au.mathworks.com/help/pdf_doc/deeplearning/nnet_ug.pdf

Three Operations

- *Convolution* puts the input through a set of convolutional filters.
- *Pooling* simplifies the output through nonlinear downsampling, reducing the number of parameters that the network needs to be trained.
- *ReLU* (Rectified Linear Unit) allows for fast and effective training by mapping negative values to zero and maintaining positive ones.

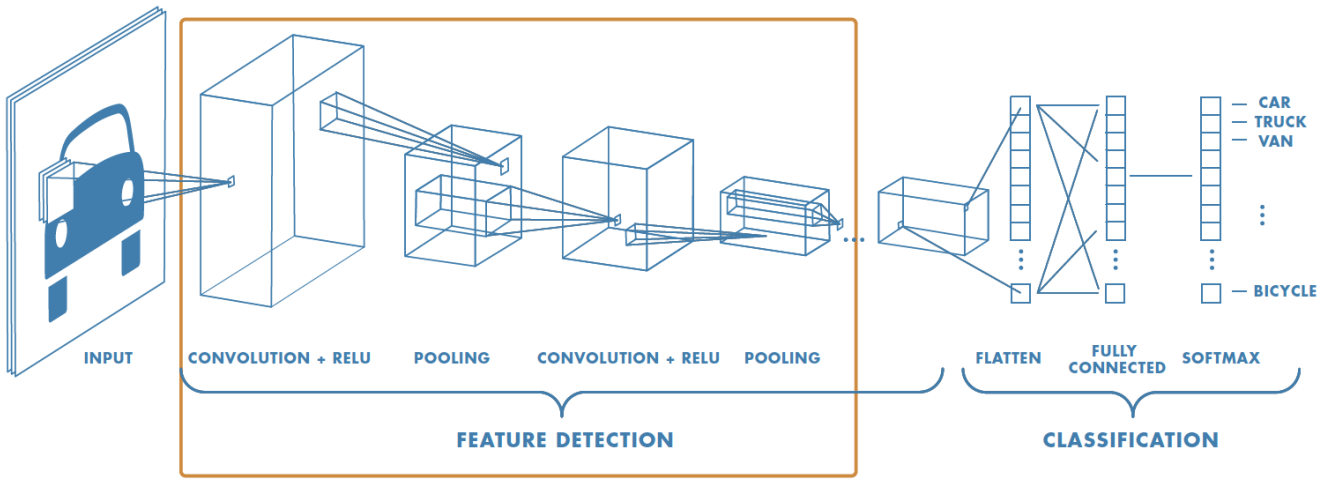


$$f(x) = \begin{cases} 0 & \text{for } x \leq 0 \\ x & \text{for } x > 0 \end{cases}$$

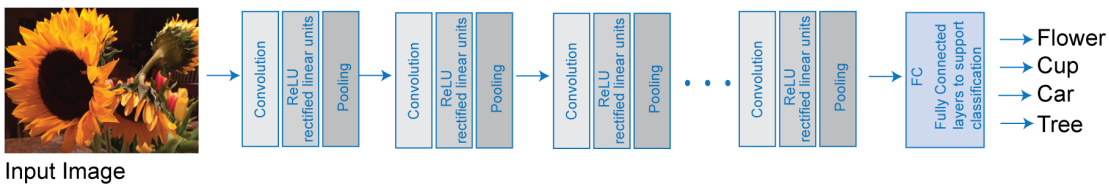
Web: https://en.wikipedia.org/wiki/Activation_function

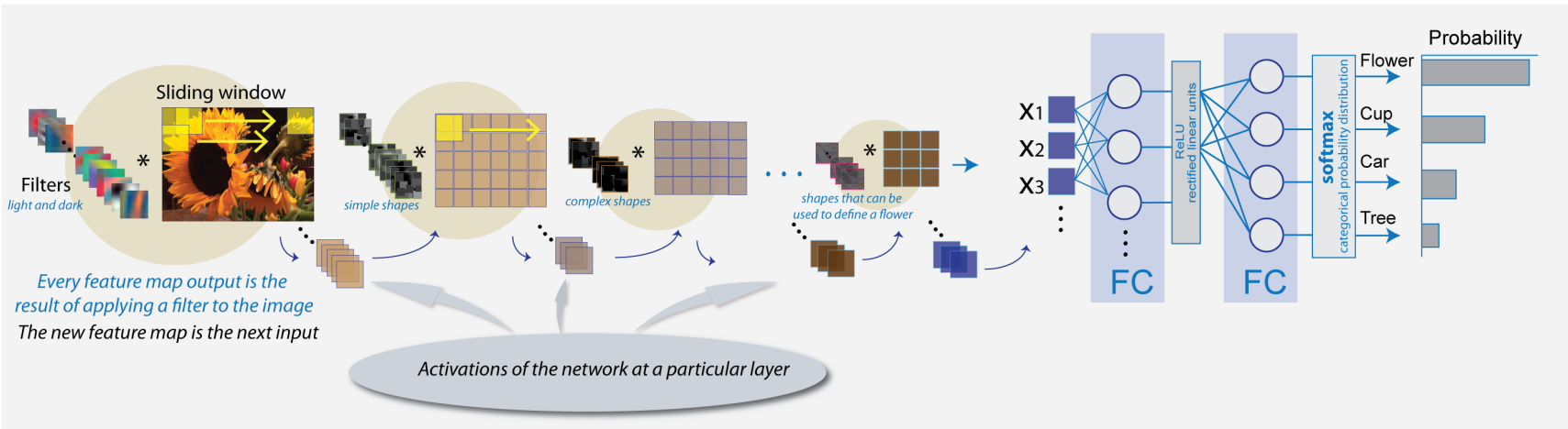
The Classification

- A fully connected layer (FC) outputs a vector of k dimensions where k is the number of classes that the net is able to predict.
- The vector contains the probabilities for each class of any images being classified.
- The final layer of the CNN architecture uses a softmax function to provide the classification output.



