Cancer Diagnosis in Medical Imaging

Kenneth Chen, Yajaira Gonzalez, Christoph Muus, Matthew Stewart, and Claire Stolz

Motivation

250,000 new cases of breast cancer per year

40,000 deaths from breast cancer each year

2nd most common cancer in women

2nd highest mortality rate of cancers for women

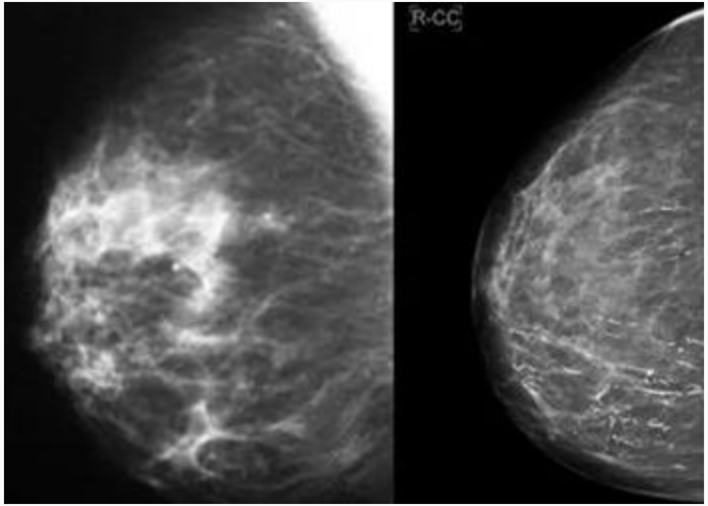
Motivation

Diagnosed primarily through medical imaging (mammograms)

High false positive rate: “**One in 10 women undergoing screening mammography are recalled for diagnostic workup, of which fewer than 5 percent will eventually be found to have cancer.**”

2 classes:

Malignant vs. Nonmalignant



Approach

Use 3 different networks to identify cancer in mammograms:

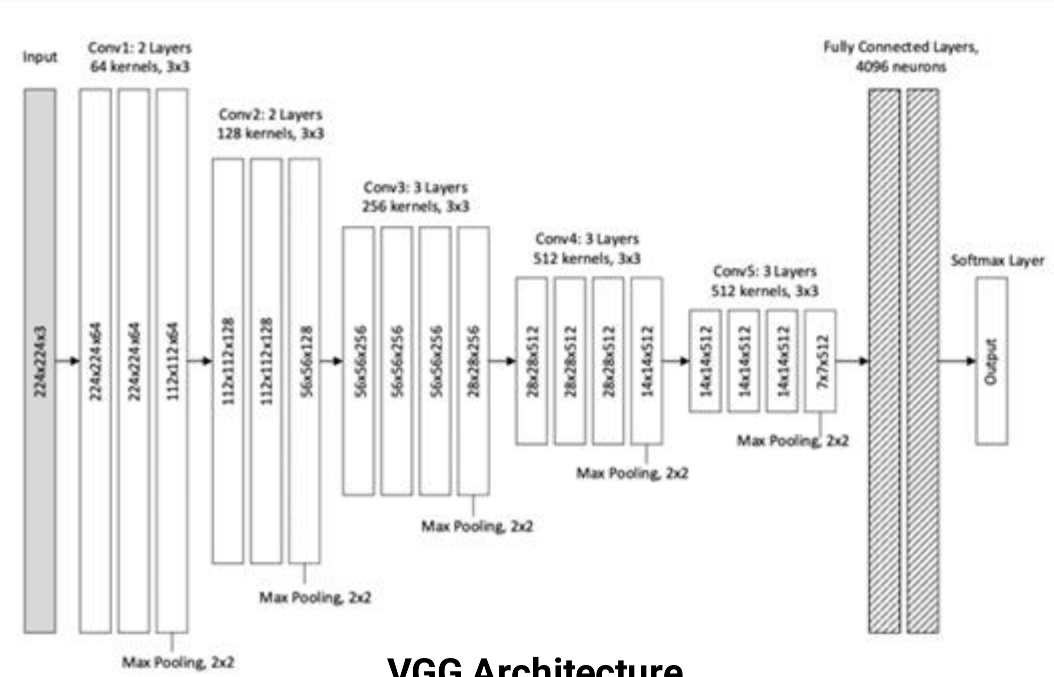
1. VGG
2. U-Net
3. ResNet

In addition we implemented:

