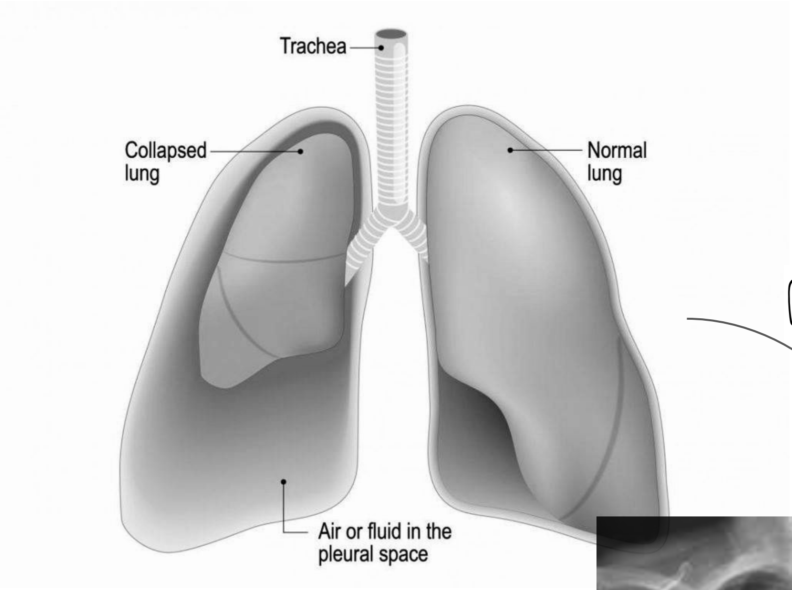
Chest X-Ray Pneumothorax images segmentation

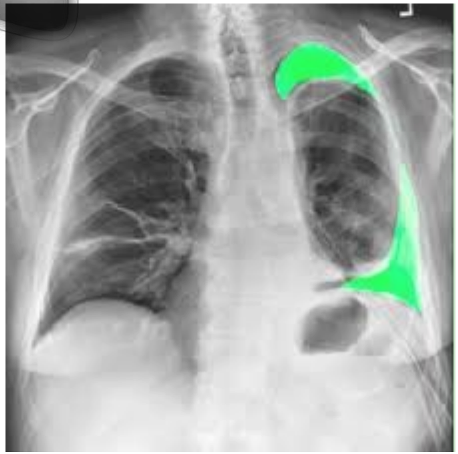
by Aleksandra Shchetinina (a.shchetinina@innopolis.ru)
& Anastasia Pichka (a.pichka@innopolis.ru)
BS17-RO

CV: Check Point, Fall 2020

Intro

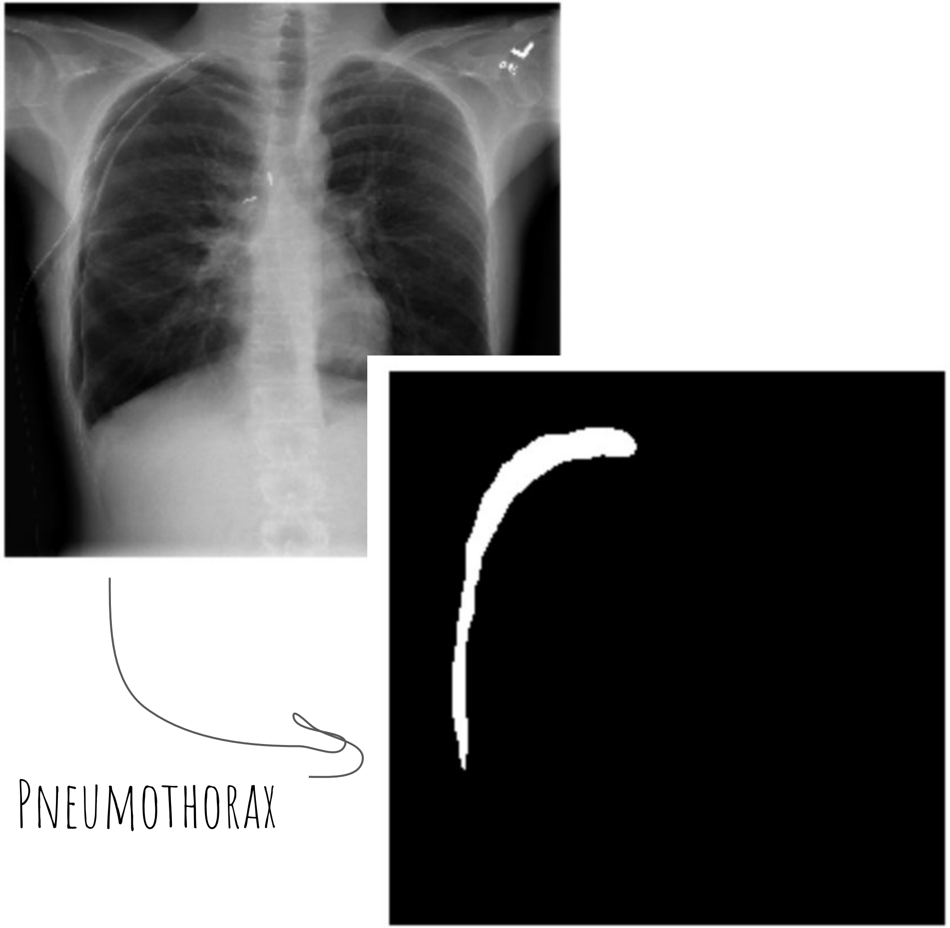


Chest X-Ray



Pneumothorax

Collapse of the lung, occurs when the chest and lungs are injured, as well as certain lung conditions (eg, tuberculosis, pneumonia)



