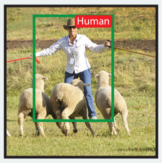
# Mordern History of Object Recognition Infographics

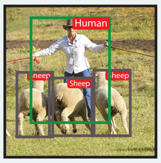
## MiniMap

2012 AlexNet RCNN OverFeat 2013 ZFNet SPPNets
YOLO Fast RCNN InceptionNet VGGNet 2014 MultiBox
2015 ResNet Faster RCNN 2016 SSD 2017 MaskRCNN

**Image Classification**
Classify an image based on the dominant object inside it.
**datasets:** MNIST, CIFAR, ImageNet




**Object Localization**
Predict the image region that contains the dominant object. Then image classification can be used to recognize object in the region
**datasets:** ImageNet




**Object Recognition**
Localize and classify all objects appearing in the image. This task typically includes: proposing regions then classify the object inside them.
**datasets:** PASCAL, COCO

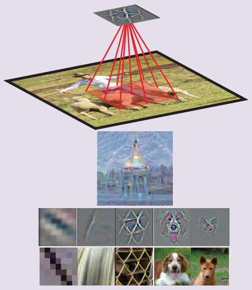
**Semantic Segmentation**
Label each pixel of an image by the object class that it belongs to, such as human, sheep, and grass in the example.
**datasets:** PASCAL, COCO

**Instance Segmentation**
Label each pixel of an image by the object class and object instance that it belongs to. Background pixels are usually ignored.
**datasets:** PASCAL, COCO

**Keypoint Detection**
Detect locations of a set of predefined keypoints of an object, such as keypoints in a human body, or a human face.
**datasets:** COCO

1958
1968
1970s
1989
1993
1995
2000s
2006 2010
2012

## Important CNN Concepts



### Feature[4,5,8] *(pattern, activation of a neuron, feature detector)*

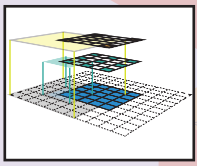
A hidden neuron that is activated when a particular pattern (feature) is presented in its input region (receptive field).

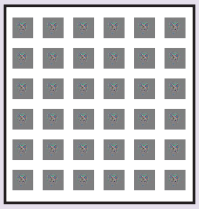
The pattern that a neuron is detecting can be visualized by (1) optimizing its input region to maximize the neuron's activation (deep dream), (2) visualizing the gradient or guided gradient of the neuron activation on its input pixels (back propagation and guided back propagation), (3) visualizing a set of image regions in the training dataset that activate the neuron the most.

### Receptive Field[2] *(input region of a feature)*

The region of the input image that affects the activation of a feature. In other words, it is the region that the feature is looking at.

Generally, a feature in a higher layer has a bigger receptive field, which allows it to learn to capture a more complex/abstract pattern. The ConvNet architecture determines how the receptive field change layer by layer.



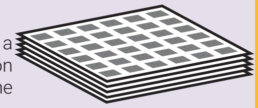


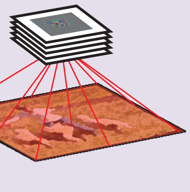
### Feature Map[1] *(a channel of a hidden layer)*

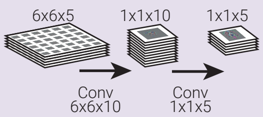
A set of features that created by applying the same feature detector at different locations of an input map in a sliding window fashion (i.e. convolution). Features in the same feature map have the same receptive size and look for the same pattern but at different locations. This creates the spatial invariance properties of a ConvNet.

### Feature Volume[3] *(a hidden layer in a ConvNet)*

A set of feature maps, each map searches for a particular feature at a fixed set of locations on the input map. All features have the same receptive field size.





