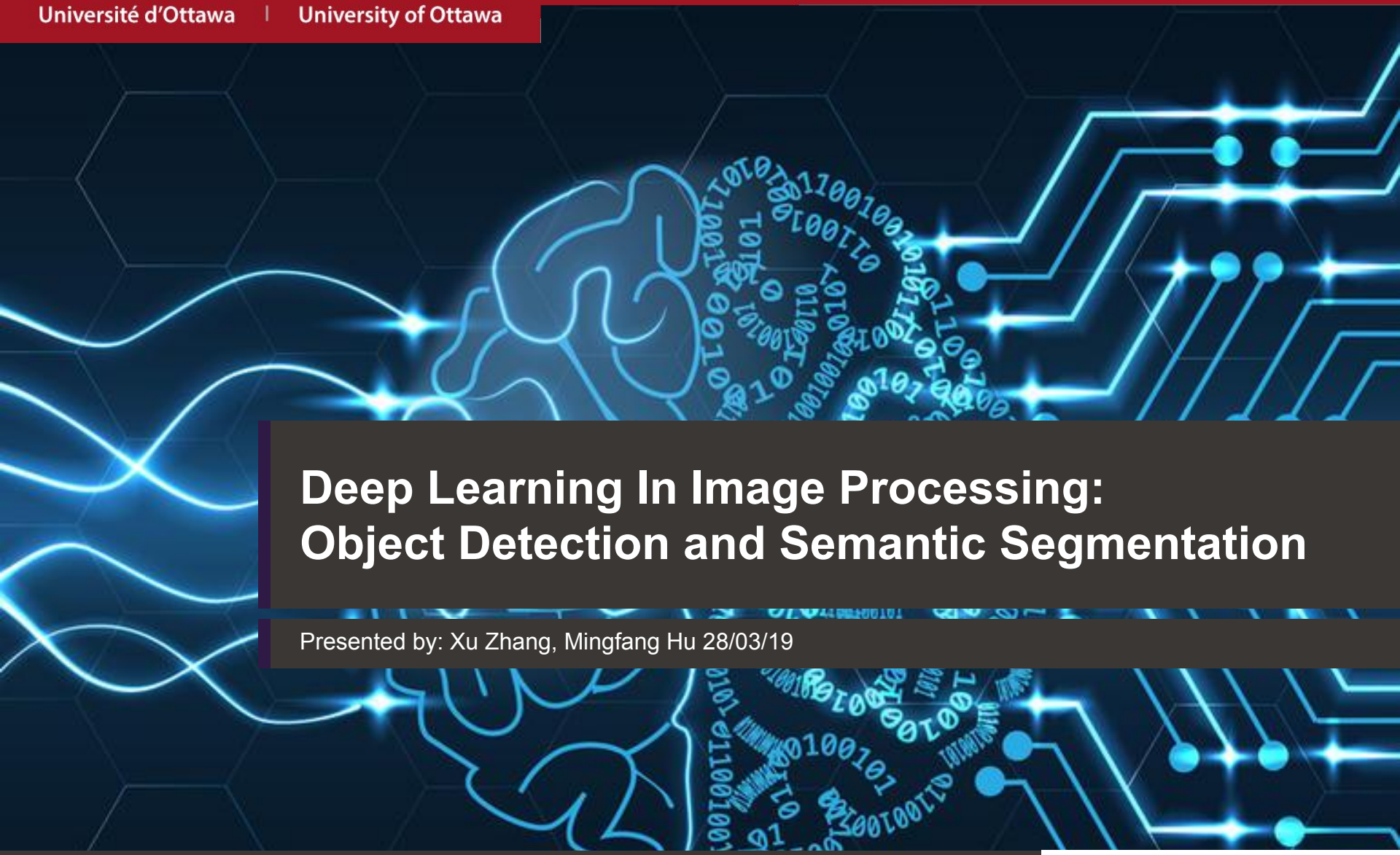


Deep Learning In Image Processing: Object Detection and Semantic Segmentation

Presented by: Xu Zhang, Mingfang Hu 28/03/19

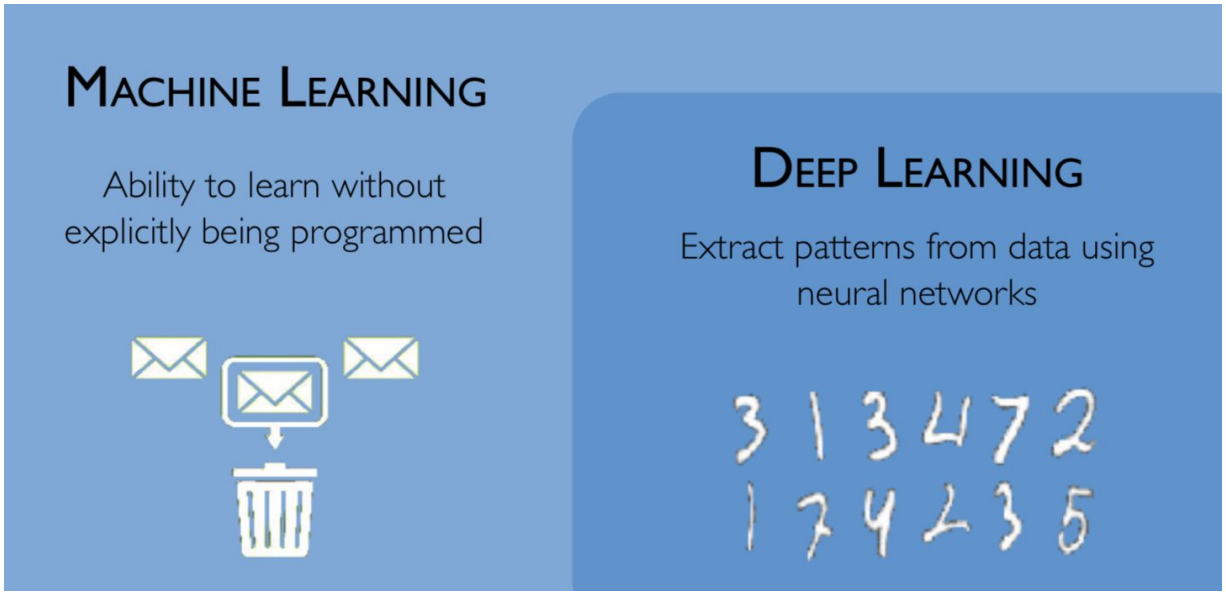
Outline

- Backgrounds
- Object Detection
- Semantic Segmentation

Backgrounds

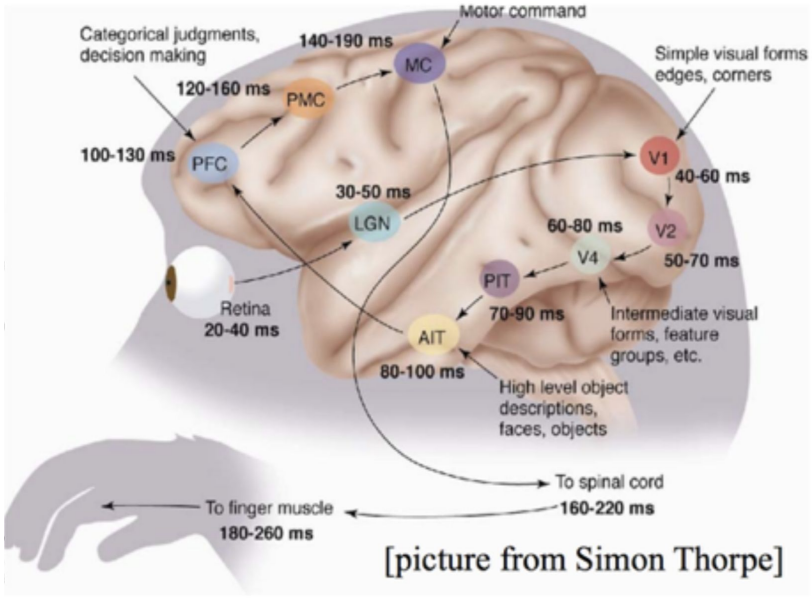
What Is Deep Learning?

- Subfield of the machine learning based on learning data representations.



What Is Deep Learning?