- Transfer learning
- Image annotation for interpretability

VGG16 Architecture

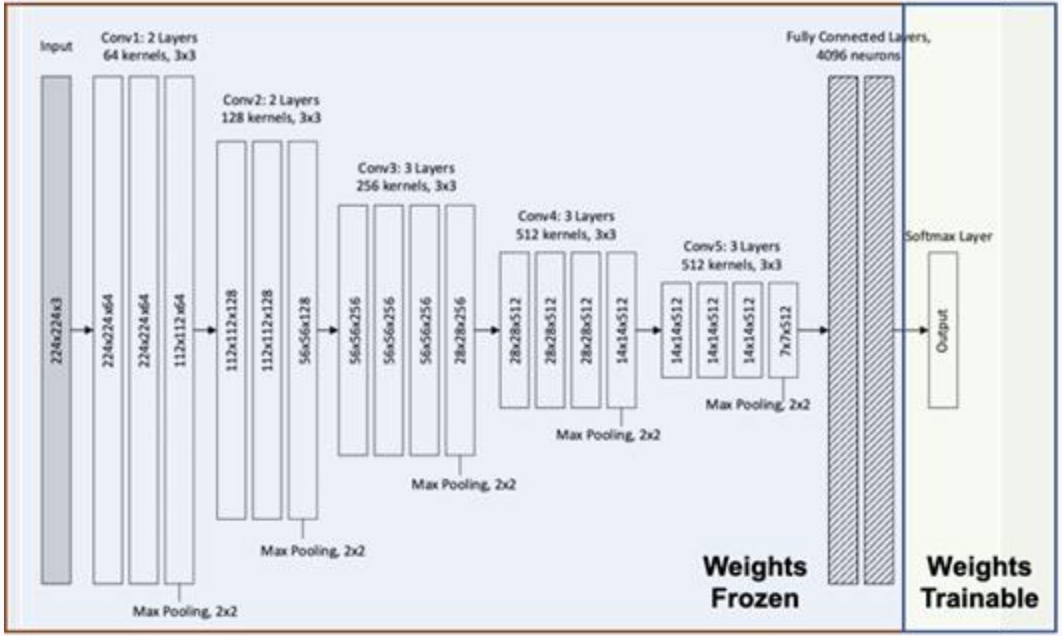
- Convolutional blocks with max pooling layers between each block
- Designed for RGB images
- Fully connected layers before output classifier
- Softmax function for multiclass classification



VGG Architecture

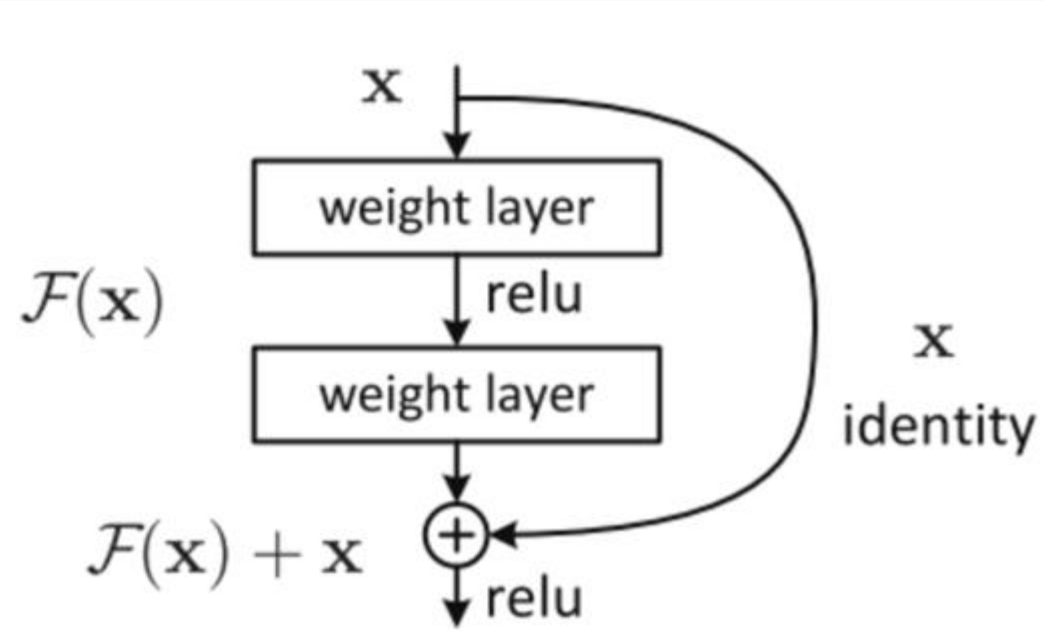
Transfer Learning with VGG16

- VGG16 pre-trained on Imagenet used for classification
- Output layer retrained on resized breast cancer image data
- Network restructured to accept grayscale images
- Output activation function changed from softmax to sigmoid for binary classification