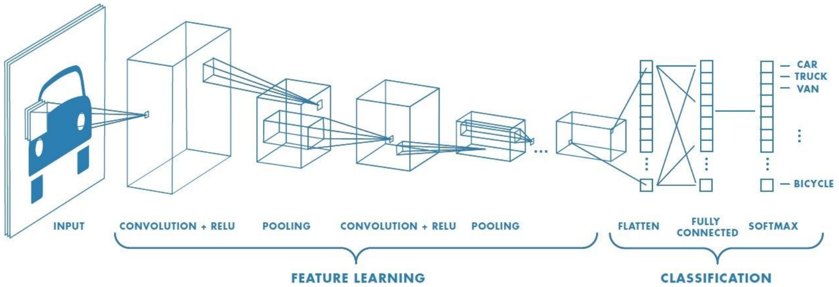
Dataset

12,047 chest x-ray images
12,047 pneumothorax masks

Size - 1024x1024, in png format

Segmentation

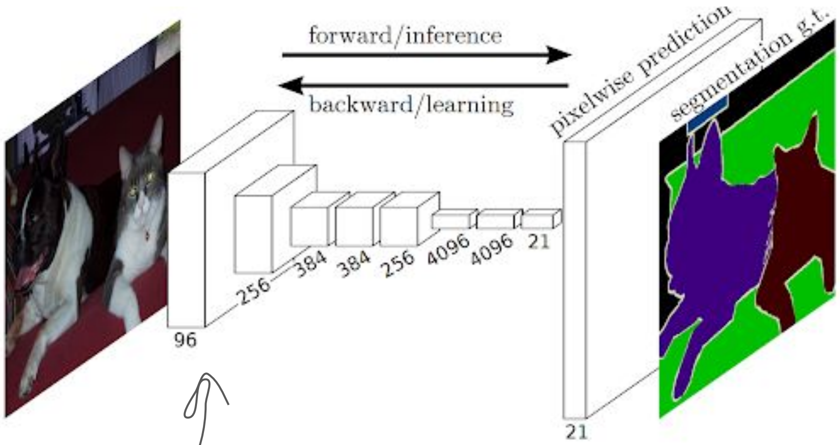
Process of partitioning a digital image into multiple segments (sets of pixels, also known as image objects)



CNN

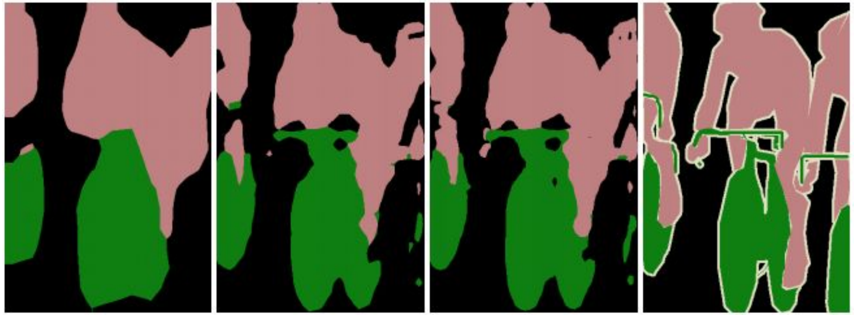
FCN - Fully Convolutional Network

Last fully connected layer of such network was replaced with a fully convolutional layer, produce a dense pixel-wise prediction



FCN

Related Work



	FCN-AlexNet	FCN-VGG16	FCN-GoogLeNet[4]
mean IU	39.8	**56.0**	42.5
forward time	50 ms	210 ms	59 ms
conv. layers	8	16	22
parameters	57M	134M	6M
rf size	355	404	907
max stride	32	32	32

Fully Convolutional Networks for Semantic Segmentation

Jonathan Long, Evan Shelhamer, Trevor Darrell

8 Mar 2015

AlexNet, VGG, GoogLeNet can be adopted using transfer learning by fine-tuning and used for segmentation

Idea Explanation

	new_filename	ImageId	has_pneumo
0	0_train_0_.png	1.2.276.0.7230010.3.1.4.8323329.5597.151787518...	0
1	1_train_0_.png	1.2.276.0.7230010.3.1.4.8323329.12515.15178752...	0
2	2_train_1_.png	1.2.276.0.7230010.3.1.4.8323329.4904.151787518...	1
3	3_train_1_.png	1.2.276.0.7230010.3.1.4.8323329.32579.15178751...	1
4	4_train_1_.png	1.2.276.0.7230010.3.1.4.8323329.1314.151787516...	1
5	5_train_0_.png	1.2.276.0.7230010.3.1.4.8323329.11364.15178752...	0
6	6_train_0_.png	1.2.276.0.7230010.3.1.4.8323329.4541.151787518...	0
7	7_train_1_.png	1.2.276.0.7230010.3.1.4.8323329.4440.151787518...	1
8	8_train_1_.png	1.2.276.0.7230010.3.1.4.8323329.4982.151787518...	1
9	9_train_0_.png	1.2.276.0.7230010.3.1.4.8323329.31759.15178751...	0
10	10_train_1_.png	1.2.276.0.7230010.3.1.4.8323329.12743.15178752...	1
11	11_train_1_.png	1.2.276.0.7230010.3.1.4.8323329.11633.15178752...	1
12	12_train_0_.png	1.2.276.0.7230010.3.1.4.8323329.11512.15178752...	0
13	13_train_1_.png	1.2.276.0.7230010.3.1.4.8323329.2663.151787517...	1
14	14_train_0_.png	1.2.276.0.7230010.3.1.4.8323329.10206.15178752...	0

