Applied Deep Learning
Segmentation Networks

Mark Schutera

Karlsruhe Institute of Technology | ZF Friedrichshafen AG | Hochschule Ravensburg-Weingarten

Course overview

1. Deep Learning Foundations

3. Transfer Learning and Object Detection

5. Segmentation Networks

8. Deep Reinforcement Learning

10. Generative Adversarial Networks

12. Recurrent Neural Networks

Course overview

Project proposal

One page on introduction, methods, dataset

Deadline 3. Lecture

Intermediate presentation

Ten minutes on achievements, problems, next steps

Due 7. Lecture

Final presentation

Code and results

Due 14. Lecture

Final documentation

Paper and code on github or jupyter notebook

Deadline 14. Lecture

Course features

Sli.do

Every question matters.

Get the app.

Ask questions (with slide number) or vote on other students' questions during the lecture.

And give direct feedback.

#TOBEDETERMINED

Questions will be covered immediately or in the next lecture in more depth.

Github

Find slides, tutorials, flashcards and references on Github.

https://github.com/schutera/DeepLearningLecture_Schutera

You found typos, additional material such as links, algorithms, papers, literature or want to contribute to the slides and lecture notes..

..Feel free to contribute, e-mail me.

Course features

Sli.do

Every question matters.

Get the app.

Ask questions (with slide number) or vote on other students' questions during the lecture.

#TOBEDETERMINED

Questions will be covered directly or in the next lecture in more depth.

Github

Typos, additional material such as links, algorithms, paper, literature, lecture notes..

..Feel free to contribute.

Grade Bonus .3

Prepare flashcards based on Ian Goodfellow's Deep Learning Book

- Commit to flashcard set by emailing me, first come first serve
- Must be comprehensive

This lecture in one slide

Introduction to segmentation

Segmentation a problem statement

Applications for segmentation

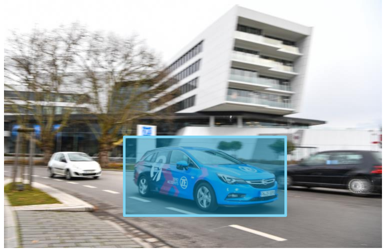
Conventional segmentation approaches

Segmentation with neural networks

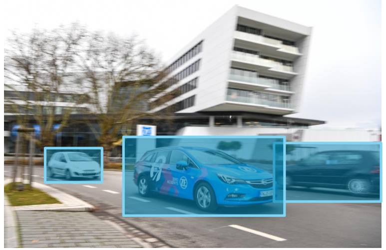
Introduction – Segmentation a problem statement

Classification

Classification
+ Localization



Object Detection



Object Tracking

Introduction – Segmentation a problem statement

Segmentation is classification on pixel-level, which results in super-pixels or segments or groups of pixels based on some criteria.

Semantic Segmentation

Instance Segmentation

Introduction – Applications for segmentation

Autonomous Driving

- Scene understanding
- Understanding of shapes
- Supports sensor fusion with point cloud sensors
- Free space detection

Geo Analytics

Medical Imaging

https://www.cityscapes-dataset.com/

Introduction – Applications for segmentation

Autonomous Driving

Geo Analytics

- Building structures
- Road network analysis
- Wildfire detection
- Water supply tracking
- Real time crisis management
- Weather prediction

https://github.com/mapbox/robosat

Medical Imaging

Introduction – Applications for segmentation

Autonomous Driving
Geo Analytics

Medical Imaging

- Tissue localization and analysis
- Volume approximations
- Surgery planning
- Temporal tumor or tissue development
- Tooling for drug testing

https://osf.io/snb6p/

Introduction – Conventional segmentation approaches

Image segmentation is a well researched field.

In order to design neural networks it is a good thing to really understand the task at hand.

Thresholding
Edge detection
Clustering
Region growing

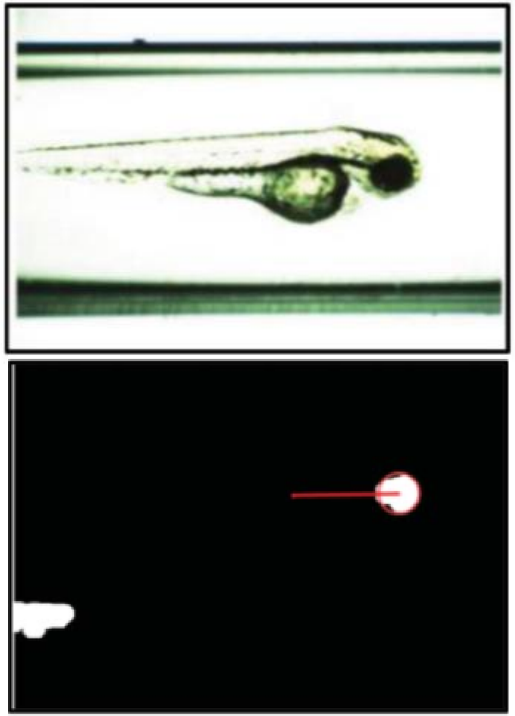
Thresholding

The simplest method of image segmentation is called the thresholding method.

This method is based on a threshold value to turn a gray-scale image into a binary image (mask).

Usually this is just one step of many.

Edge detection
Clustering
Region growing



Thresholding on Zebrafish for eye segmentation

Lena Test image

Lena , the ‘hello world!’ of image processing. 330x330

Cover photo of 1972 Playboy magazine of the Swedish model Lena Söderberg.

Since then Lena was a guest at several IEEE conferences. The image also sparked discussions on gender-equality in the male-dominated field of engineering.

It is a good test image because of its detail, flat regions, shading, and texture.

https://en.wikipedia.org/wiki/Lenna

Introduction – Conventional segmentation approaches

Thresholding

Edge detection

Segment boundaries and edges are closely related.

Since there is often a large gradient at the segment boundaries.

Canny Edge Detection

$$L_x = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} L \quad \text{and} \quad L_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} L.$$

Sobel Operators for Edge Detection

Clustering

Region growing

Introduction – Conventional segmentation approaches

Thresholding

Edge detection

Often edge detectors are combined with morphological operators to close the detected edges.

Clustering

Region growing

Edge Detection for segmentation

Introduction – Conventional segmentation approaches

Thresholding

Edge detection

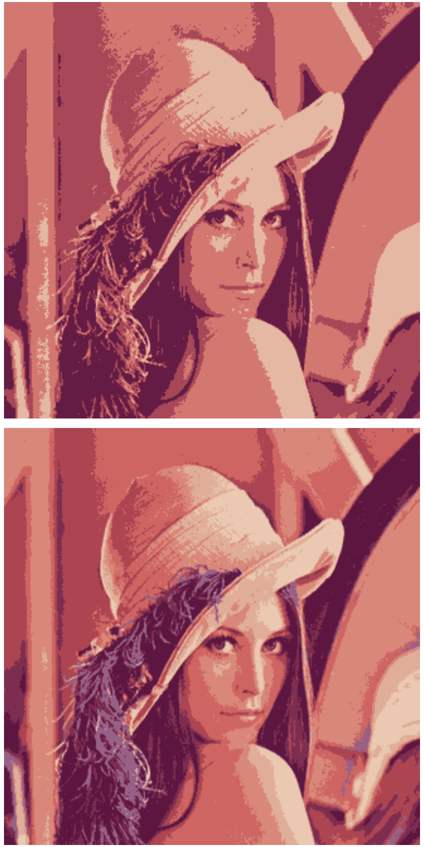
Clustering (Color quantization)

K-means with 3 features (R,G,B) and K centroids.

The centroids are iteratively adjusted until convergence.

After the clustering, the centroid values are applied to the pixels in their cluster.

Region growing



Clustering for K=4 (top) and K=8 (bottom)

Introduction – Conventional segmentation approaches

Thresholding
Edge detection
Clustering

Thresholding, Find valleys, Region growing for Segmentation

Region growing

Any image can be viewed as a topographic surface due to the gradients in the image.

You start by finding valleys and filling them with different colored water (labels). As the water rises, depending on the peaks (gradients) nearby, water from different valleys, will get in contact, that is where you build barriers. The barriers give the resulting segmentation

References

[1] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[2] Andrej Karpathy. Cs231n: Convolutional neural networks for visual recognition. http://cs231n.github.io/neural-networks-3/, 2018. Zugriff: 20.01.2018.

[3] Schutera, Mark, Steffen Just, Jakob Gierten, Ralf Mikut, Markus Reischl, and Christian Pylatiuk. 2019. "Machine Learning Methods for Automated Quantification of Ventricular Dimensions." OSF. March 28. osf.io/snb6p.

[4] Mark Schutera, Thomas Dickmeis, Marina Mione, Ravindra Peravali, Daniel Marcato, Markus Reischl, Ralf Mikut, and Christian Pylatiuk. Automated phenotype pattern recognition of zebrafish for high-throughput screening. *Bioengineered*, 7(4):261–265, 2016.

[5] Canny, J., *A Computational Approach To Edge Detection*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 8(6):679–698, 1986.

This lecture in one slide

Introduction to segmentation

Segmentation with neural networks

Basic structure

Overview state-of-the-art

Datasets and benchmarking

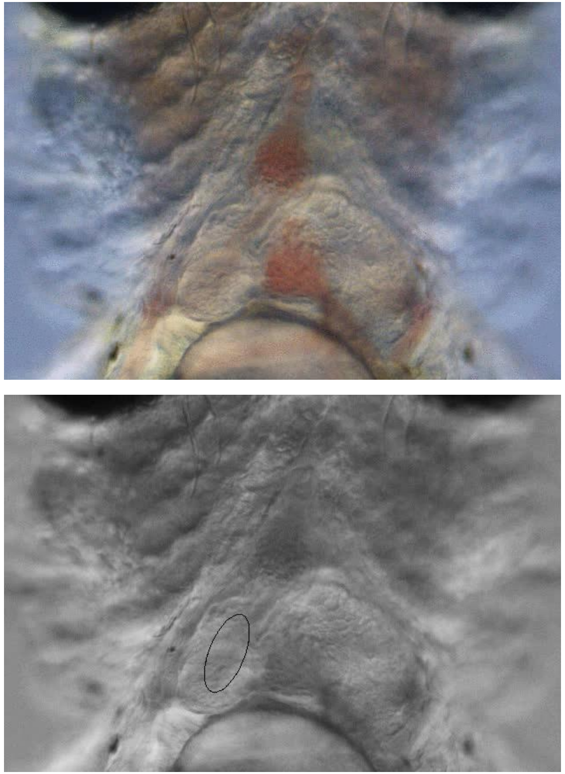
Deep dive U-Net

Neural Network Segmentation - Basic structure

Why classical segmentation approaches

- Interpretability
- Only a few samples needed
- No labeling needed
- No training needed
- Usually better runtime during inference

Why not?



https://osf.io/snb6p/

Why classical segmentation approaches

Why segmentation by neural networks?

- Do generalize better
- Feature engineering has a limited capacity to capture semantics
- Feature engineering is expensive and time consuming

https://osf.io/snb6p/

Feature Representation by Convolution

Idea is to classify each pixel of an input image by representation learning

Downsampling

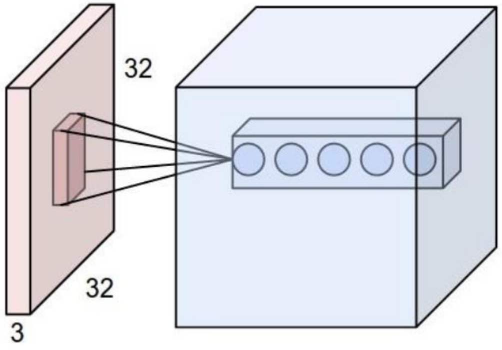
Upsampling

Parameter sharing

Feature Representation by Convolution

A convolutional layer is not fully connected, but has a narrowed down receptive field (e.g. 3x3).

The parameters of each filter are spatially shared: A feature that is useful in one place, ought to be useful in another, too.



- **Depth:** number of filters
- **Stride:** filter step size (when we “slide” it)
- **Padding:** zero-pad the input

https://selfdrivingcars.mit.edu/

Downsampling
Upsampling
Parameter sharing

Feature Representation by Convolution

0	0	0	0	0	0	0
0	1	2	0	0	0	0
0	2	2	0	0	0	0
0	2	2	2	1	0	0
0	1	0	0	2	0	0
0	1	1	0	1	0	0
0	0	0	0	0	0	0

Zero-Padding is adding zero-valued pixel to the image border (gray area).

Downsampling
Upsampling
Parameter sharing

Feature Representation by Convolution

0	0	0	0	0	0	0
0	1	2	0	0	0	0
0	2	2	0	0	0	0
0	2	2	2	1	0	0
0	1	0	0	2	0	0
0	1	1	0	1	0	0
0	0	0	0	0	0	0

Zero-padded image

0	0	-1
-1	0	0
-1	-1	-1

Filter

0

Bias

-4	-4	0
-3	-4	-3
0	-3	-1

Output

Downsampling

Upsampling

Parameter sharing

Feature Representation by Convolution

0	0	0	0	0	0	0
0	1	2	0	0	0	0
0	2	2	0	0	0	0
0	2	2	2	1	0	0
0	1	0	0	2	0	0
0	1	1	0	1	0	0
0	0	0	0	0	0	0

0	0	-1
-1	0	0
-1	-1	-1

Filter

0

Bias

-4	-4	0
-3	-4	-3
0	-3	-1

Output

Amount of filters or convolution depth: 1
Filter step size or Stride: 2
Zero-padded image

Downsampling
Upsampling
Parameter sharing

Feature Representation by Convolution

0	0	0	0	0	0	0
0	1	2	0	0	0	0
0	2	2	0	0	0	0
0	2	2	2	1	0	0
0	1	0	0	2	0	0
0	1	1	0	1	0	0
0	0	0	0	0	0	0

0	0	-1
-1	0	0
-1	-1	-1

Filter

0

Bias

-4	-4	0
-3	-4	-3
0	-3	-1

Output

Review edge detector:
Similar idea, now the parameters of the filters are learned. We want a lot of filters!

$$L_x = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} L \quad \text{and} \quad L_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} L.$$

Downsampling
Upsampling
Parameter sharing

Feature Representation by Convolution

0	0	0	0	0	0	0
0	1	2	0	0	0	0
0	2	2	0	0	0	0
0	2	2	2	1	0	0
0	1	0	0	2	0	0
0	1	1	0	1	0	0
0	0	0	0	0	0	0

0	0	-1
-1	0	0
-1	-1	-1

Filter

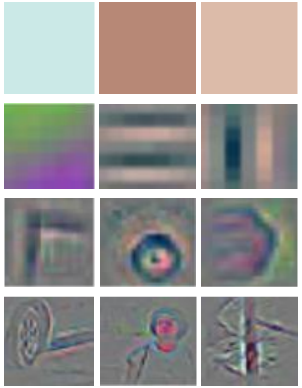
0

Bias

-4	-4	0
-3	-4	-3
0	-3	-1

Output

Review edge detector:
Similar idea, now the parameters of the filters are learned. And we want to go deep!



Downsampling
Upsampling
Parameter sharing

Convolutions

Downsampling

Convolutions at original image resolution are computational expensive:
Filter dimensions x *image dimensions* x *number of filters* x *number of input channels.*

Motivating a convolutional encoder-decoder structure and Downsampling.

