Segmentation of Breast Cancer Masses in Digital Mammograms: A Convolutional Network

Progress Report

Erick Cobos[1]

[1]Centro de Sistemas Intelligentes
Tecnologico de Monterrey

September, 2016

Table of Contents

Thesis objective

The main goal is to segment breast cancer lesions using convolutional networks, an end-to-end learnable model.

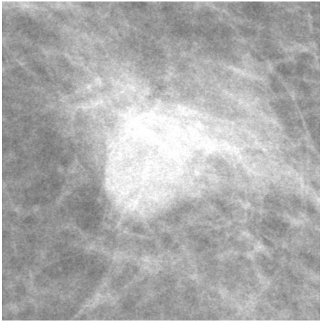
- Obtain and process the mammographic database.
- Develop software to handle the database and train new deep learning models.
- Train modern, fine-tuned convolutional networks.
- Test the viability of convolutional networks for breast cancer research.
- Propose ideas for future research in the area.

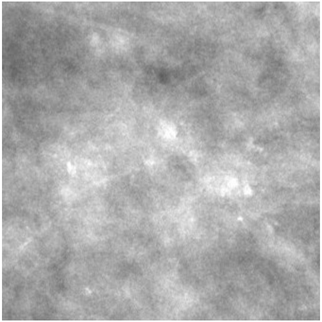
Why breast cancer

Breast cancer is the most commonly diagnosed cancer among women.
For women, death rates are the second highest of any cancer. However, survival rate when detected early is close to 100%.
On-going project at the institution.

Breast cancer signs

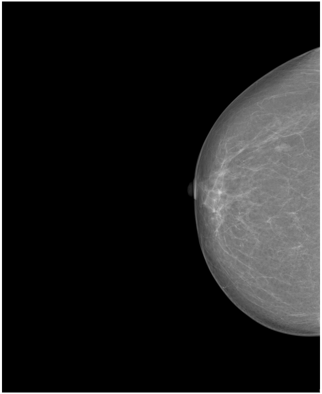
Masses vs. microcalcifications



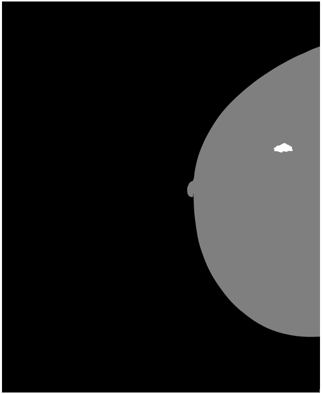


Detection vs. diagnosis

Lesion segmentation



(a) Mammogram



(b) Segmentation

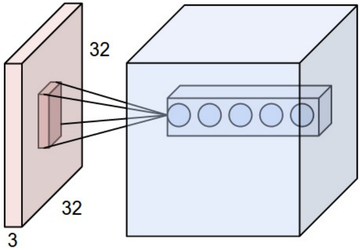
Why convolutional networks?

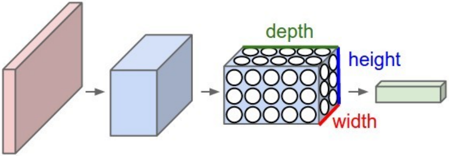
- Convnets have showed great results in image classification tasks.
- Convnets learn which features are important for the classification.
- We dont need experts to carefully handcraft and select features.

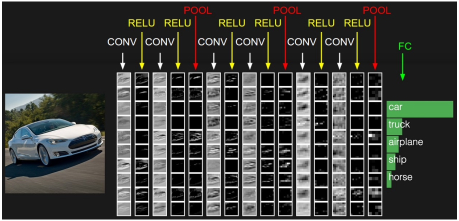
Cons: Need processing power, data.

Convolutional networks

Learnable filters: spatially local and simple in early layers, global and complex in deeper layers.

