Attention U-Net: Looking for Where to look for the Pancreas

[Ozan Oktay, Jo Schlemper, Loic Le Folgoc, Matthew Lee, Mattias Heinrich, Kazunari Misawa, Kensaku Mori, Steven McDonagh, Nils Y Hammerla, Bernhard Kainz, Ben Glocker, and Daniel Rueckert]

Summarized by:

Raaghav Radhakrishnan – 246097
Data Analytics II – 07.05.2019

Overview

1. Introduction
2. Related Work
3. Problem Definition
4. Baseline Methods
5. Attention U – Net
6. Experiment and Results
7. Future Work
8. Conclusion

1. Introduction

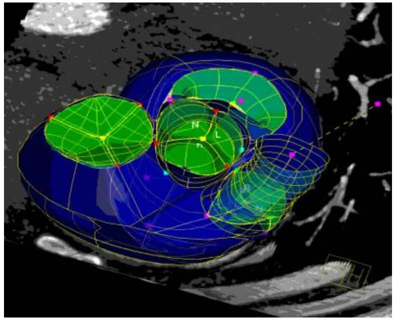
Image Segmentation for Medical Imaging

Introduction

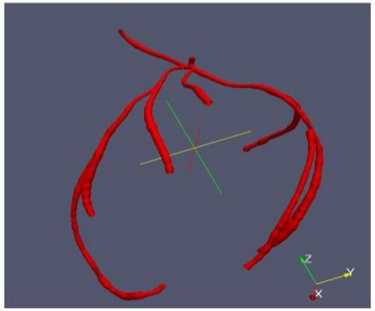
- Medical Image Segmentation is the process of automatic or semi-automatic detection of boundaries
- It is extensively studied because manual, dense labelling large amounts of data is a tedious and error prone task
- A major difficulty is the high variability in medical images
- Possible applications are automatic detection/measurement of organs, cell counting, or simulations based on the extracted boundary information

Introduction

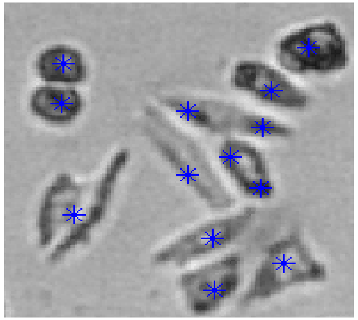
Heart Vessels and Valves



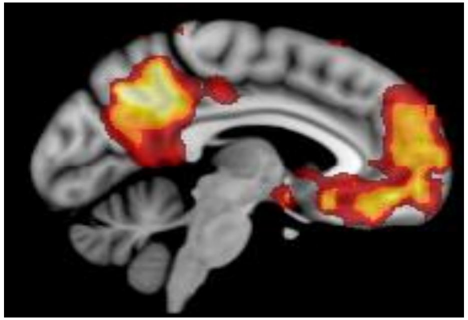
Coronary Vessel



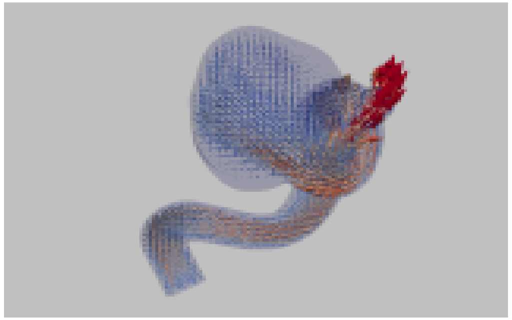
Cell Detection



Resting state Magnetic Resonance



Vessel Seg. And Blood Flow



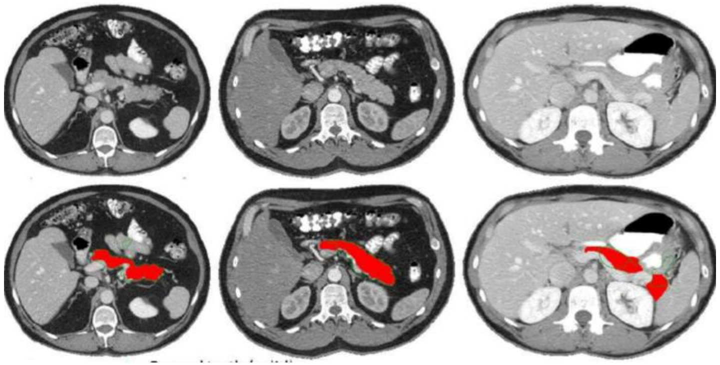
2. Related Work

State-of-the-art CT pancreas segmentation and Attention methods

Related Work

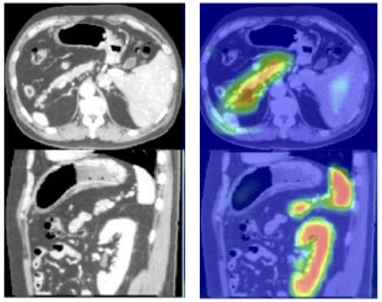
CT Pancreas Segmentation

- U-Net
- Hierarchical 3D FCN
- Dense – Dilated FCN
- Holistically Nested FCN
- FCN 2D
- Single and Multi-Model 2D FCN



Attention Gates

- Learn to Pay Attention
- Attention in CNNs



Related Work

CT Pancreas Segmentation

- U-Net
- Hierarchical 3D FCN
- Dense – Dilated FCN
- Holistically Nested FCN
- FCN 2D
- Single and Multi-Model 2D FCN

Baseline Methods

Attention Gates

- Learn to Pay Attention
- Attention in CNNs

3. Problem Definition

Problem Definition

FCNs and the U-Net rely on multi-cascaded CNNs:

- Extract an ROI and make dense predictions on it
- Excessive and redundant use of computational resources and model parameters
- Similar low-level features are repeatedly extracted by models
- Use of explicit external tissue/organ localisation modules
- Difficult to reduce false-positive predictions for small objects that show large shape variability

