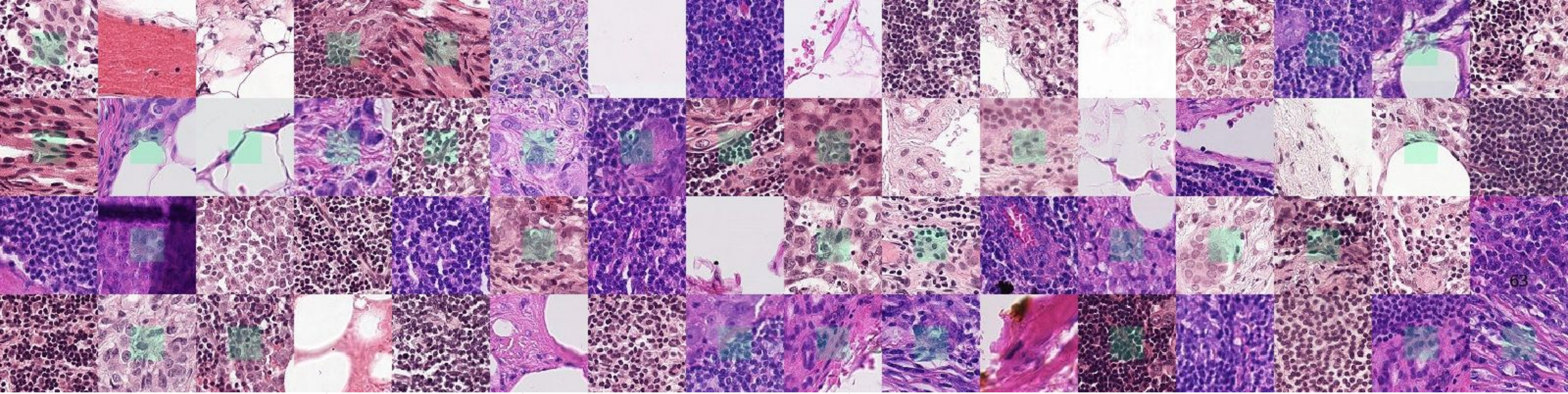


Histopathologic cancer detection

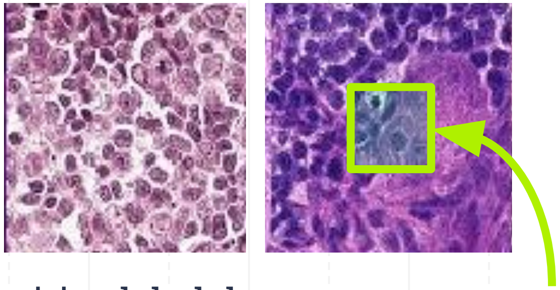
Capybara Team

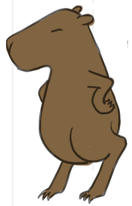
Andrea Lorenzon, Leonardo Stincone, Gabriele Sarti

DATA SCIENCE & SCIENTIFIC COMPUTING

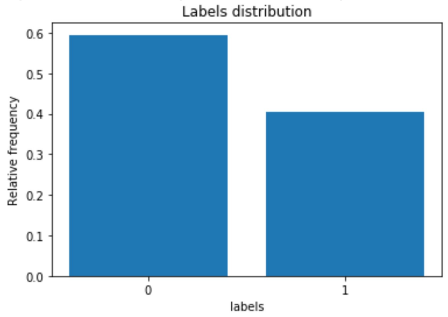
Introduction

- **Aim**: Identify metastatic infiltration in lymph nodes.
- **Data:** ~220.000 images, 96x96 px RGB, from PatchCamelyon (PCam) dataset.
  - 90 % used for training.
  - 10 % used for testing.
- **Problem:** Supervised image classification.