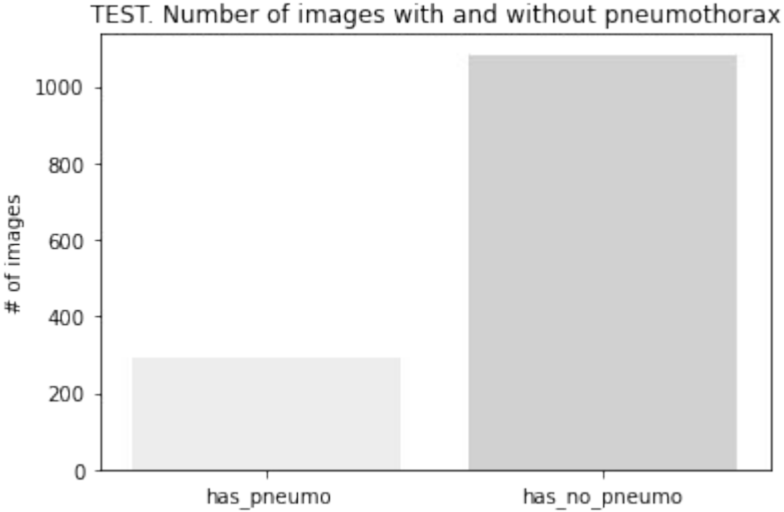
Data Preparation

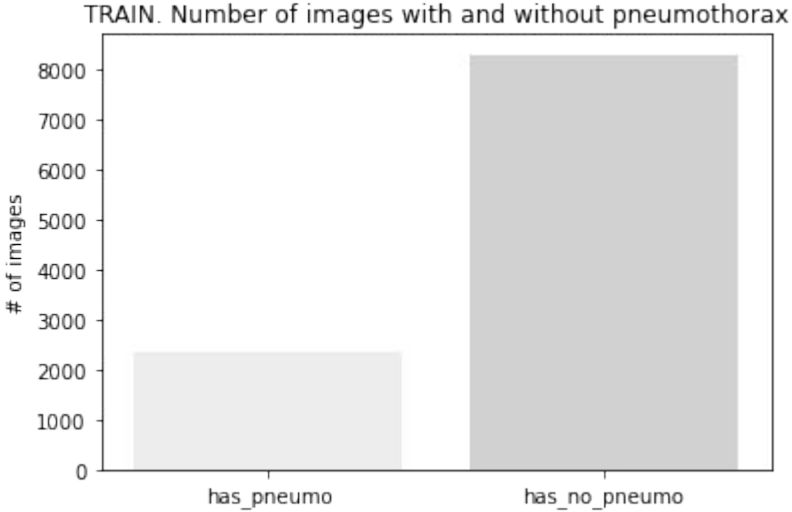
The data is taken from the dataset and prepared (normalized, resized, divided into test and train, etc). Masks (segmented image) is also prepared