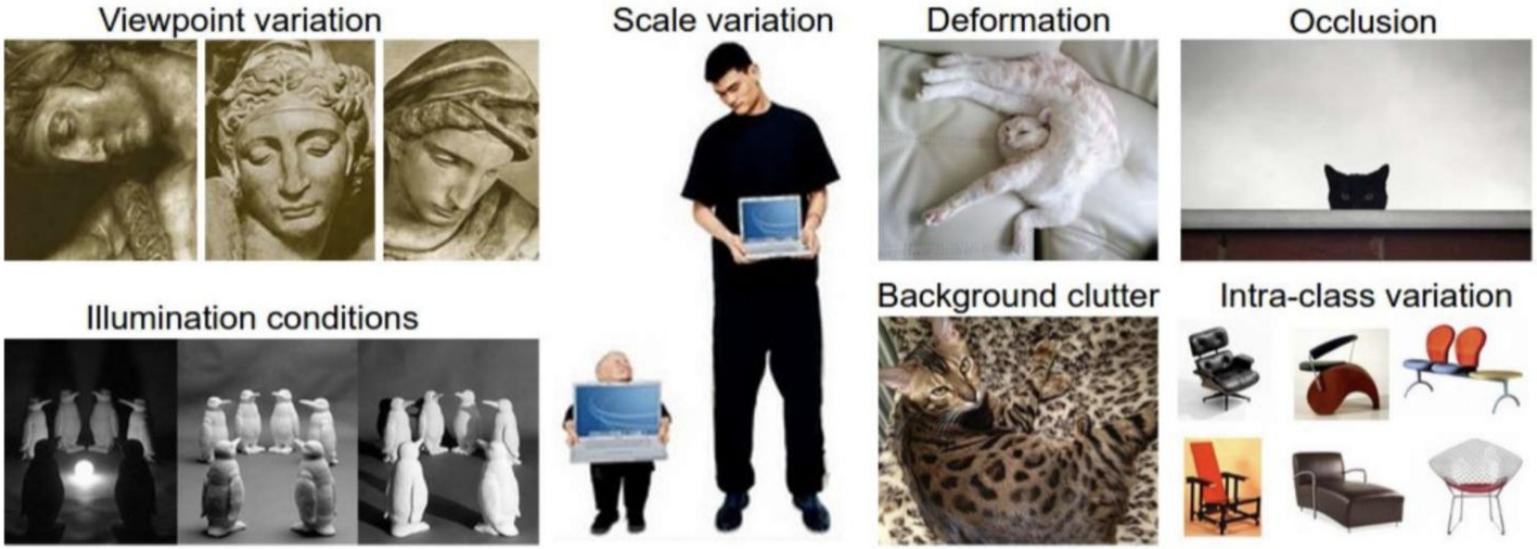
- Motivated by using a hierarchy of multiple layers that mimic the neural networks of our brain.



The first hierarchy of neurons that receives information in the visual cortex are sensitive to specific edges while brain regions further down the visual pipeline are sensitive to more complex structures such as faces.

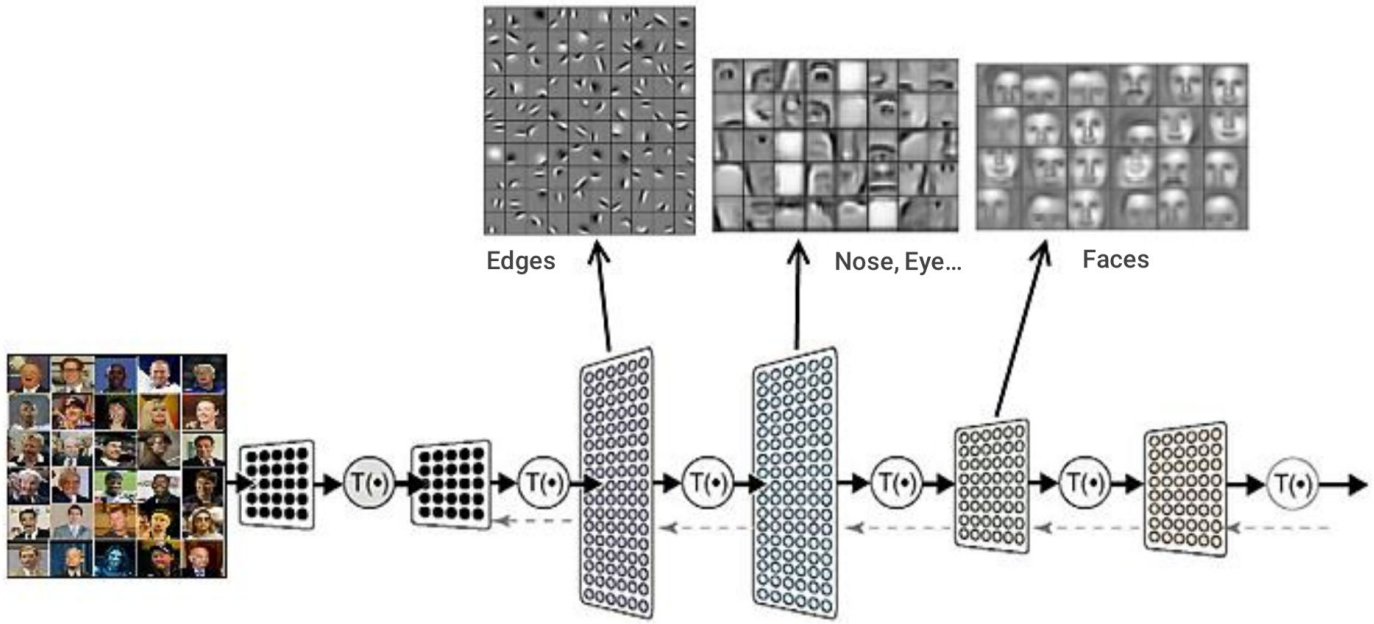
Why Deep Learning?

- Hand engineered features are time consuming, brittle and not scalable in practice.



Why Deep Learning?

- Can we learn the underlying features directly from data?



The Big Players

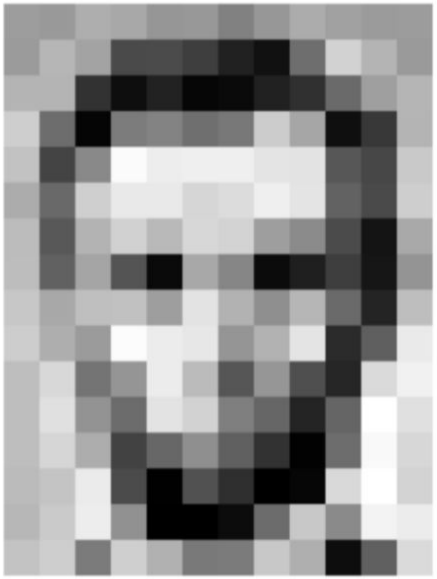
Deep Learning is an algorithm which has no theoretical limitations of what it can learn; the more data you give and the more computational time you provide, the better it is – *Geoffrey Hinton (Google)*

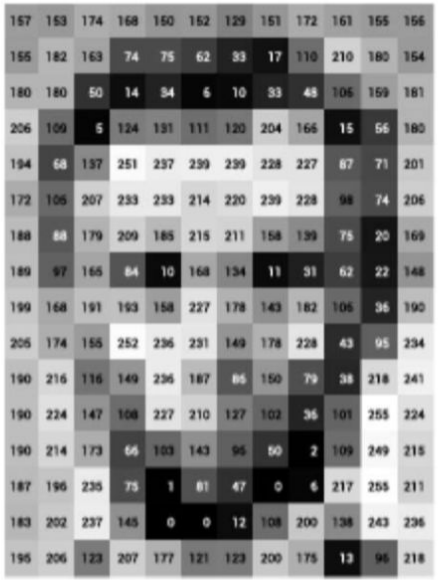
I have worked all my life in Machine Learning, and I've never seen one algorithm knock over benchmarks like Deep Learning
– *Andrew Ng (Stanford & Baidu)*

For a very long time it will be a complementary tool that human scientists and human experts can use to help them with the things that humans are not naturally good – *Demis Hassabis (Co-Founder* DeepMind)