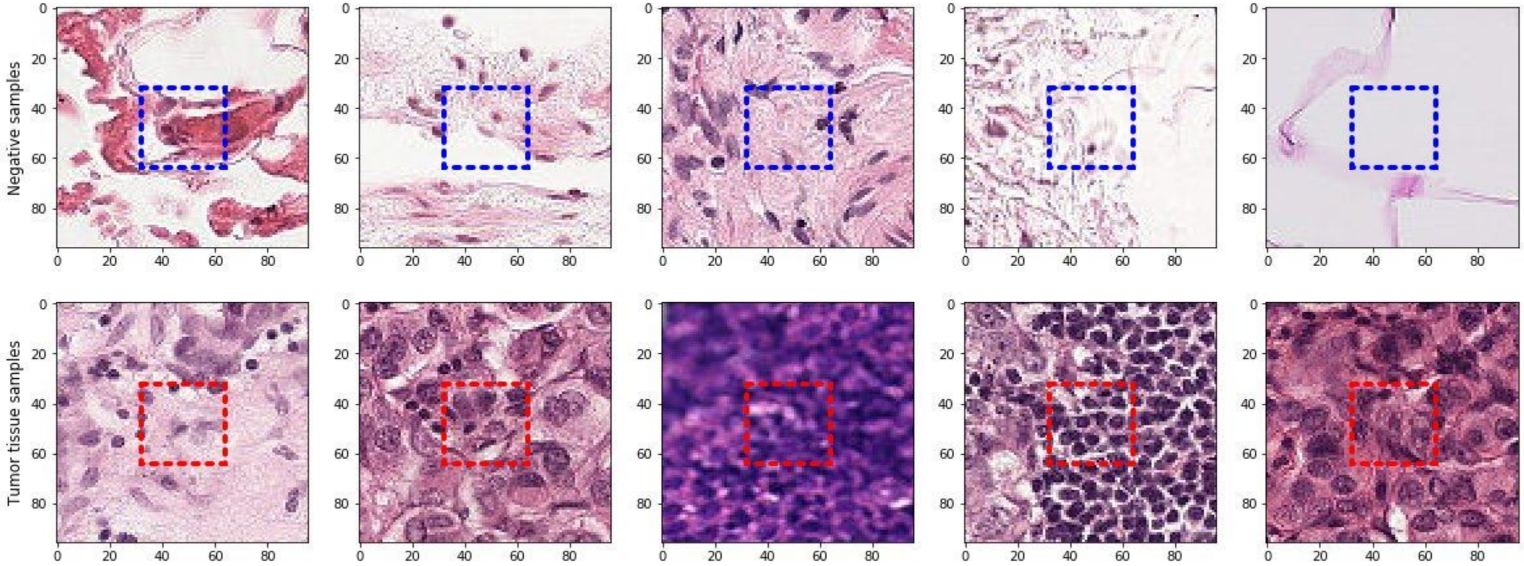


A positive label denotes cancer presence in the central 32x32 square



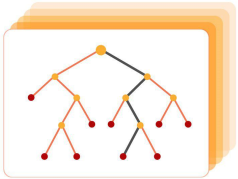
Examples

Low resolution, high within-group variance, artifacts, random blur...

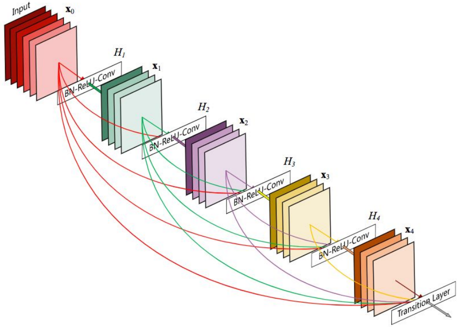


Approaches

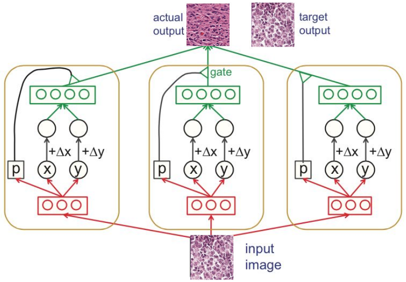
Unsupervised Segmentation and Random Forest



Pretrained DenseNet with FastAI



Deep/Shallow Capsule Network with Keras



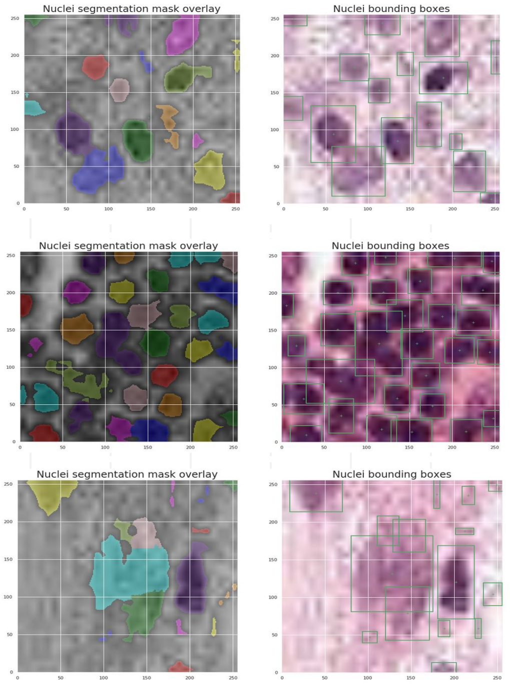
Segmentation and Random Forest

Classical approach to **nuclei segmentation**:

- ❖ Supervised, hand-made.
- ❖ Operator bias.
- ❖ Very time consuming.