Convolutional Neural Networks





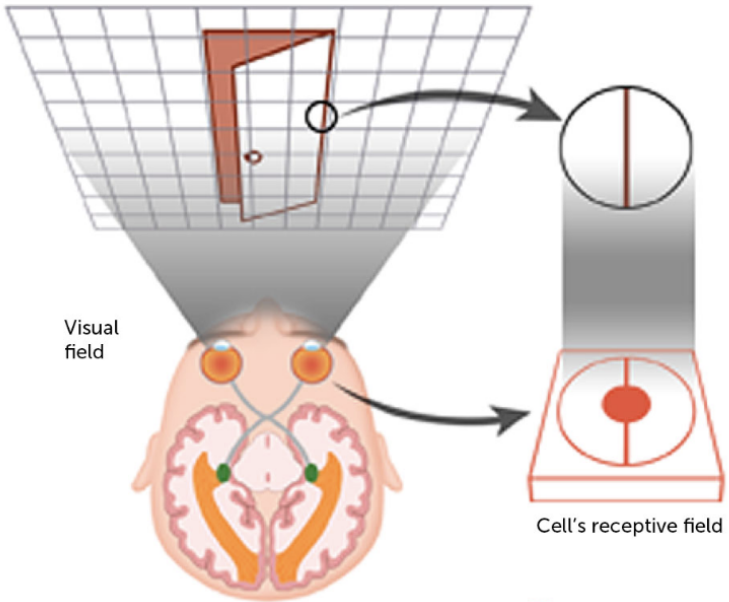
CNNs or ConvNets

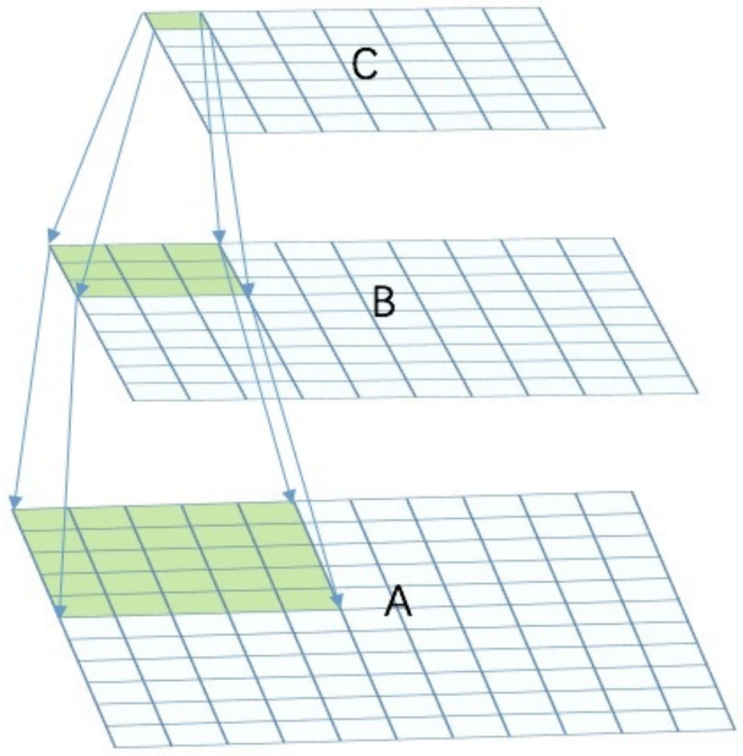
Convolutional Neural Networks

- ConvNets are inspired from the biological structure of a visual cortex, which contains arrangements of simple and complex cells.
- These cells are activated based on the subregions of a visual field, i.e., **receptive field**.
- A ConvNet reduces the number of parameters with the number of connections, shared weights, and downsampling.
- A ConvNet consists of multiple layers, such as convolutional layers, max pooling or average pooling layers, and fully connected layers.