The Big Players

Deep Learning In Image Processing



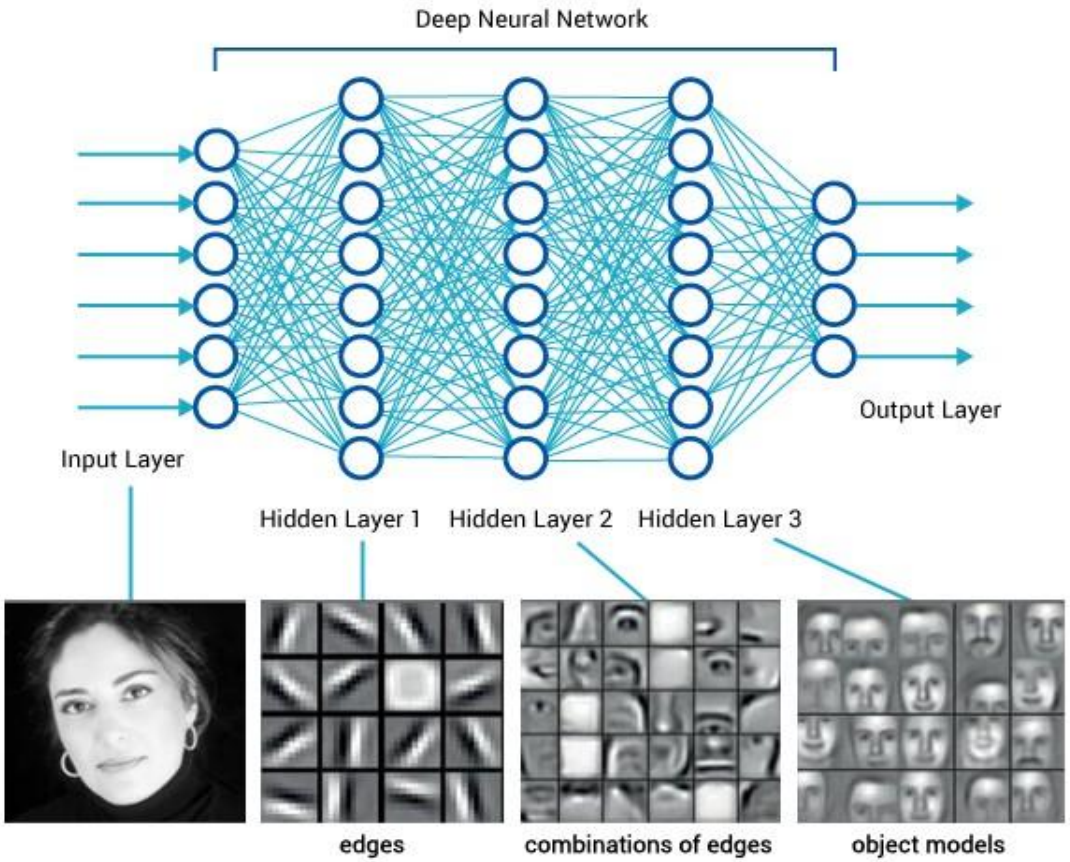


What the computer sees

157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
196	206	123	207	177	121	123	200	175	13	96	218

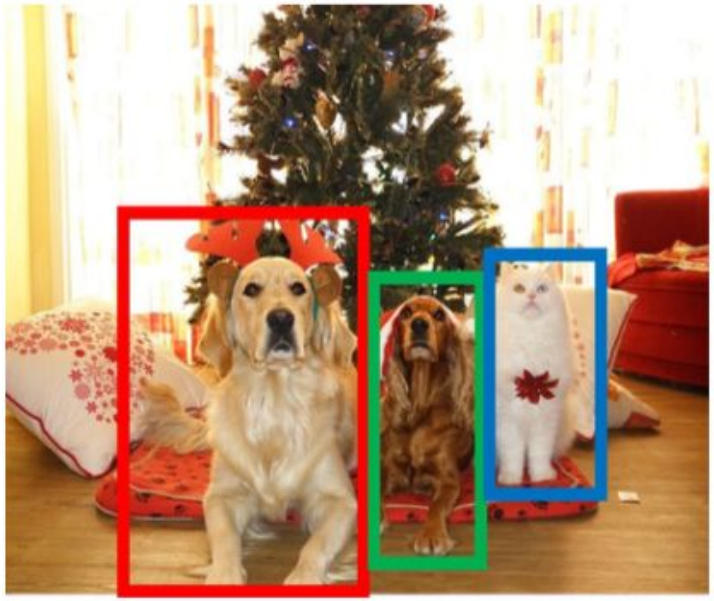
An image is just a matrix of numbers [0,255]!
i.e., 1080x1080x3 for an RGB image