Imbalanced data

Test

Train





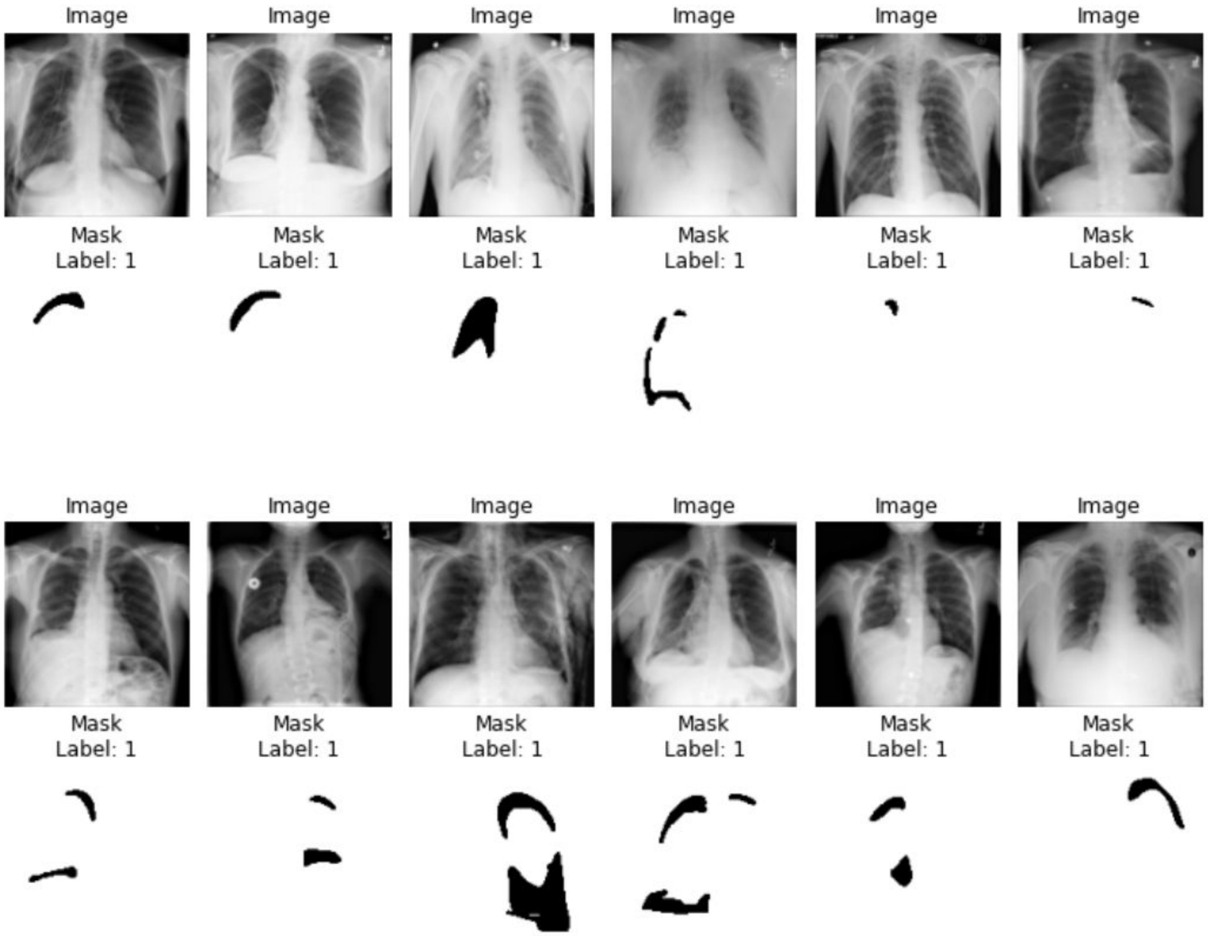
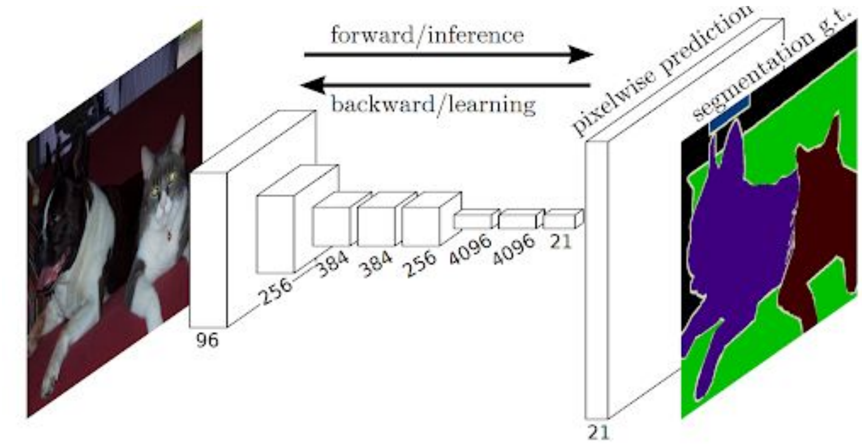
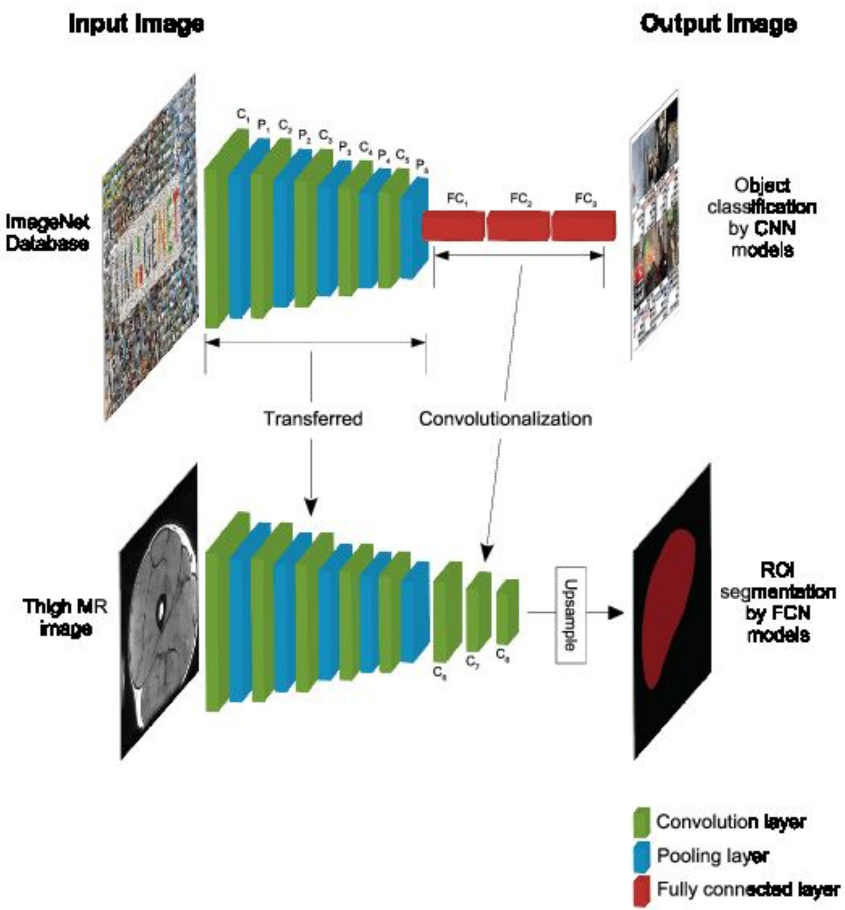


Image
Image
Image
Image
Image
Image
Mask
Label: 1
Mask
Label: 1
Mask
Label: 1
Mask
Label: 1
Mask
Label: 1
Mask
Label: 1
Image
Image
Image
Image
Image
Image
Mask
Label: 1
Mask
Label: 1
Mask
Label: 1
Mask
Label: 1
Mask
Label: 1
Mask
Label: 1

Model

The basic idea behind a fully convolutional network is that it is “fully convolutional”, so all of its layers are convolutional layers. FCNs don’t have any of the fully-connected layers at the end, which are typically used for classification. Instead, FCNs use convolutional layers to classify each pixel in the image.