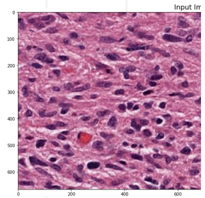
Random forest suits well our problem:

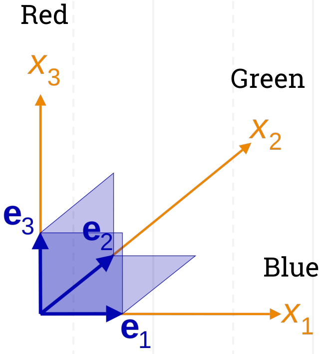
- ❖ General purpose classification algorithm.
- ❖ Requires few optimizable parameters.
- ❖ Robust to correlation between features.

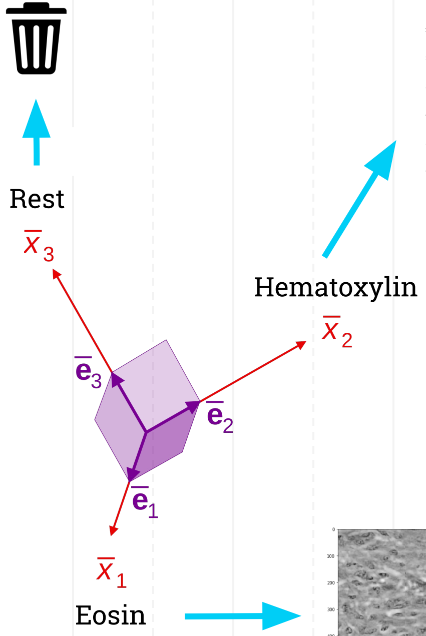


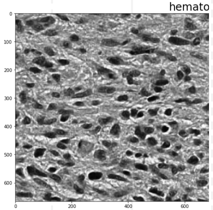
Segmentation Pipeline

Step 1
Center crop
32x32

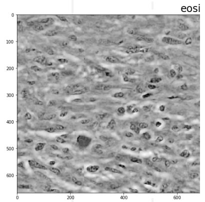


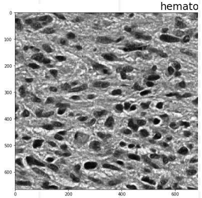




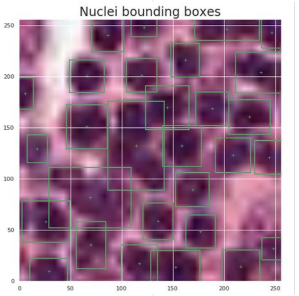


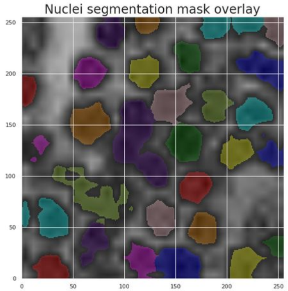
Step 2
Color deconvolution
in dye color space



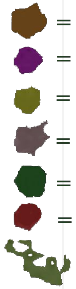


Step 3
Nuclei segmentation (local max clustering)



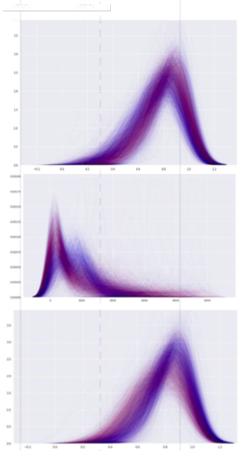


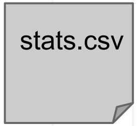
Step 4
Extract an array of attributes for each nucleus



= [eccentricity, area, solidity,...]
= [eccentricity, area, solidity,...]
= [eccentricity, area, solidity,...]
= [eccentricity, area, solidity,...]
= [eccentricity, area, solidity,...]
= [eccentricity, area, solidity,...]
= [eccentricity, area, solidity,...]