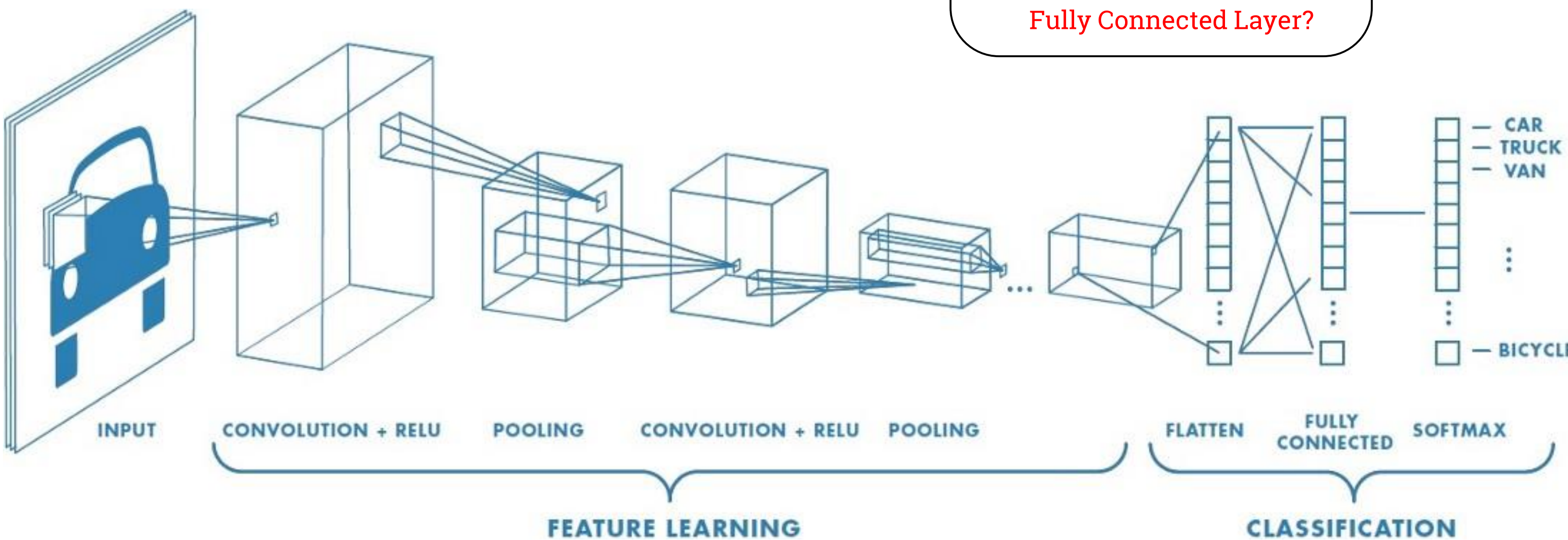
4. Baseline Methods

Basic Concepts, U – Net and Attention Gates

Basic Concepts

What is

- Kernel?
- Convolution Layer?
- Pooling and Up sampling?
- Loss Function?
- Fully Connected Layer?



Neural Network Architecture

Basic Concepts

Kernel

- A small matrix
- Used for blurring, edge detection, sharpening and more
- Accomplished by convolving kernel over image

$$\begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix}$$

Edge Detection

$$\begin{pmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{pmatrix}$$

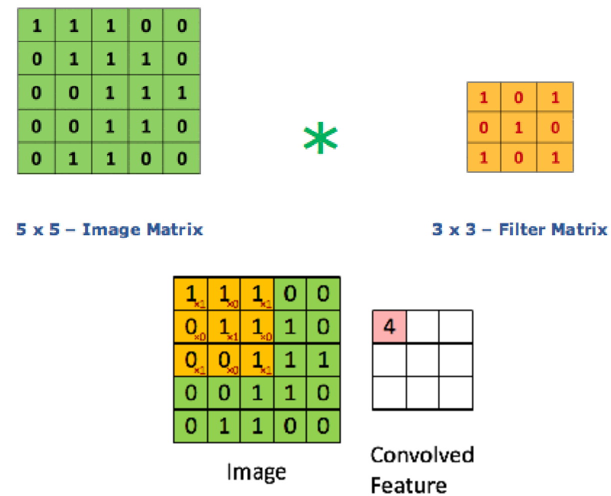
Sharpening

$$\begin{pmatrix} 1/16 & 1/8 & 1/16 \\ 1/8 & 1/4 & 1/8 \\ 1/16 & 1/8 & 1/16 \end{pmatrix}$$

Gaussian Blur (3x3)

Convolution Layer

- Layer to extract features from an input image
- Preserves the relationship between pixels by learning image features
- Multiplication of image matrix with kernel



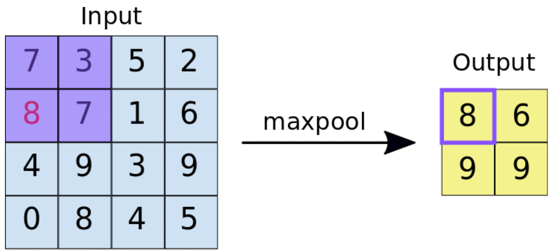
Basic Concepts

Pooling – Down Sampling

- Reduce the number of parameters
- Reduces the dimensionality but retains important information