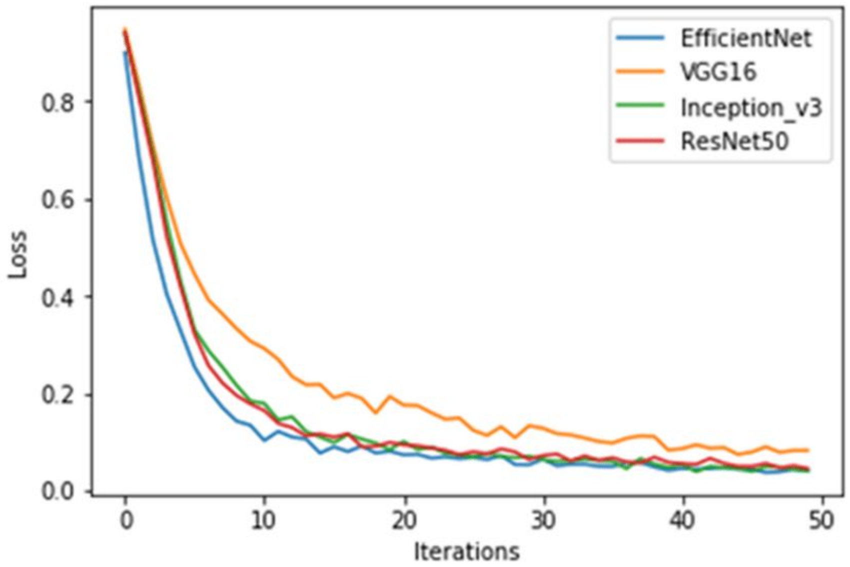
Transfer Learning

Research problem in machine learning (ML) that focuses on storing knowledge gained while solving one problem and applying it to a different but related problem

Architecture

Architecture:
• Pre-trained classifier without fully-connected layers (transferred, frozen)
• Convolution layers
• Upsampling(Deconvolution)
• Softmax layer

→ VGG
→ ResNet
→ InceptionV3



VGG

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
vgg19 (Functional)           (None, 4, 4, 512)         20024384
_________________________________________________________________
conv2d_transpose (Conv2DTran (None, 16, 16, 2)         262146
_________________________________________________________________
dropout (Dropout)            (None, 16, 16, 2)         0
_________________________________________________________________
conv2d_transpose_1 (Conv2DTr (None, 64, 64, 2)         4098
_________________________________________________________________
dropout_1 (Dropout)          (None, 64, 64, 2)         0
_________________________________________________________________
conv2d_transpose_2 (Conv2DTr (None, 128, 128, 2)       16386
_________________________________________________________________
activation (Activation)      (None, 128, 128, 2)       0
=================================================================
Total params: 20,307,014
Trainable params: 282,630
Non-trainable params: 20,024,384
_________________________________________________________________
```

ResNet

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
resnet50 (Model)             (None, 4, 4, 2048)        23587712
_________________________________________________________________
conv2d_transpose_14 (Conv2DT (None, 8, 8, 2)           65538
_________________________________________________________________
dropout_10 (Dropout)         (None, 8, 8, 2)           0
_________________________________________________________________
conv2d_transpose_15 (Conv2DT (None, 16, 16, 2)         1026
_________________________________________________________________
dropout_11 (Dropout)         (None, 16, 16, 2)         0
_________________________________________________________________
conv2d_transpose_16 (Conv2DT (None, 32, 32, 2)         4098
_________________________________________________________________
dropout_12 (Dropout)         (None, 32, 32, 2)         0
_________________________________________________________________
conv2d_transpose_17 (Conv2DT (None, 64, 64, 2)         4098
_________________________________________________________________
dropout_13 (Dropout)         (None, 64, 64, 2)         0
_________________________________________________________________
conv2d_transpose_18 (Conv2DT (None, 128, 128, 2)       16386
_________________________________________________________________
activation_4 (Activation)    (None, 128, 128, 2)       0
=================================================================
Total params: 23,678,858
Trainable params: 91,146
Non-trainable params: 23,587,712
_________________________________________________________________
```