Deep Learning In Image Processing



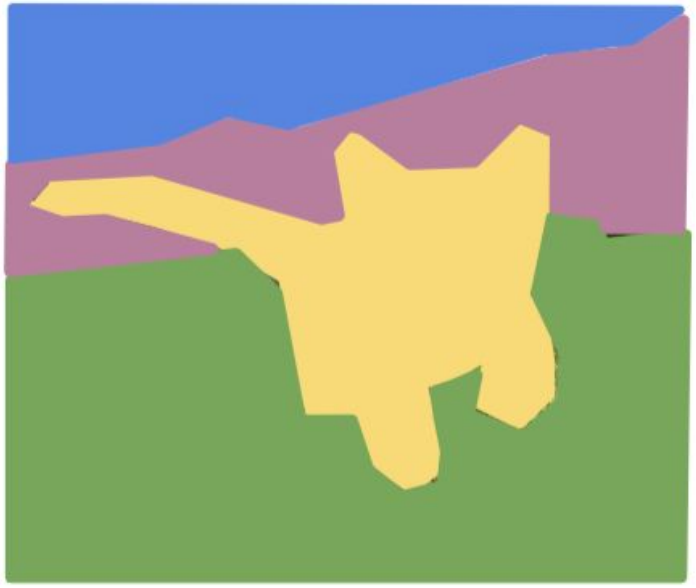
Applications Of DL In DIP

Object Detection



DOG, DOG, CAT

Semantic Segmentation



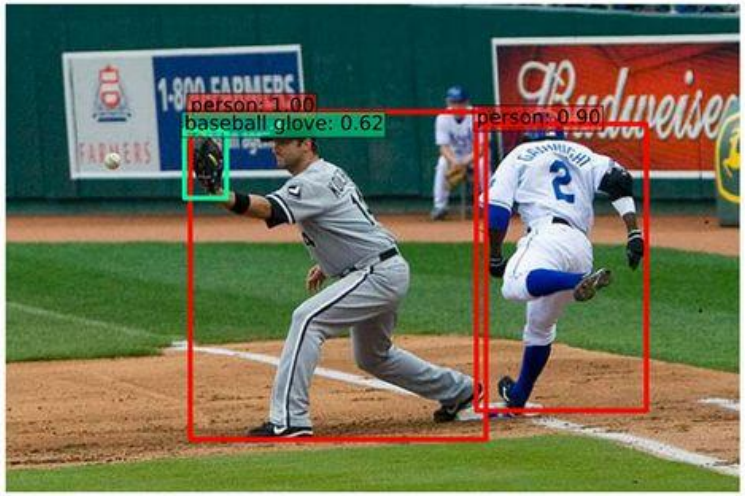
GRASS, CAT,TREE, SKY

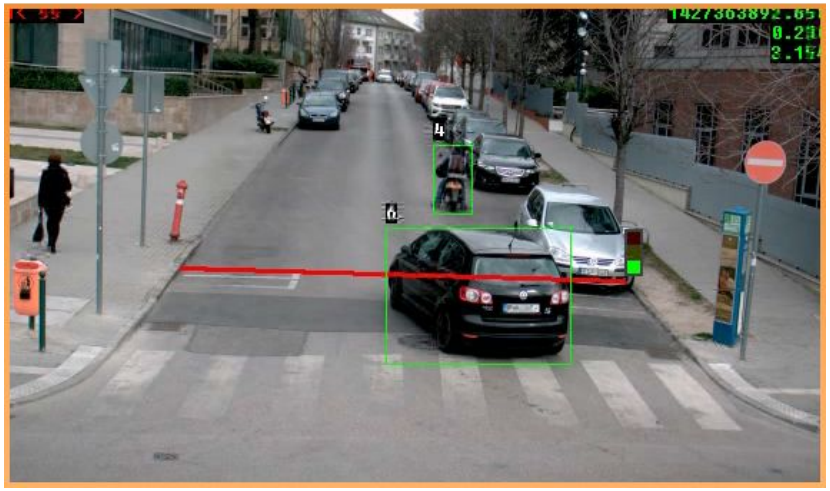
Object Detection

Applications of Object Detection

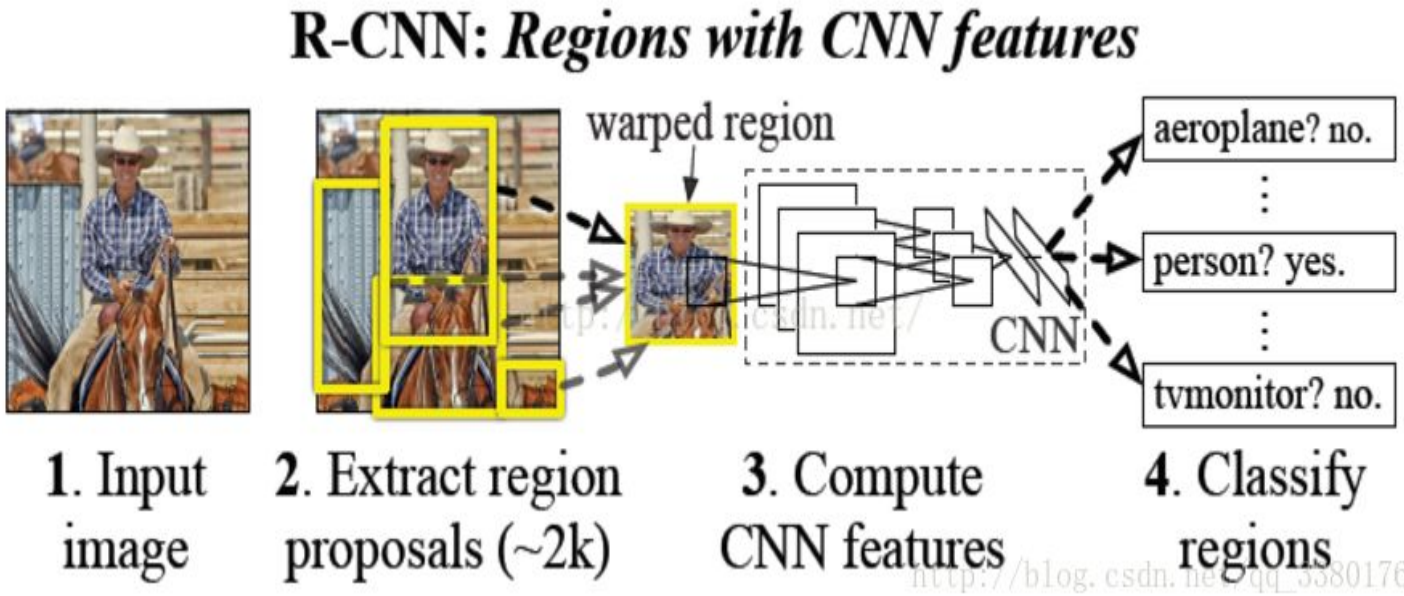
The most popular application in computer vision field:

- Sports video
- Find the lost key
- For the police, to judge which car is in traffic violations

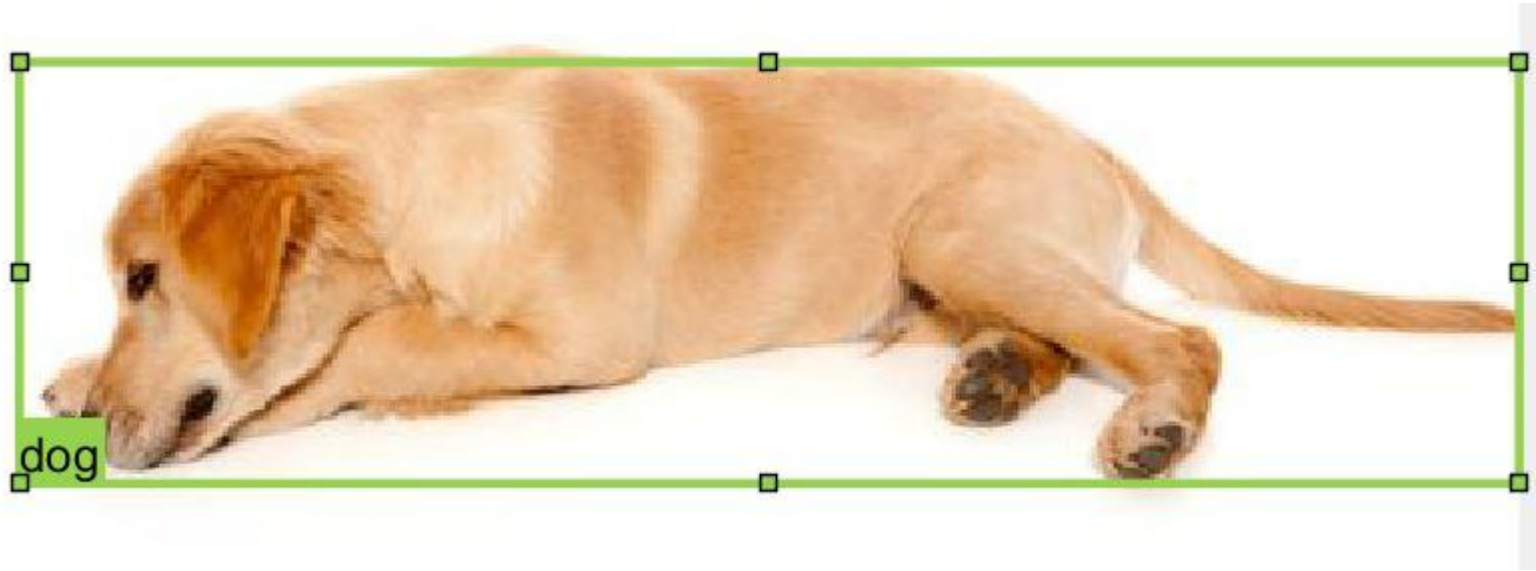




Object Detection Using R-CNN



Label Image



Export the Label to Get the Table

	1 imageFilename	2 dog	3
1	'F:\paper\dog.jpg'	[35,51,426,...	
2	'F:\paper\dog1.j...	[287,87,17...	
3	'F:\paper\dog2.j...	[51,188,44...	
4	'F:\paper\dog3.j...	[35,10,303,...	
5	'F:\paper\dog4.j...	[76,48,366,...	
6	'F:\paper\dog5.j...	[196,62,85...	
7	'F:\paper\dog6.j...	[64,25,159...	
8	'F:\paper\dog7.j...	[74,27,363,...	
9	'F:\paper\dog8.j...	[451,146,8...	
10	'F:\paper\dog9.j...	[17,8,475,4...	
11	'F:\paper\dog10...	[10,39,386,...	
12	'F:\paper\dog11...	[138,120,1...	
13	'F:\paper\dog12...	[130,8,306,...	
14	'F:\paper\dog13...	*4x4 double*	
15	'F:\paper\dog14...	[162,53,16...	
16	'F:\paper\dog15...	[651,616,1...	
17	'F:\paper\dog16...	[198,23,30...	
18	'F:\paper\dog17...	[172,69,43...	
19	'F:\paper\dog18...	[202,111,7...	