Upsampling

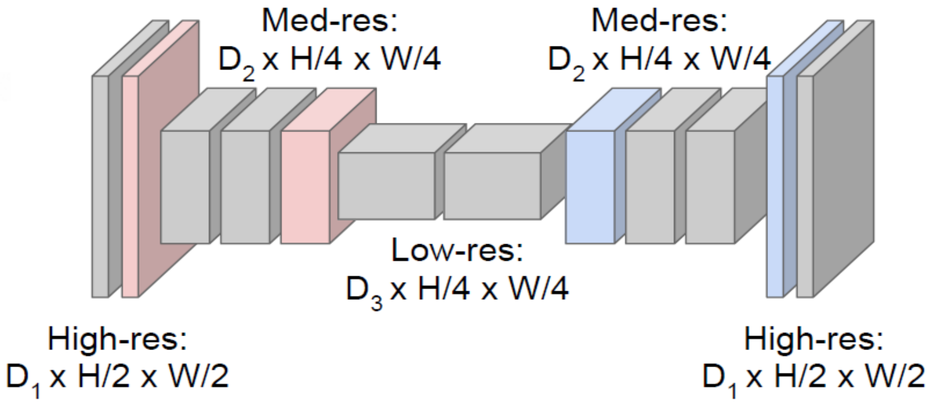
Parameter sharing

Convolutions

Downsampling

Convolutions at original image resolution are computational expensive:
Filter dimensions x *image dimensions* x *number of filters* x *number of input channels.*

Motivating a convolutional encoder-decoder structure and Downsampling.



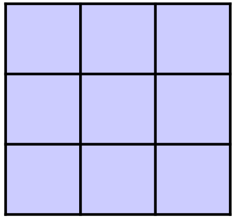
http://cs231n.github.io/

Upsampling
Parameter sharing

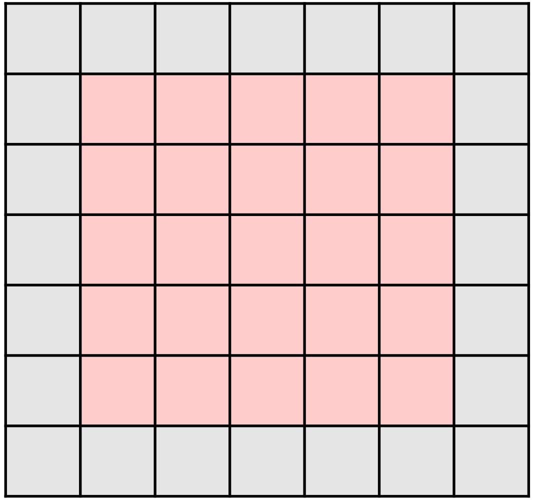
Convolutions

Downsampling

Strided convolutions



Filter 3x3x1



Zero-padded image

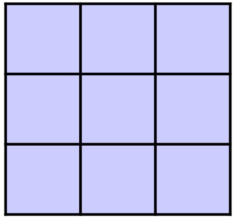
Upsampling

Parameter sharing

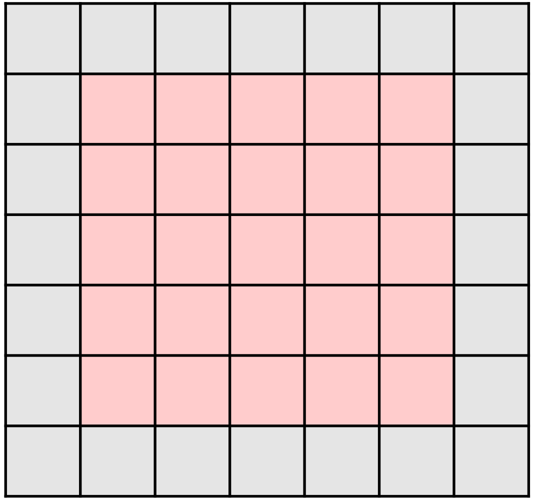
Convolutions

Downsampling

Strided convolutions

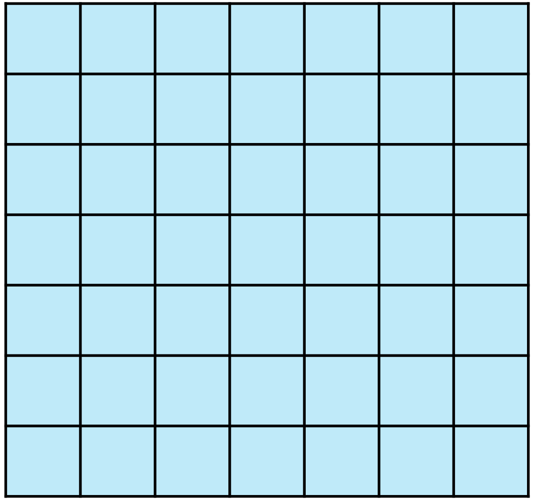


Filter 3x3x1



Zero-padded image

Stride 1



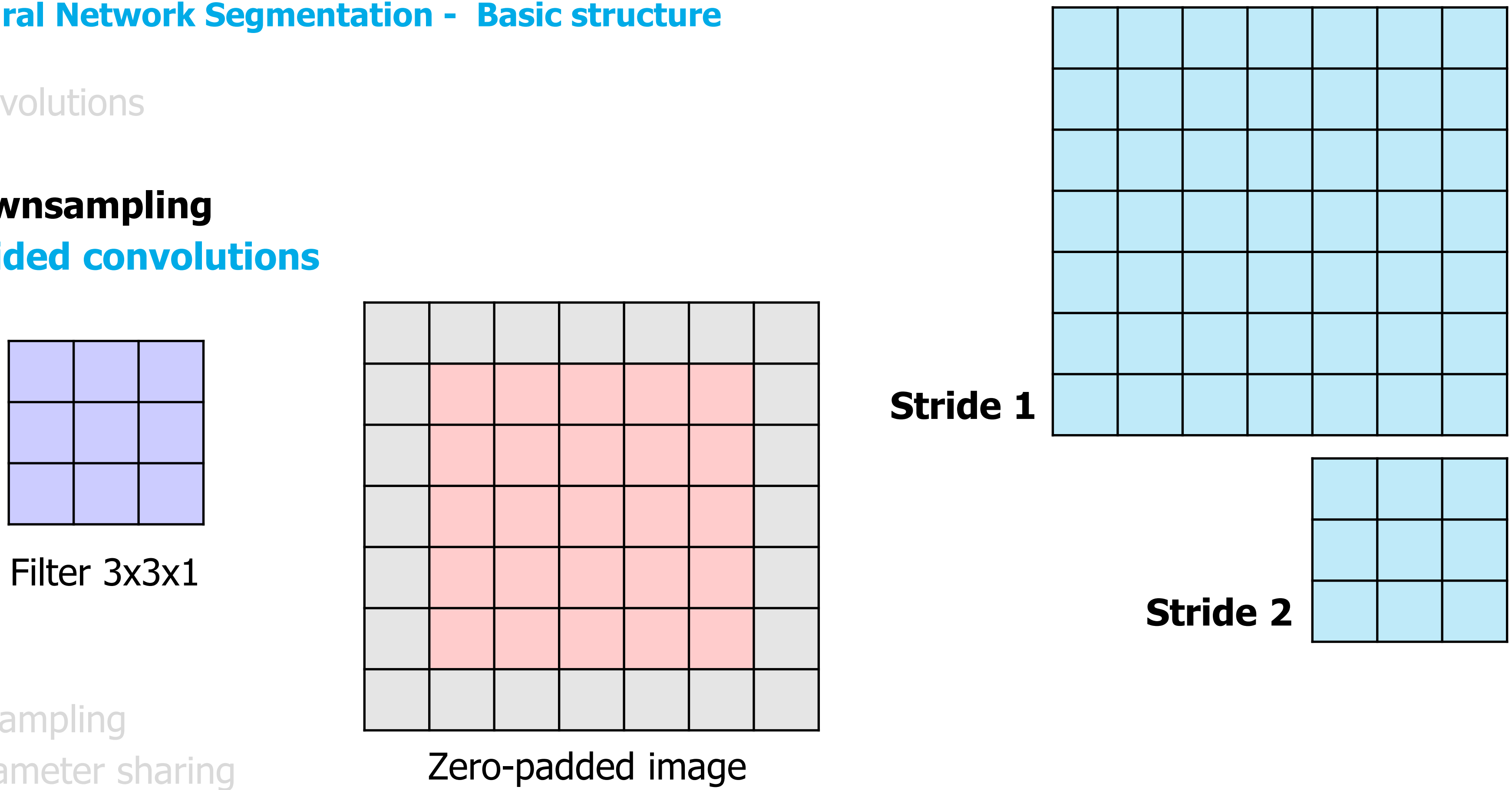
Upsampling

Parameter sharing

Convolutions

Downsampling

Strided convolutions



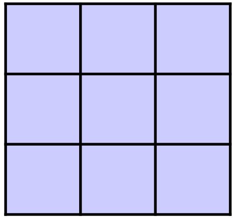
Upsampling

Parameter sharing

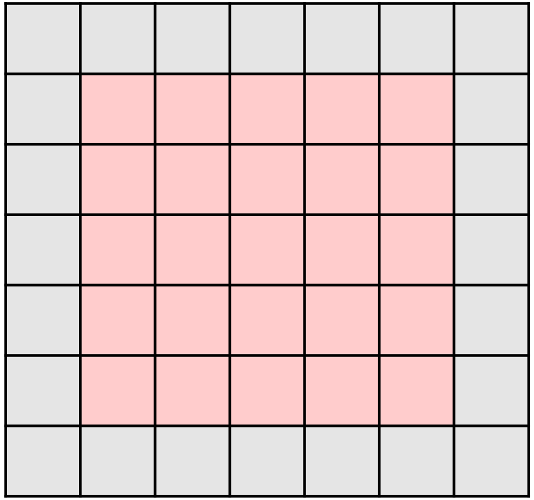
Convolutions

Downsampling

Strided convolutions

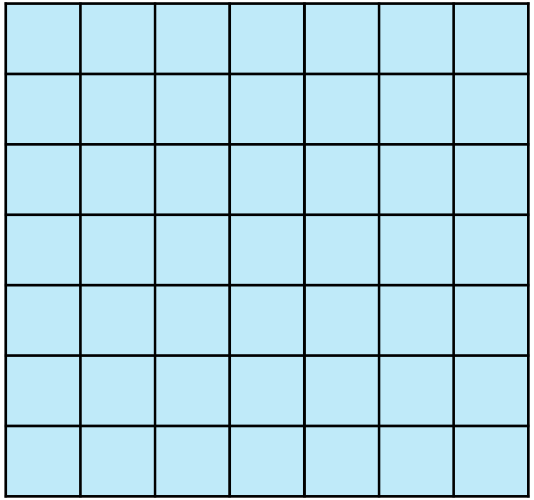


Filter 3x3x1

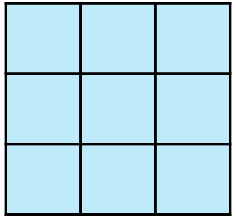


Zero-padded image

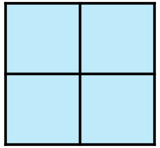
Stride 1



Stride 2



Stride 4



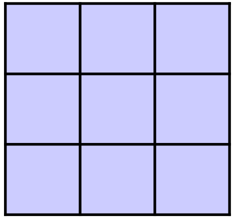
Upsampling

Parameter sharing

Convolutions

Downsampling

Max Pooling



Max Pooling

3x3 Stride 3

5	4	6	3	1	7
3	4	1	6	4	4
6	2	5	6	6	4
2	6	9	8	6	3
4	8	5	6	8	2
3	1	7	8	6	3

→

6	7
9	8

Upsampling

Parameter sharing

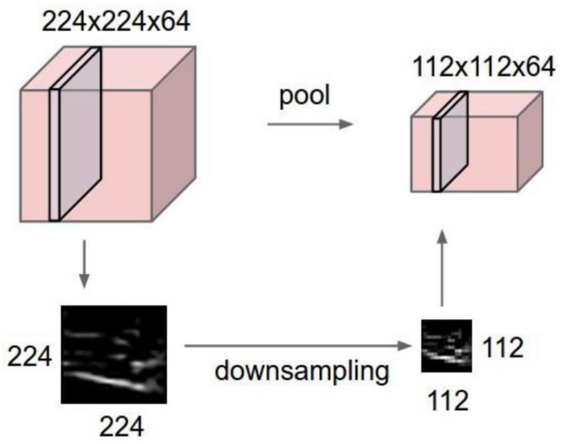
Convolutions

Downsampling

Max Pooling

Intuition is to decrease the resolution while keeping the strongest features of each channel.

Introducing a location invariance.



https://selfdrivingcars.mit.edu/

Upsampling
Parameter sharing

Convolutions
Downsampling

Upsampling

Classification needs to happen in original image resolution

Motivating Upsampling inside the network structure.

Parameter sharing

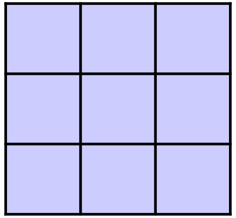
Neural Network Segmentation - Basic structure

Convolutions

Downsampling

Upsampling

Nearest neighbor



3x3 Stride 3

6	7
9	8

6	6	6	7	7	7
6	6	6	7	7	7
6	6	6	7	7	7
9	9	9	8	8	8
9	9	9	8	8	8
9	9	9	8	8	8

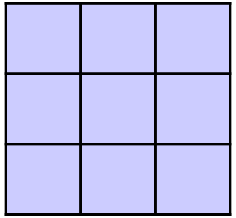
Parameter sharing

Convolutions

Downsampling

Upsampling

Bed of Nails



3x3 Stride 3

6	7
9	8

6	0	0	7	0	0
0	0	0	0	0	0
0	0	0	0	0	0
9	0	0	8	0	0
0	0	0	0	0	0
0	0	0	0	0	0

Parameter sharing

Neural Network Segmentation - Basic structure

Convolutions

Downsampling

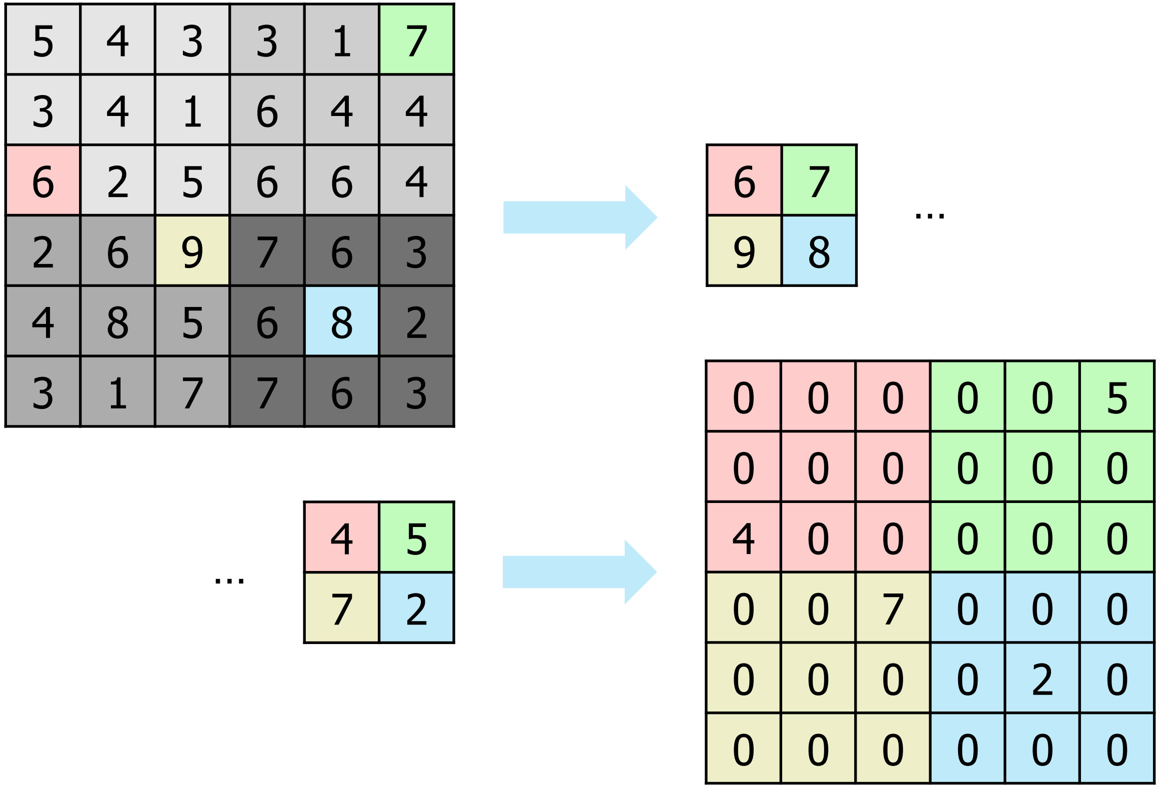
Upsampling

Max Unpooling

Corresponding pairs of downsampling and upsampling layers.