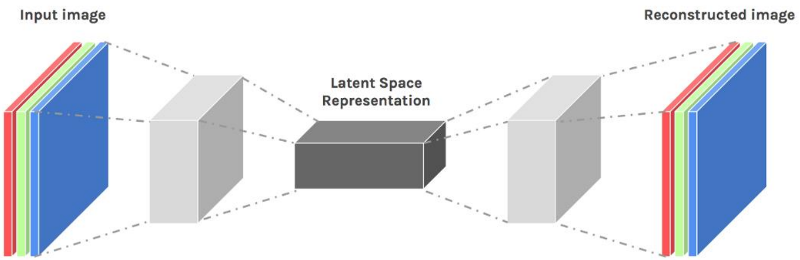
Types

- Max Pooling
- Average Pooling and more

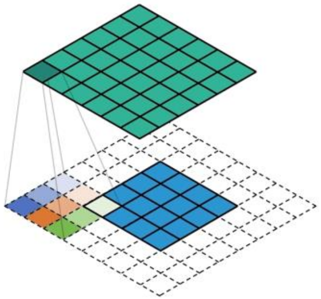


Up sampling – Transposed Convolution

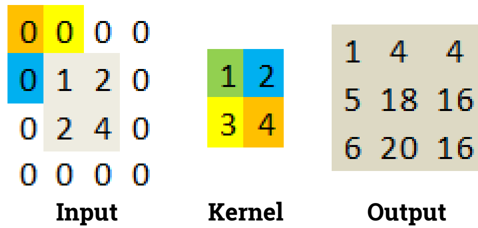
- To decompress abstracted representation into a different domain
- Example: Semantic Segmentation



Auto-encoding an RGB Image



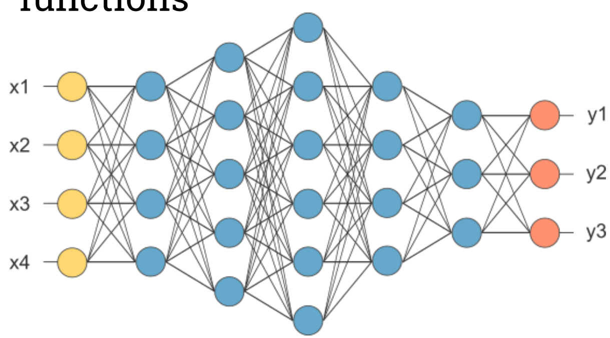
Transposed Convolution



Basic Concepts

Fully Connected Layer

- Converts feature map to vector of features and combines them to create a model and uses activation functions



$$f(x) = \max(0, x)$$

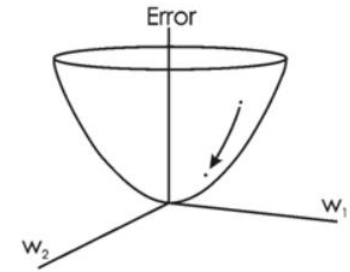
ReLU

$$f(x) = \frac{1}{1 + e^{-x}}$$

Sigmoid

Loss Function

- Evaluates how well an algorithm models the given data
- A measure of how good a prediction model predicts the expected target



- Through Back Propagation Weights has to be adjusted in such a way that the error decreases

$$L = \sum_{i=1}^{n} \frac{1}{2} (target \; - output)^2 \qquad w_{new} = \mathrm{w}_{\mathrm{old}} - \eta . \frac{\partial L}{\partial w}$$

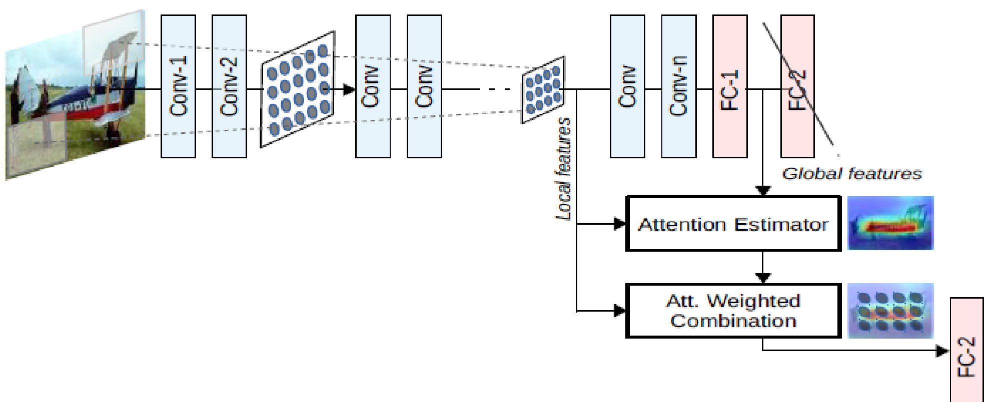
Attention Gates

- A scalar matrix representing the relative importance of layer activations at spatial locations with respect to the target task
- Provides a straight forward way of determining the location of the object of interest and/or its segmentation mask
- Helps to identify discriminative visual properties across classes
- Training the smaller networks to mimic the attention maps of higher-performance network leads to gain in classification accuracy
- Trainable attention in CNNs falls under two categories:
    - Hard Attention
    - Soft Attention

Attention Gates

- **Hard Attention** – Stochastic method where a decision is made by using an image region (low-order parameterisation)

- **Soft Attention** – Deterministic method that is probabilistic and can be trained by backpropagation