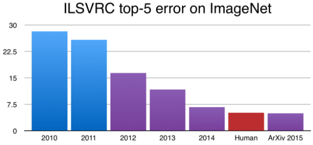


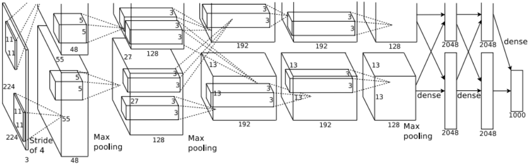




[Karpathy et al., 2016]

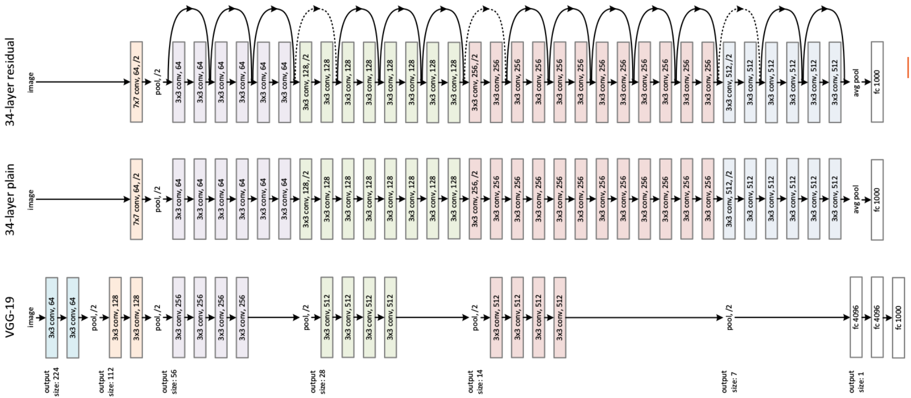
What have they done





[Nvidia Corp., 2015], [Krizhevsky et al., 2012]

Architectures

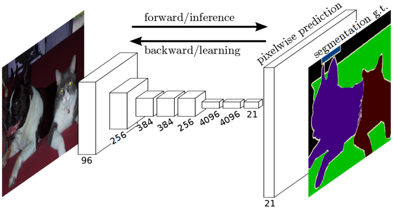


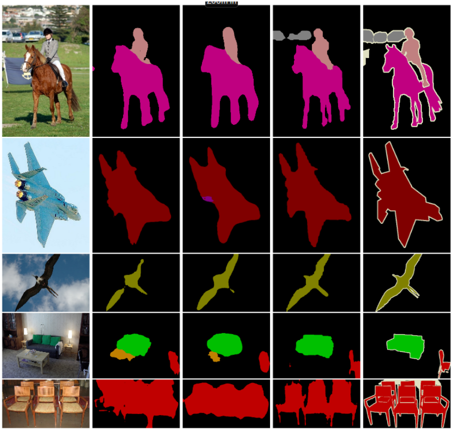
Deeper, no pooling layers, no fully-connected layers, no dropout, all 3x3 kernels, batch normalization, residual connections, attention, memory...

[He et al., 2015]

Segmentation

Solved by upsampling, deconvolution or dilated convolutions.





(a) Image (b) FCN-8s (c) DeepLab (d) Our front end (e) Ground truth

[Longet al., 2015], [Yu and Koltun, 2016]

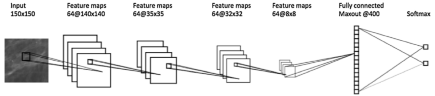
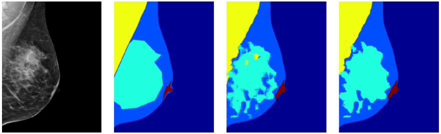


Table 2 – Summary of results in terms of AUC in the test set. Best results are shown in bold typeface and (*) signals scores with no evidence of differences from the highest ($\rho < 0.1$).