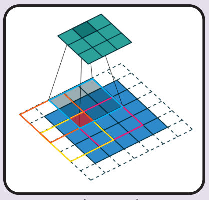



### Fully connected layer as Feature Volume[2]

Fully connected layers (usually attached to the end of a ConvNet for classification) with k hidden nodes can be seen as a *1x1xk* feature volume. This feature volume has one feature in each feature map, and its receptive field covers the whole image. The weight matrix W in an FC layer can be converted to a CNN filter. Convolving the filter kernel *wxhxk* to a CNN feature volume wxhxd creates a *1x1xk* feature volume (k-node FC layer). Convolving a *1x1xk* filter kernel to a *1x1xd* feature volume creates a *1x1xk* feature volume. Replacing fully connected layers by convolution layers allows us to apply a ConvNet to an image with arbitrary size.

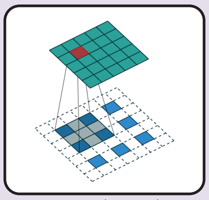
### Transposed Convolution[1] *(fractional strided convolution, deconvolution, upsampling)*

The operation that back-propagates the gradient of a convolution operation. In other words, it is the backward pass of a convolution layer. A transposed convolution can be implemented as a normal convolution with zero inserted between the input features. A convolution with filter size k, stride s and zero padding p has an associated transposed convolution with filter size k'=k, stride s'=1, zero padding p'=k-p-1, and s-1 zeros inserted between each input unit.



Normal Convolution
k = 3, s=2, p = 1

On the left, the red input unit contributes to the activation of the 4 top left output units (through the 4 colored squares), therefore it receives gradient from these output units. This gradient backpropagation can be implemented by the transposed convolution shown on the right.



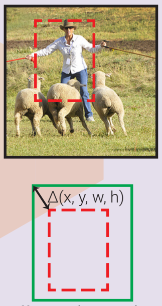
Transposed Convolution
k' = 3, s'=1, p = 1, insert=1

### End-To-End object recognition pipeline *(end-to-end learning/system)*

A object recognition pipeline that all stages (pre-processing, region proposal generation, proposal classification, post-processing) can be trained altogether by optimizing a *single objective function*, which is a differentiable function of all stages' variables. This is the opposite of the traditional object recognition pipeline, which connects stages in a non-differentiable fashion. In these systems, we do not know how changing a stage's variable can affect the overall performance, so that each stage must be trained (or heuristically programmed) independently from each other.

J. Schmidhuber
Yoshua Bengio
Yann Lecun
Geoffrey Hinton
Alex Graves
Alex Krizhevsky
Ilya Sutskever
Andrej Karpathy
Christopher Olah
Ross Girshick
Matthew Zeiler
Rob Fergus
Kaiming He
Pierre Sermanet
Christian Szegedy
Joseph Redmon
Shaoqing Ren
Wei Liu
Karen Simonyan
Andrew Zisserman
Evan Shelhamer
Jonathan Long
Trevor Darrell
Springenberg
Mordvintsev
V. Dumoulin
Francesco Visin
Adit Deshpande
et al.

## Important Object Recognition Concepts




Offset =Δ(x, y, w, h)
Distance = ||Δ||

### Bounding box proposal *(region of interest, region proposal, box proposal)*

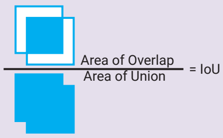
A rectangular region of the input image that potentially contains an object inside. These proposals can be generated by some heuristics search: objectness, selective search, or by a region proposal network

A bounding box can be represented as a 4-element vector, either storing its two corner coordinates (x0, y0, x1, y1), or (commonly) storing its center location and its width and height (x, y, w, h). A bounding box is also usually accompanied by a confidence score of how likely the box contains an object, which NMS requires.

The difference between two bounding boxes is usually measured by the L2 distance of their vector representations. w and h can be log-transformed before the distance calculation.

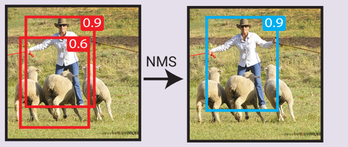
### Intersection over Union *(IoU, Jaccard similarity)*:

A metric that measures the similarity between two bounding boxes = their overlapping area over their union area.