VGG16 Architecture

ResNet - Architecture



- Skips connections, allowing jumps over some layers.
- Does this to avoid vanishing gradients by reusing activations
- Skipped layers are reintroduced once weights are learned to improve feature space

U-Net - Architecture

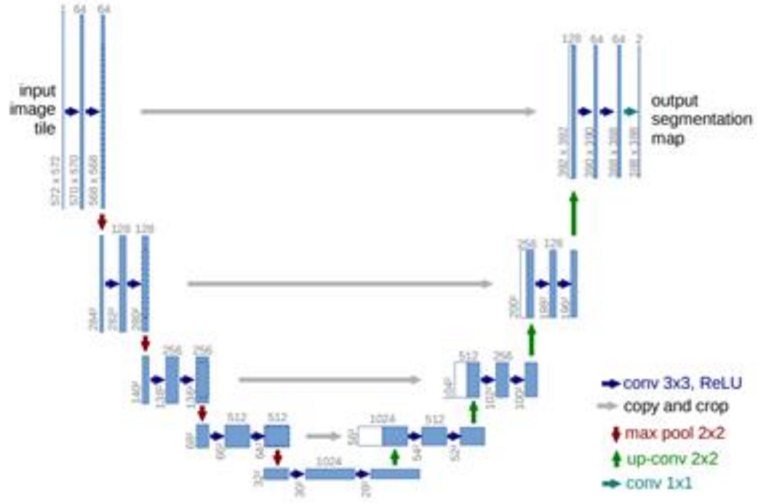
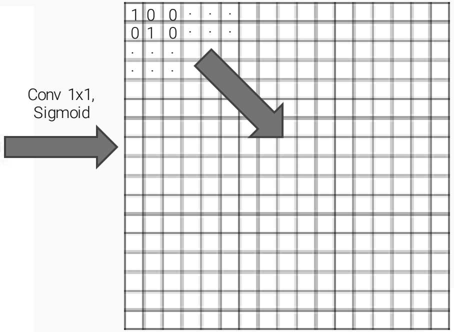


Fig. 1. U-net architecture (example for 32x32 pixels in the lowest resolution). Each blue box corresponds to a multi-channel feature map. The number of channels is denoted on top of the box. The x-y-size is provided at the lower left edge of the box. White boxes represent copied feature maps. The arrows denote the different operations.



Our U -Net: Additional
Flatten+Dense Layer

Binary: 0 or 1

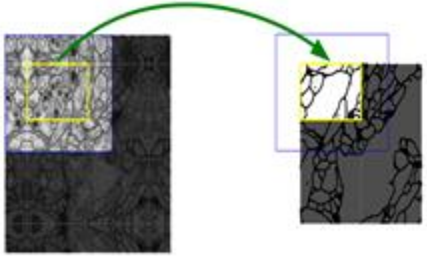
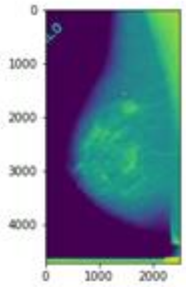
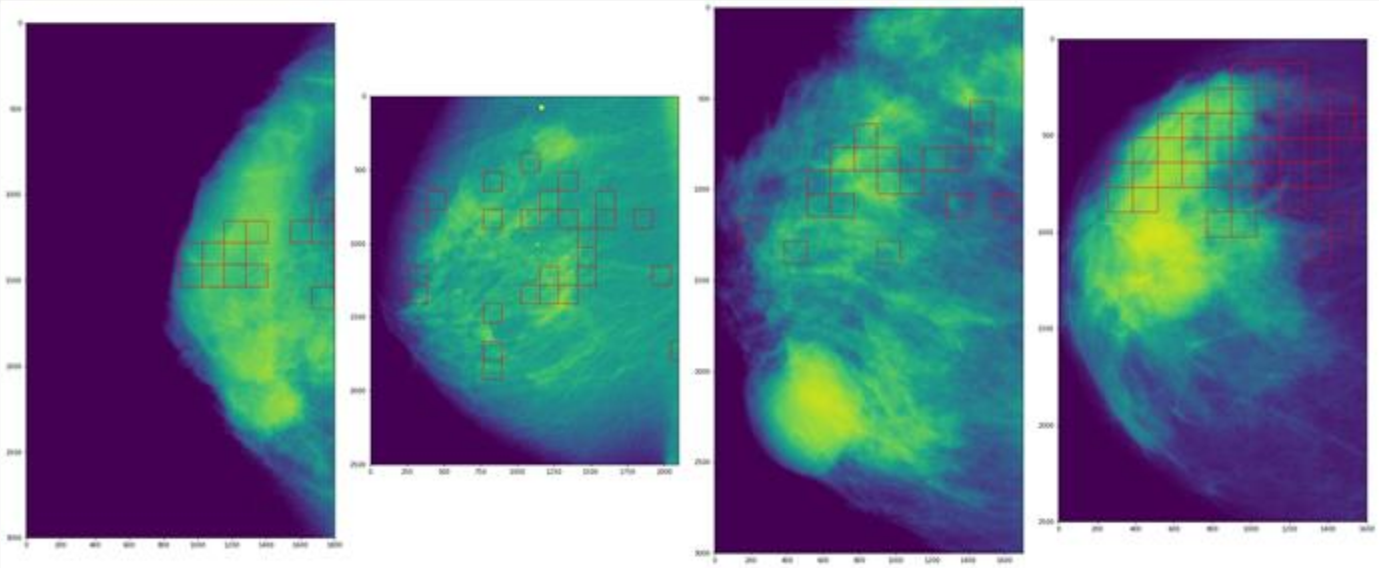