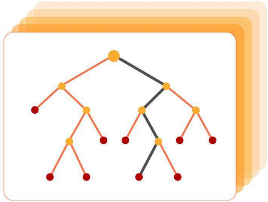
Step 5
Summarize attributes for each image with statistics



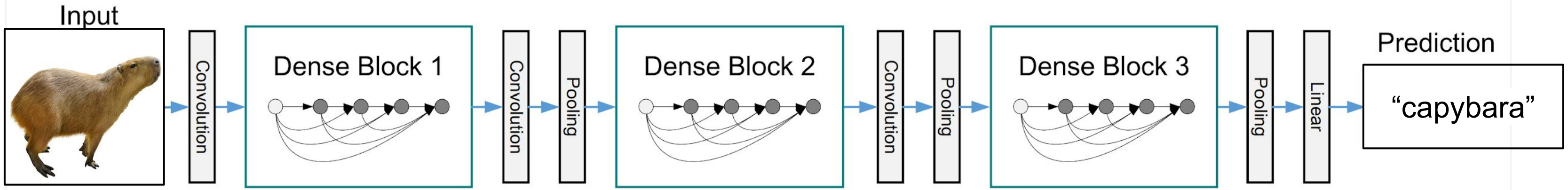


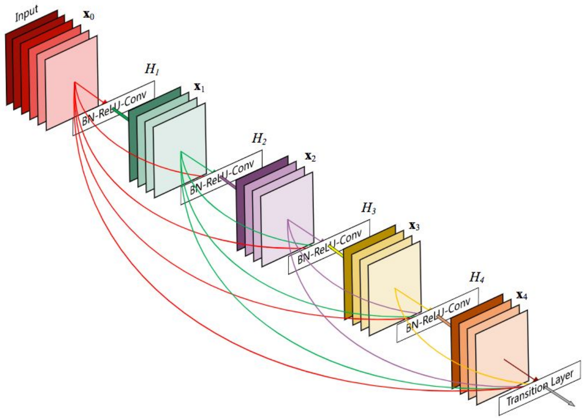
Step 6
Build the Random Forest

Random search for hyperparameter optimization



DenseNet



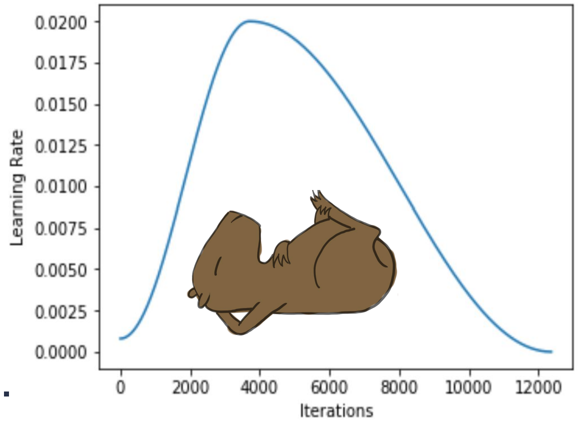


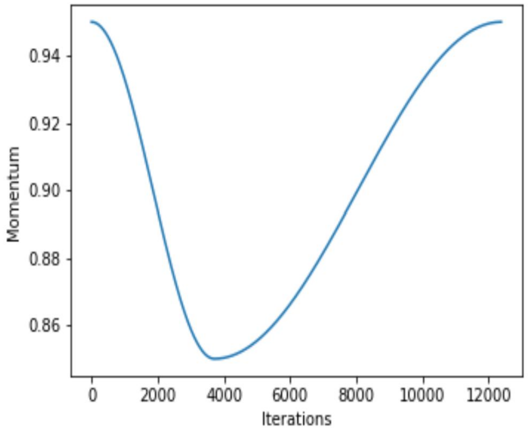
Layers	Output Size	DenseNet-121	DenseNet-169	DenseNet-201	DenseNet-264
Convolution	112 × 112	7 × 7 conv, stride 2			
Pooling	56 × 56	3 × 3 max pool, stride 2			
Dense Block (1)	56 × 56	[1 × 1 conv, 3 × 3 conv] × 6	[1 × 1 conv, 3 × 3 conv] × 6	[1 × 1 conv, 3 × 3 conv] × 6	[1 × 1 conv, 3 × 3 conv] × 6
Transition Layer (1)	56 × 56	1 × 1 conv			
	28 × 28	2 × 2 average pool, stride 2			
Dense Block (2)	28 × 28	[1 × 1 conv, 3 × 3 conv] × 12	[1 × 1 conv, 3 × 3 conv] × 12	[1 × 1 conv, 3 × 3 conv] × 12	[1 × 1 conv, 3 × 3 conv] × 12
Transition Layer (2)	28 × 28	1 × 1 conv			
	14 × 14	2 × 2 average pool, stride 2			
Dense Block (3)	14 × 14	[1 × 1 conv, 3 × 3 conv] × 24	[1 × 1 conv, 3 × 3 conv] × 32	[1 × 1 conv, 3 × 3 conv] × 48	[1 × 1 conv, 3 × 3 conv] × 64
Transition Layer (3)	14 × 14	1 × 1 conv			
	7 × 7	2 × 2 average pool, stride 2			
Dense Block (4)	7 × 7	[1 × 1 conv, 3 × 3 conv] × 16	[1 × 1 conv, 3 × 3 conv] × 32	[1 × 1 conv, 3 × 3 conv] × 32	[1 × 1 conv, 3 × 3 conv] × 48
Classification Layer	1 × 1	7 × 7 global average pool			
		1000D fully-connected, softmax			