Use position of pooling layer for unpooling

Parameter sharing



Neural Network Segmentation - Basic structure

Convolutions

Downsampling

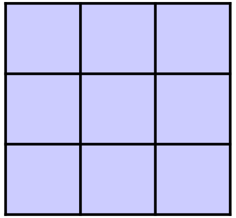
Upsampling

Transpose convolution

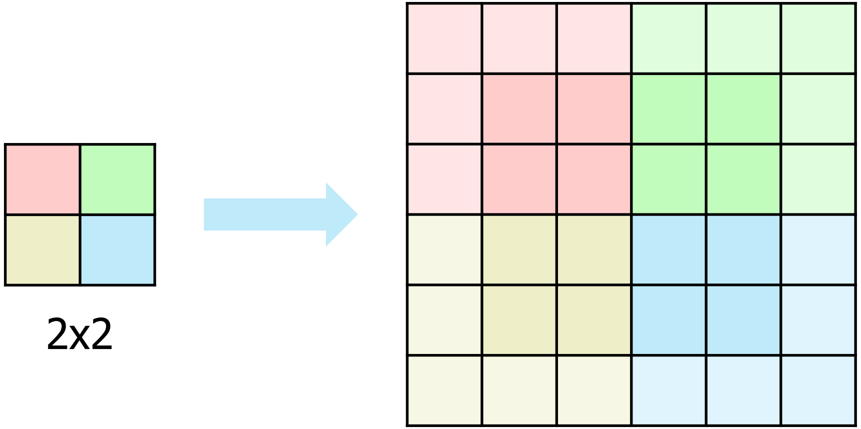
Learnable Upsampling, also known as: Upconvolution, or Deconvolution (bad terminology)

Stride: 3

Padding: 1



3x3



2x2

4x4

Parameter sharing

Neural Network Segmentation - Basic structure

Convolutions
Downsampling
Upsampling

Skip connections

Trade-off between classification and localization

- High level features from later in the network, enable high classification performance, since they are more discriminative and contain more useful semantic information.
- On the other hand, those deep features have low resolution and, thus pose a problem for localization performance.

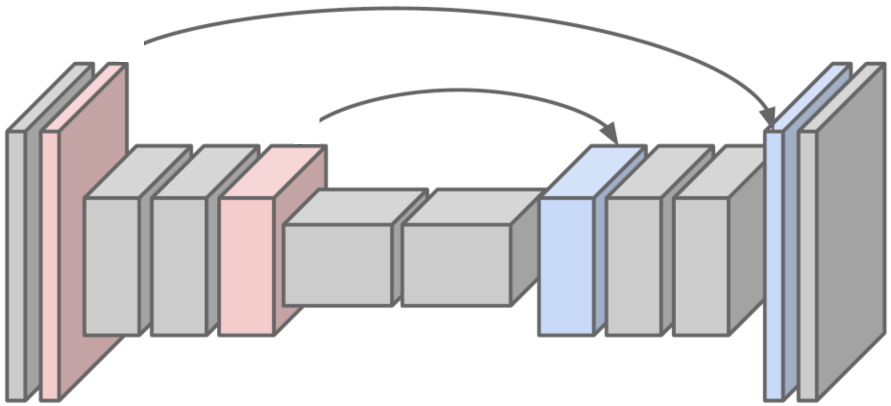
Neural Network Segmentation - Basic structure

Convolutions
Downsampling
Upsampling

Skip connections

Combining low-level features, which have high localization accuracy

With the high-level features, which have are descriptive but low-resolution.

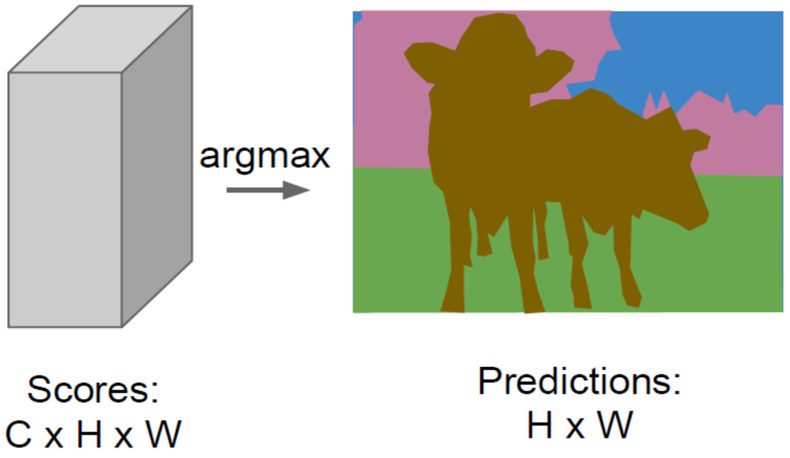


http://cs231n.github.io/

Last layer

Last Layer results in a tensor with H x W image resolution and a depth of
C: Number of classes to segment.

The last layer should encode the values into a range of values of (0;1).
Either by softmax or sigmoid function.



http://cs231n.github.io/

Cross-entropy
Dice-coefficient

Neural Network Activation Functions - Review

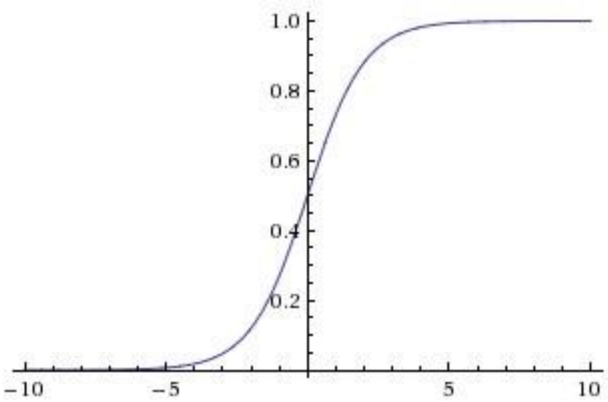
Sigmoid

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

$$f'(z) = (1 - f(z))f(z)$$

Binary classification only.

The probability sum does not need to be one.



http://cs231n.github.io/neural-networks-1/

Softmax

Normalized exponential function

$$\sigma(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

Used for multi-class segmentation.

Probability sum will be 1.

before	after
2.0	0.7
1.0	0.2
0.1	0.1

Last layer

Binary Cross-entropy

$$J = -\frac{1}{N}\sum_{n=1}^{N}\left[y_n \log \hat{y}_n + (1 - y_n)\log(1 - \hat{y}_n)\right]$$

Heavily penalizes opposing predictions.

And gives rise to the problem of imbalanced classes.

Dice-coefficient

Neural Network Segmentation - Optimization

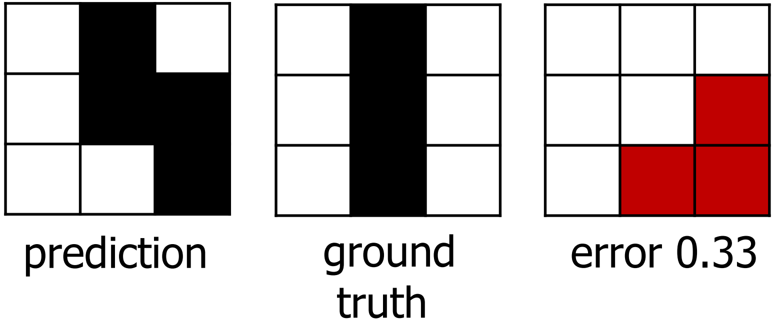
Last layer

Binary Cross-entropy

$$J = -\frac{1}{N}\sum_{n=1}^{N}\left[y_n \log \hat{y}_n + (1 - y_n)\log(1 - \hat{y}_n)\right]$$

Heavily penalizes opposing predictions.

And gives rise to the problem of imbalanced classes.



Dice-coefficient

Neural Network Segmentation - Optimization

Last layer
Binary Cross-entropy

$$dice\ loss = 1 - \frac{2\sum(\tilde{\mathbf{y}} \odot \mathbf{y}_{seg}) + 1}{\sum(\tilde{\mathbf{y}}^2) + \sum(\mathbf{y}_{seg}^2) + 1}.$$

Dice-coefficient

Similar to the IoU (Intersection over union), and thus easy to interpret.

+1 is a smoothing factor for numeric stability.

More robust with respect to imbalanced classes.

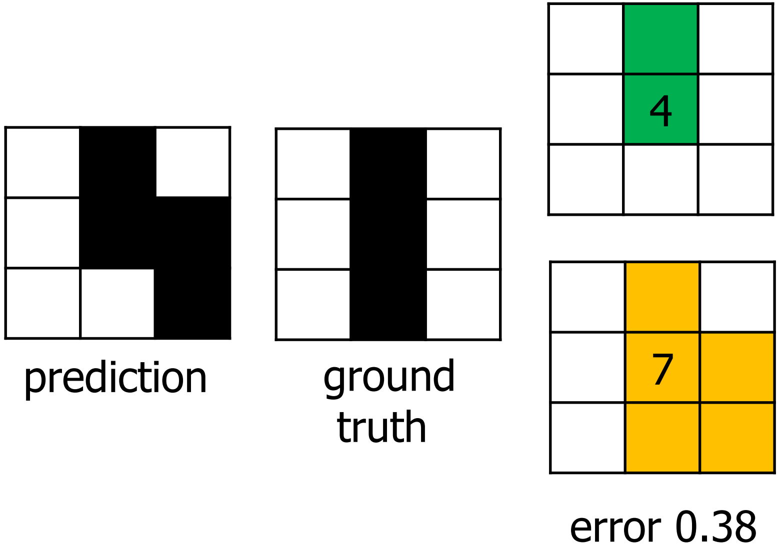
Last layer
Binary Cross-entropy

Dice-coefficient

Similar to the IoU (Intersection over union), and thus better interpretability.

More robust with respect to imbalanced classes.

$$dice\ loss = 1 - \frac{2\sum(\tilde{\mathbf{y}} \odot \mathbf{y}_{seg}) + 1}{\sum(\tilde{\mathbf{y}}^2) + \sum(\mathbf{y}_{seg}^2) + 1}.$$

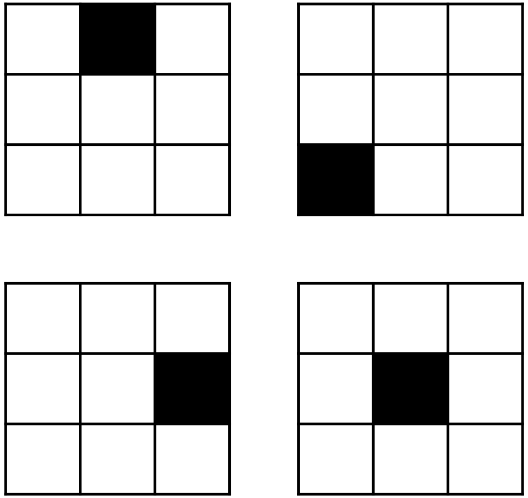


Thought experiment

Assumption:

The maximum number of class 1 pixels in a single sample is 1.

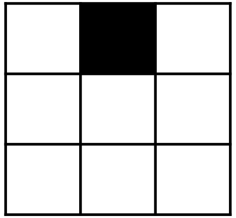
This simulates an extreme class imbalance ratio of 1 to 8.



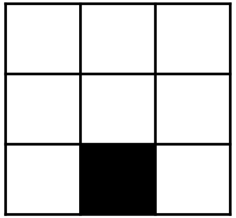
Thought experiment

Assumption:

The maximum number of class 1 pixels in a single sample is 1



prediction



ground truth

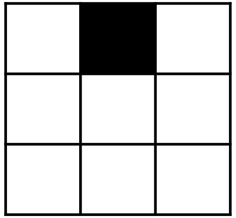
What is the expected error if the model tries to predict a single pixel of class 1 and fails?

Thought experiment

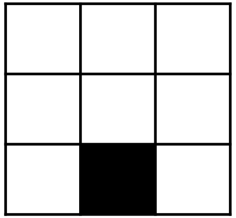
Assumption:

The maximum number of class 1 pixels in a single sample is 1

What is the expected error if the model tries to predict a single pixel of class 1 and fails?



prediction



ground truth

BCE
0.22

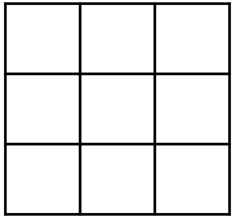
DL
0.66

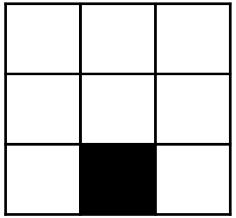
Thought experiment

Assumption:
The maximum number of class 1 pixels in a single sample is 1

What is the expected error if the model tries to predict a single class 1?

What is the maximal expected error if the model predicts class 2 only?

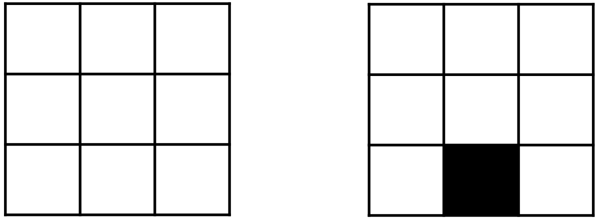




Thought experiment

Assumption:
The maximum number of class 1 pixels in a single sample is 1

What is the expected error if the model tries to predict a single class 1?



What is the maximal expected error if the model predicts class 2 only?

BCE	DL
0.11	0.5

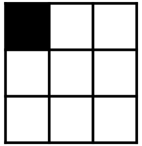
Thought experiment

How high is the pressure to get locked in a local minimum if predictions are initially random?

Thought experiment

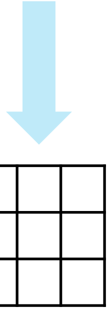
How high is the pressure to get locked in a local minimum if predictions are initially random?

Binary cross-entropy



$0.22 * 8 + 0.00 * 1 = 1.76$

$0.11 * 9 = 1.00$



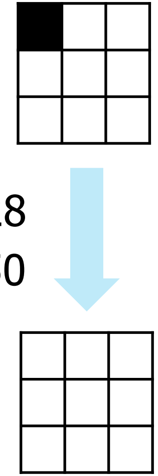
43 %

Thought experiment

How high is the pressure to get locked in a local minimum if predictions are initially random?

Dice-loss

$$0.66 \cdot 8 + 0.00 \cdot 1 = 5.28$$
$$0.50 \cdot 9 = 4.50$$



15 %

How to deal with imbalanced classes

Choose dice-loss over cross-entropy.

How to deal with imbalanced classes

Choose dice-loss over cross-entropy.

Balance your cross-entropy according to the class imbalance.

In our case β *=7/8*

$$\mathrm{BCE}\left(p, \hat{p}\right) = -\left(\beta p \log(\hat{p}) + (1-\beta)(1-p)\log(1-\hat{p})\right)$$

How to deal with imbalanced classes

Choose dice-loss over cross-entropy.

Balance your cross-entropy according to the class imbalance.