InceptionV3

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
inception_v3 (Model)         (None, 2, 2, 2048)        21802784
_________________________________________________________________
conv2d_transpose_27 (Conv2DT (None, 8, 8, 2)           1048578
_________________________________________________________________
dropout_15 (Dropout)         (None, 8, 8, 2)           0
_________________________________________________________________
conv2d_transpose_28 (Conv2DT (None, 32, 32, 2)         4098
_________________________________________________________________
dropout_16 (Dropout)         (None, 32, 32, 2)         0
_________________________________________________________________
conv2d_transpose_29 (Conv2DT (None, 64, 64, 2)         4098
_________________________________________________________________
dropout_17 (Dropout)         (None, 64, 64, 2)         0
_________________________________________________________________
conv2d_transpose_30 (Conv2DT (None, 128, 128, 2)       16386
_________________________________________________________________
activation_289 (Activation)  (None, 128, 128, 2)       0
=================================================================
Total params: 22,875,944
Trainable params: 1,073,160
Non-trainable params: 21,802,784
_________________________________________________________________
```

Accuracy

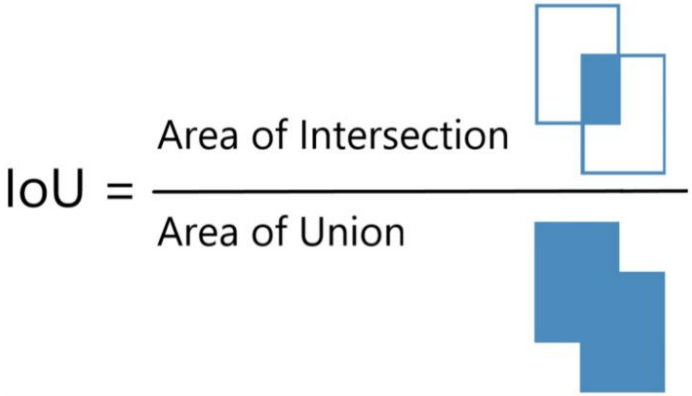