Pretrained weights

1-Cycle Policy

- ❖ Hyper-parameters
  - ➢ Learning rate
  - ➢ Momentum
  - ➢ Weight decay
- ❖ Learning in 2 steps
  - ➢ Freeze first layers and fit only last one.
  - ➢ Unfreeze and fit all parameters.

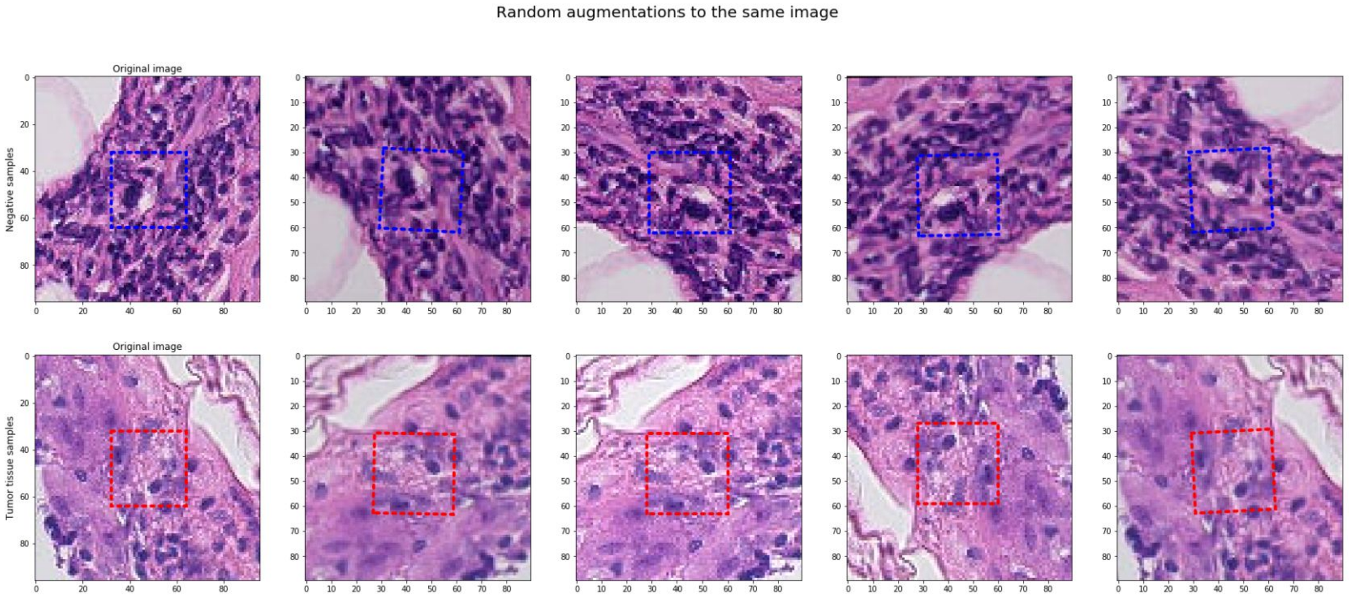




Data augmentation

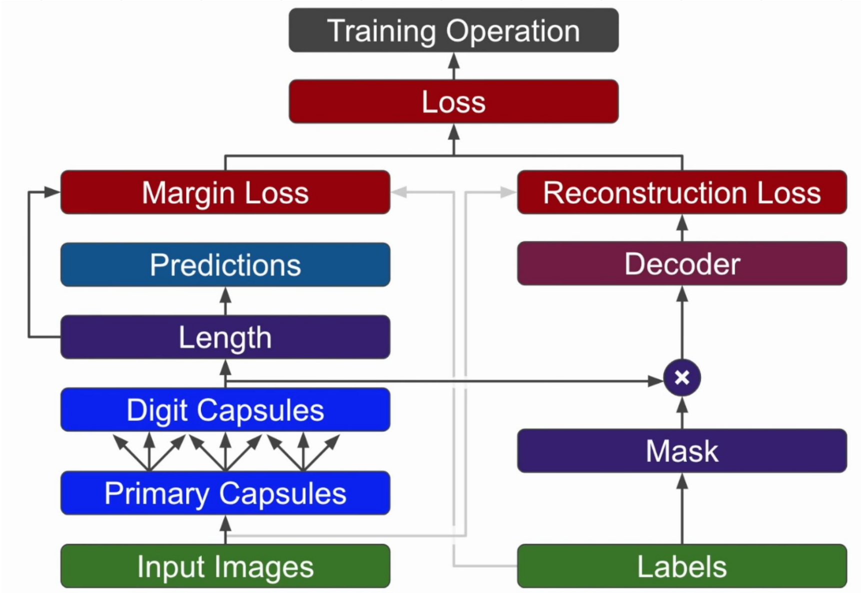
Augmentation performed:

- ❖ Flip
- ❖ Rotation
- ❖ Shift
- ❖ Brightness
- ❖ Contrast
- ❖ Cropping



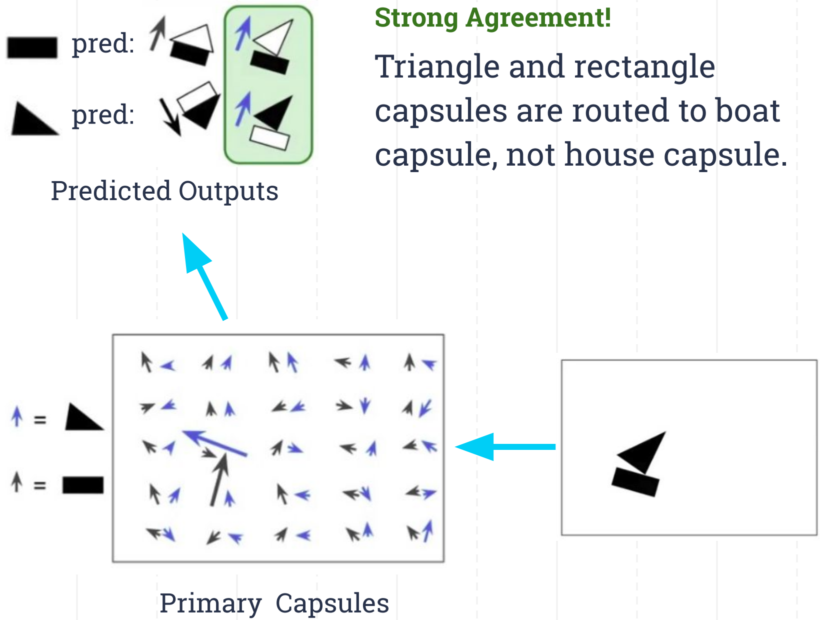
Capsule Network

- **Purpose:** Preserve spatial relations and achieve **equivariance**.
  - Different POV, different activation, same label.
- **Capsule layers** have vectorial input/output that are **squashed** to normalize their length.
- Great for segmentation and crowded scenes, but slow training.



Dynamic Routing

- **Routing by agreement:** Information propagated to the most relevant capsule, hierarchical tree of capsules.
- Part/whole pairs of transformation matrices are learned to distinguish inputs.
- Cleaner signal, reconstructible part hierarchy and parsing scenes.



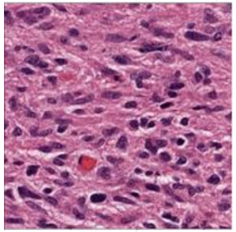
Strong Agreement!

Triangle and rectangle capsules are routed to boat capsule, not house capsule.

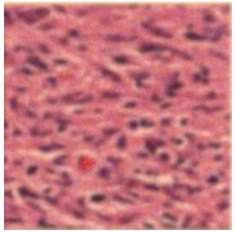
Decoder Network and Loss

- **Decoder:** 3 fully connected layers + 1 softmax.
  - Reconstruct input and computes squared difference between original and reconstructed input (**RecLoss**).
- **Loss:** Marginal loss + (α x RecLoss)
- **Result:** Low-level information is preserved to reconstruct the image up to the topmost layers.