Web: https://au.mathworks.com/help/pdf_doc/deeplearning/nnet_ug.pdf

Receptive Field





Web: https://brainconnection.brainhq.com/2004/03/06/overview-of-receptive-fields/

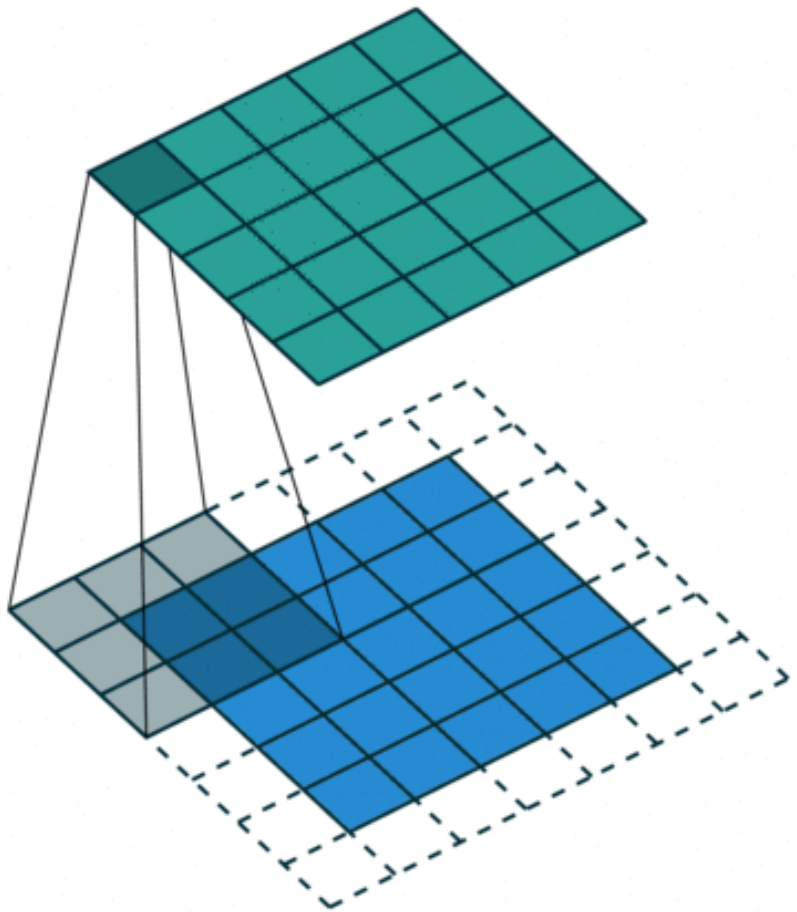
CNNs or ConvNets

Concepts

- **Receptive field**: A region of the original image corresponding to a pixel of the feature map of a filter or kernel
- **Feature map**: The output of convolution operations
- **Stride**: The step length of convolution operations
- **Fsize**: The size of convolution kernels or filters
- **Padding**: The filled region of an image boundary
- **Top to down**: From a deep layer to its next layer

Web: https://au.mathworks.com/help/pdf_doc/deeplearning/nnet_ug.pdf

Padding



Web: https://au.mathworks.com/help/pdf_doc/deeplearning/nnet_ug.pdf

Evaluations

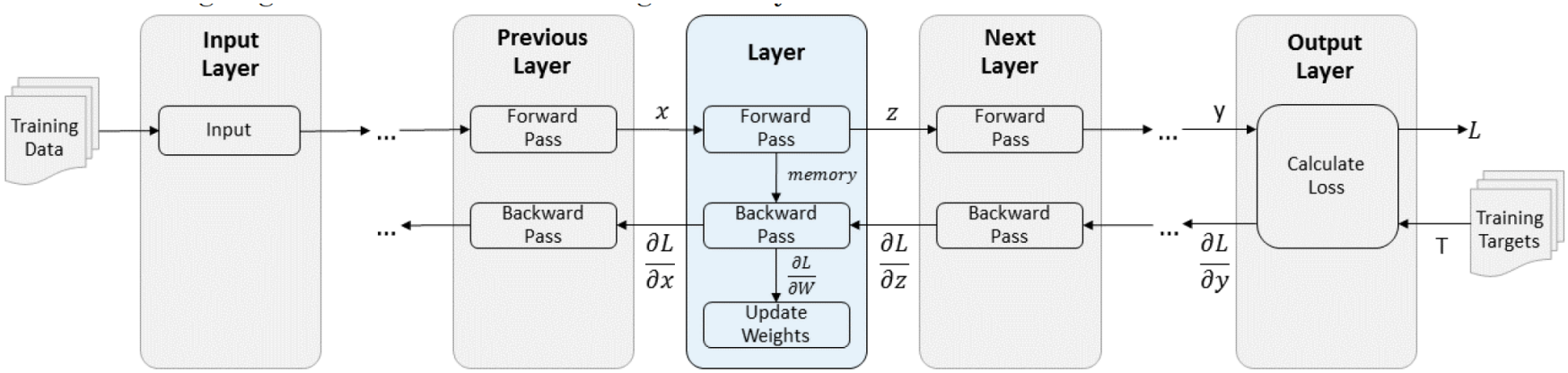
- A full pass through the whole dataset is called an *epoch*.
- Iteration in deep learning is the number of batches needed to complete one epoch.
- What a larger *learning rate* is gradually reduced during optimization enables smaller steps towards to optimum value.
- Performing *validation* at regular intervals during training can determine whether the network is *overfitting* over the training data.
- To check whether a network is overfitting, compare *training loss* and *accuracy* corresponding to validation metrics.

Forward Pass and Backward Pass

- A layer has two main components: The forward pass, the backward pass.
- During the forward pass, the layer takes the output of the previous layer, applies functions, and outputs (forward propagates) the result.
- At the end of a forward pass, the network calculates the loss between the predictions and the true target.
- During the backward pass of a network, each layer takes the derivatives of the loss, computes the derivatives of the loss, and then outputs (backpropagation) results to the previous layer.