U-Net

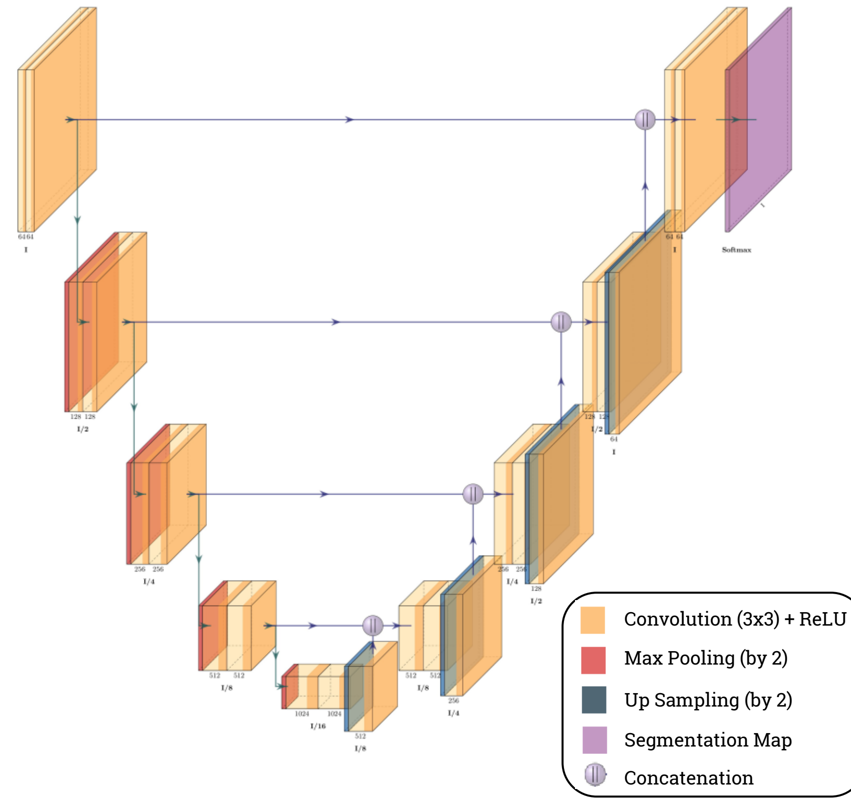
Conv 1 → Max Pooling → Conv 2 → Max Pooling → Conv 3 → Max Pooling → Conv 4 → Max Pooling → Conv 5

Conv 5 → Up Sampling → Deconv 4 → Up Sampling → Deconv 3 → Up Sampling → Deconv 2 → Up Sampling → Deconv `1

Conv 2 → Deconv 2

Conv 3 → Deconv 3

Conv 4 → Deconv 4



U-Net

- Extended FCN with few training images and yields more precise segmentations
- A contracting network supplemented by successive layers
- Pooling operators are replaced by upsampling operators resulting increase in resolution of the output
- To localize, high resolution features from the contracting path are combined with the upsampled output
- Upsampling part has a large number of feature channels that allows the network to propagate context information to high resolution layers
- This is followed by a convolution layer to assemble more precise output