### Non Maxium Suppression *(NMS)*

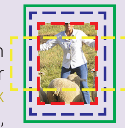
A common algorithm to merge overlapping bounding boxes (proposals or detections). Any bounding box that significantly overlap (IoU > IoU_threshold) with a higher-confident bounding box is suppressed (removed).




### Bounding box regression *(bounding box refinement )*


By looking at an input region, we can infer the bounding box that better fit the object inside, even if the object is only partly visible. The example on the right illustrates the possibility of inferring the ground truth box only by looking at part of an object. Therefore, one regressor can be trained to predict the offset Δ(x, y, w, h) between the input bounding box (input region) and the ground truth box. if we have one regressor for each object class, it is called class-specific regression, otherwise (one regressor for all) it is class-agnostic. A bounding box regressor is often accompanied by a ***bounding box classifier*** to estimate the confidence of object existence in the predicted box. The classifier can also be class-specific or class-agnostic. Without prior box, the input region here plays the role of a prior box.

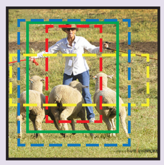
### Prior box *(default box, anchor box)*



Instead of using the input region as the only prior box, we can train multiple bounding box regressors, each has a different prior box and learns to predict the offset between its own prior box and the ground truth box. By looking at the same input region, regressors with different prior boxes can learn to predict bounding boxes with different properties (aspect ratio, scale, locations). Prior boxes can be predefined relatively to the input region, or learned by clustering. An appropriate box matching strategy is crucial to make the training converge.

### Box Matching Strategy

We cannot expect a bounding box regressor to be able to predict a bounding box of an object that is too far away from its input region (input matching) or its prior box (prior matching, more popular). Therefore, we need a box matching strategy to decide which prior box is matched with a ground truth box. Each match is a training example to regress. Possible strategies: (Multibox) matching each ground truth box with one prior box with highest IoU; (SSD, FasterRCNN) matching a prior box with any ground truth with IoU higher than 0.5.



One region proposal with 3 prior boxes and one ground truth box

The 3 bounding box regressors only see the input region and try to infer the ground truth box from their prior boxes

In Multibox strategy, the ground truth box is matched with the prior box with highest IoU

### Hard negative example mining

For each prior box, there is a bounding box classifier that estimates the likelihood of having an object inside. After box matching, all matched prior boxes are positive examples for the classifier. All other prior boxes are negatives. We cannot use all the hard negative examples, since there is a significant imbalance between the positives and negatives. Possible solutions: pick randomly negative examples (FasterRCNN), or pick the ones that the classifier makes the most serious error (SSD), so that the ratio between the negatives and positives is at roughly 3:1.

## Region Proposals or Sliding Windows

RCNN and OverFeat represent two competing ways to do object recognition: either classify regions proposed by another method (RCNN, FastRCNN, SPPNet), or classify a fixed set of evenly spaced square windows (OverFeat). The first approach has region proposals that fit the objects better than the second grid-like candidate windows, but is two orders of magnitude slower. The second approach takes advantage of the convolution operation to quickly regress and classify objects in sliding-windows fashion.

This competition was ended when Multibox introduced the ideas of prior box and region proposal network. Since then, all state-of-the-art methods now has a set of prior boxes (generated based on a set of sliding windows or by clustering ground-truth boxes) from which bounding box regressors are trained to propose regions that better fit the object inside. Object recognition then can be done in direct classification (YOLO, SSD), or refined classification way (FasterRCNN, MaskRCNN).

## 2012

## Everything is started here!

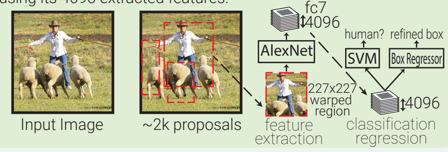
The modern history of object recognition goes along with the developement of ConvNets, which was all started here in 2012 when AlexNet won the ILSVRC 2012 by a large margin. Note that all the object recognition approaches are orthogonal to the specific ConvNet designs (any ConvNet can be combined with any object recognition approach). ConvNets are used as general image feature extractor.