Web: https://au.mathworks.com/help/pdf_doc/deeplearning/nnet_ug.pdf

Forward Pass and Backward Pass



Web: https://au.mathworks.com/help/pdf_doc/deeplearning/nnet_ug.pdf

Questions?

R-CNN

R-CNN (Region-based CNN) is a two-stage detection algorithm:

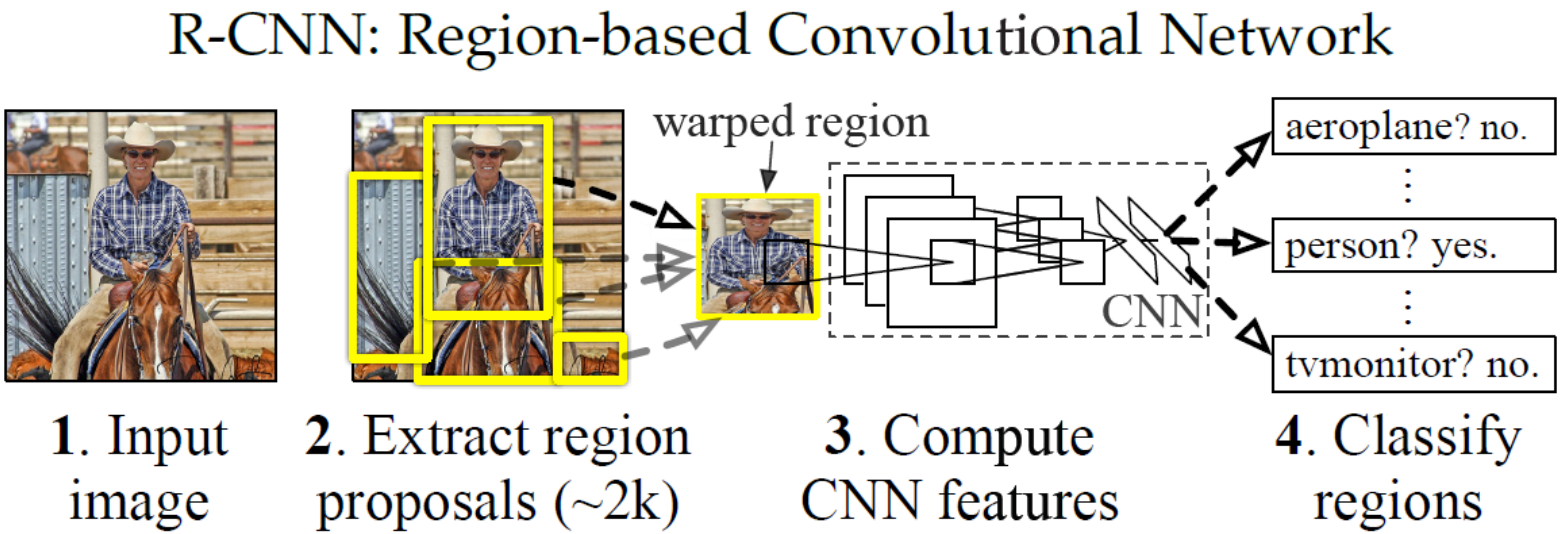
- The first stage identifies a subset of regions in an image that might contain an object.
- The second stage classifies the object in each region.

Visual object detection using R-CNN is based on three processes:

- Find regions in the image that might contain an object. These regions are called region proposals.
- Extract CNN features from the region proposals.
- Classify the objects using the extracted features.

Web:https://au.mathworks.com/help/vision/ug/getting-started-with-r-cnn-fast-r-cnn-and-faster-r-cnn.html