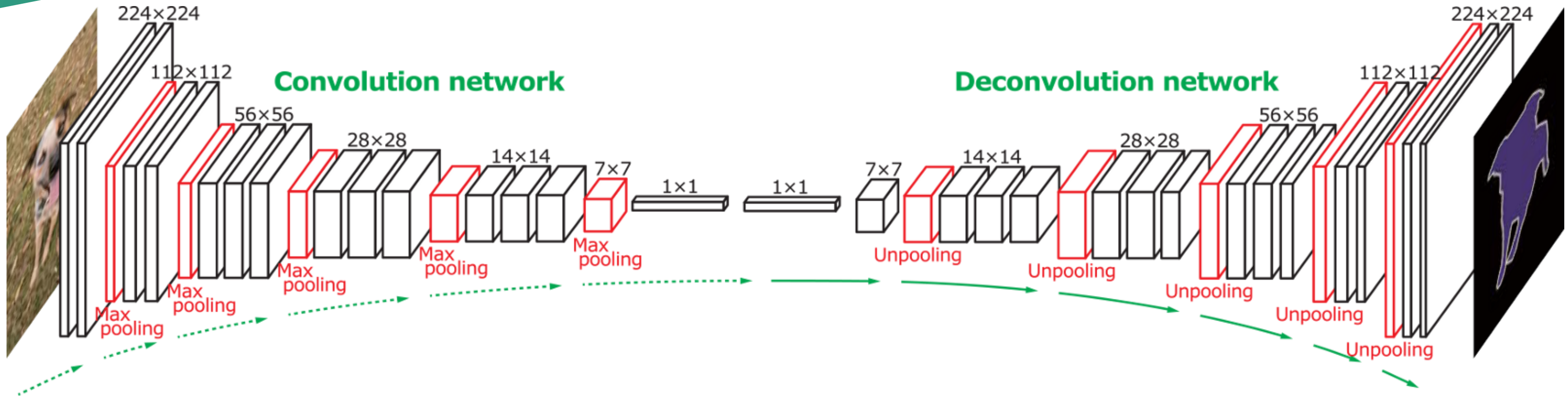
5. Attention U - Net

Methodology and Code review

Attention U-Net

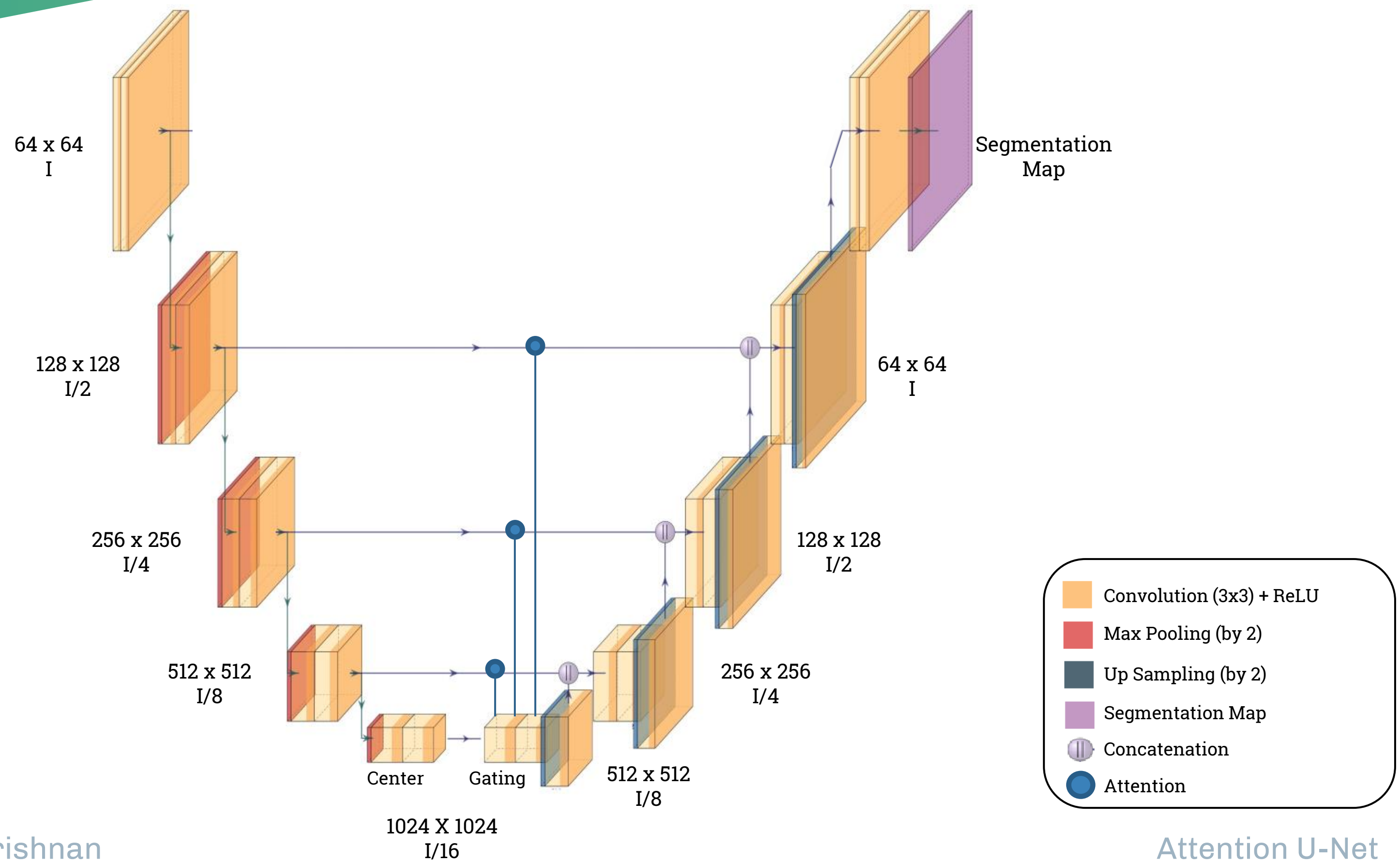
- An extension to standard U-Net model that improves model sensitivity to foreground pixels
- Uses grid-based gating that allows attention coefficients to be more specific to local regions
- Input image is progressively filtered and downsampled in the encoding part
- Attention gates filter the features propagated through skip connections
- The bottleneck layer is upsampled and concatenated with attention layer for better representation of the region of interest

Attention U-Net



- FCNs outperform traditional approaches due to the fact that
    1. Domain specific image features are learnt using SGD optimisation
    2. Learnt kernels are shared across pixels
    3. Exploit structural information and achieve robust and accurate performance

Architecture



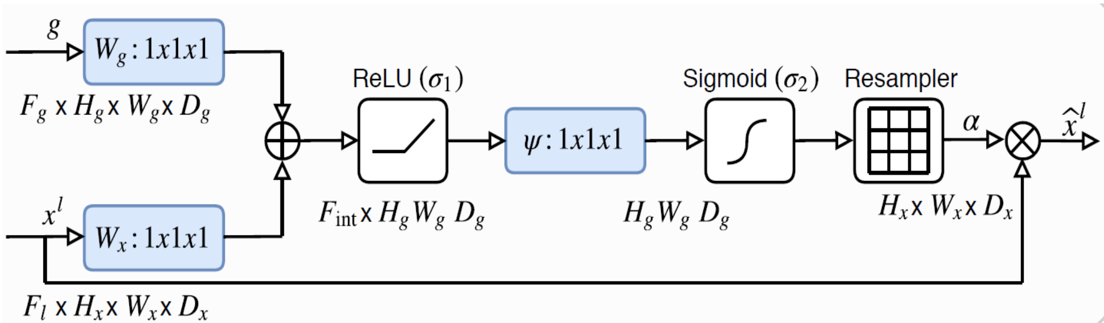
Attention Gates in U-Net

- Features on the coarse spatial grid level model location and relationship between tissues at global scale
- Remains difficult to reduce false-positive predictions of objects that show large shape-variability
- Integrating AGs in CNNs progressively suppress feature responses in irrelevant background regions without cropping an ROI
- Highlights salient features that are passed through skip connections
- Information extracted from coarse scale is used in gating to eliminate irrelevant and noisy responses in skip connections

Attention Gates in U-Net

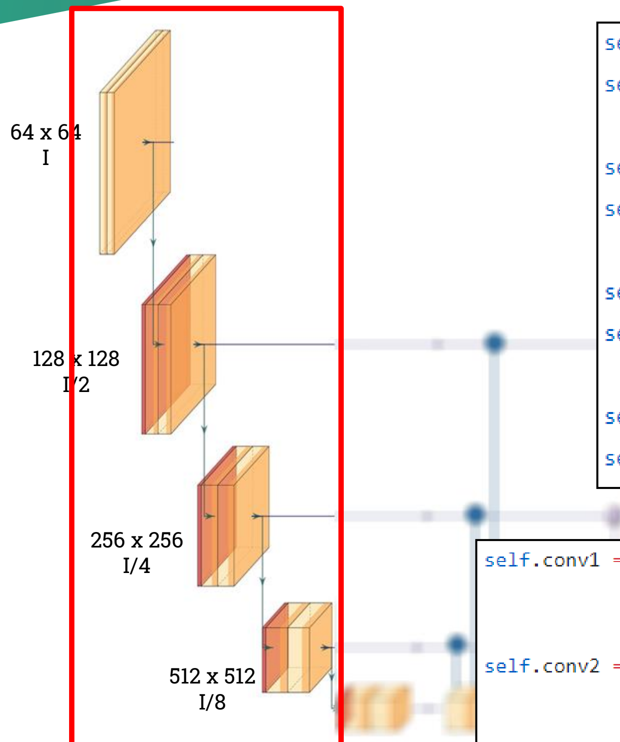
- Input features (x^l) are scaled with attention coefficients (α)
- Spatial regions are selected by analysing activations and information provided by gating signal