### AlexNet

AlexNet bases on the decades-old LeNet, conbined with data augumentation, ReLU, dropout, and GPU implementation. It proved the effectiveness of ConvNet, kicked off the glorious combeback, and opened a new era for computer vision.

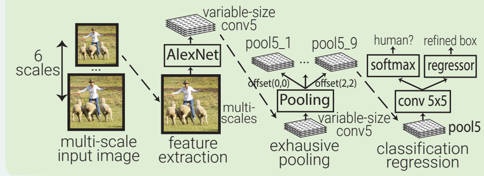
### RCNN

Region-based ConvNet (RCNN) is a natural combination of heuristic region proposal method and ConvNet feature extractor. From an input image, ~2000 bounding box proposals are generated using selective search. Those proposed regions are cropped and warped to a fixed-size 227x227 image. AlexNet is then used to extract 4096 features (fc7) for each warped image. A SVM model is then trained to classify object in the warped image using its 4096 features. Multiple class-specific bounding box regressors are trained to refine the bounding box proposal using its 4096 extracted features.




### OverFeat

OverFeat uses AlexNet to extract features at multiple evenly-spaced square windows in the image (receptive fields of features in AlexNet Pool5), and over multiple scales (by scaling input image). An object classifier and a class-agnotic box regressor are then trained to classify object and refine bounding box for every 5x5 region in the Pool5 layer (339x339 window). OverFeat replaces fc layers by 1x1xn conv layers to predict for multi-scale images. Because receptive field moves 36 pixels when moving one pixel in the Pool5, the windows are usually not well aligned with the objects. OverFeat introduces exhausive pooling scheme: Pool5 is applied at every offset of its input, which results in 9 Pool5 volumes. The windows are now only spaced 12pixels instead of 36pixels

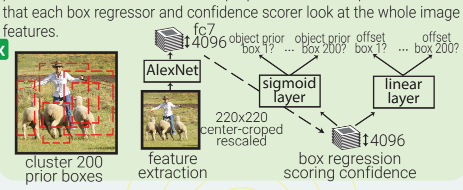



### ZFNet 2013

ZFNet is the ILSVRC 2013 winner, which is basically AlexNet with minor modification: use 7x7 kernel instead of 11x11 kernel in the first Conv layer to retain more information.

### MultiBox

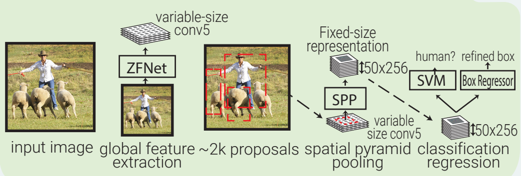
MultiBox is not an object recognition but a ConvNet-based region proposal solution. It popularized the ideas of region proposal network (RPN) and prior box, and proved that ConvNet can be trained to propose better region proposals than heuristic approaches. Since then, heuristic approaches have been fading out (also partly due to its untrainable nature). MultiBox first clusters all ground truth box locations in the whole dataset to find 200 centroids that it uses as prior boxes' centers. Each input image is center croped and rescaled to 220x220. Then it uses AlexNet to extract 4096 features (fc7). A 200-sigmoid layer is added to predict the object confidence score, and 4x200-linear layer is added to predict center offset and scale of proposal from each prior box. Note that each box regressor and confidence score look at the whole image features.




### SPPNet

SPPNet (Spatial Pyramid Pooling net) is essentially an enhanced version of RCNN by introducing two important concepts: adaptively-sized pooling (the SPP layer), and computing feature volume only once. In fact, the Fast-RCNN embraced these ideas to fasten RCNN with minor modifications.

SPPNet uses selective search to propose 2000 region proposals per image. It then extracts a common global feature volume from the entire image using ZFNet-Conv5. For each region proposal, SPPNet use spatial pyramid pooling (SPP) to pool features in that region from the global feature volume to generate its fixed-length representation. This representation is used for training the object classifier and box regressors. Pooling features from a common global feature volume rather than pushing all image crops through a full CNN like RCNN brings two orders of magnitude speed up. Note that although SPP operation is differentiable, the authors did not do that, so the ZFNet was only trained on ImageNet.