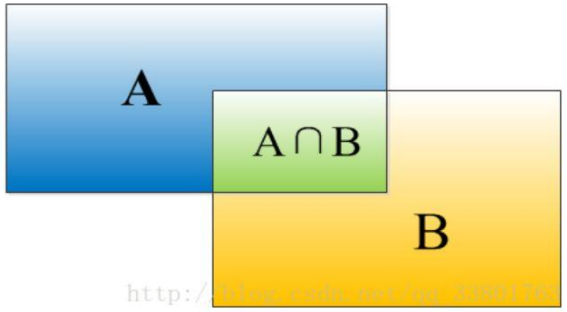
1.Input image

2.Regional proposal

Selective search: regional proposal

IOU: $IOU=(A \cap B)/(A \cup B)>0.5$

object boxing



A：regional proposal boxing

B: label bounding boxing

3.CNN for feature extraction

- a self-learning feature extraction + softmax classifier
- In traditional image processing: sobel -convolution kernel (extract feature)

way: features which are gotten by the traditional way, it transfer the features into neural network

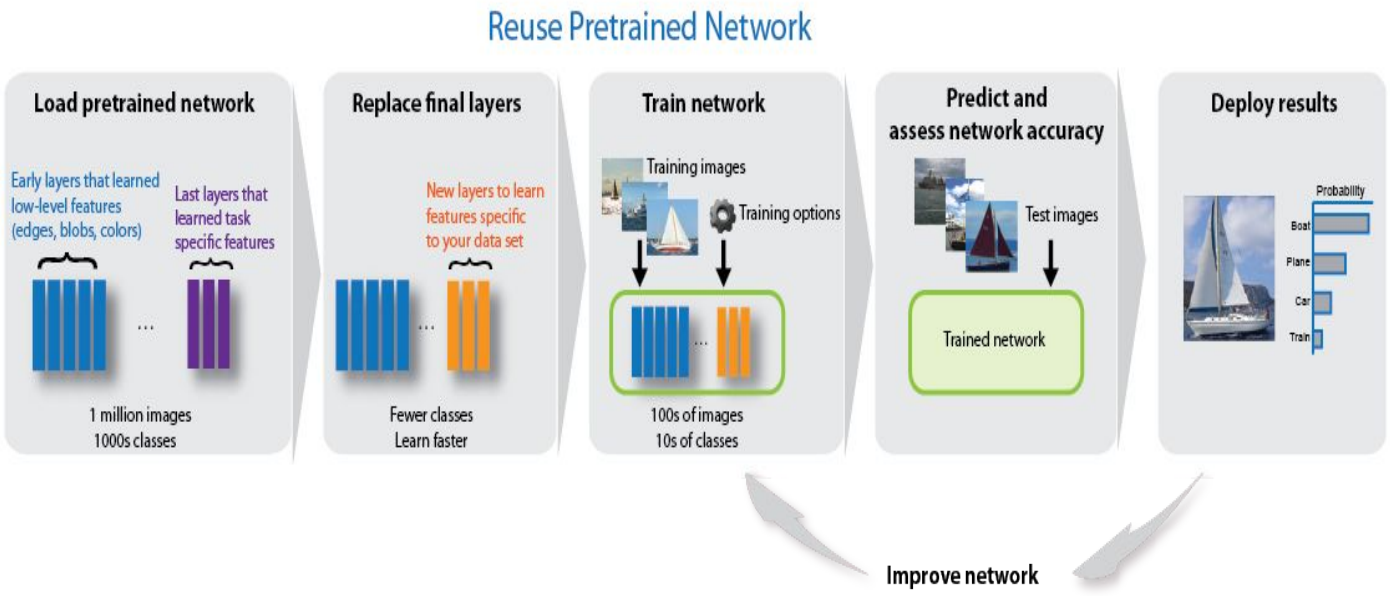
3. CNN feature extraction

In CNN: the size of convolutional kernel (by us) (learning process)

way:use the picture to be convoluted with the neural network , make the result more precise

Alexnet

- have been trained by 100 million images



Train RCNN(transfer learning using alexnet)

main function:

detector = trainRCNNObjectDetector (groundTruth, network, options)

need to detect one object and the background ,so there is 2 classes

```
x=alexnet.Layers(1:end-3);
numClasses=2;
lastlayers = [
fullyConnectedLayer(numClasses,'WeightLearnRateFactor',20,'BiasLearnRateFactor',20)
    softmaxLayer
    classificationLayer];

mylayers=[x;lastlayers];
```

Options

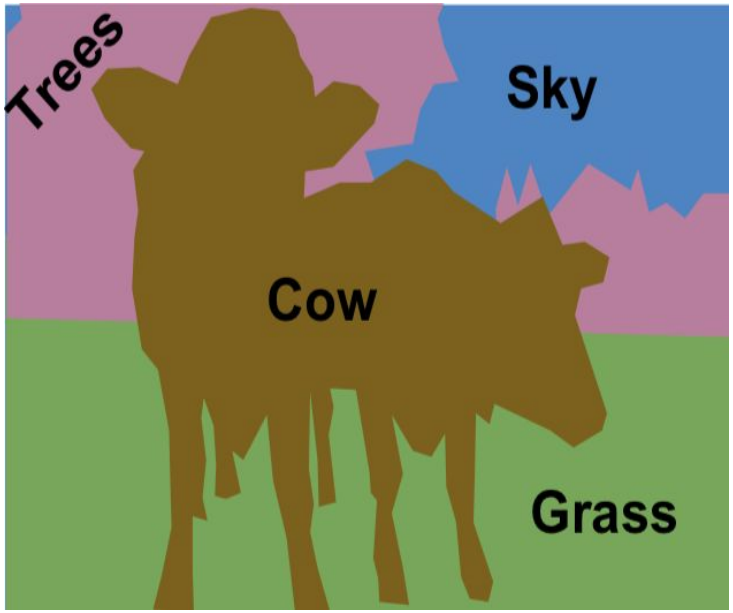
```
options = trainingOptions('sgdm', ...
    'MiniBatchSize',10, ...
    'MaxEpochs',6, ...
    'InitialLearnRate',1e-4, ...
    'Shuffle','every-epoch', ...
    'ValidationData',augimdsValidation, ...
    'ValidationFrequency',3, ...
    'Verbose',false, ...
    'Plots','training-progress');
```

The expected test result after training RCNN

Semantic Segmentation

What Is Semantic Segmentation?

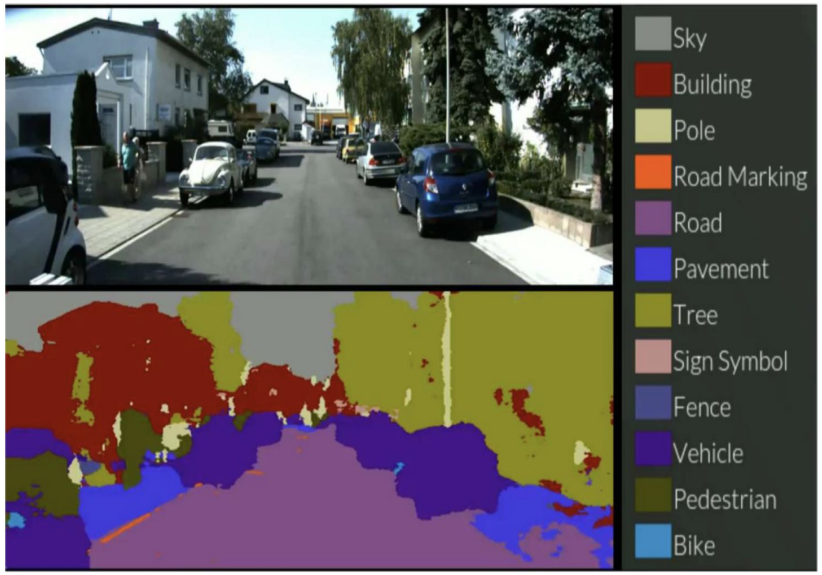
- Label each pixel in the image with a category label
- Don't differentiate instances, only care about pixels



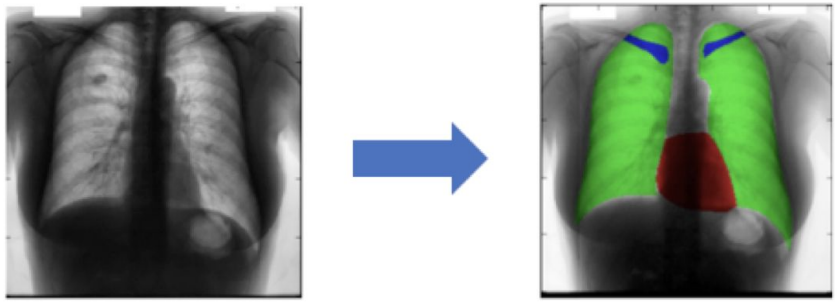
Why Semantic Segmentation?