Extend the dice-loss to the Tverski loss.
Which weighs the influence
of the False Positives
and False Negative.

$$\mathrm{DL}\left(\mathbf{p},\hat{\mathbf{p}}\right)=\frac{\langle\mathbf{p},\hat{\mathbf{p}}\rangle}{\langle\mathbf{p},\hat{\mathbf{p}}\rangle+\beta\langle\mathbf{1}-\mathbf{p},\hat{\mathbf{p}}\rangle+(1-\beta)\langle\mathbf{p},\mathbf{1}-\hat{\mathbf{p}}\rangle}$$

In our case $\beta = 1/8$

PASCAL Visual Object Classes

Pixel-wise segmentation of objects from a number of visual object classes in realistic scenes (i.e. not pre-segmented objects).

Annotations

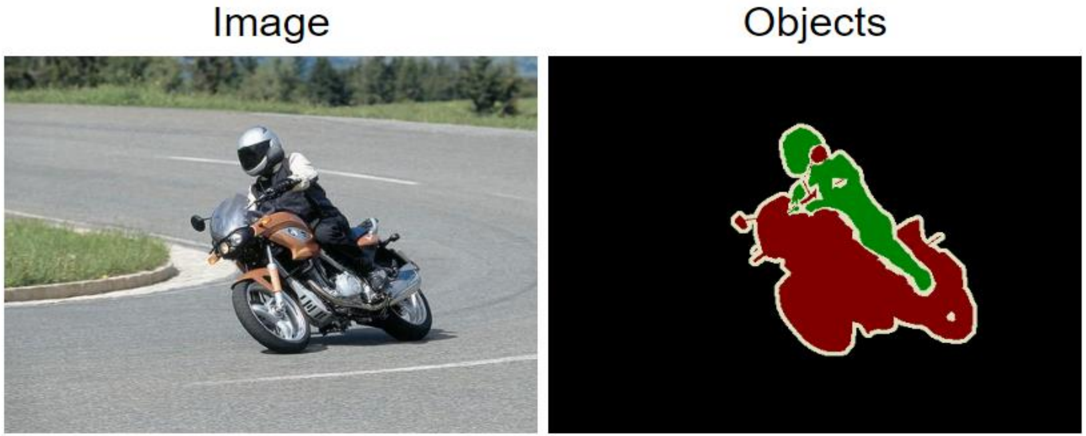
Person, animals, vehicles, indoor.

Number of samples

6929 Pixel-wise
instance level annotations.

Metric

Mean Intersection over Union.



http://host.robots.ox.ac.uk/pascal/VOC/voc2012/index.html

Common Objects in Context

COCO-Stuff augments 164K images with pixel-level stuff annotations for semantic segmentation.

Annotations

91 stuff classes (wall, grass, etc.) and 80 thing classes (person, elephant, etc.), as well as captions.

Number of samples

164000 dense pixel-level annotations and instance level annotations for things.

Metric

Mean Intersection over Union.

https://github.com/nightrome/cocostuff

MedakaHeart

Dataset to enable heart ventricle segmentation in Medaka fish for quantification of ventricular dimensions.

Annotations

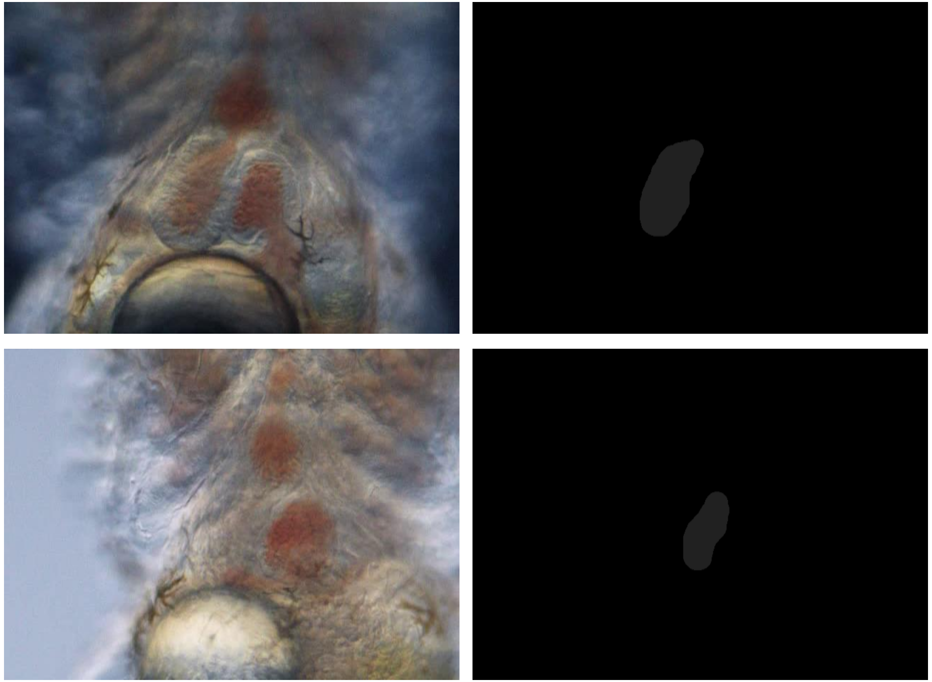
Medaka heart ventricle.

Number of samples

1725 binary pixel-level annotations.

Metric

Dice coefficient, ventricular dimensions.



https://osf.io/snb6p/

Cityscapes

The Cityscapes Dataset focuses on semantic understanding of urban street scenes.

Annotations

City scene semantic and instance-wise pixel annotations (road, person, pole, etc.).

https://www.cityscapes-dataset.com/

Number of samples

30 classes in 5000 fine and 20000 coarse annotated images.

Metric

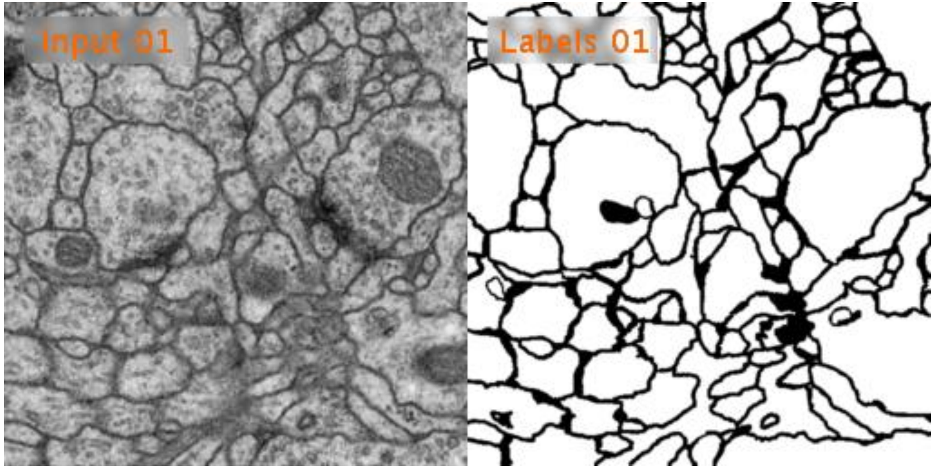
Mean Intersection over Union and Instance Intersection over Union.

ISBI Segmentation Challenge

Goals is the automatic segmentation of neural structure.

Annotations

Transmission Electron Microscopy (ssTEM) of the Drosophila first instar larva ventral nerve cord (VNC).



http://brainiac2.mit.edu/isbi_challenge/

Number of samples

30 consecutive images and the labelled boundary map.

Metric

Rand Scoring (similarity measurement between partitions).

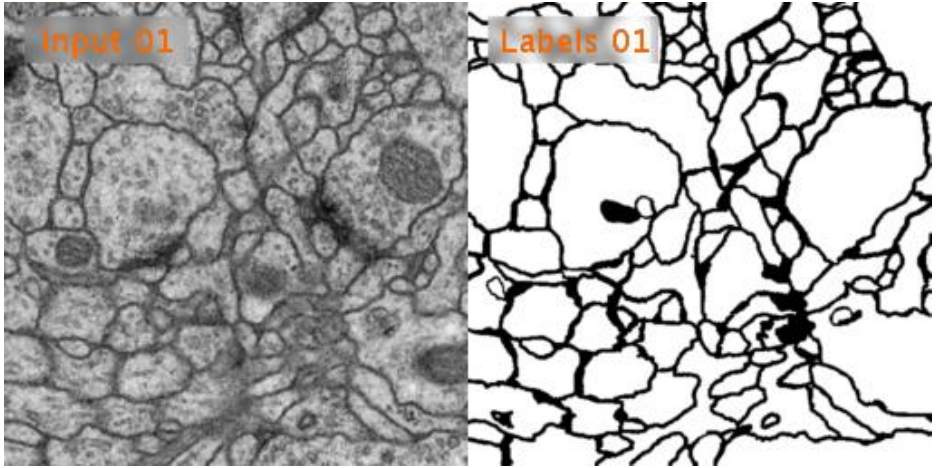
So how many samples do we need for segmentation?

So how many samples do we need for segmentation?,

Is an ill-posed question.

So how many samples do we need for segmentation?,
Is an ill-posed question.

Variance of the dataset
If your data is sequential or otherwise very similar you will need more samples.



http://brainiac2.mit.edu/isbi_challenge/

Instances per sample
Complexity of the task

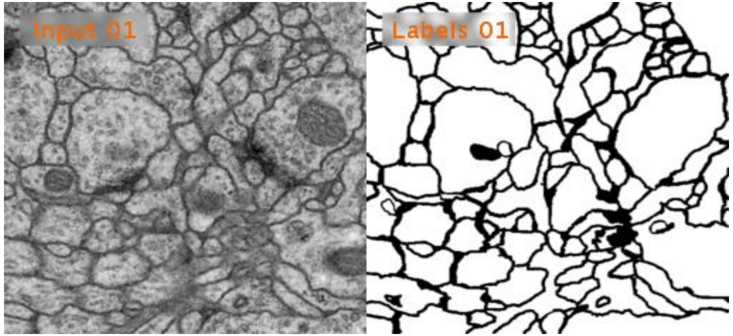
Neural Network Segmentation - Datasets and benchmarking

So how many samples do we need for segmentation?,
Is an ill-posed question.

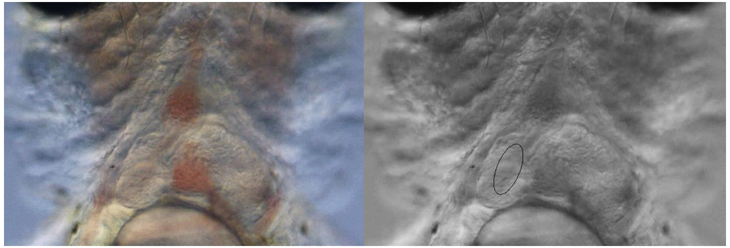
Variance of the dataset

Instances per sample

Data augmentation approaches, such as cropping can harness the availability of multiple instances in one sample.



http://brainiac2.mit.edu/isbi_challenge/



https://osf.io/snb6p/

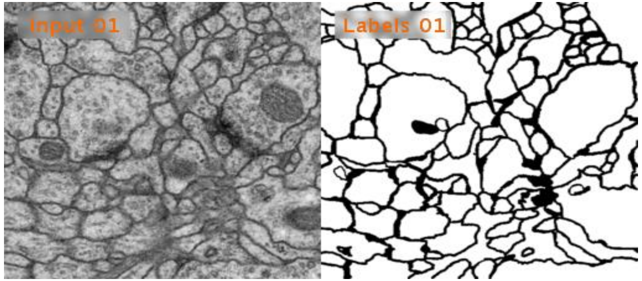
Complexity of the task

Neural Network Segmentation - Datasets and benchmarking

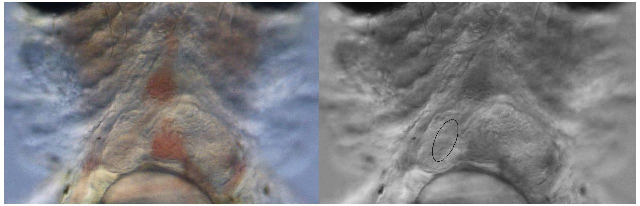
So how many samples do we need for segmentation?,
Is an ill-posed question.

Variance of the dataset
Instances per sample

Complexity of the task
The more complex the task, such as variance in shape, orientation, or intra sample variance,
the more data is necessary to generalize well enough.



http://brainiac2.mit.edu/isbi_challenge/



https://osf.io/snb6p/

Architectures over time

Fully Convolutional Network	2015
ParseNet	2015
Convolutional and Deconvolutional Networks	2015
U-Net	2015
Feature Pyramid Network	2016
Mask R-CNN	2017
DeepLab	2017

Fully Convolutional Network

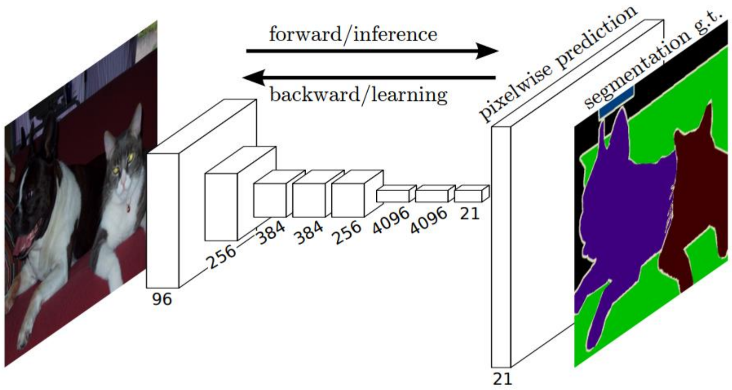
First end-to-end trained Fully Convolutional Network for image segmentation.

Transfer Learning approach, modifying well known architectures (such as VGG16).

Ending with an upsampling layer with one channel per class.

Fully Convolutional Networks for Semantic Segmentation

Jonathan Long* Evan Shelhamer* Trevor Darrell
UC Berkeley
{jonlong,shelhamer,trevor}@cs.berkeley.edu

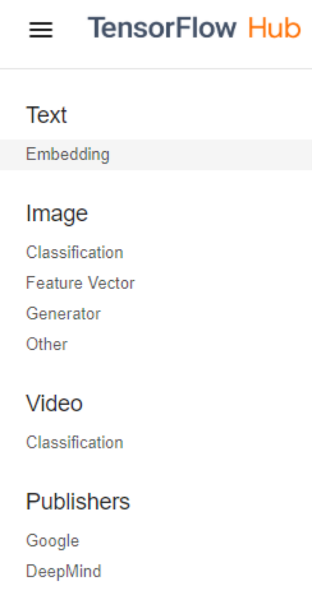


https://people.eecs.berkeley.edu/~jonlong/long_shelhamer_fcn.pdf

TensorFlow Hub

A library *tensorflow_hub* for reusable machine learning modules in TensorFlow.

Such as text embedding, image feature vectors, and video classification.



https://tfhub.dev/

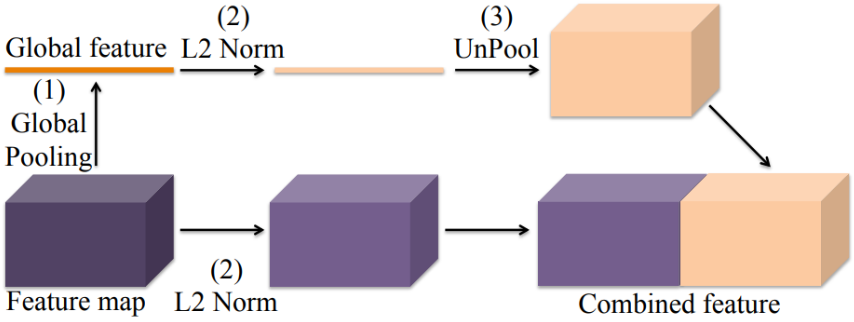
```python
import tensorflow_hub as hub


def create_module_graph(module_spec):
    """Creates a graph and loads Hub Module into it.
    Args:
      module_spec: the hub.ModuleSpec for the image module being used.
    Returns:
      graph: the tf.Graph that was created.
      bottleneck_tensor: the bottleneck values output by the module.
      resized_input_tensor: the input images, resized as expected by the module.
      wants_quantization: a boolean, whether the module has been instrumented
        with fake quantization ops.
    """
    height, width = hub.get_expected_image_size(module_spec)
    with tf.Graph().as_default() as graph:
      resized_input_tensor = tf.placeholder(tf.float32, [None, height, width, 3])
      m = hub.Module(module_spec)
      bottleneck_tensor = m(resized_input_tensor)
      wants_quantization = any(node.op in FAKE_QUANT_OPS
                               for node in graph.as_graph_def().node)
    return graph, bottleneck_tensor, resized_input_tensor, wants_quantization
```

https://github.com/tensorflow/hub/blob/master/examples/image_retraining/retrain.py

ParseNet

The global context of an image helps segmentation.