Fig. 2. Overlap-tile strategy for seamless segmentation of arbitrary large images (here segmentation of neuronal structures in EM stacks). Prediction of the segmentation in the yellow area, requires image data within the blue area as input. Missing input data is extrapolated by mirroring

Annotations





1) Crop Full Image to exclude non-natural high-density artifacts
2) Chop up into 128x128 subimages
3) Apply model and box if class=1

The DDSM dataset we used was provided in **TFRecords format**.

- Many machine learning frameworks have their own binary format, **TFRecords are the tensorflow binary format.**
- Binary formats
  - make better use of disk cache
  - allow faster data transfer
  - store different kinds of data (i.e. labels and images) in same place

https://keras.io/examples/mnist_tfrecord/

Tensor Processing Unit (TPU) as alternative to GPU

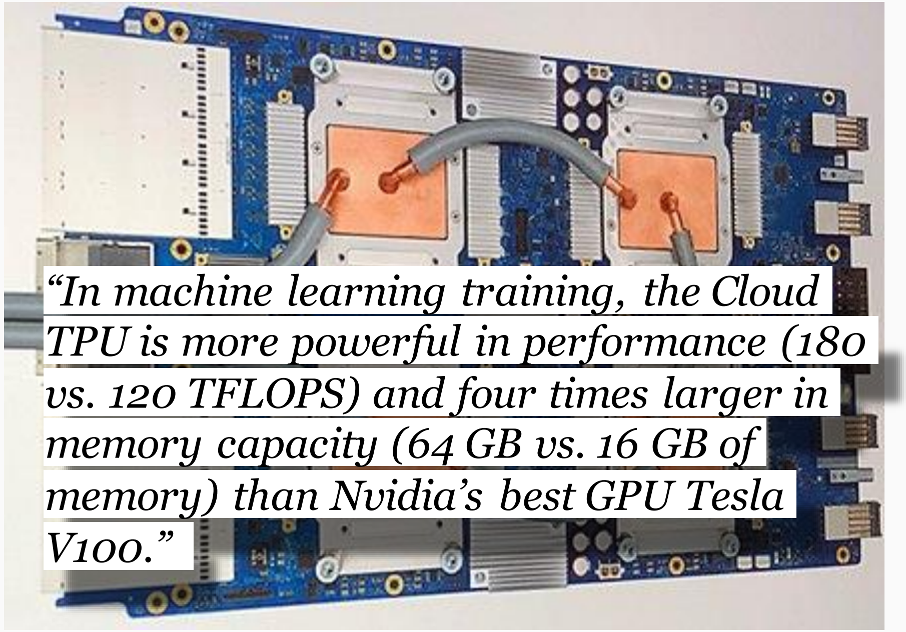
- **Application-specific integrated circuit (ASIC)** developed specifically for neural network machine learning

- **More powerful** in performance and memory than Nvidia's best GPU

MNIST on TPU or GPU using tf.Keras and tf.data.Dataset

https://colab.research.google.com/
github/tensorflow/tpu/

TensorFlow



"In machine learning training, the Cloud TPU is more powerful in performance (180 vs. 120 TFLOPS) and four times larger in memory capacity (64 GB vs. 16 GB of memory) than Nvidia's best GPU Tesla V100."

Model Comparison

|	Accuracy	F1 Score	Precision	Recall
VGG16	89.91 %	0.283	22.57 %	38.18 %
ResNet	93.97 %	0.227	34.52 %	16.92 %
U-Net	86.31	0.131	38.04 %	7.93 %
Transfer Learning	92.96%	0.033	21.55 %	7.65 %