Representation	Standalone	Combined with HCfeats
CNN3	**0.82 ± 0.03**	0.82 ± 0.03 (*)
CNN2	0.76 ± 0.05	0.78 ± 0.04
HGD	0.78 ± 0.04	**0.83 ± 0.04**
HOG	0.77 ± 0.03	0.81 ± 0.03 (*)
DeCAF	0.79 ± 0.05	0.82 ± 0.03 (*)
HCfeats	0.77 ± 0.02	–

(a) Mass diagnosis (4 layers, 3.4M params)

Layer	1	2	3	4	5	6	7 - Output
Stage	conv+relu+max	conv+relu+max	conv+relu+max	dropout	full+relu	full+relu	full
# channels	16	16	16	16	128	16	4
Filter size	7 × 7	5 × 5	5 × 5	-	-	-	-
Pooling size	3 × 3	3 × 3	3 × 3	-	-	-	-
Pooling stride	2	2	2	-	-	-	-
Dropout factor	-	-	-	0.5	-	-	-
Spatial input size	61 × 61	27 × 27	11 × 11	3 × 3	3 × 3	1 × 1	1 × 1



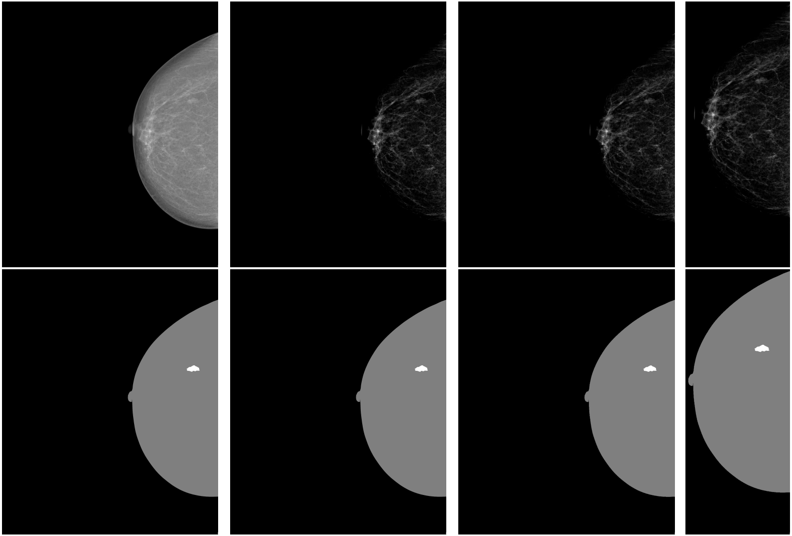
	Pectoral muscle	Fibroglandular tissue	Nipple	Breast tissue (fat, skin)	Average over four tissues
Raw DNN output	0.78	0.60	0.47	0.80	0.66
Post-processed DNN output	0.85	0.61	0.56	0.81	0.71
Raw DNN output, with (x, y) coordinates	0.78	0.60	0.56	0.77	0.68
Post-processed, with (x, y) coordinates	0.79	0.61	0.57	0.77	0.69

Table 2. Average Dice coefficients obtained using the proposed methods for different breast tissues, and average coefficients over the four tissues.

(b) Tissue segmentation (6 layers, 34K params)

[Arevalo et al., 2016], [Dubrovina et al.,2015]