Pixel accuracy is conceptually the easiest metric. It is the percent of pixels in image that are classified correctly. The pixel accuracy is commonly reported for each class separately as well as globally across all classes. Also, this metric can be calculated through binary mask (positives and negatives):

$$\text{accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$$

Mean Intersection over Union

Mean IoU is a common evaluation metric for semantic segmentation. Firstly, it computes the IOU for each semantic class and then computes the average over classes. Formula for IOU is the following:

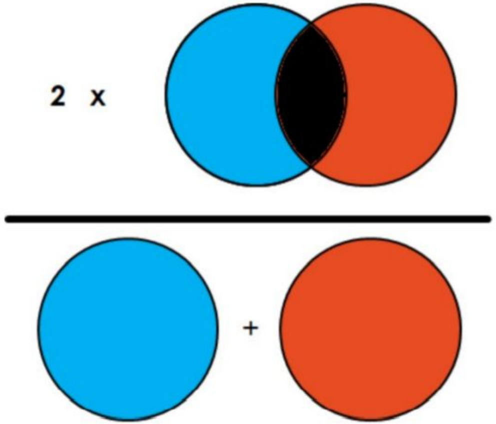
$$\text{IOU} = \frac{truepositive}{(truepositive + falsepositive + falsenegative)}$$



Dice coefficient

Dice Coefficient is the area of overlap multiplied by 2 and divided by the total number of pixels in both images. It ranges from 0 to 1, 0 means the smallest similarities between predicted and true, and 1 means the largest.

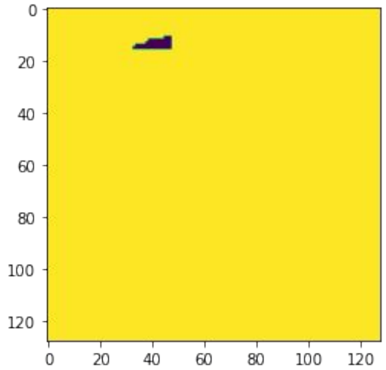
$$\frac{2*\text{area of overlap}}{\text{total number of pixels}}$$



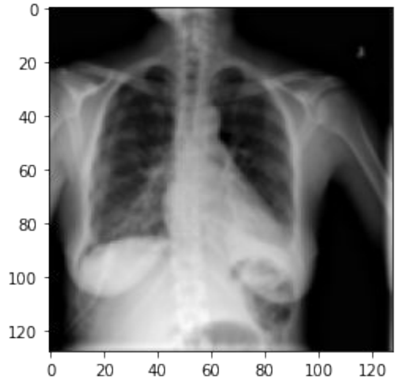
Results

Results, we've got

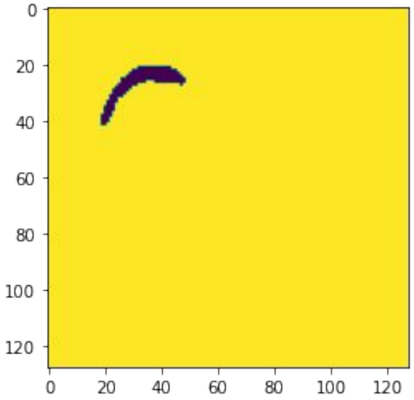
after some training



X_TEST



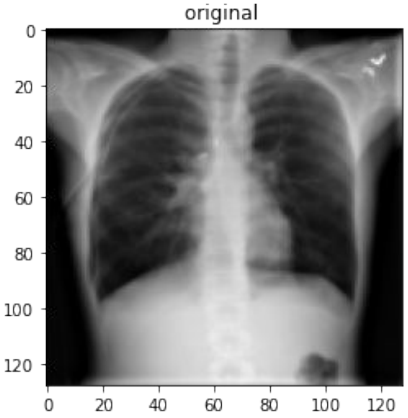
Y_TEST

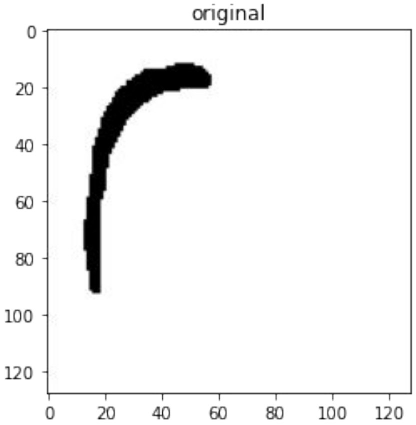


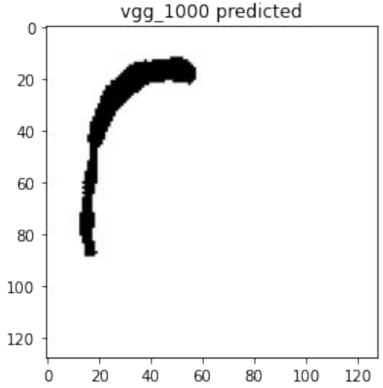
Example#1

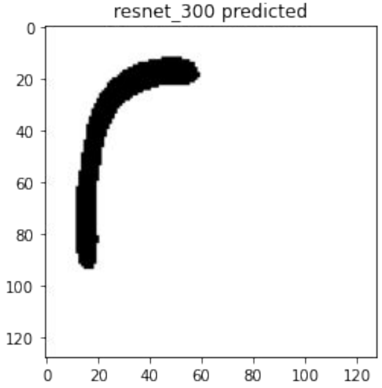
FROM TRAIN

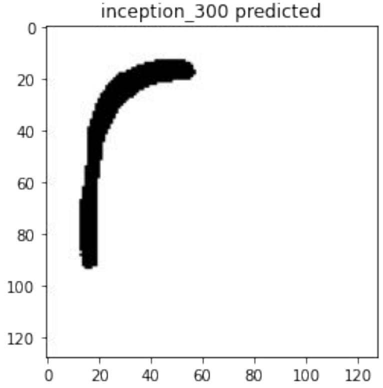
Results