- Segmentation models are useful for a variety of tasks, including:

Autonomous Driving



Medical Image Diagnostics

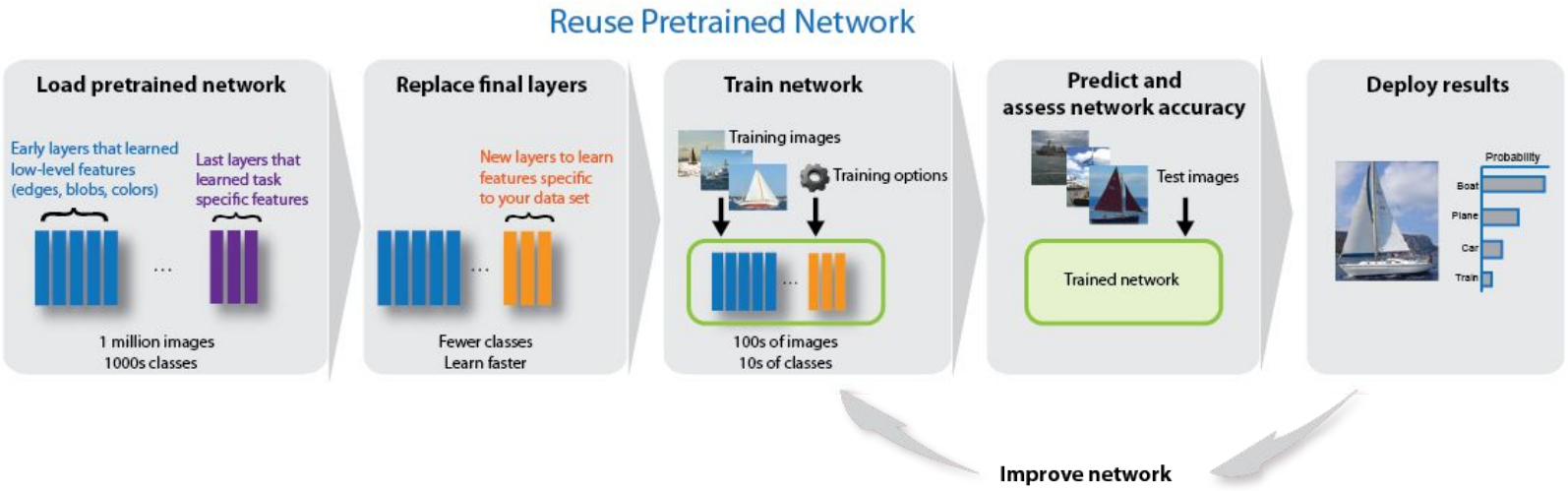


Input Image

Segmented Image

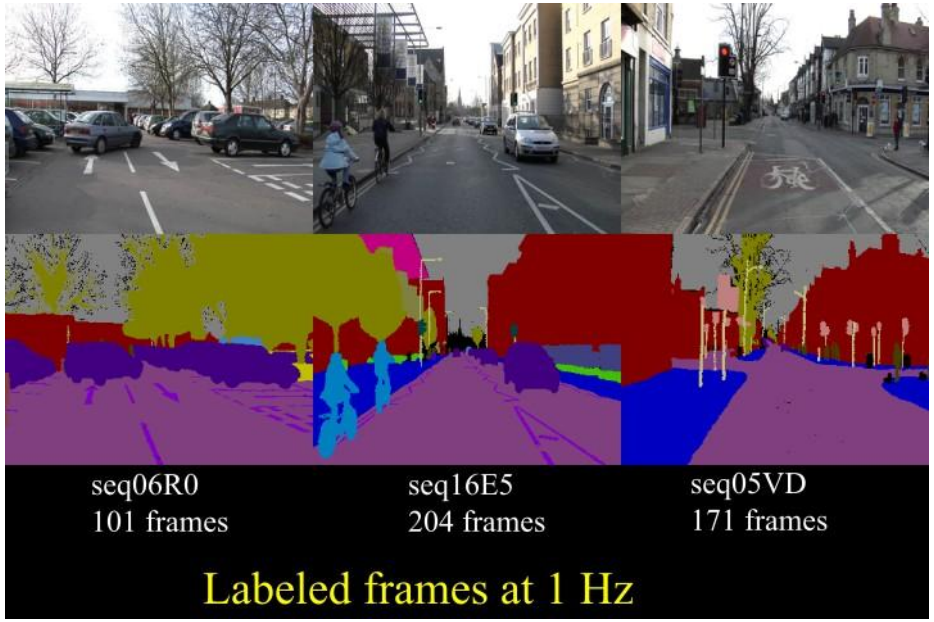
Build Model Using Transfer Learning

- Transfer learning is a deep learning approach in which a model that has been trained for one task is used as a starting point to train a model for similar task.



Camvid Datasets

- The Cambridge-driving Labeled Video Database (CamVid) is the first collection of videos with object class semantic labels, complete with metadata. The database provides ground truth labels that associate each pixel with one of 32 semantic classes.



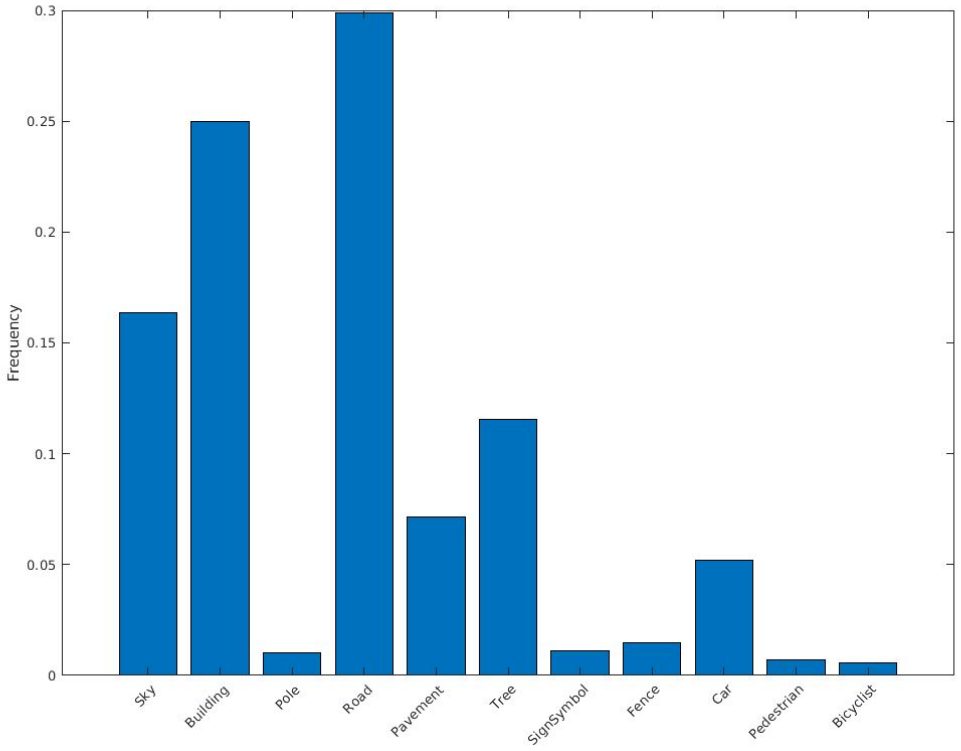
Analyze Datasets

- We use following codes to generate the histogram To show the distribution of class labels in the dataset.

```
%% ============================ Analyze Dataset ============================
% Visualize the distribution of class labels in the CamVid dataset
tbl = countEachLabel(pxds);
frequency = tbl.PixelCount/sum(tbl.PixelCount);

bar(1:numel(classes),frequency)
xticks(1:numel(classes))
xticklabels(tbl.Name)
xtickangle(45)
ylabel('Frequency')
```

Analyze Datasets

- The result is shown below:



Balance Dataset

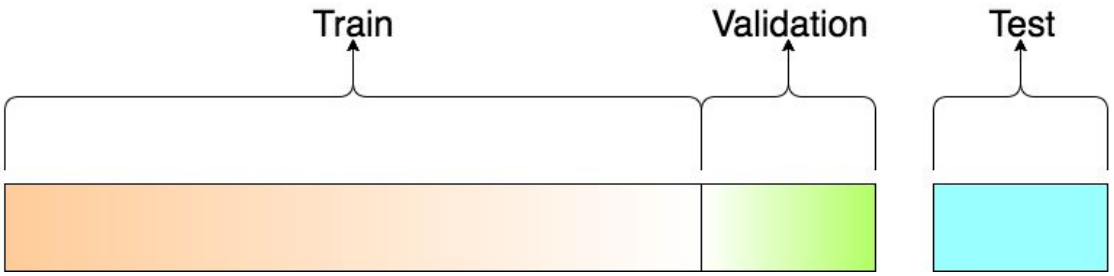
- As we can see, the classes in dataset are not balanced.
- Unbalanced classes in the training data might lead to bias.
- To improve training, we use class weighting to balance the classes.

```
%% ================ Balance Classes Using Class Weighting =================
% Get the imageFreq using the data from the countEachLabel function
imageFreq = tbl.PixelCount ./ tbl.ImagePixelCount;

% The higher the frequency of a class the smaller the classWeight
classWeights = median(imageFreq) ./ imageFreq
```

Splitting Datasets

- We use following codes to split our datasets into training sets(60%), cross-validation sets(20%) and test sets(20%).