Database preprocessing



(a) Original image (b) Enhancement (c) Downsampling (d) Final

Software

TensorFlow

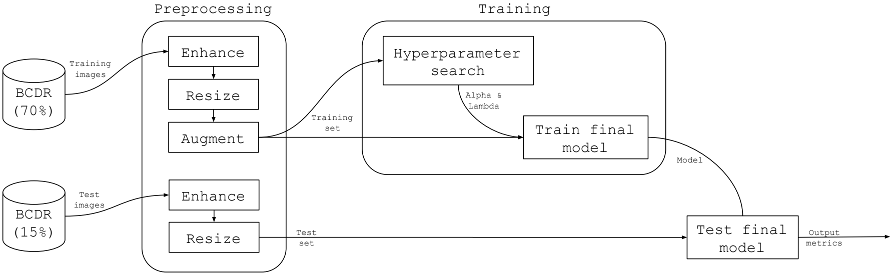
Unwatch 1 | Star 10 | Fork 6

model_v3.py

```python
    """
    # Create filter and biases
    filter = tf.Variable(initialize_weights(filter_shape), name='weights')
    biases = tf.Variable(tf.zeros([filter_shape[3]]), name='biases')

    # Add weights to the weights collection (for regularization)
    tf.add_to_collection(tf.GraphKeys.WEIGHTS, filter)

    # Perform dilated 2d convolution
    w_times_x = tf.nn.atrous_conv2d(input, filter, dilation, padding='SAME')
    output = tf.nn.bias_add(w_times_x, biases)

    return output

def leaky_relu(x, alpha=0.1):
    """ Leaky ReLU activation function."""
    with tf.name_scope('leaky_relu'):
        output = tf.maximum(tf.mul(alpha, x), x)
    return output

def dropout(x, keep_prob):
    """ Performs dropout if training. Otherwise, returns original."""
    output = tf.cond(drop, lambda: tf.nn.dropout(x, keep_prob), lambda: x)
    return output

# Create a batch with a single image
batch = tf.expand_dims(image, 0)

# Define the architecture
with tf.name_scope('conv1'):
    conv = conv_op(batch, [6, 6, 1, 32], [1, 2, 2, 1])
    relu = leaky_relu(conv)
    conv1 = dropout(relu, keep_prob=0.9)
with tf.name_scope('conv2'):
    conv = conv_op(conv1, [3, 3, 32, 32])
    relu = leaky_relu(conv)
    conv2 = dropout(relu, keep_prob=0.9)

with tf.name_scope('conv3'):
    conv = conv_op(conv2, [3, 3, 32, 64], [1, 2, 2, 1])
    relu = leaky_relu(conv)
    conv3 = dropout(relu, keep_prob=0.8)
```

Python ▾ Tab Width: 4 ▾ Ln 49, Col 14 INS

Overview of the solution



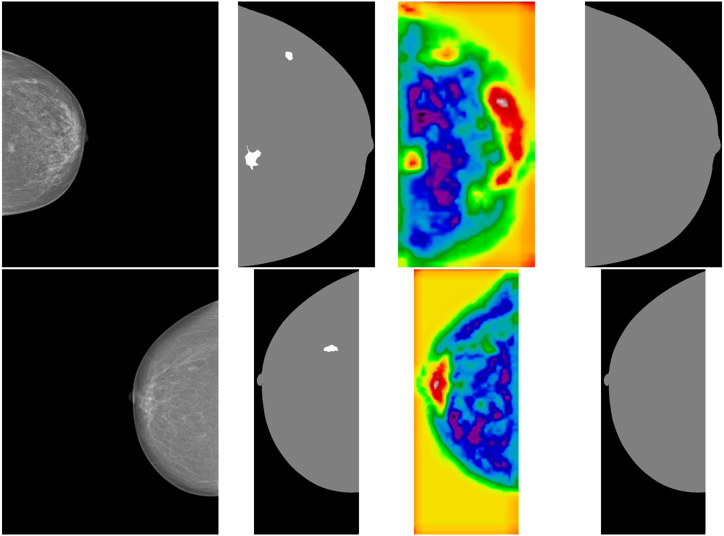
Experiment 1

Modelled on a VGG network, winner of the 2014 ImageNet.

Layer	Filter	Stride	Pad	Volume	Parameters
`INPUT`	-	-	-	$112 \times 112 \times 1$	-
`CONV -> Leaky RELU`	6×6	2	2	$56 \times 56 \times 56$	2 072
`CONV -> Leaky RELU`	3×3	1	1	$56 \times 56 \times 56$	28 280
`MAXPOOL`	2×2	2	0	$28 \times 28 \times 56$	-
`CONV -> Leaky RELU`	3×3	1	1	$28 \times 28 \times 84$	42 420
`CONV -> Leaky RELU`	3×3	1	1	$28 \times 28 \times 84$	63 588
`MAXPOOL`	2×2	2	0	$14 \times 14 \times 84$	-
`CONV -> Leaky RELU`	3×3	1	1	$14 \times 14 \times 112$	84 784
`CONV -> Leaky RELU`	3×3	1	1	$14 \times 14 \times 112$	113 008
`CONV -> Leaky RELU`	3×3	1	1	$14 \times 14 \times 112$	113 008
`MAXPOOL`	2×2	2	0	$7 \times 7 \times 112$	-
`FC -> Leaky RELU`	7×7	1	3	$7 \times 7 \times 448$	2 459 072
`FC`	1×1	1	0	$7 \times 7 \times 1$	449
`BILINEAR (x16)`	-	-	-	$112 \times 112 \times 1$	-