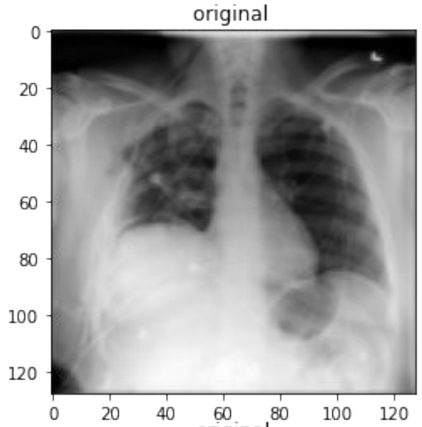


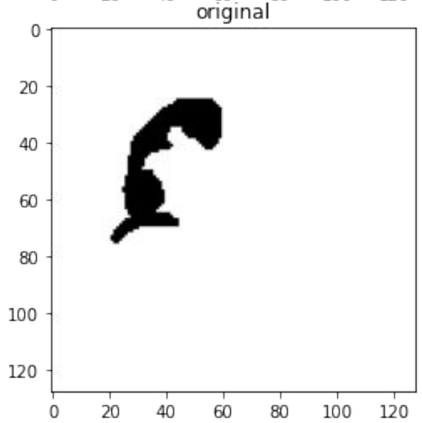


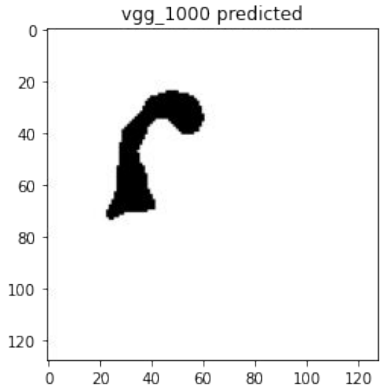
Example#2

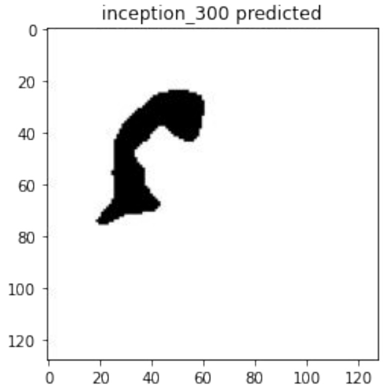
FROM TRAIN

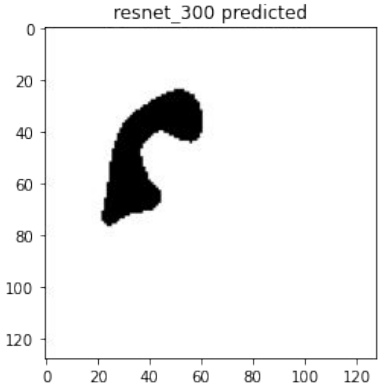
Results







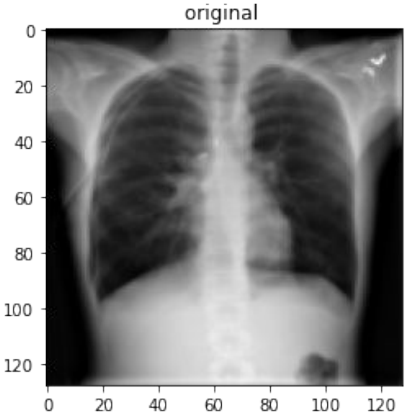


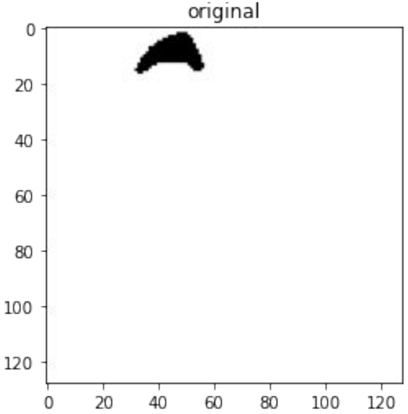


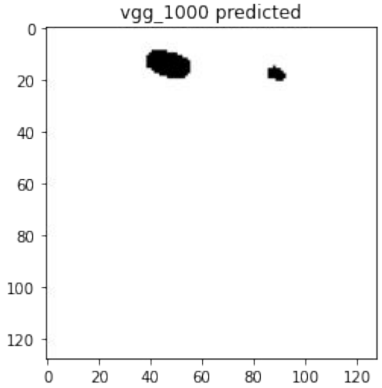
Results

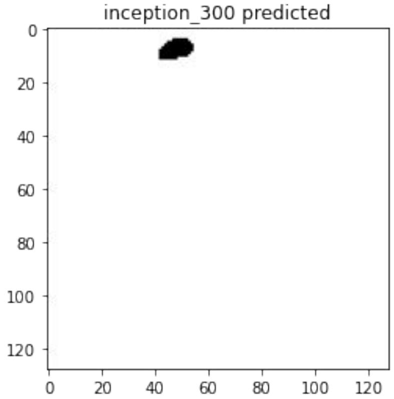
Example#1

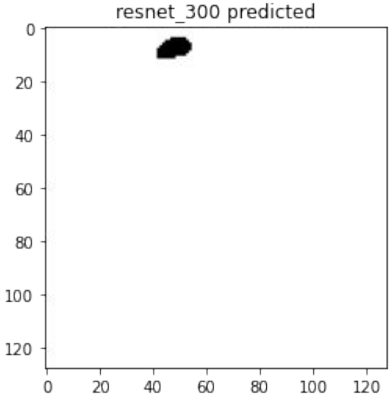
FROM TEST







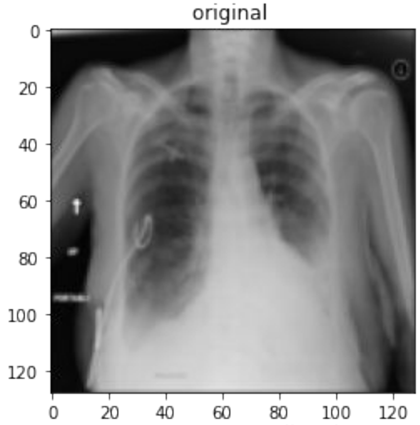


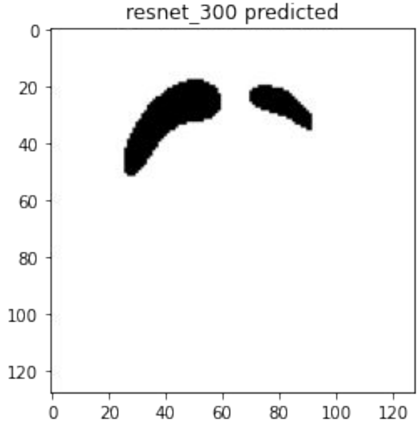


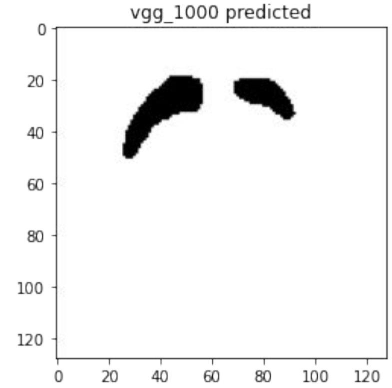
Example#2

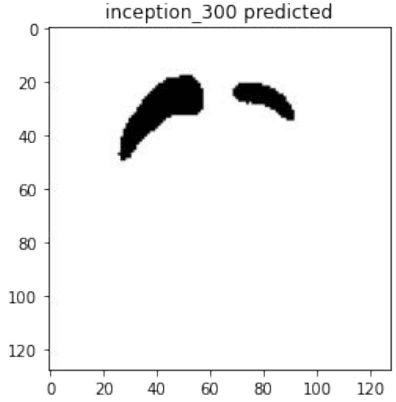
FROM TEST

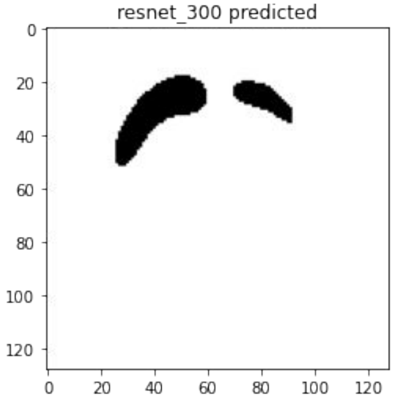
Results











Accuracy

During training and testing we understood that "accuracy" and "dice" are always high

Mean IoU

Epoch 50/100

483/483 [==============================] - 13s 28ms/step - loss: 0.0444 - accuracy: 0.9840 - mean_io_u: 0.2523 - dice: -0.9764 - val_loss: 0.0160 - val_accuracy: 0.9954 - val_mean_io_u: 0.2509 - val_dice: -0.9881

Results

50 100 300

	train	val	test
VGG19_300	0.3012	0.3054	0.3190
ResNet50_300	0.4102	0.4109	0.4302
InsNet_300	0.4409	0.4501	0.4514

Mean IoU

Conclusions

- Transfer learning and FCN can be used for segmentation
- We can continue training to get even better results
- Also, we can use different configurations of FCN, different loss, optimizers, etc

Thank you