$$q_{att}^l = \psi^T \left(\sigma_1 \left(W_x^T x_i^l + W_g^T g_i + b_g \right) \right) + b_\psi$$
$$\alpha_i^l = \sigma_2(q_{att}^l(x_i^l \,, g_i \,; \Theta_{att})),$$



Proposed Additive Attention Gate

Code Review - Encoder




```python
self.conv1 = UnetConv3(self.in_channels, filters[0], self.is_batchnorm)
self.maxpool1 = nn.MaxPool3d(kernel_size=(2, 2, 1))

self.conv2 = UnetConv3(filters[0], filters[1], self.is_batchnorm)
self.maxpool2 = nn.MaxPool3d(kernel_size=(2, 2, 1))

self.conv3 = UnetConv3(filters[1], filters[2], self.is_batchnorm)
self.maxpool3 = nn.MaxPool3d(kernel_size=(2, 2, 1))

self.conv4 = UnetConv3(filters[2], filters[3], self.is_batchnorm)
self.maxpool4 = nn.MaxPool3d(kernel_size=(2, 2, 1))
```

1

```python
self.conv1 = nn.Sequential(nn.Conv3d(in_size, out_size, kernel_size, init_stride, padding_size),
                           nn.BatchNorm3d(out_size),
                           nn.ReLU(inplace=True),)
self.conv2 = nn.Sequential(nn.Conv3d(out_size, out_size, kernel_size, 1, padding_size),
                           nn.BatchNorm3d(out_size),
                           nn.ReLU(inplace=True),)
```

2

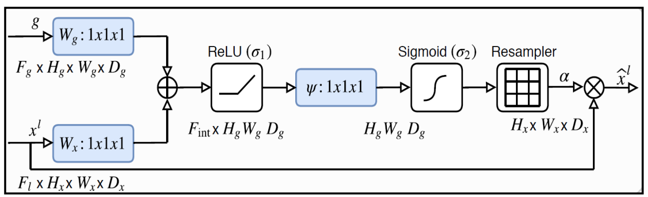
Code Review – Bottle-neck

①

```python
self.center = UnetConv3(filters[3], filters[4], self.is_batchnorm)
self.gating = UnetGridGatingSignal3(filters[4], filters[3], kernel_size=(1, 1, 1), is_batchnorm=self.is_batchnorm)
```

②

```python
# Attention Mechanism
g_conv4, att4 = self.attentionblock4(conv4, gating)
g_conv3, att3 = self.attentionblock3(conv3, gating)
g_conv2, att2 = self.attentionblock2(conv2, gating)
```



③

```python
theta_x = self.theta(x)
theta_x_size = theta_x.size()
```

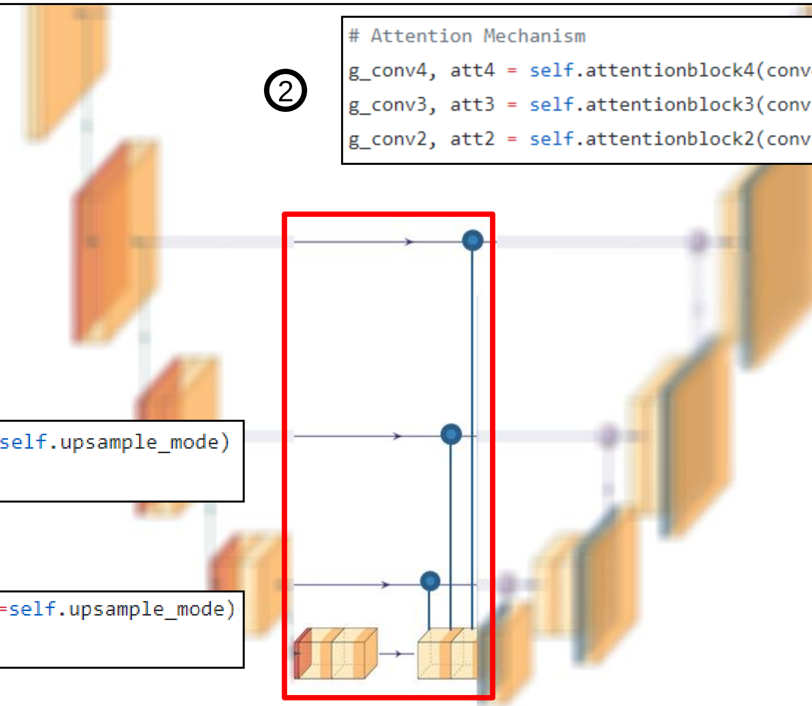
④

```python
phi_g = F.upsample(self.phi(g), size=theta_x_size[2:], mode=self.upsample_mode)
f = F.relu(theta_x + phi_g, inplace=True)
```

⑤

```python
sigm_psi_f = F.sigmoid(self.psi(f))
```

```python
sigm_psi_f = F.upsample(sigm_psi_f, size=input_size[2:], mode=self.upsample_mode)
y = sigm_psi_f.expand_as(x) * x
```