IOU	F1-score	G-mean	Accuracy	Sensitivity	Specificity	Precision	Recall
0.109	0.197	0.49	0.98	0.243	0.987	0.165	0.243

Qualitative results



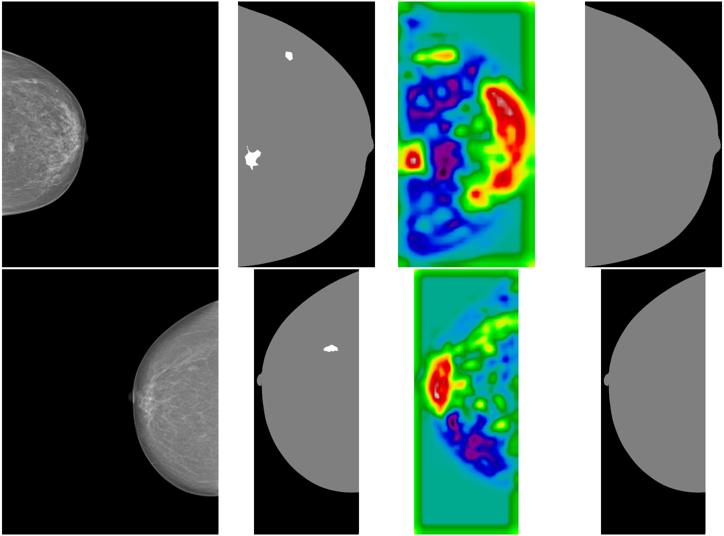
(a) Original (b) Label (c) Prediction (d) Segmentation

Experiment 2

To combat class imbalance, we use a loss function that weights errors over breast masses higher than errors over breast tissue. Architecture as in Experiment 1.

IOU	F1-score	G-mean	Accuracy	Sensitivity	Specificity	Precision	Recall
0.117	0.21	0.493	0.981	0.246	0.989	0.184	0.246

Qualitative results



(a) Original (b) Label (c) Prediction (d) Segmentation

Experiment 3

Modelled on the Residual network, winner of the 2015 ImageNet. Deeper, fewer parameters and better resolution (4x).

Layer	Filter	Stride	Pad	Dilation	Volume	Parameters
`INPUT`	-	-	-	-	$128 \times 128 \times 1$	-
`CONV -> LRELU`	6×6	2	2	1	$64 \times 64 \times 32$	1 184
`CONV -> LRELU`	3×3	1	1	1	$64 \times 64 \times 32$	9 248
`CONV -> LRELU`	3×3	2	1	1	$32 \times 32 \times 64$	18 496
`CONV -> LRELU`	3×3	1	1	1	$32 \times 32 \times 64$	36 928
`CONV -> LRELU`	3×3	1	2	2	$32 \times 32 \times 128$	73 856
`CONV -> LRELU`	3×3	1	2	2	$32 \times 32 \times 128$	147 584
`CONV -> LRELU`	3×3	1	2	2	$32 \times 32 \times 128$	147 584
`CONV -> LRELU`	3×3	1	2	2	$32 \times 32 \times 128$	147 584
`CONV -> LRELU`	3×3	1	4	4	$32 \times 32 \times 256$	295 168
`CONV`	8×8	1	14	4	$32 \times 32 \times 1$	16 385
`BILINEAR (x4)`	-	-		-	$128 \times 128 \times 1$	-

IOU	F1-score	G-mean	Accuracy	Sensitivity	Specificity	Precision	Recall
-	-	-	-	-	-	-	-

Writing the thesis

65 pages and counting...

Future work

- Train networks with increasing number of examples
- Train simpler network with all data
- Write thesis

?