R-CNN



G. Ross (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. IEEE CVPR, pp580–587.

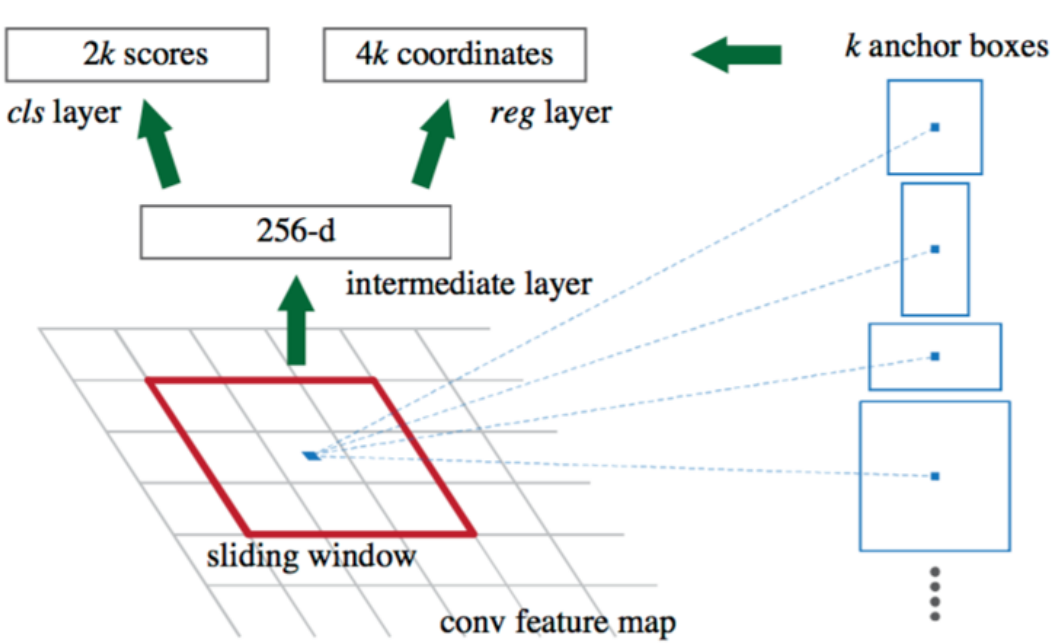
R-CNN

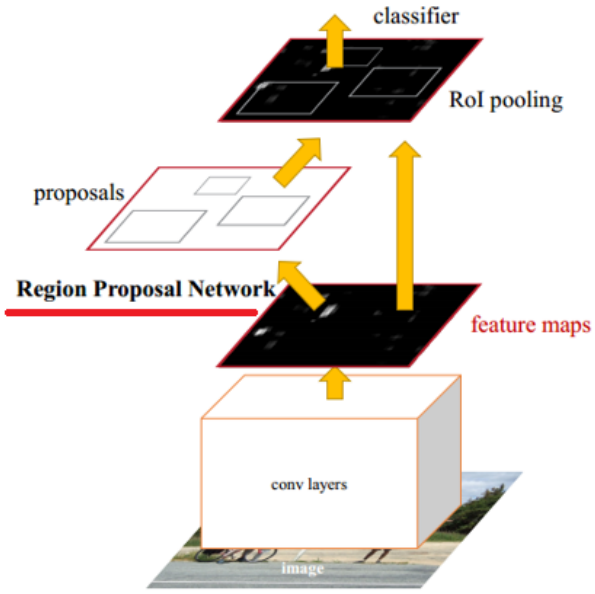
R-CNN combines rectangular region proposals with ConvNets features:

- R-CNN detector firstly generates region proposals using an algorithm such as Edge Boxes.
- The proposal regions are cropped out of the image and resized.
- CNN classifies the cropped and resized regions.
- The bounding boxes of region proposals are refined by using support vector machine (SVM) that is trained by using CNN features.

G. Ross (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. IEEE CVPR, pp580–587.

Fast R-CNN





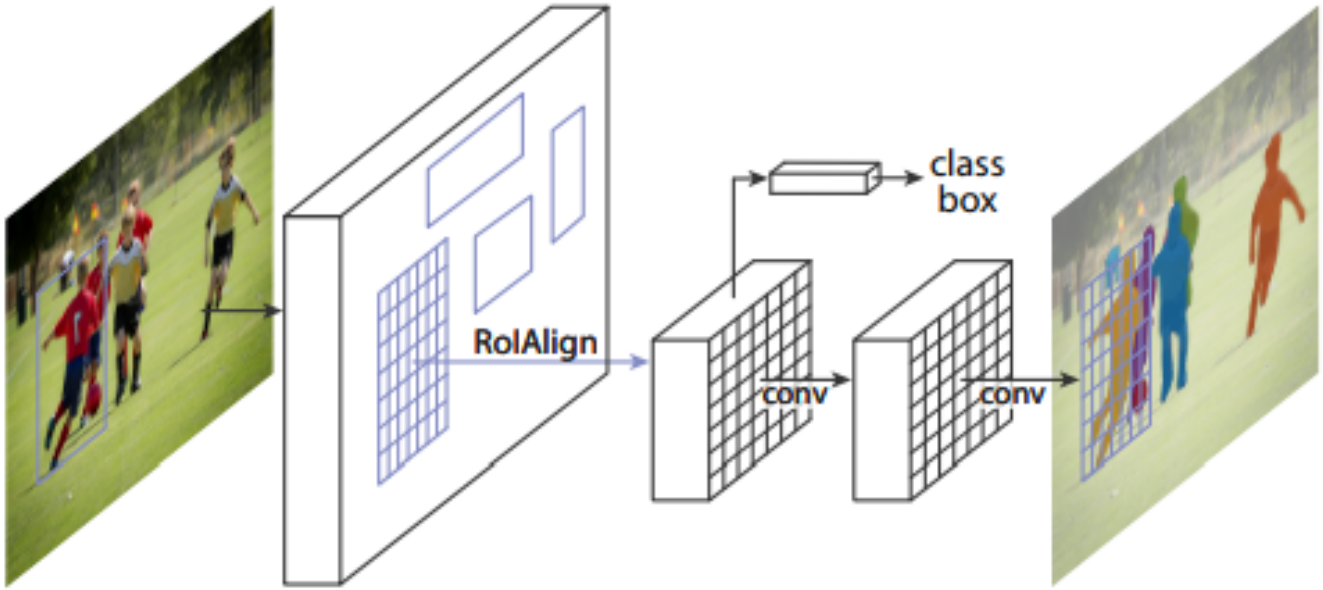
R. Girschick (2015). Fast R-CNN. IEEE ICCV, pp.1440–1448.

Fast R-CNN and Faster R-CNN

- Faster R-CNN detector adds a region proposal network (**RPN**) to generate region proposals directly in the net instead of using an external algorithm like Edge Boxes.
- RPN uses *anchor boxes* for visual object detection. Generating region proposals in the network is faster and better tuned to the data.
- Fast R-CNN and Faster R-CNN detectors were designed to improve detection performance with a large number of regions.