Code Review – Decoder

1

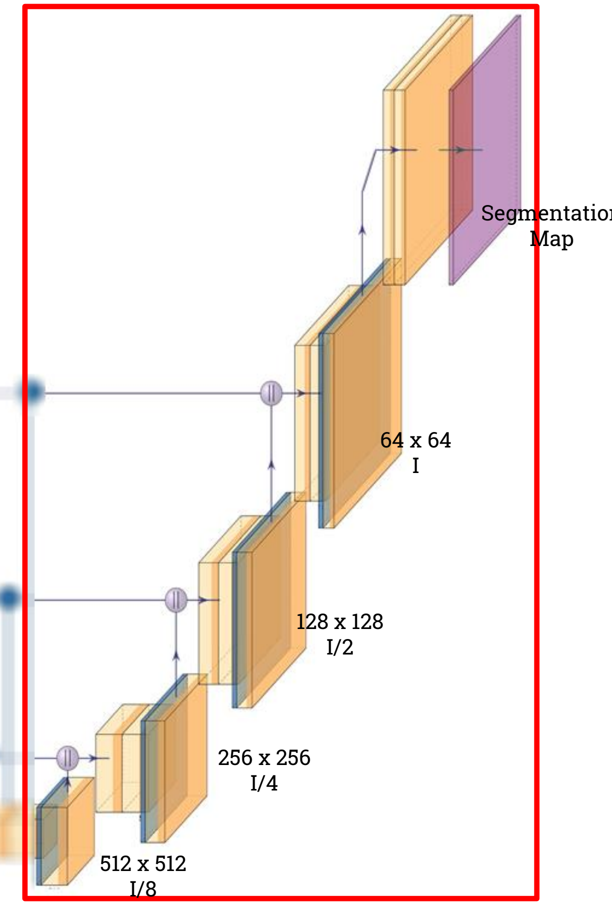
```
up4 = self.up_concat4(g_conv4, center)
up3 = self.up_concat3(g_conv3, up4)
up2 = self.up_concat2(g_conv2, up3)
up1 = self.up_concat1(conv1, up2)
```

2

```
self.up_concat4 = UnetUp3(filters[4], filters[3], self.is_deconv, self.is_batchnorm)
self.up_concat3 = UnetUp3(filters[3], filters[2], self.is_deconv, self.is_batchnorm)
self.up_concat2 = UnetUp3(filters[2], filters[1], self.is_deconv, self.is_batchnorm)
self.up_concat1 = UnetUp3(filters[1], filters[0], self.is_deconv, self.is_batchnorm)
```

3

```
outputs2 = self.up(inputs2)
offset = outputs2.size()[2] - inputs1.size()[2]
padding = 2 * [offset // 2, offset // 2, 0]
outputs1 = F.pad(inputs1, padding)
return self.conv(torch.cat([outputs1, outputs2], 1))
```



6. Experiments and Results

Experiments and comparisioins with baseline results

Experiments

- **Evaluation Datasets**

CT – 150	150 abdominal 3D CT scans (Gastric cancer)
CT – 82	82 contrast enhanced 3D CT scans

- **Implementation Details**

ML Library	PyTorch
Data-augmentation	Yes
Loss Function	Sorensen – Dice Loss

Sorensen – Dice Loss

$$D = \frac{2\sum_i^n p_i g_i\ )}{\sum_i^n p_i^2 + \sum_i^n g_i^2}$$

Results

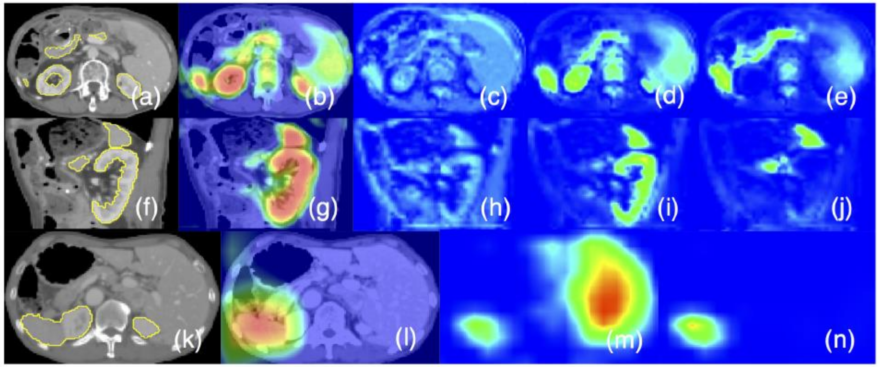


Figure: Axial and sagittal views with feature activations

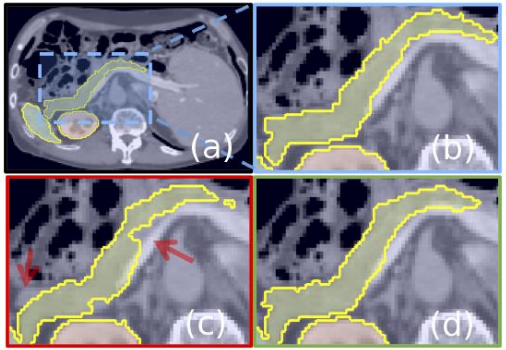


Figure: a,b: Ground Truth
c,d: U-Net and Attention U-Net

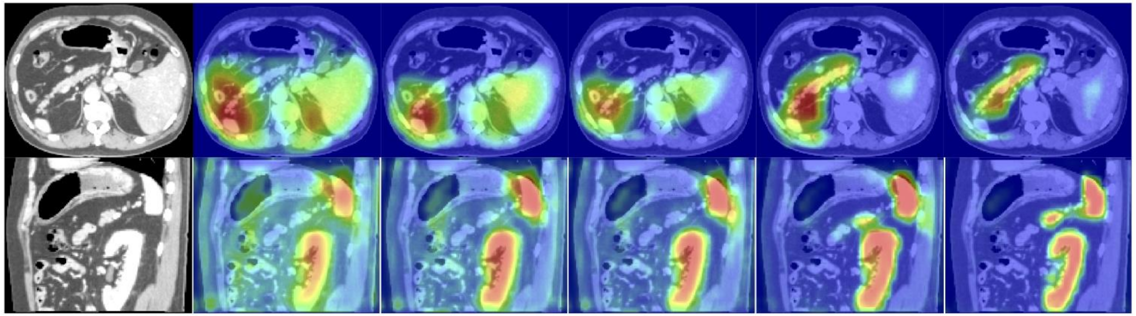


Figure: Attention coefficients across epochs (3,6,10,60,150)