This is done by depicting the context in a global feature, which is later combined with the feature map.



https://arxiv.org/pdf/1506.04579.pdf

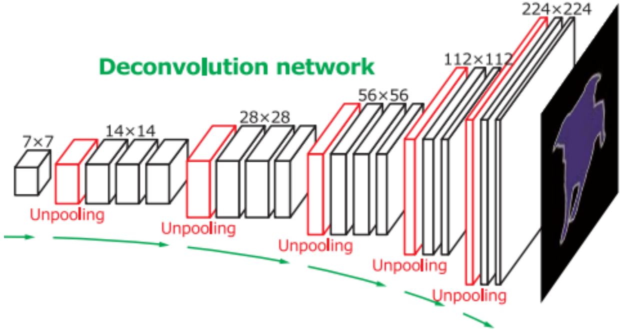
Neural Network Segmentation - Overview state-of

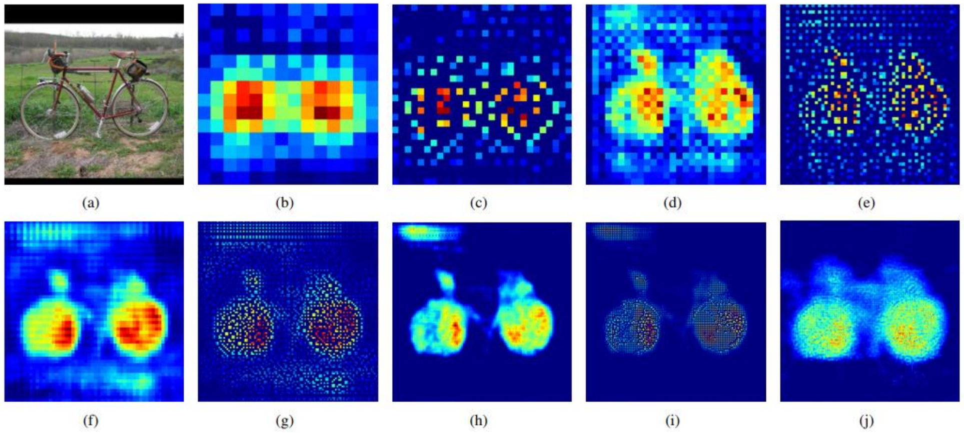
Convolutional and Deconvolutional Networks

Introducing a encoder-decoder architecture.

From the convolutional encoding, the deconvolution branch generates a dense pixel-wise class probability map, by successive:

Unpooling, deconvolutions, and rectifications.





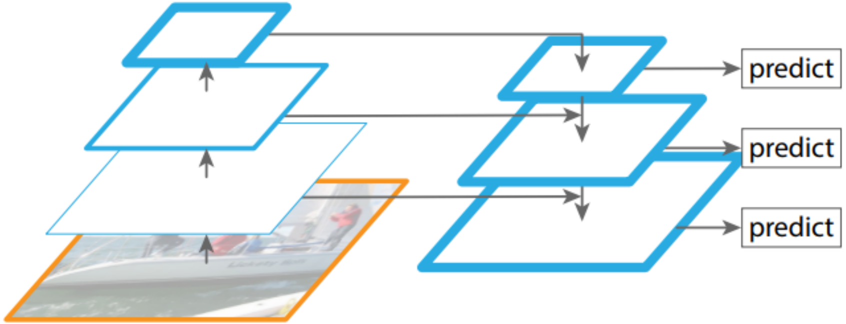
https://arxiv.org/pdf/1505.04366.pdf

Feature Pyramid Network

Introducing a bottom-up pathway to generate features at different scales.

Combined with a top-down pathway doing the upsampling, while infusing the high-level features through lateral connections.

On each stage of the pyramid a segmentation mask is predicted.

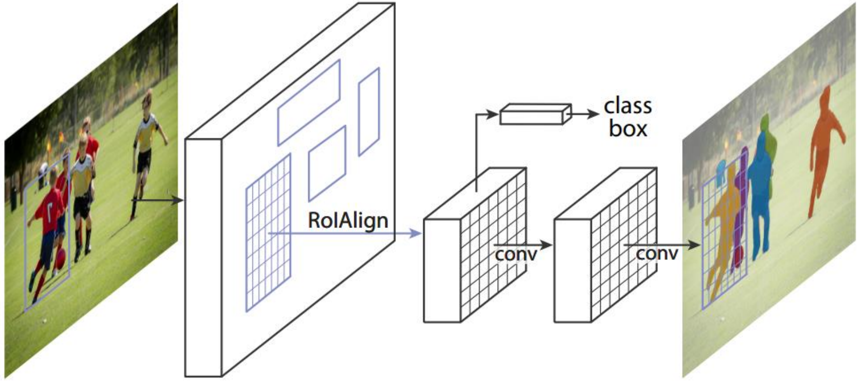


https://arxiv.org/pdf/1612.03144.pdf

Mask R-CNN

Using a Region Proposal Network to extract a manageable number of regions of interest.

A multi-task loss, combining losses for the bounding box coordinates, the class prediction and the segmentation mask improve the performance.

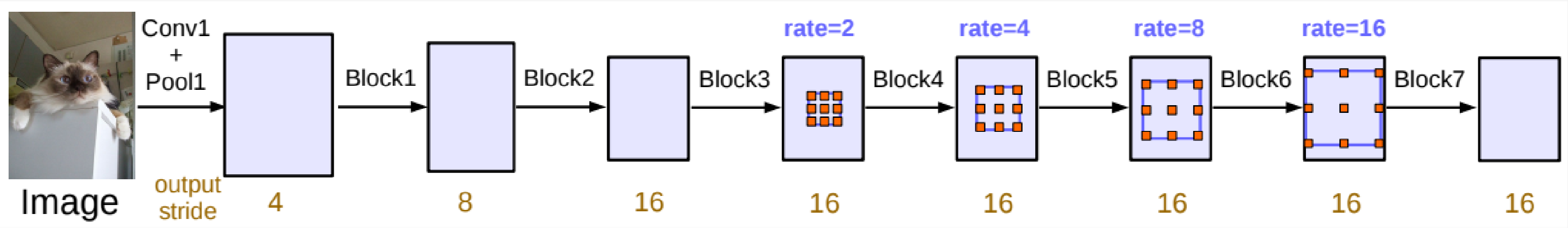


https://arxiv.org/pdf/1703.06870.pdf

DeepLabv3

Combining Atrous Convolutions (dilated convolutions) with a pyramidal architecture.

Atrous convolutions replace a combination of pooling, convolution and unpooling.



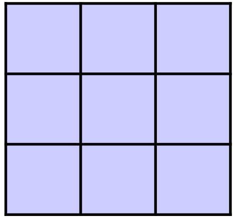
https://arxiv.org/pdf/1706.05587.pdf

Atrous Convolution

Introducing an additional parameter, called the dilation rate or rate.

Defining a spacing between the values in a filter map.

Atrous Convolution



Filter 3x3

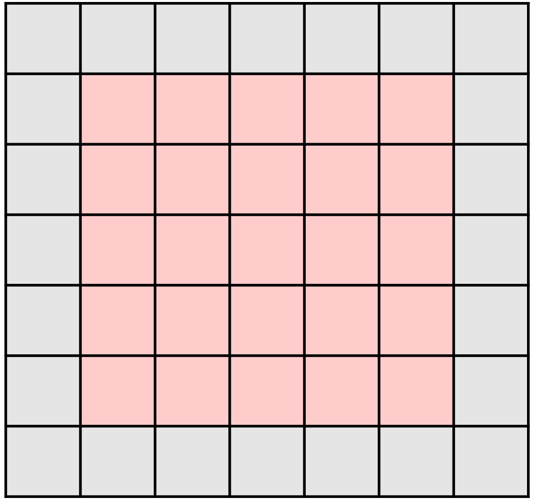
Dilation rate: 2

Stride: 1

Padding: 1

Introducing an additional parameter, called the dilation rate or rate.

Defining a spacing between the values in a filter map.

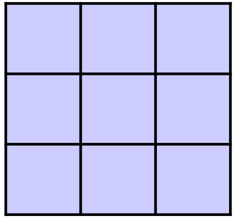


Atrous Convolution

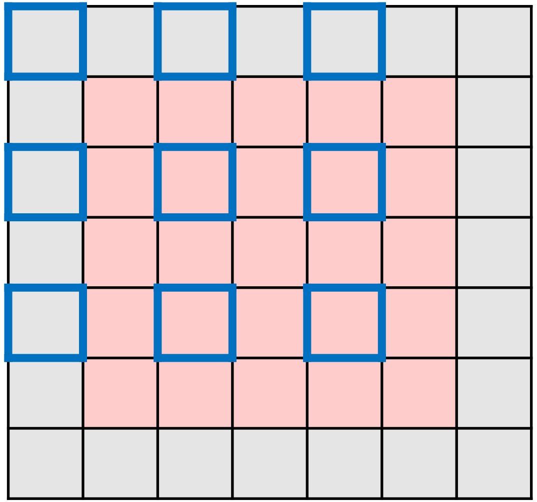
Introducing an additional parameter, called the dilation rate or rate.

Defining a spacing between the values in a filter map.

This enhances the field of view while keeping the computational cost low.



Filter 3x3
Dilation rate: 2
Stride: 1
Padding: 1



Machine Learning Methods for Automated Quantification of Ventricular Dimensions

Mark Schutera[1], Steffen Just[2], Jakob Gierten[3, 4], Ralf Mikut[1], Markus Reischl[1], Christian Pylatiuk[1*]

1 Institute for Automation and Applied Informatics (IAI), Karlsruhe Institute of Technology (KIT), Hermann-von-Helmholtz-Platz 1, 76344 Eggenstein, Germany
2 Department of Internal Medicine II, University of Ulm, Albert-Einstein-Allee 23, 89081 Ulm, Germany
3 Department of Pediatric Cardiology, University Hospital Heidelberg, Im Neuenheimer Feld 430, 69120 Heidelberg, Germany
4 Centre for Organismal Studies Heidelberg, Heidelberg University, Im Neuenheimer Feld 230, 69120 Heidelberg, Germany

Machine Learning Methods
for Automated Quantification of Ventricular Dimensions

Project Description

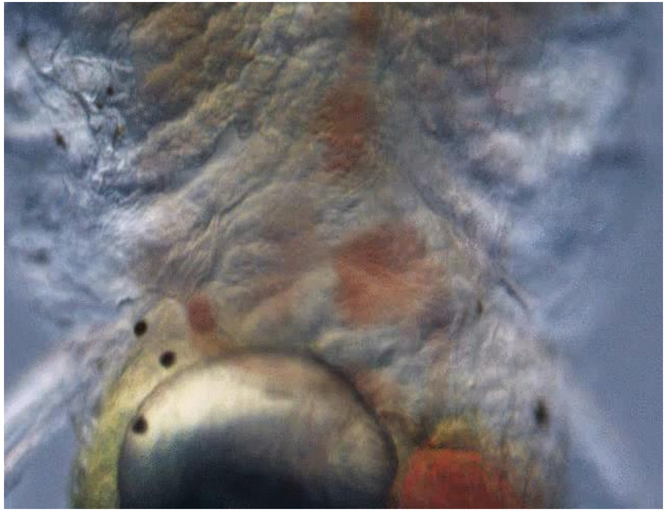
Medaka fish contribute to the understanding of human cardiovascular diseases.
In this context the quantification of cardiac functional parameters is fundamental.

Project Description

Medaka fish contribute to the understanding of human cardiovascular diseases.

In this context the quantification of cardiac functional parameters is fundamental.

Medaka prove to be a valuable whole organism-based in vivo model due to the visibility of the heart and other organs.



Project Description

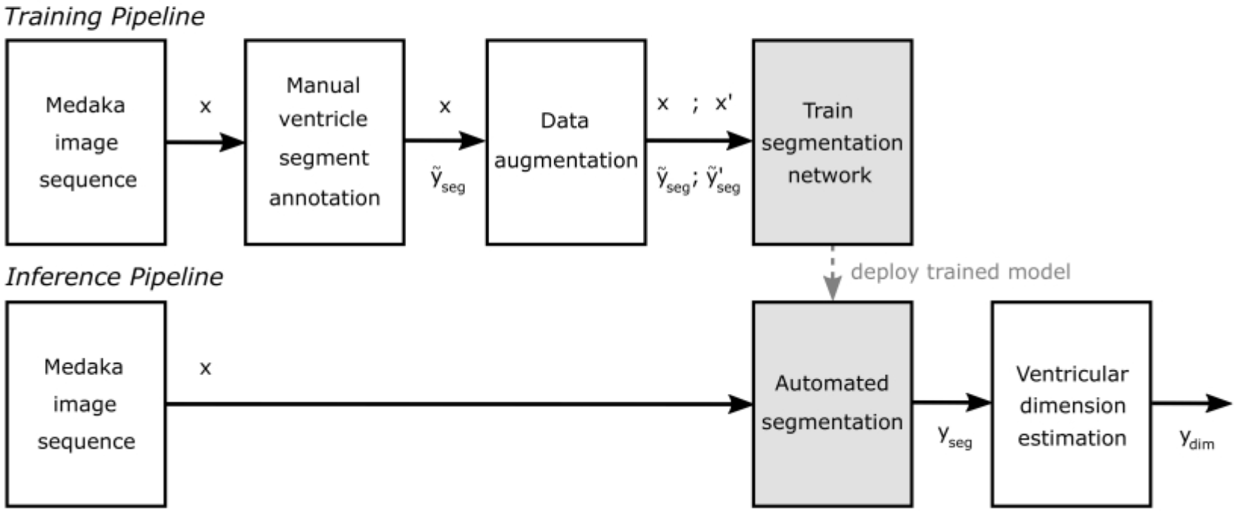
Medaka fish contribute to the understanding of human cardiovascular diseases.
In this context the quantification of cardiac functional parameters is fundamental.

Medaka prove to be a valuable whole organism-based in vivo model due to the visibility of the heart and other organs.

Automated screening pipelines are thus invaluable for biomedical research.

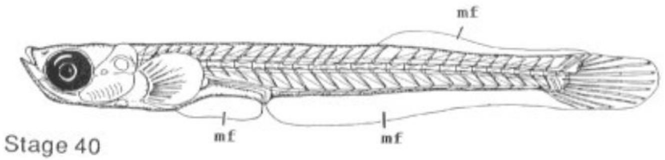
Development of a pipeline for automated measurement and quantification of important cardiac functional parameters.

Project Description



Data Recording

Medaka larvae were imaged 1-2 days after hatching.