R. Girschick (2015). Fast R-CNN. IEEE ICCV, pp.1440–1448.

Mask R-CNN



The **Mask R-CNN** framework for instance segmentation.

K. He, G. Gkioxari, P. Dollar, R. Girshick (2017) Mask R-CNN, ICCV, pp. 2980 - 2988.
(**ICCV 2017 Best Paper Award**)

Mask R-CNN

- Mask R-CNN efficiently detects visual objects in an image while simultaneously generating a high-quality segmentation mask for each instance.
- Mask R-CNN extends Faster R-CNN by adding a branch for predicting an object mask in parallel with the existing branch for bounding box recognition.
- Mask R-CNN is simple to train and add only a small overhead to Faster R-CNN.
- Mask R-CNN is easy to generalize to other tasks: Segmentation, bounding box, and keypoint of visual object.
- Mask R-CNN outperforms all existing, single-model entries on every task.

K. He, G. Gkioxari, P. Dollar, R. Girshick (2017) Mask R-CNN, ICCV, pp. 2980 - 2988.
(**ICCV 2017 Best Paper Award**)

Questions?

SSD and YOLO

SSD: Single Shot MultiBox Detector

- SSD is similar to the Faster R-CNN and simultaneously produces a score for each object in each box.
- SSD skips the proposal step and predicts bounding boxes & confidences scores for multiple classes.
- SSD uses default bounding boxes of **different aspect ratios** on each location from multiple feature maps.

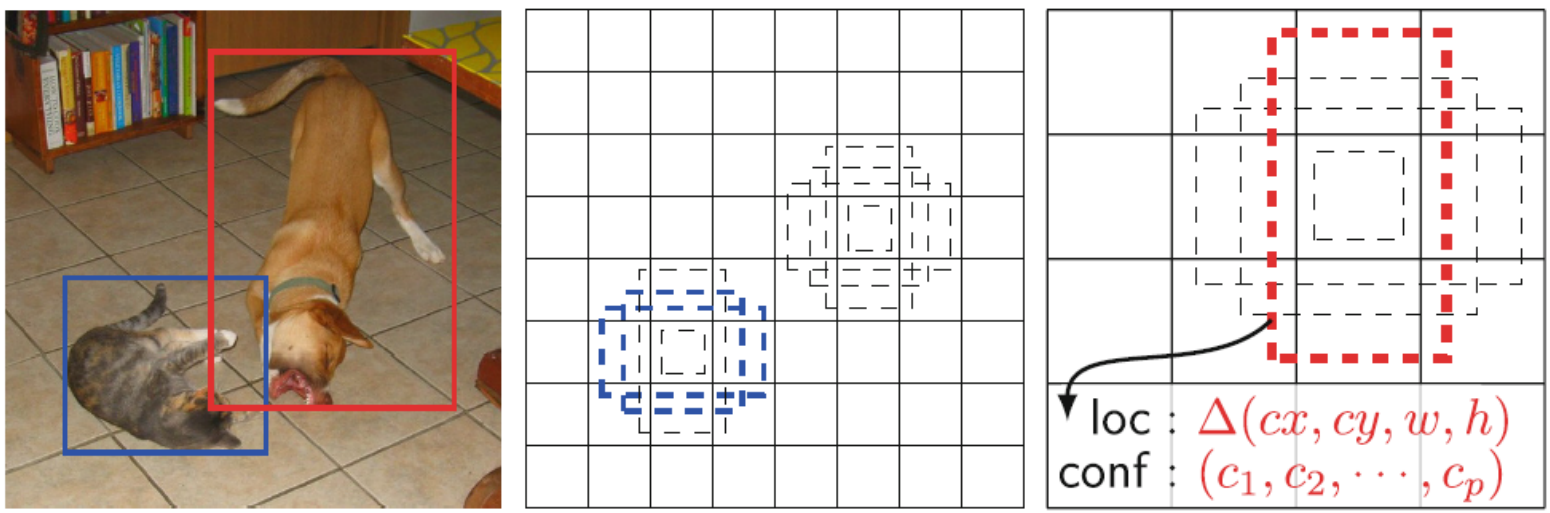
SSD: Concepts

- Single shot: The tasks of object detection and classification are conducted in a single forward pass of the network.
- Multibox: Designed for bounding box regression.
- Detector: A neural network for classifying the target visual objects.

W. Liu, et al. (2016) SSD: Single Shot MultiBox Detector. ECCV, pp.21-27.

SSD: Single Shot MultiBox Detector

SSD framework.