### VGGNet

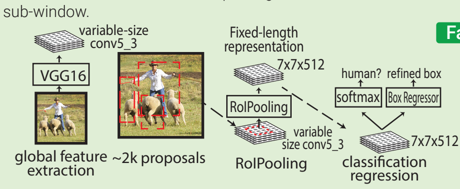
Although not an ILSVRC winner, VGG is still one of the most common ConvNet architecture today thanks to its simplicity and effectiveness. The main idea is to replace large-kernel conv by stacking several small-kernel conv. It strictly uses 3x3 conv with stride and padding of 1, along with 2x2 maxpooling layers with stride 2.

### 2014 Inception

Inception (GoogLeNet) is the winner of ILSVRC 2014. Instead of traditionally stacking up conv and maxpooling layer sequencially, it stacks up Inception modules, which consists of multiple parallel conv and maxpooling layers with different kernel sizes. It uses 1x1 conv layer (network in network idea) to reduce the depth of feature volume output. There are many InceptionNet versions.

### FastRCNN

Fast RCNN is essentially SPPNet with trainable feature extraction network and RoIPooling in replacement of the SPP layer. RoIPooling (region of interest pooling) is simply a special case of SPP where here only one pyramid level is used. RoIPooling generates a fixed 7x7 feature volume for each RoI (region proposal) by dividing the RoI feature volume into a 7x7 grid of sub-windows and then max-pooling the values from each sub-window.

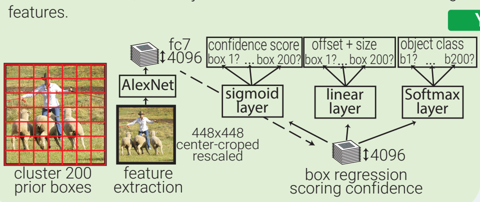


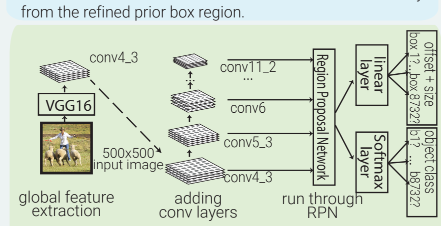

## Direct Classification or Refined Classification

Direct classification and refined classification are the two competing approaches for now. Direct classification classifies object inside the prior box directly, while the refined classification approach has an extract feature resampling stage where it pools the features of the refined prior box from a common feature volume and classify object by these features. The former is faster but less accurate since the features it uses are not extracted exactly from the refined prior box region.

### YOLO

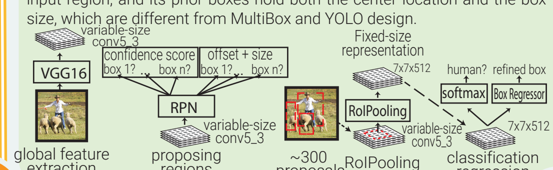
YOLO (You Only Look Once) is a direct development of MultiBox. It turns MultiBox from a region proposal solution to an object recognition method by adding a softmax layer to predicts the class of object inside the prior box. In addition, instead of clustering ground truth box locations to get the prior boxes, YOLO divides the input image into a 7x7 grid where each grid cell is a prior box. Grid cell is also used for box matching: if the center of an object falls into a grid cell, that grid cell is responsible for detecting that object. Like MultiBox, prior box only holds the center location information, not the size, so that box regressor predict the box size independent with the size of the prior box. Like MultiBox, all the box regressor, confidence scorer, and object classifier look at the whole image features.