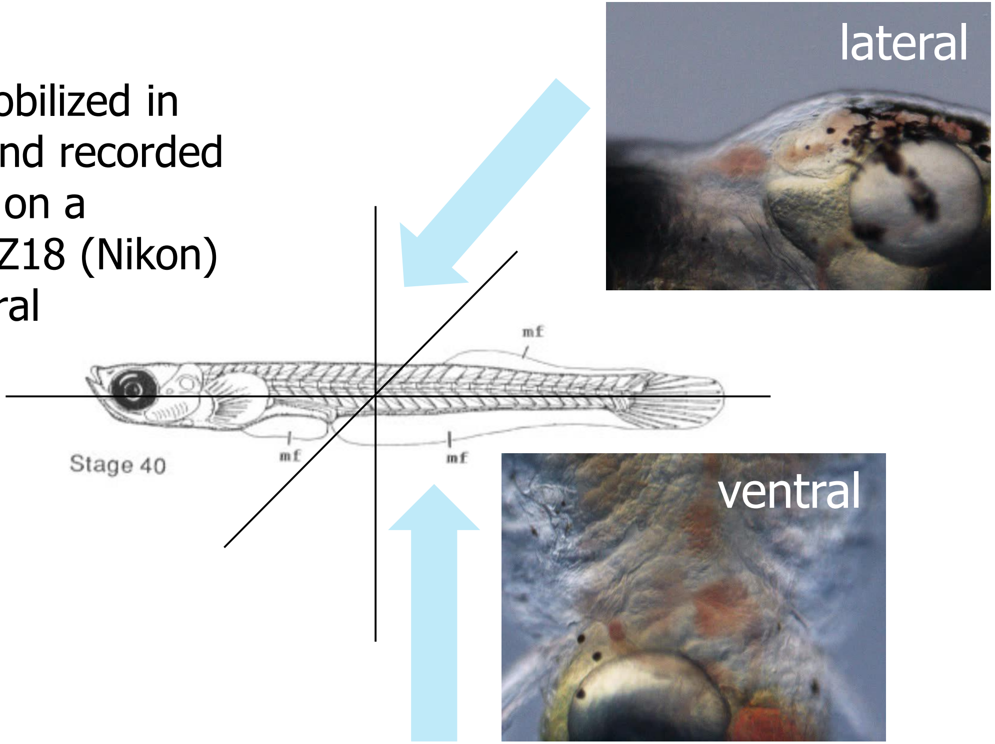
http://mepd.cos.uni-heidelberg.de/mepd/forms/developmentalStages.jsf

Medaka image sequence

Manual ventricle segment annotation

Data Recording

Hatchlings were immobilized in 3% methylcellulose and recorded at room temperature on a stereomicroscope SMZ18 (Nikon) from ventral and lateral



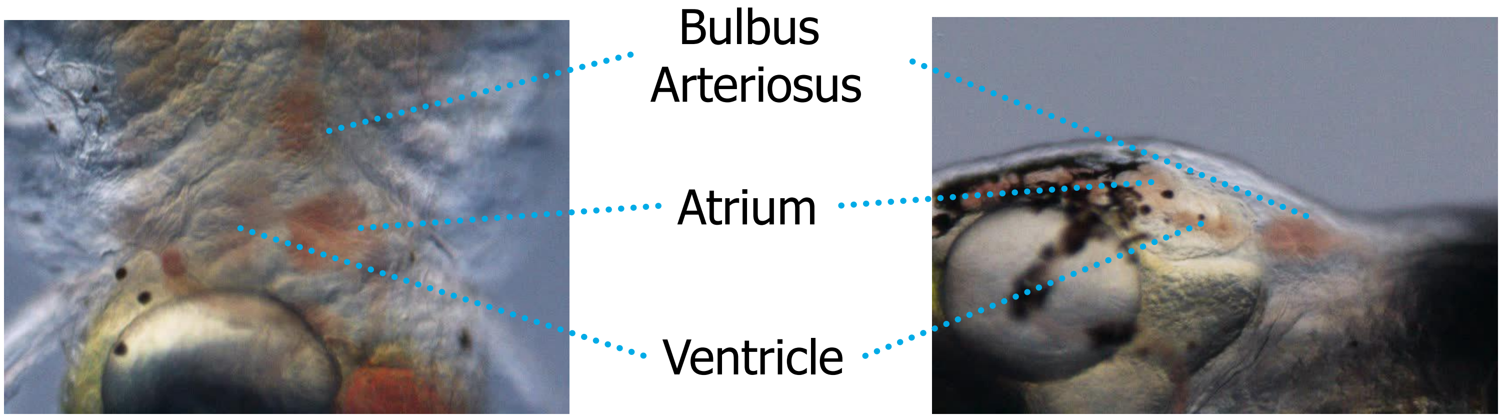
Medaka image sequence

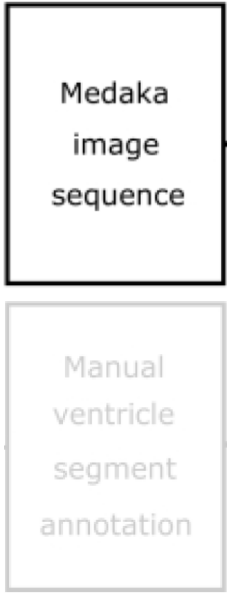
Manual ventricle segment annotation

http://mepd.cos.uni-heidelberg.de/mepd/forms/developmentalStages.jsf

Data Recording

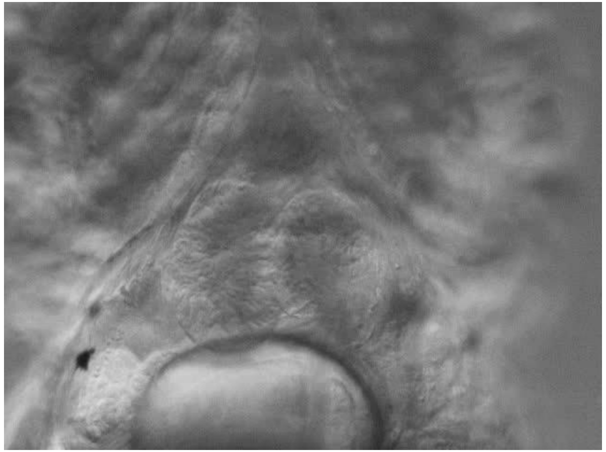
At zoom magnification of 6x with 640x480 pixels (1 px = 1,115 µm) and 15 frames per second (fps).

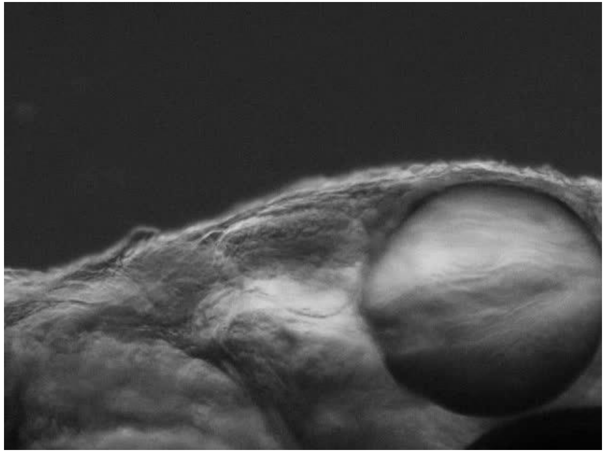


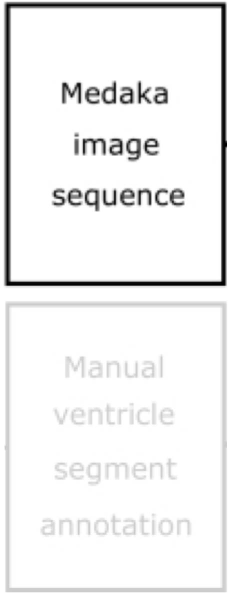


Data Recording

To reduce the computing effort, the images are converted to grayscale

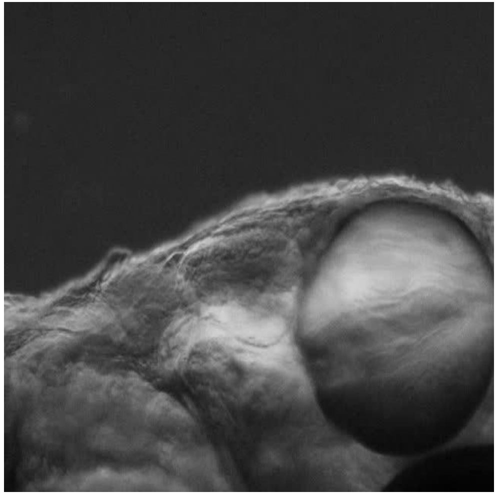


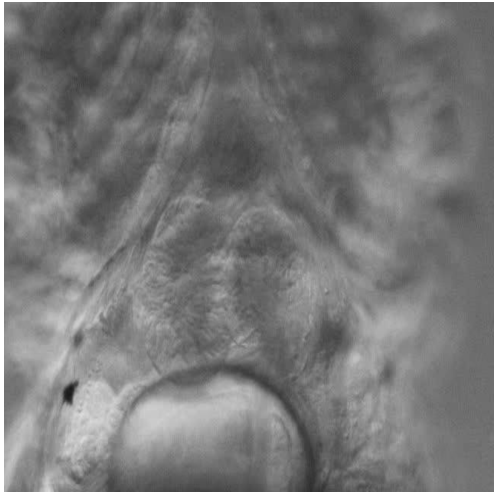




Data Recording

To reduce the computing effort, the images are converted to grayscale and rescaled to 256 × 256 px (.tif).





Medaka image sequence

Manual ventricle segment annotation

Data Recording

The data is extracted from 63 videos in total comprising approx. 115 sec of image sequences or 1725 frames.

Data Recording

The data is extracted from 63 videos in total comprising approx. 115 sec of image sequences.

Training and Validation Set

725 frames from 29 ventral sequences

500 frames from 20 lateral sequences

Medaka image sequence

Manual ventricle segment annotation

Data Recording

The data is extracted from 63 videos in total comprising approx. 115 sec of image sequences.

Test Set Ventral

150 consecutive frames from a single sequence

5x25 consecutive frames from 5 sequences

Medaka image sequence

Manual ventricle segment annotation

Data Recording

The data is extracted from 63 videos in total comprising approx. 115 sec of image sequences.

Training Set Lateral

150 consecutive frames from a single sequence

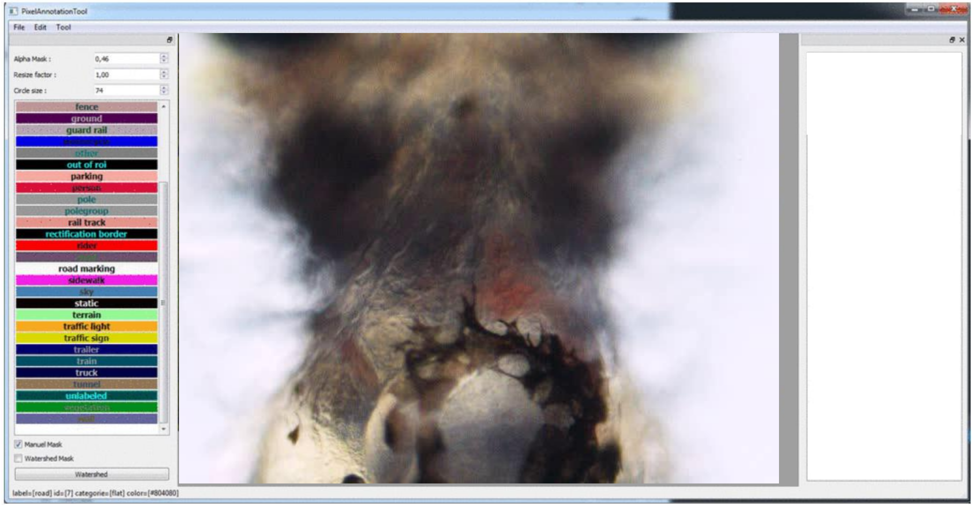
5x25 consecutive frames from 5 sequences

Medaka image sequence

Manual ventricle segment annotation

Data Annotation

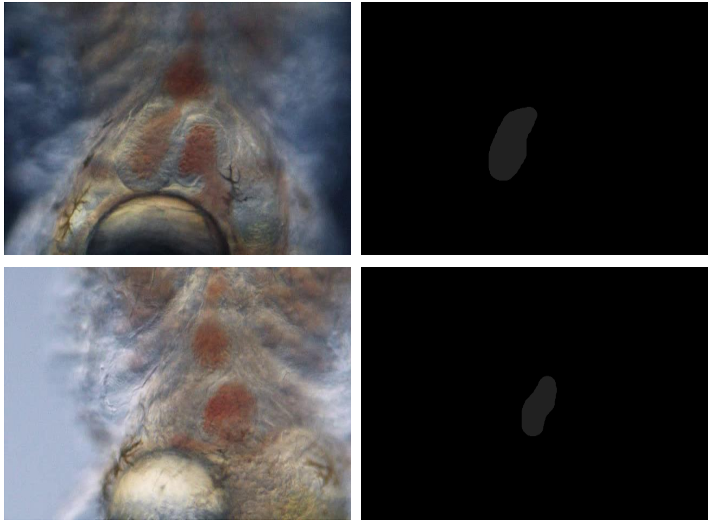
Each frame of the dataset has been annotated individually by manually labeling the present heart segment / ventricle, using the brush-tool of the pixel annotation tool v1.3.1.



https://github.com/abreheret/PixelAnnotationTool

Data Annotation

The annotations are binary image masks (.tiff) whereas ground truth pixels are assigned to 1 and background pixels to 0.



https://osf.io/snb6p/

Medaka image sequence

Manual ventricle segment annotation

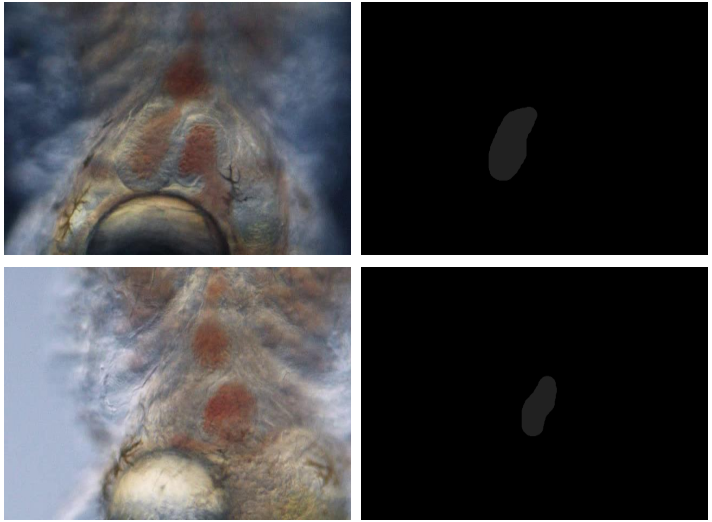
Data augmentation

Data Annotation

The annotations are binary image masks (.tif) whereas ground truth pixels are assigned to 1 and background pixels to 0.

Time devoted per frame
25 seconds

Time devoted for the dataset
14 hours



https://osf.io/snb6p/

Medaka image sequence

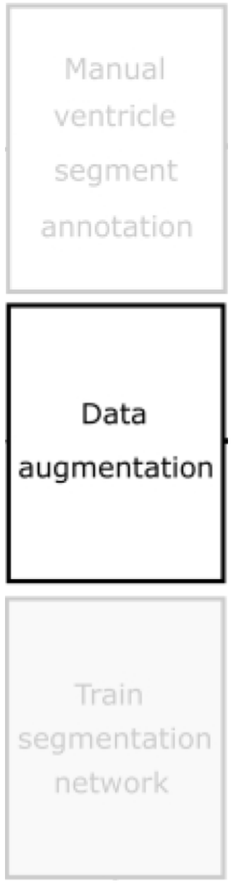
Manual ventricle segment annotation

Data augmentation

Data Augmentation

Confronted with the great effort required for data annotation, and the subsequent low amount of available ground truth (1225 frames),

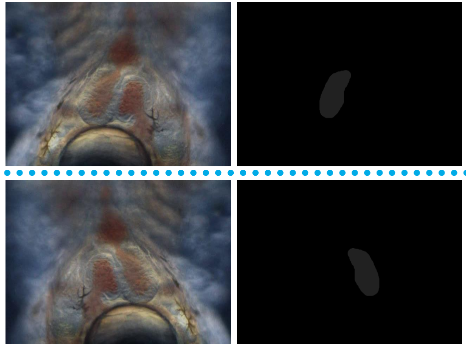
data augmentation methods are applied.