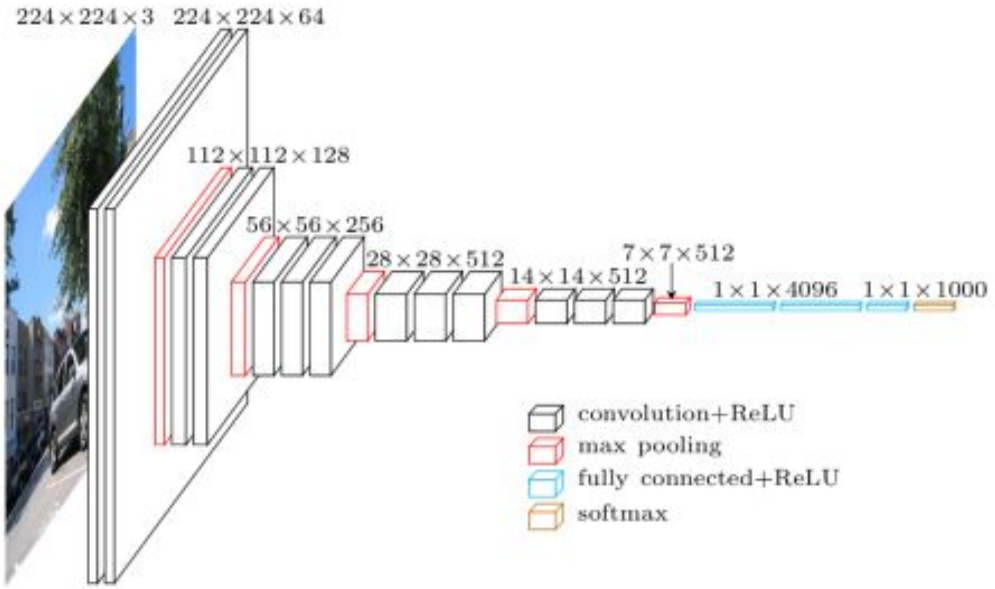
```
%% ============= Prepare Training, Validation, and Test Sets ==============
% Randomly splits the image and pixel label data into a training, validation and test set.
% Where Training Sets 60%, Validation 20% and Test Sets 20%
[imdsTrain, imdsVal, imdsTest, pxdsTrain, pxdsVal, pxdsTest] = partitionCamVidData(imds,pxds);
% numTrainingImages = numel(imdsTrain.Files)
% numValImages = numel(imdsVal.Files)
% numTestingImages = numel(imdsTest.Files)
```



VGG-16

- VGG-16 is a convolutional neural network that is trained on more than a million images from the ImageNet database.

- The network is 16 layers deep and can classify images into 1000 object categories.



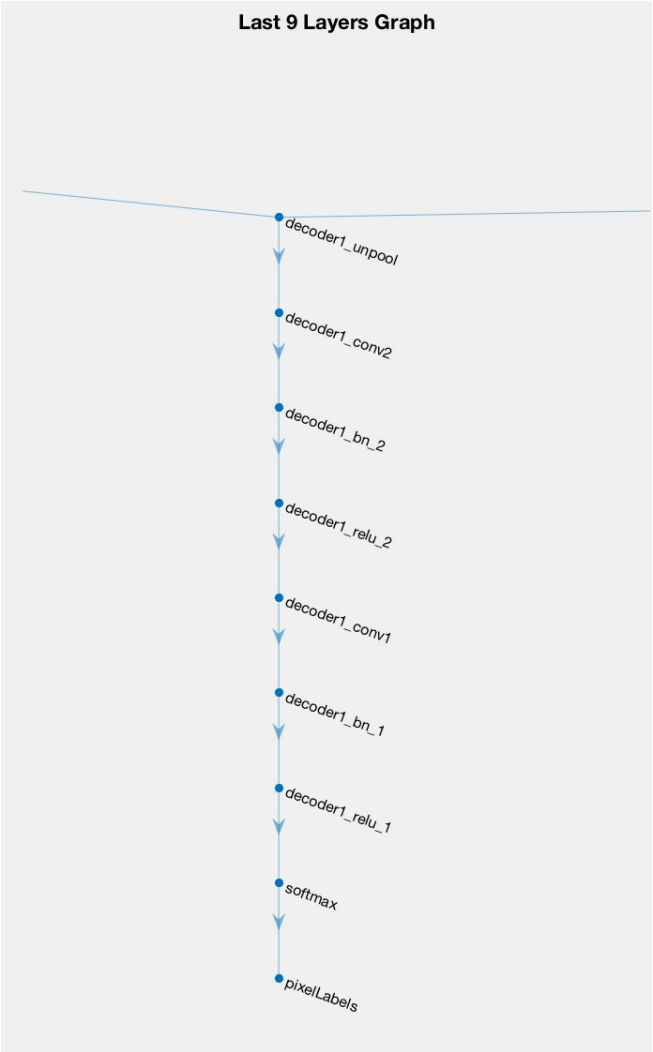
Create Model and Modify Our Model

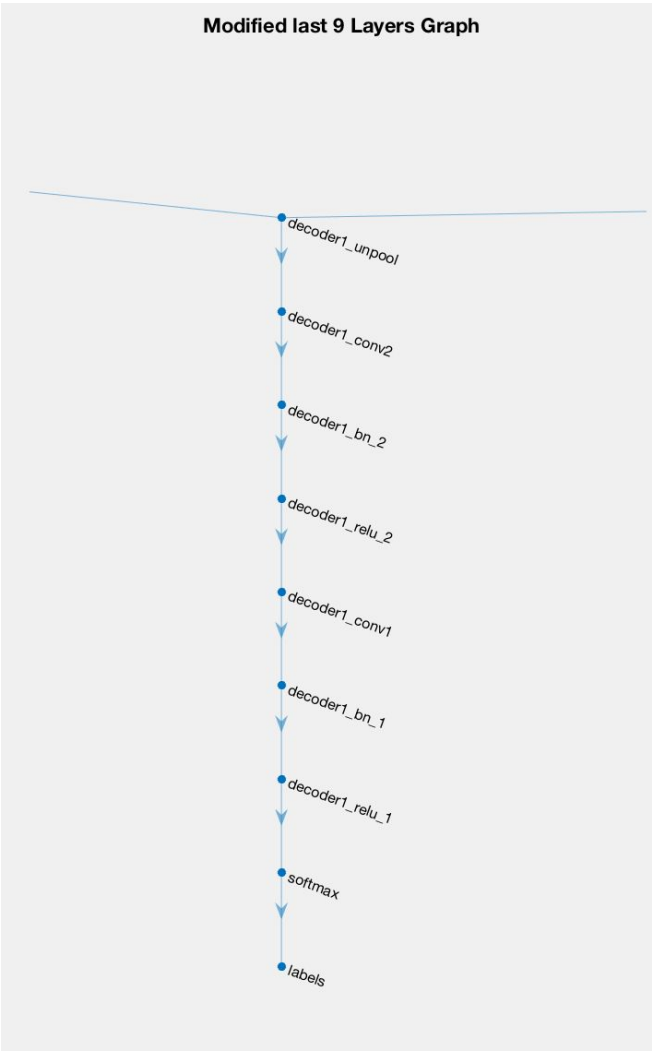
- We firstly create our model.

```
%% ========================= Create the Network ==========================
% Specify the network image size. This is typically the same as the traing image sizes.
imageSize = [720 960 3];

% Specify the number of classes.
numClasses = numel(classes);

% Create SegNet Network
lgraph = segnetLayers(imageSize,numClasses,'vgg16');
```