Results

Table: Multi-class CT abdominal segmentation results obtained on the CT-150 dataset

Method (Train/Test Split)	U-Net (120/30)	Att U-Net (120/30)	U-Net (30/120)	Att U-Net (30/120)
Pancreas DSC	0.814±0.116	**0.840±0.087**	0.741±0.137	**0.767±0.132**
Pancreas Precision	0.848±0.110	0.849±0.098	0.789±0.176	**0.794±0.150**
Pancreas Recall	0.806±0.126	**0.841±0.092**	0.743±0.179	**0.762±0.145**
Pancreas S2S Dist (mm)	2.358±1.464	**1.920±1.284**	3.765±3.452	3.507±3.814
Spleen DSC	0.962±0.013	0.965±0.013	0.935±0.095	**0.943±0.092**
Kidney DSC	0.963±0.013	0.964±0.016	0.951±0.019	0.954±0.021
Number of Params	5.88 M	6.40 M	5.88 M	6.40 M
Inference Time	0.167 s	0.179 s	0.167 s	0.179 s

Table:` Segmentation experiments on CT-150 with higher capacity U-Net models

Method	Panc. DSC	Panc. Precision	Panc. Recall	S2S Dist (mm)	# of Pars	Run Time
U-Net (120/30)	0.821±.119	0.849±.111	0.814±.125	2.383±1.918	6.44 M	0.191 s
U-Net (120/30)	0.825±.104	0.861±.082	0.807±.121	2.202±1.144	10.40 M	0.222 s

Results

Table: Pancreas segmentation on TCIA Pancreas – CT Dataset (82)

	Method	Dice Score	Precision	Recall	S2S Dist (mm)
BFT	U-Net [24]	0.690±0.132	0.680±0.109	0.733±0.190	6.389±3.900
BFT	Attention U-Net	**0.712±0.110**	0.693±0.115	**0.751±0.149**	**5.251±2.551**
AFT	U-Net [24]	0.820±0.043	0.824±0.070	0.828±0.064	2.464±0.529
AFT	Attention U-Net	**0.831±0.038**	0.825±0.073	**0.840±0.053**	**2.305±0.568**
SCR	U-Net [24]	0.815±0.068	0.815±0.105	0.826±0.062	2.576±1.180
SCR	Attention U-Net	0.821±0.057	0.815±0.093	**0.835±0.057**	**2.333±0.856**

Table: State-of-the-art CT pancreas segmentation methods on CT-150 dataset

Method	Dataset	Pancreas DSC	Train/Test	# Folds
Hierarchical 3D FCN [27]	CT-150	82.2 ± 10.2	Ext/150	-
Dense-Dilated FCN [6]	CT-82 & Synapse[3]	66.0 ± 10.0	63/9	5-CV
2D U-Net [8]	CT-82	75.7 ± 9.0	66/16	5-CV
Holistically Nested 2D FCN Stage-1[26]	CT-82	76.8 ± 11.1	62/20	4-CV
Holistically Nested 2D FCN Stage-2[26]	CT-82	81.2 ± 7.3	62/20	4-CV
2D FCN [4]	CT-82	80.3 ± 9.0	62/20	4-CV
2D FCN + Recurrent Network [4]	CT-82	82.3 ± 6.7	62/20	4-CV
Single Model 2D FCN [38]	CT-82	75.7 ± 10.5	62/20	4-CV
Multi-Model 2D FCN [38]	CT-82	82.2 ± 5.7	62/20	4-CV

7. Future Work

Future work proposed by the authors

Future Work

- Transfer learning and multi-stage training schemes
- Gates can be trained accordingly in fine-tuning stage
- To initialise attention network, pre-trained U-Net weights can be used
- Highway networks to allow better gradient backpropagation
- Improve performance by utilising fine resolution input batches

8. Conclusion

Conclusion

- A novel AG model was applied to medical image segmentation
- Eliminates the use of external object localisation model
- Focuses on the relevant region eliminating the extraneous regions
- Can be applied to image classification and regression
- Highly beneficial for tissue/organ identification and localisation

References

- Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation.In: IEEE CVPR. pp. 3431–3440 (2015)
- Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: MICCAI. pp. 234–241. Springer (2015)
- Jetley, S., Lord, N.A., Lee, N., Torr, P.: Learn to pay attention. In: International Conference on Learning Representations (2018) https://openreview.net/forum?id=HyzbhfWRW
- Milletari, F., Navab, N., Ahmadi, S.A.: V-net: Fully convolutional neural networks for volumetric medical image segmentation. In: 3D Vision. pp. 565–571. IEEE(2016)
- https://github.com/ozan-oktay/Attention-Gated-Networks

References

- https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148
- https://www.mathworks.com/solutions/deep-learning/convolutional-neural-network.html
- https://www.mathworks.com/help/deeplearning/examples/create-simple-deep-learning-network-for-classification.html
- https://adeshpande3.github.io/adeshpande3.github.io/A-Beginner%27s-Guide-To-Understanding-Convolutional-Neural-Networks/
- https://medium.com/apache-mxnet/transposed-convolutions-explained-with-ms-excel-52d13030c7e8

Thank You !!