(a) Image with GT boxes (b) 8×8 feature map (c) 4×4 feature map

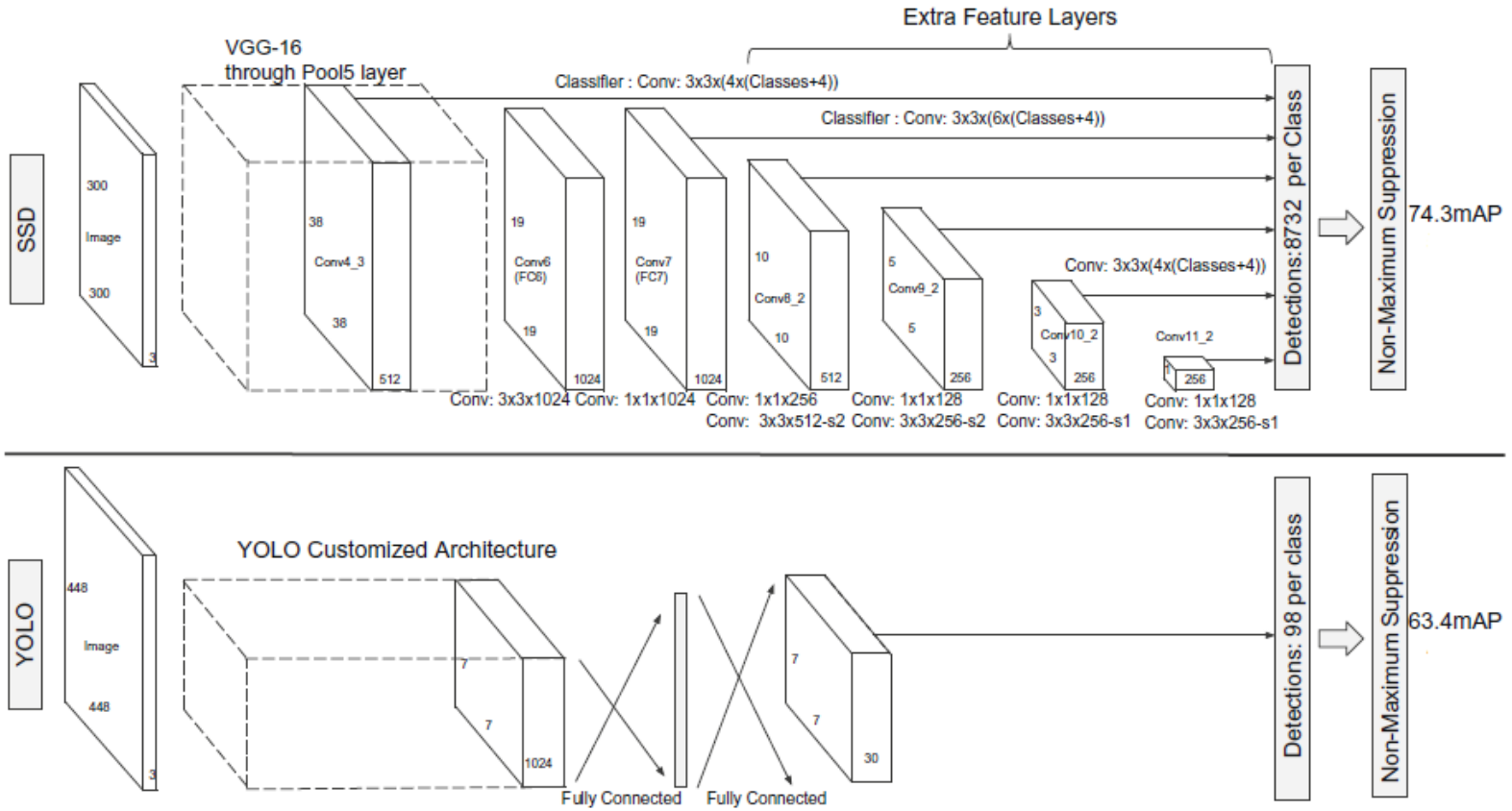
*GT: Ground Truth.