- Then, we remove last layer of vgg16 and add the new one we created.

```
%% ==================== Modify VGG16 Model For Our Task ====================
% Create a new layer with the new pixelClassificationLayer.
pxLayer = pixelClassificationLayer('Name','labels','ClassNames', tbl.Name, 'ClassWeights', classWeights)

% Remove last layer of vgg16 and add the new one we created.
lgraph = removeLayers(lgraph, {'pixelLabels'});
lgraph = addLayers(lgraph, pxLayer);

% Connect the newly created layer with the graph.
lgraph = connectLayers(lgraph, 'softmax','labels');
lgraph.Layers
subplot(1,2,2)
plot(lgraph);
xlim([2.862 3.200])
ylim([-0.9 10.9])
axis off
title(' Modified last 9 Layers Graph')
```


Last 9 Layers Graph
decoder1_unpool
decoder1_conv2
decoder1_bn_2
decoder1_relu_2
decoder1_conv1
decoder1_bn_1
decoder1_relu_1
softmax
pixelLabels


Modified last 9 Layers Graph
decoder1_unpool
decoder1_conv2
decoder1_bn_2
decoder1_relu_2
decoder1_conv1
decoder1_bn_1
decoder1_relu_1
softmax
labels

Optimization Algorithm

SGDM

- Stochastic Gradient Descent with Momentum (SGDM) is an iterative method for optimizing a differentiable objective function, a stochastic approximation of gradient descent optimization.

- The hyper-parameters are specified as:

Momentum = 0.9

InitialLearnRate = 1e-2

L2Regularization = 0.0005

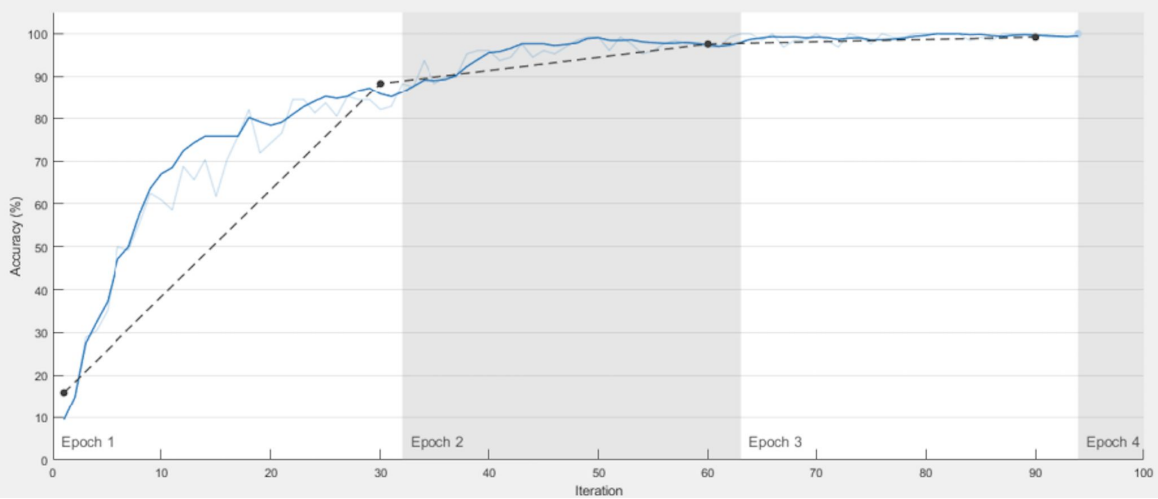
MaxEpochs = 120

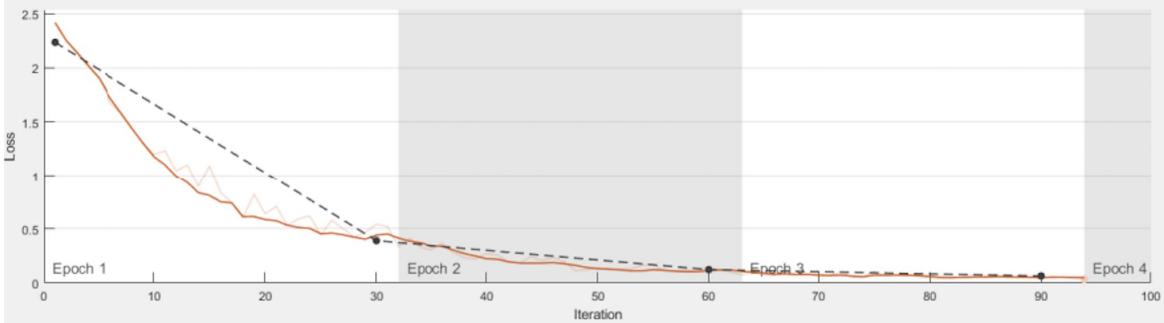
MiniBatchSize = 8

```
options = trainingOptions('sgdm', ...
    'Momentum', 0.9, ...
    'InitialLearnRate', 1e-2, ...
    'L2Regularization', 0.0005, ...
    'MaxEpochs', 120,...
    'MiniBatchSize', 4, ...
    'Shuffle', 'every-epoch', ...
    'Verbose', false,...
    'Plots','training-progress');
```

```
%% ============================ Start Training ==============================
% Combine the training data and data augmentation selections
pximds = pixelLabelImageSource(imdsTrain,pxdsTrain,'DataAugmentation',augmenter);

% Trains the network for semantic segmentation
[net, info] = trainNetwork(pximds,lgraph,options);
disp('NN trained');
```

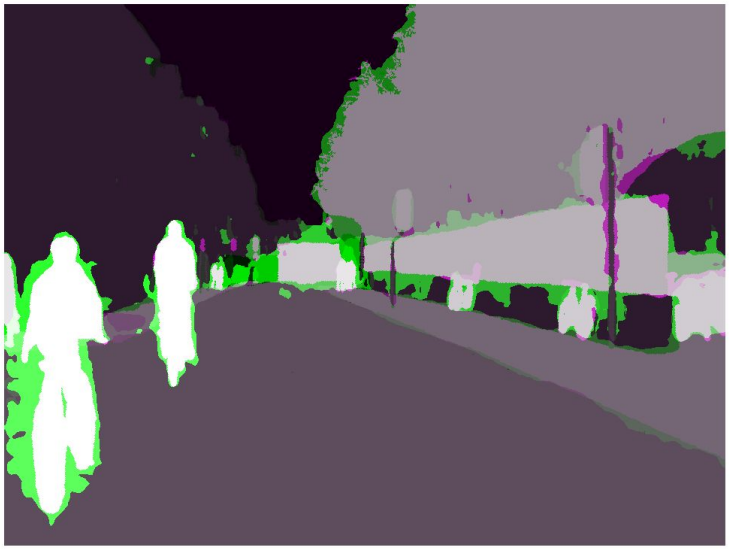
Train Our Model





Visually Test Our Model

- To quickly check our model, we select one picture from our results and compared it with original labeled image.



Future Work

- So far, we've used transfer learning technique to build and trained our model.
- However, we just measured our model visually, and just based on one certain sample in our results.
- In the future, we prepare to use confusion matrix to evaluate our model numerically.

		Prediction outcome: positive	Prediction outcome: negative	
Actual value	positive	TP	FN	$TP + FN$
	negative	FP	TN	$FP + TN$
		$TP + FP$	$FN + TN$	

References

- 1: Girshick, R., J. Donahue, T. Darrell, and J. Malik. "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation." Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition. Columbus, OH, June 2014, pp. 580-587.
- 2: Pathak, A. R., Pandey, M., & Rautaray, S. (2018). Application of deep learning for object detection. Procedia Computer Science, 132, 1706-1717. doi:10.1016/j.procs.2018.05.144.
- 3: Xu, X., Li, Y., Wu, G., & Luo, J. (2017). Multi-modal deep feature learning for RGB-D object detection. Pattern Recognition, 72, 300-313. doi:10.1016/j.patcog.2017.07.026.
- 4: Jonathan Long, Evan Shelhamer, Trevor Darrell. Fully Convolutional Networks for Semantic Segmentation. UC Berkeley. 2014.
- 5: Olaf Ronneberger, Philipp Fischer, Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. 2015.
- 6: Brostow, G. J., J. Fauqueur, and R. Cipolla. "Semantic object classes in video: A high-definition ground truth database." Pattern Recognition Letters. Vol. 30, Issue 2, 2009, pp 88-97.

Thank You!