Original Input

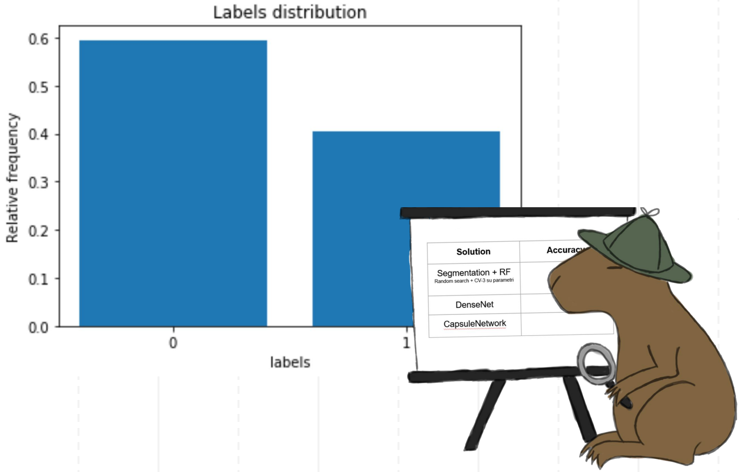


Reconstructed Input



Performance Assessment

- **Method:** Accuracy on test set.
- Test set taken from original training set
  - 90 % used for training.
  - 10 % used for testing.
- **Stratified sampling** for the test set to preserve the 60-40 label distribution.



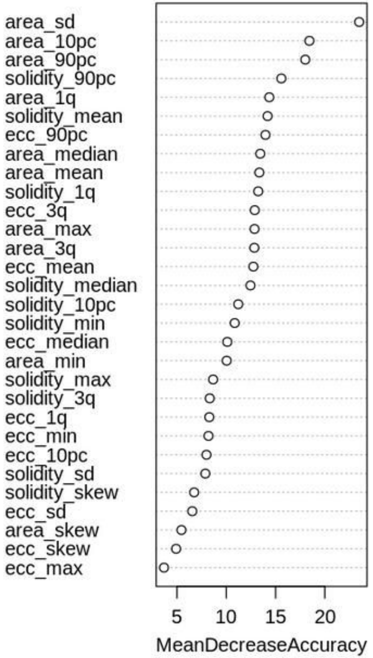
Results

Solution	Test Accuracy	Training Duration
Segmentation + Random Forest Random search + CV-3	71 %	6 + 6 hours On Kaggle 4-core server and an i5-5440@3.1GHz
DenseNet Pretrained with LR tuning	96%	2.5 hours On a Nvidia K80 server with pre-trained weights
Capsule Network Shallow with 1 Conv2D	82 %	4.5 days On an Azure K80 server (~60$)
Capsule Network Deep with 3 Conv2D	91 %	5.5 hours On an Azure K80 server

Discussion: Random Forest

Variable importance is supported by bibliography: **higher cell dimension variance** and **dysmorphism** are known to be correlated with cancer status, and commonly used by histopathologists as metrics.

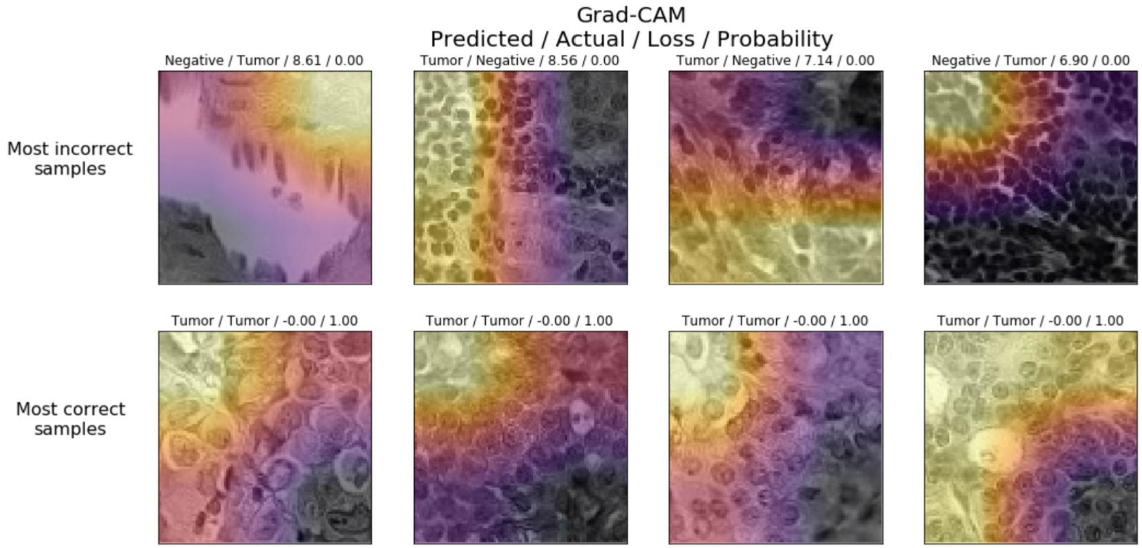
- Limitations:
  - Higher interpretability, but less predictive.
- Advantages:
  - Justification of variable importance.