W. Liu, et al. (2016) SSD: Single Shot MultiBox Detector. ECCV, pp.21-27.

SSD: Single Shot MultiBox Detector

W. Liu, et al. (2016) SSD: Single Shot MultiBox Detector. ECCV, pp.21-27.

SSD: Single Shot MultiBox Detector



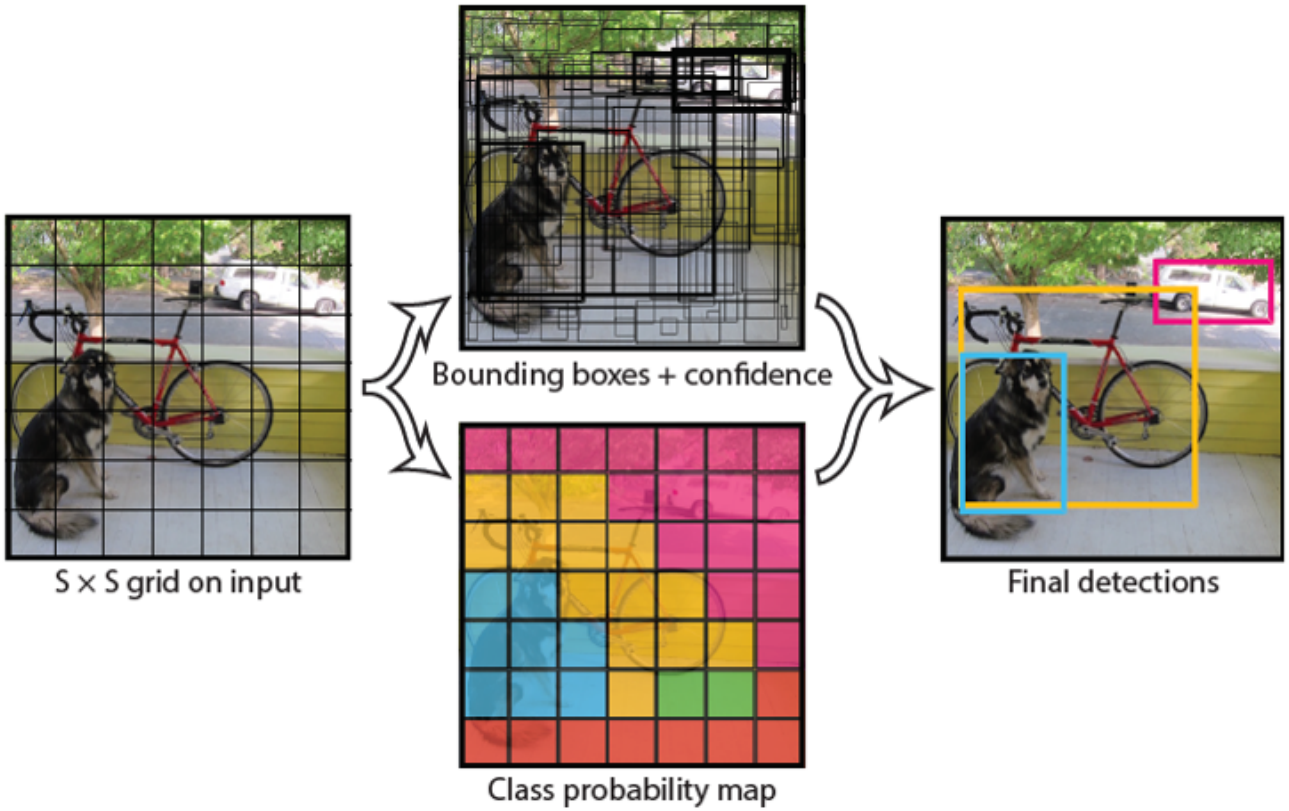
J. Redmon, et al. (2016) You Only Look Once: Unified, Real-Time Object Detection. CVPR.

YOLO: You Only Look Once

- YOLO: A single neural network predicts bounding boxes and class probabilities directly from full images.
- YOLO is trained based on full images that directly optimizes detection performance.
- YOLO achieves more than twice of the mean average precision of other real-time methods.
- YOLO sees the entire image during training and testing time so that it encodes **contextual information** of all classes.

J. Redmon, et al. (2016) You Only Look Once: Unified, Real-Time Object Detection. CVPR.

YOLO: You Only Look Once



J. Redmon, et al. (2016) You Only Look Once: Unified, Real-Time Object Detection. CVPR.

YOLO: You Only Look Once

- YOLO **predicts** what objects present and where they are.
- A single convolutional network simultaneously **predicts** multiple bounding boxes and class probabilities for those boxes.
- YOLO is trained based on full images that directly optimizes detection performance.
- YOLO runs a neural network on a new image at testing time to predict detections.
- YOLO is highly generalizable and is less likely to break down when applied to new domains or unexpected inputs.

J. Redmon, et al. (2016) You Only Look Once: Unified, Real-Time Object Detection. CVPR.

YOLO9000: Better, Faster, Stronger

- YOLO9000 is faster than other detection systems before.
- YOLO9000 is a real-time framework for detecting more than 9000 object classes by jointly optimizing the detection and classification.
- YOLO9000 improves recall and object locating while maintaining classification accuracy.

J. Redmon, A. Farhadi, (2017) YOLO9000: Better, Faster, Stronger. CVPR.

SSD and YOLO

MATLAB YOLOv2

- YOLOv2 uses a single stage object detection network.
- YOLOv2 is faster than other two-stage deep learning object detectors, such as Faster R-CNN.
- YOLOv2 runs a CNN in deep learning based on an input image to produce network prediction.
- YOLOv2 uses *anchor boxes* to detect classes of objects in an image.

YOLOv2 Predictions

- Intersection over Union (IoU): Predict the objectiveness score of each anchor box.
- Anchor box offset: Refine the anchor box position.
- Class probability: Predict the class label assigned to each anchor box.

MATLAB YOLOv3

- YOLOv3 improves upon YOLOv2 by adding detection at multiple scales to detect smaller objects.
- The loss function of YOLOv3 for training is separated into mean squared error for bounding box regression and binary cross entropy for object classification to improve detection accuracy.
- YOLOv3 detector utilizes anchor boxes estimated through training data to have better initial priors and predict the boxes accurately.

Web:
https://au.mathworks.com/help/vision/ug/object-detection-using-yolo-v3-deep-learning.html

MATLAB YOLOv4

- YOLOv4 is a one-stage object detection network that is composed of three parts: Backbone, neck, and head.
- CSPDarkNet-53 is used as the backbone for extracting features from the input images.
- YOLOv4 takes use of anchor boxes to detect classes of objects in an image.
- Similar to YOLOv3, YOLOv4 predicts these three attributes for each anchor box: Intersection over union (IoU), anchor box offsets, class probability.

Web: https://au.mathworks.com/help/vision/ug/getting-started-with-yolo-v4.html

Questions?

Learning Objectives

- Design and analyse algorithms of deep neural networks.
- Demonstrate advanced understanding of the state-of-the-art in the practice of deep learning.