### FasterRCNN

Faster RCNN is Fast RCNN with heuristic region proposal is replaced by region proposal network (RPN) inspired by MultiBox. In Faster RCNN, RPN is a small ConvNet (3x3 conv -> 1x1 conv -> 1x1 conv) looking at the conv5_3 global feature volume in the sliding window fashion. Each sliding window has 9 prior boxes that relative to its receptive field (3 scales times 3 aspect ratios). RPN does bounding box regression and box confidence scoring for each prior box. The whole pipeline is trainable by combining the loss of box regression, box confidence scoring, and object classification into one common global objective function. Note that here, RPN only looks at a small input region, and its prior boxes hold both the center location and the box size, which are different from MultiBox and YOLO design.




### SSD

SSD leverages the Faster RCNN's RPN, using it to directly classify object inside each prior box instead of just scoring the object confidence (similar to YOLO). It improves the diversity of prior boxes' resolutions by running the RPN on multiple conv layers at different depth levels.

### Mask RCNN

Mask RCNN extends Faster RCNN by adding a branch for predicting an object mask (instance segmentation task) in parallel with the existing branch for bounding box regression. Since RoIPool is not designed for pixel-to-pixel alignment between network inputs and outputs, MaskRCNN replaces it with RoIAlign, which use bilinear interpolation to compute the exact values of the input features at four sampled locations in each ROI instead of maxpooling.

### ResNet 2015

ResNet won the ILSVRC 2015 competition with an unbelievable 3.6% error rate (human performance is 5-10%). Instead of transforming the input representation to output representation, ResNet sequentially stacks residual blocks, each computes the change (residual) it want to make to its input, and add that to its input to produce its output representation. This is slightly related to boosting.

Faster RCNN 2016 SSD 2017 Mask RCNN

## References


1 Dumoulin, Vincent, and Francesco Visin. *"A guide to convolution arithmetic for deep learning."* Conv

2 The-Hien Dang-Ha, *"A guide to receptive field arithmetic for CNN"* ReceptiveField

3 Karpathy, Andrej. "Cs231n: Convolutional neural networks for visual recognition." DetailSummary

4 Zeiler, Matthew D., and Rob Fergus. *"Visualizing and understanding convolutional networks."* BackProp

5 Mordvintsev, Alexander, Christopher Olah, and Mike Tyka. *"Inceptionism: Going deeper into neural networks."* DeepDream

6 Adit Deshpande, *"The 9 Deep Learning Papers You Need To Know About"* Summary

7 Shelhamer, Evan, Jonathan Long, and Trevor Darrell. *"Fully convolutional networks for semantic segmentation."* Deconv

8 Springenberg, Jost Tobias, et al. *"Striving for simplicity: The all convolutional net."* GuidedBackProp

9 Dhruv Parthasarathy *"A Brief History of CNNs in Image Segmentation: From R-CNN to Mask R-CNN"* Summary

1 Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." AlexNet

2 Zeiler, Matthew D., and Rob Fergus. *"Visualizing and understanding convolutional networks."* ZFNet

3 Simonyan, Karen, and Andrew Zisserman. *"Very deep convolutional networks for large-scale image recognition."* VGGNet

4 Szegedy, Christian, et al. *"Going deeper with convolutions."* Inception

5 He, Kaiming, et al. *"Deep residual learning for image recognition."* ResNet

1 Girshick, Ross, *et al. "Rich feature hierarchies for accurate object detection and semantic segmentation."* RCNN

2 Sermanet, Pierre, et al. *"Overfeat: Integrated recognition, localization and detection using convolutional networks."* OverFeat

3 He, Kaiming, et al. *"Spatial pyramid pooling in deep convolutional networks for visual recognition."* SPPNet

4 Szegedy, Christian, et al. *"Scalable, high-quality object detection."* MultiBox

5 Girshick, Ross. *"Fast r-cnn."* FastRCNN

6 Redmon, Joseph, et al. *"You only look once: Unified, real-time object detection."* YOLO

7 Ren, Shaoqing, et al. *"Faster r-cnn: Towards real-time object detection with region proposal networks."* FasterRCNN

8 Liu, Wei, et al. *"SSD: Single shot multibox detector."* SSD

9 He, Kaiming, et al. *"Mask R-CNN."* MaskRCNN



Nikasa
https://nikasa1889.github.io/
https://medium.com/@nikasa1889/