Discussion: DenseNet

DenseNet strengths:

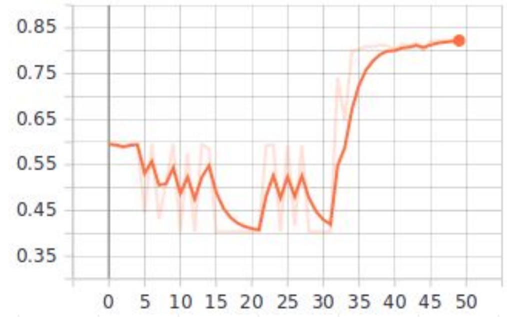
- ❖ Shortcut connections prevent vanishing gradient issues.
- ❖ Pre-trained weights.
- ❖ Gradient-weighted Class Activation Mapping for model checking.

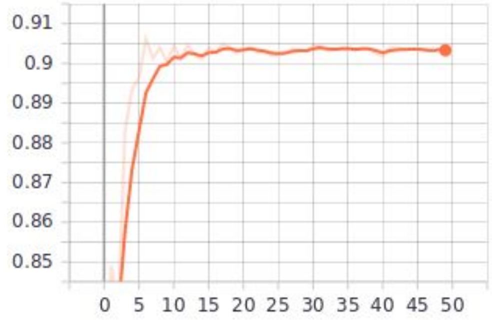


Discussion: Capsule Network

- Shallow net architecture was used for MNIST classification, **not deep enough** to grasp **problem complexity.**
- **Depth** and **pretraining** of DenseNet justify its superior performances with respect to the Capsule Networks.
- Potential to achieve even better scores for larger Capsule Networks.

Shallow
CapsNet
Test Accuracy





Deep CapsNet
Test Accuracy

Future Perspectives

- Use higher-resolution images for the training
- Segment nuclei from high res images and use the same pipeline to discriminate single cancer cells
- Analyze cancerous inputs reconstructed by a Capsule Network to retrieve insights about characters of interest.

Reproducibility

https://www.github.com/gsarti/cancer-detection

Bibliography (1)

❖ **Introduction**

- ➢ Challenge “**Histopathological Cancer Detection**”, Kaggle.
- ➢ B.S. Veeling, **PatchCamelyon Dataset**, Github.
- ➢ B. S. Veeling et al. "**Rotation Equivariant CNNs for Digital Pathology**", ArXiv.
- ➢ Ehteshami Bejnordi et al. “**Diagnostic Assessment of Deep Learning Algorithms for Detection of Lymph Node Metastases in Women With Breast Cancer**”. JAMA: The Journal of the American Medical Association, 2199–2210.

❖ **Unsupervised Segmentation and Random Forest**

- ➢ J. Diamond, N. H. Anderson, P. H. Bartels, R. Montironi, and P. W. Hamilton, “**The use of morphological characteristics and texture analysis in the identification of tissue composition in prostatic neoplasia,**” Human Pathology
- ➢ S. Doyle, M. Hwang, K. Shah, A. Madabhushi, M. Feldman, and J. Tomaszeweski, “**Automated grading of prostate cancer using architectural and textural image features**,” in Proceedings of the 4th IEEE International Symposium on Biomedical Imaging: From Nano to Macro
- ➢ **HistomicsTK**, an unsupervised cell microscopy segmentation library for Python

Bibliography (2)

- **DenseNet**
    - G. Huang, Z. Liu, L. van der Maaten, K. Q. Weinberger, "**Densely Connected Convolutional Networks**", 2018.
    - Leslie N. Smith, "**A disciplined approach to neural network hyper-parameters**", 2018.
    - Sylvain Gugger, "**The 1 cycle policy**", 2018.
    - R. R. Selvaraju et al. "**Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization**", 2017.
    - Jeremy Howard, "**Practical Deep Learning for Coders, v3**", 2019.

- **Capsule Networks**
    - S. Sabour, N. Frosst, G. E. Hinton, "**Dynamic Routing Between Capsules**", In Proceedings of NeurIPS 2017.
    - A. Géron, "**Capsule Networks Tutorial**", "**How to Implement CapsNets in Tensorflow**", Youtube, accessed June 20, 2019.
    - M. Pechyonkin "**Understanding Hinton's Capsule Networks**", Medium series, accessed June 24th, 2019.
    - T. Iesmantas et al., "**Convolutional capsule network for classification of breast cancer histology images**", ArXiv.
    - V. Davydov, "**GPU + Azure + Deep Learning with minimum pain**", Medium post, accessed June 16th, 2019.
    - "**Keras Example: Capsule Network on CIFAR10**", Github.