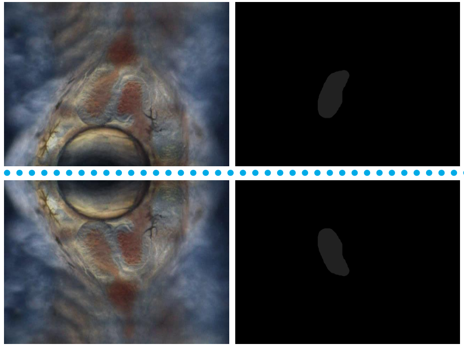
Data Augmentation

The samples and the according ground truth are randomly augmented during training, by applying a set of augmentation methods.

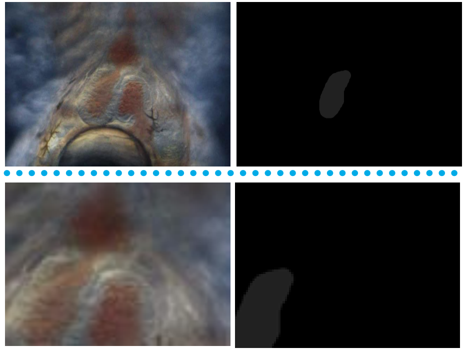
horizontal flips



vertical flips



Zoom [0;0.1]



Manual ventricle segment annotation

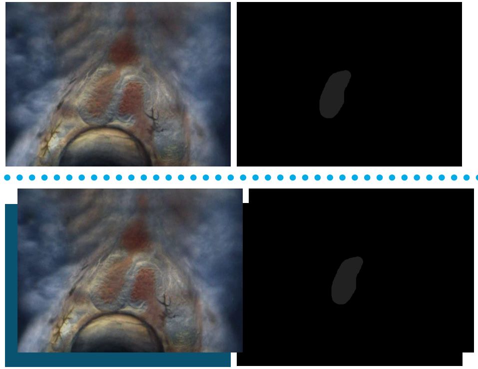
Data augmentation

Train segmentation network

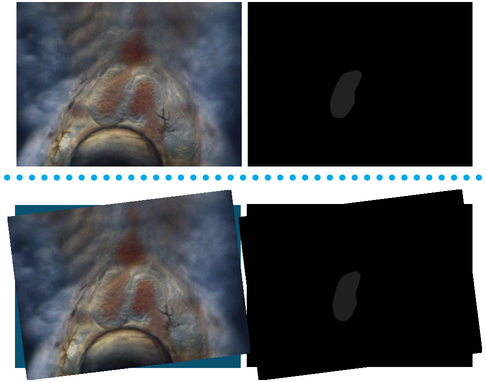
Data Augmentation

The samples and the according ground truth are randomly augmented during training, by applying a set of augmentation methods.

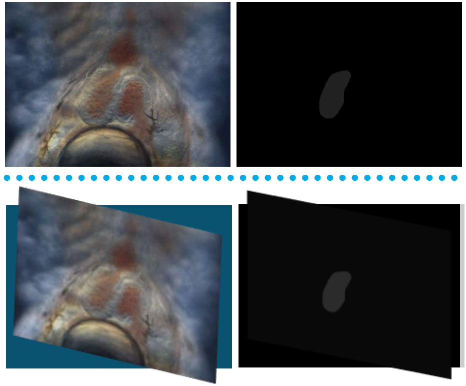
height and width shift [0;0.05]



rotation [0;0.2]



shear [0;0.1]



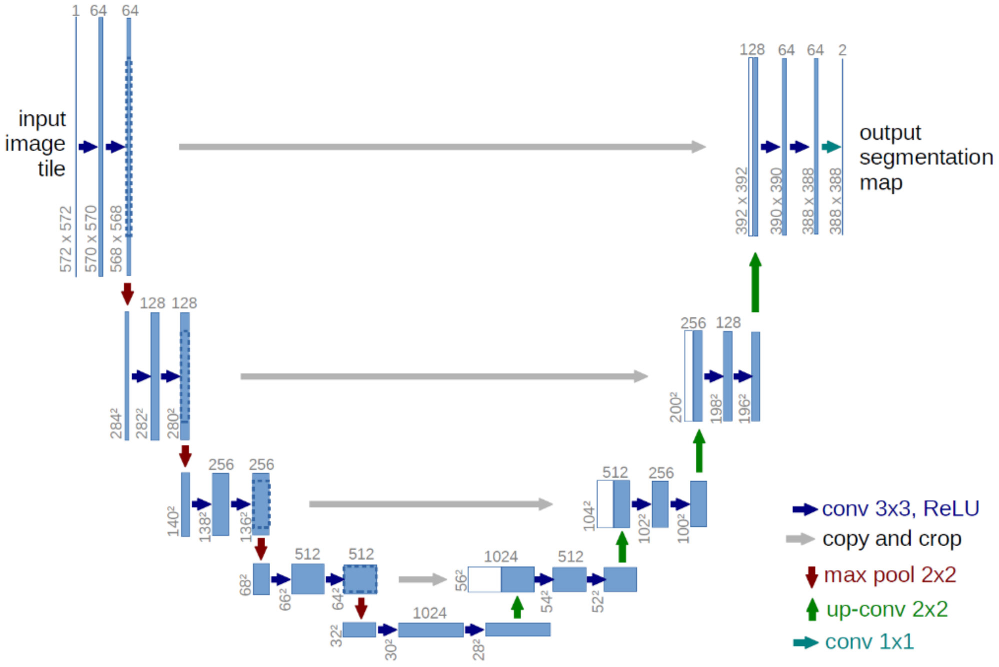
Manual ventricle segment annotation

Data augmentation

Train segmentation network

Deep Dive U-Net

The U-Net is a symmetric, deep convolutional neural network.



https://lmb.informatik.uni-freiburg.de/people/ronneber/u-net/

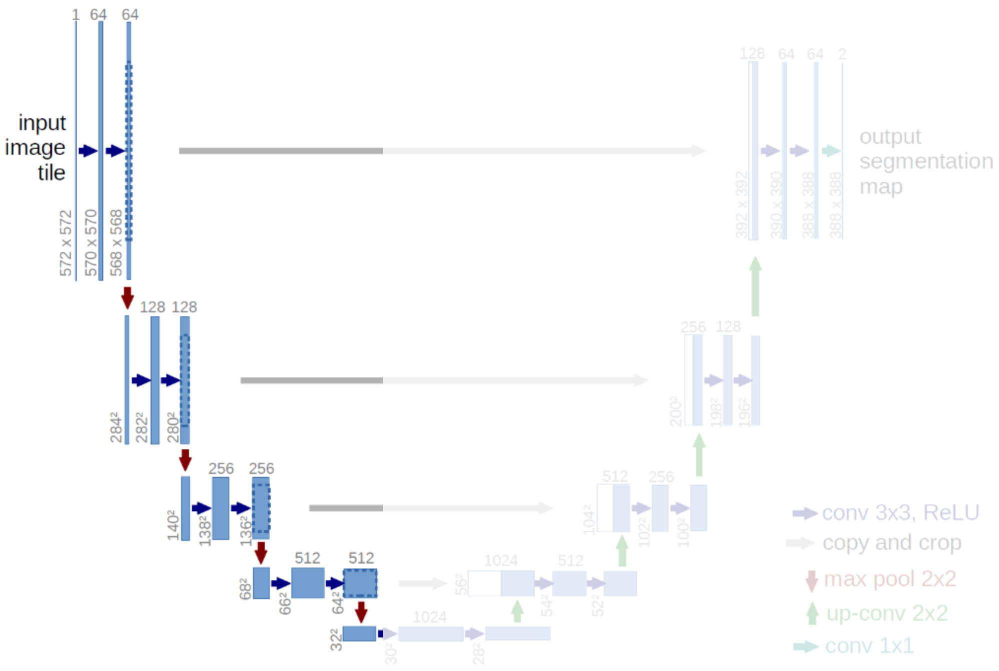
Data augmentation

Train segmentation network

Ventricular dimension estimation

Deep Dive U-Net

The down sampling path computes high-level features with semantic information.



https://lmb.informatik.uni-freiburg.de/people/ronneber/u-net/

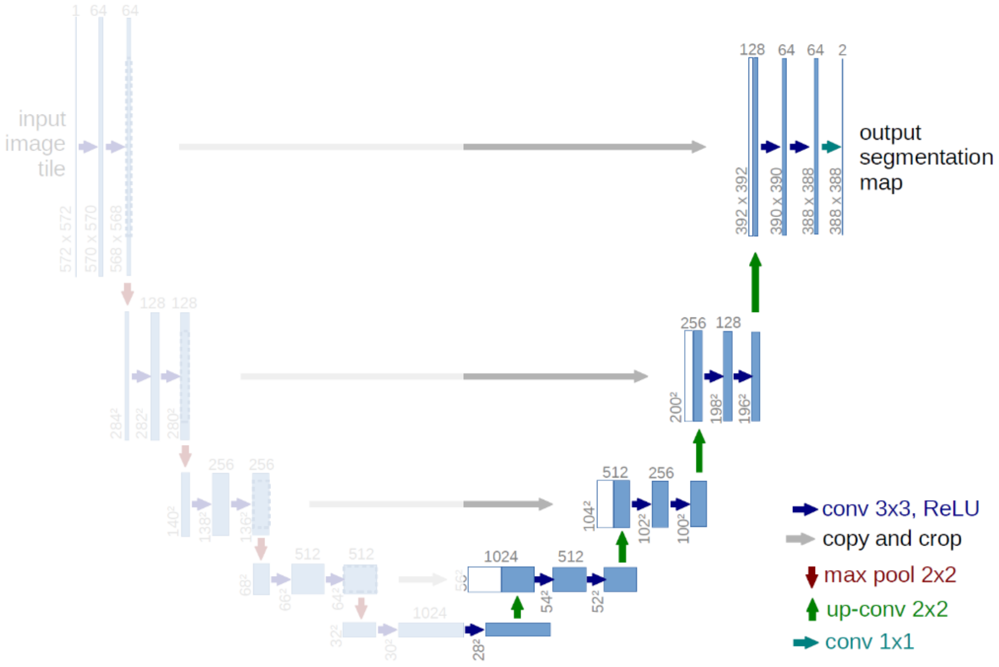
Data augmentation

Train segmentation network

Ventricular dimension estimation

Deep Dive U-Net

The up sampling path computes spatially localizes patterns in the image.



https://lmb.informatik.uni-freiburg.de/people/ronneber/u-net/

Data augmentation

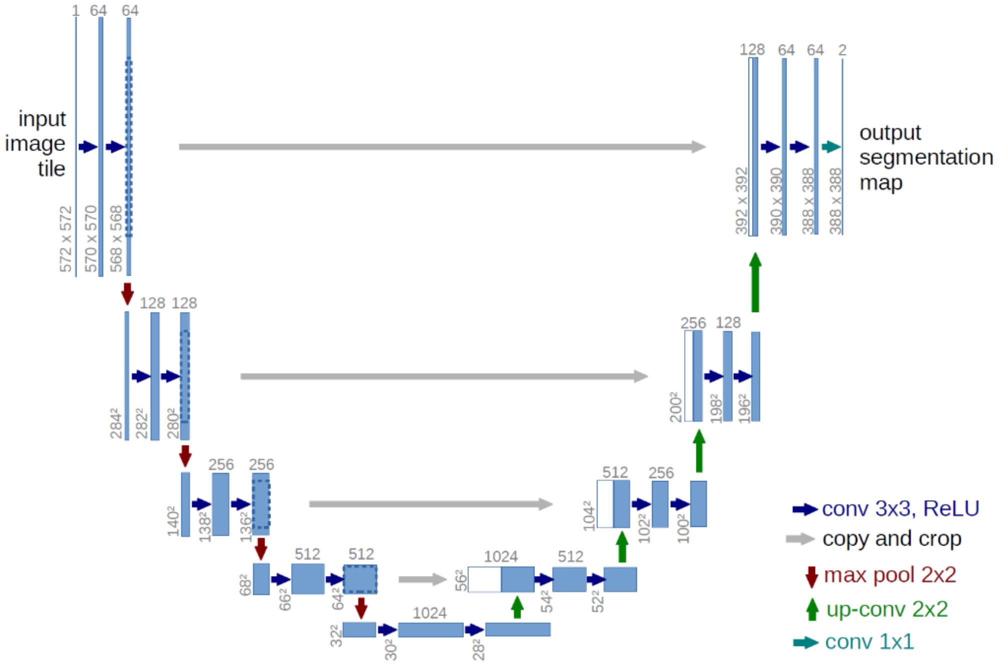
Train segmentation network

Ventricular dimension estimation

Deep Dive U-Net

Both paths are brought together by skip connections.

Combining semantic patterns with localization information.



https://lmb.informatik.uni-freiburg.de/people/ronneber/u-net/

Data augmentation

Train segmentation network

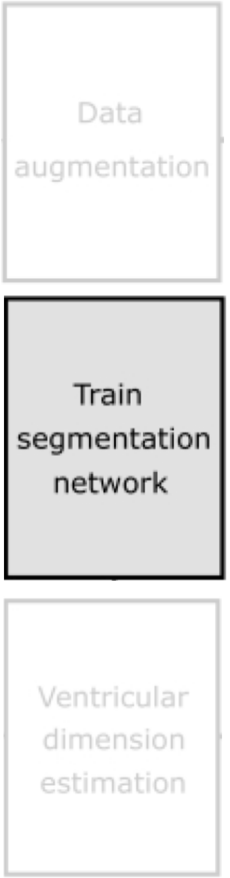
Ventricular dimension estimation

Neural Network Segmentation - Deep Dive U-Net

Training Parameters

Adam optimizer with a learning rate of $\epsilon = 1e^{-5}$
Batch size of 8
Training schedule 60 epochs
300 iterations per epoch

Data augmentation
Batch normalization in the convolution layers
Dropout in the bottleneck layers



Neural Network Segmentation - Deep Dive U-Net

Dice loss

Similar to the IoU (Intersection over union), and thus easy to interpret.

+1 is a smoothing factor for numeric stability.

Robust with respect to imbalanced classes.

$$dice\ loss = 1 - \frac{2\sum(\tilde{\mathbf{y}} \odot \mathbf{y}_{seg}) + 1}{\sum(\tilde{\mathbf{y}}^2) + \sum(\mathbf{y}_{seg}^2) + 1}.$$

Data augmentation

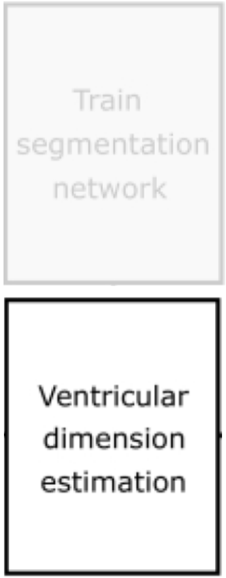
Train segmentation network

Ventricular dimension estimation

Ventricular dimensions

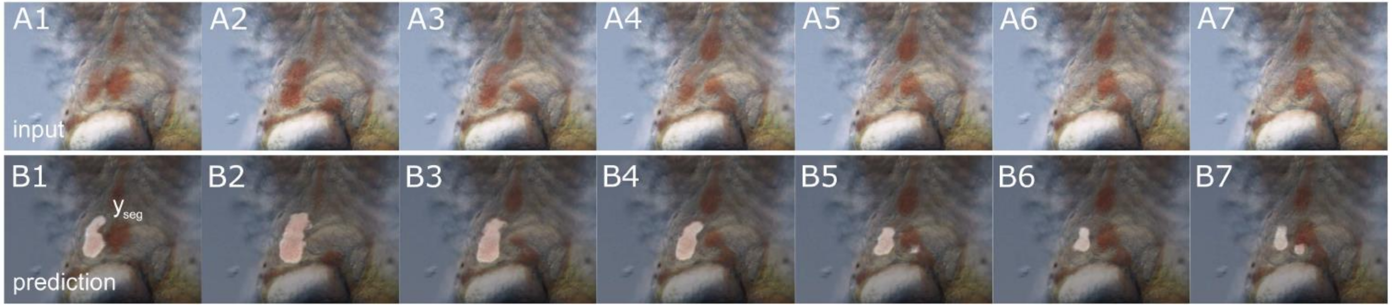
Heart rate
Ventricular volumes
Fractional shortenings

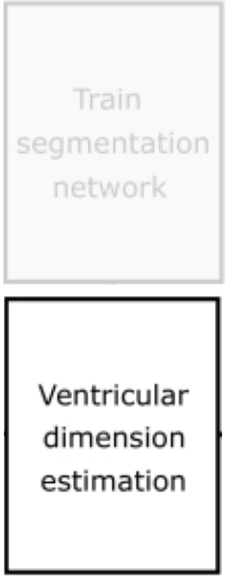
are linked to the **cardiovascular cycle**.



Neural Network Segmentation - Ventricular dimensions

To estimate the ventricular dimensions, the frame-based predicted segments have to be **transferred into the temporal domain.**





Neural Network Segmentation - Ventricular dimensions

To estimate the ventricular dimensions, the frame-based predicted segments have to be transferred into the temporal domain.

And we need to determine the cardiovascular cycle frequency.

For example through a feature such as the **segment area *a* in px over time**

$$a^{(t)} = \sum_{u=0}^{\substack{image\\width}} \sum_{v=0}^{\substack{image\\height}} (\mathbf{y}_{seg}(u,\ v))^{(t)}$$

Train segmentation network

Ventricular dimension estimation

Neural Network Segmentation - Ventricular dimensions

To estimate the ventricular dimensions, the frame-based predicted segments have to be transferred into the temporal domain.

And we need to determine the cardiovascular cycle frequency.

Train segmentation network

Ventricular dimension estimation

For example through a feature such as the **segment area *a* in px over time**

$$a^{(t)} = \sum_{u=0}^{image\ width} \sum_{v=0}^{image\ height} (\mathbf{y}_{seg}(u,\ v))^{(t)}$$

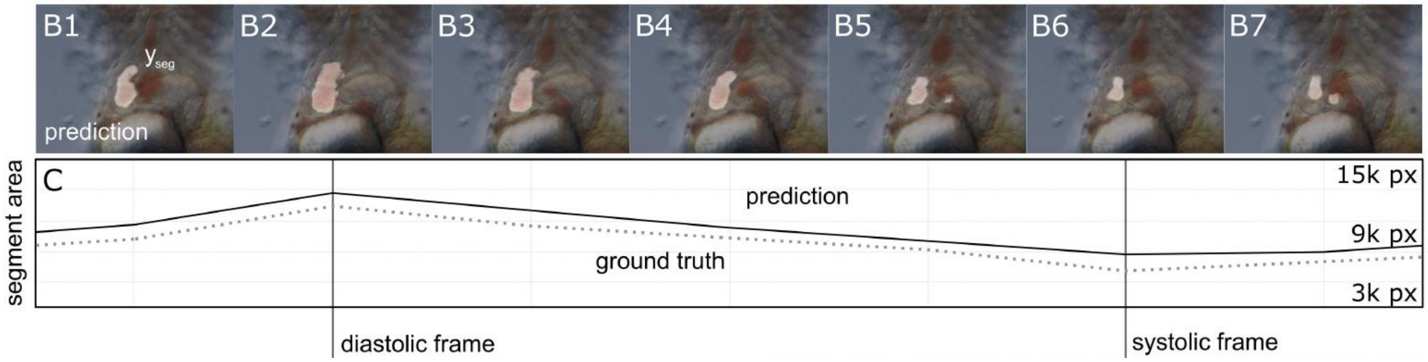
Other features could be minor axis, major axis, equivalent diameter, and so on

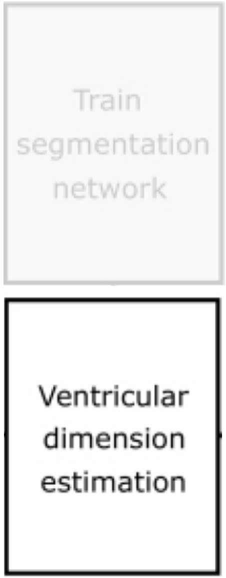
Neural Network Segmentation - Ventricular dimensions

To estimate the ventricular dimensions, the frame-based predicted segments have to be transferred into the temporal domain.

And we need to determine the cardiovascular cycle frequency.

And a sliding window approach for **segment area peak detection**

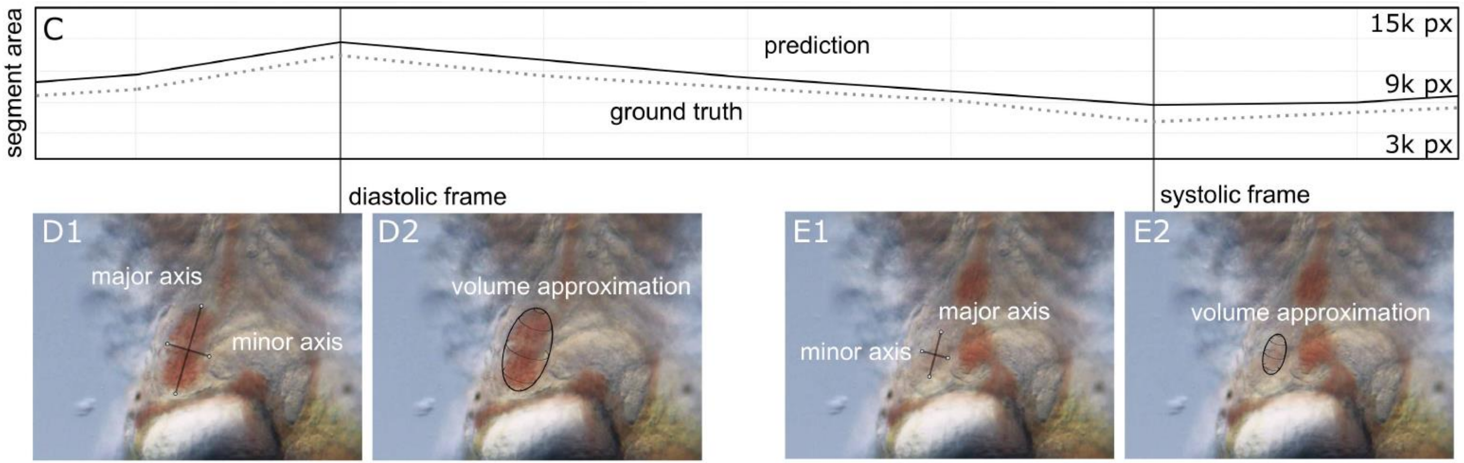


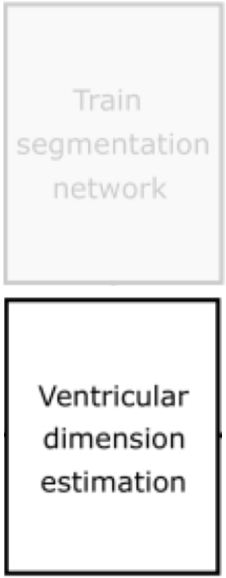


Neural Network Segmentation - Ventricular dimensions

What do the peaks mean?

Diastolic frame: Maximum heart relaxation with maximal heart volume
Systolic frame: Maximum heart tension with minimal heart volume





Heart rate

The heart rate (HR) is consequently estimated from,

offset o the amount of frames between two systolic or diastolic frames, which correspond to one heartbeat.

The beats per frame are subsequently transformed into the time space by compensating for the sample frequency f = 15fps.

$$HR = \frac{f}{o}$$

Ventricular volumes

Fractional shortenings

Train segmentation network

Ventricular dimension estimation

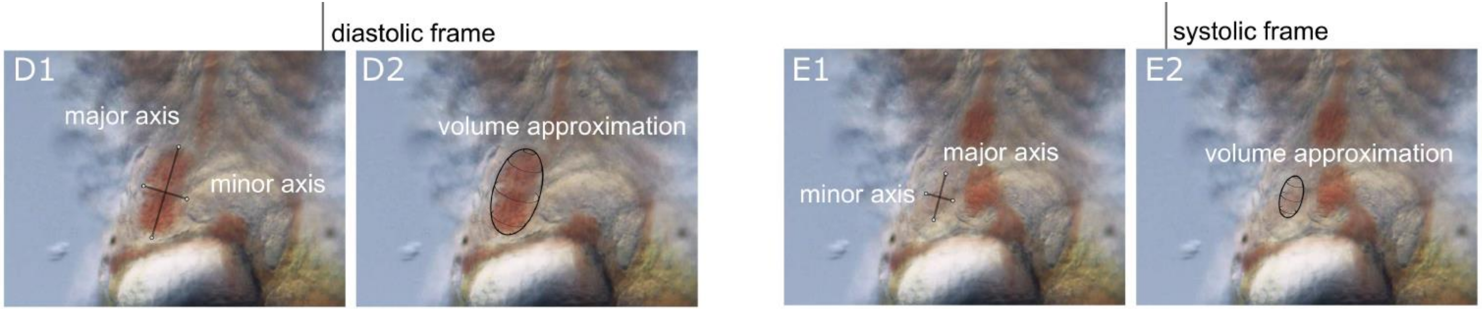
Neural Network Segmentation - Ventricular dimensions

Heart rate

Train segmentation network

Ventricular dimension estimation

Ventricular volumes



are approximated by a prolate spheroidal shape, based on the major and minor axis of the predicted heart segment. The minor axis e_{minor} and major axis e_{major} serve as semi-diameter:

$$V = \frac{1}{6}\,\pi\, e_{minor}^2\, e_{major}$$

Fractional shortenings

Neural Network Segmentation - Ventricular dimensions

Heart rate

Ventricular volumes

Fractional shortenings

is the fraction of a diastolic dimension (D_{dia}) that is lost in systole (D_{sys}).

The most important fractional shortening is the shortening of the ventricles volume, or ejection fraction is thus given by the percentage of volume change between a diastolic frame and its subsequent systolic frame.

$$V = \frac{V_{dia} - V_{sys}}{V_{dia}}$$

Train segmentation network

Ventricular dimension estimation

Neural Network Segmentation - Deep Dive U-Net

Results

Automated heart segmentation

Detection rate for diastolic / systolic frame pairs is 100%.

Mean error 0.53 frames and max error is 1.0 frames.

(Ground truth is in brackets)

Current heart beat	diastolic frame; systolic frame	heart rate [*bps*]	minor fractional shortening [%]	major fractional shortening [%]
1.	2; 6 (2; 7)	1.50 (1.50)	41.33 (47.29)	52.18 (38.80)
2.	11; 15 (11; 16)	1.67 (1.67)	40.87 (30.90)	57.58 (38.69)
3.	19; 23 (19; 24)	1.58 (1.58)	28.57 (48.92)	53.46 (34.57)
4.	28; 31 (28; 32)	1.46 (1.67)	29.37 (43.21)	47.16 (28.68)
5.	36; 42 (36; 40)	1.69 (1.50)	42.56 (45.43)	60.73 (41.05)
6.	45; 49 (45; 49)	1.52 (1.50)	39.36 (42.56)	58.66 (31.75)
7.	53; 59 (54; 58)	1.82 (1.67)	33.81 (45.71)	61.26 (34.37)
8.	62; 65 (62; 66)	1.58 (1.52)	38.61 (47.74)	58.32 (49.29)
9.	70; 74 (70; 76)	1.52 (1.58)	50.33 (43.99)	53.06 (37.22)
10.	78; 84 (79; 84)	1.58 (1.50)	44.15 (49.85)	55.30 (31.20)
11.	87; 92 (88; 93)	1.58 (1.67)	39.93 (45.48)	43.54 (39.53)
12.	96; 100 (96; 101)	1.58 (1.67)	44.27 (47.76)	58.00 (44.15)
13.	104; 109 (104; 109)	1.58 (1.58)	47.96 (42.90)	59.27 (41.57)
14.	112; 118 (113; 117)	1.58 (1.58)	38.71 (46.30)	64.93 (20.96)
15.	121; 126 (122; 125)	1.58 (1.52)	41.75 (44.24)	55.37 (22.83)
16.	130; 134 (130; 135)	1.58 (1.58)	20.39 (52.80)	59.46 (37.15)
17.	138; 143 (139; 143)	1.67 (1.52)	43.92 (48.11)	52.80 (37.61)
18.	146; 151 (147; 153)	1.50 (1.50)	35.01 (36.93)	56.59 (35.89)
mean		1.59 (1.57)	38.93 (45.01)	55.98 (35.85)
std		0.08 (0.07)	7.10 (4.65)	4.98 (6.62)
median		1.59 (1.58)	38.93 (45.59)	55.98 (37.19)
lower quartile		1.53 (1.50)	7.10 (43.40)	4.98 (32.41)
upper quartile		1.58 (1.65)	7.10 (47.75)	4.98 (39.35)

Results

Automated heart segmentation

Detection rate 100%

Max detection error is 1 frame

Ventricular dimension estimation is robust over multiple cardiovascular cycles.

Current heart beat	diastolic frame; systolic frame	heart rate [*bps*]	minor fractional shortening [%]	major fractional shortening [%]
1.	2; 6 (2; 7)	1.50 (1.50)	41.33 (47.29)	52.18 (38.80)
2.	11; 15 (11; 16)	1.67 (1.67)	40.87 (30.90)	57.58 (38.69)
3.	19; 23 (19; 24)	1.58 (1.58)	28.57 (48.92)	53.46 (34.57)
4.	28; 31 (28; 32)	1.46 (1.67)	29.37 (43.21)	47.16 (28.68)
5.	36; 42 (36; 40)	1.69 (1.50)	42.56 (45.43)	60.73 (41.05)
6.	45; 49 (45; 49)	1.52 (1.50)	39.36 (42.56)	58.66 (31.75)
7.	53; 59 (54; 58)	1.82 (1.67)	33.81 (45.71)	61.26 (34.37)
8.	62; 65 (62; 66)	1.58 (1.52)	38.61 (47.74)	58.32 (49.29)
9.	70; 74 (70; 76)	1.52 (1.58)	50.33 (43.99)	53.06 (37.22)
10.	78; 84 (79; 84)	1.58 (1.50)	44.15 (49.85)	55.30 (31.20)
11.	87; 92 (88; 93)	1.58 (1.67)	39.93 (45.48)	43.54 (39.53)
12.	96; 100 (96; 101)	1.58 (1.67)	44.27 (47.76)	58.00 (44.15)
13.	104; 109 (104; 109)	1.58 (1.58)	47.96 (42.90)	59.27 (41.57)
14.	112; 118 (113; 117)	1.58 (1.58)	38.71 (46.30)	64.93 (20.96)
15.	121; 126 (122; 125)	1.58 (1.52)	41.75 (44.24)	55.37 (22.83)
16.	130; 134 (130; 135)	1.58 (1.58)	20.39 (52.80)	59.46 (37.15)
17.	138; 143 (139; 143)	1.67 (1.52)	43.92 (48.11)	52.80 (37.61)
18.	146; 151 (147; 153)	1.50 (1.50)	35.01 (36.93)	56.59 (35.89)
mean		1.59 (1.57)	38.93 (45.01)	55.98 (35.85)
std		0.08 (0.07)	7.10 (4.65)	4.98 (6.62)
median		1.59 (1.58)	38.93 (45.59)	55.98 (37.19)
lower quartile		1.53 (1.50)	7.10 (43.40)	4.98 (32.41)
upper quartile		1.58 (1.65)	7.10 (47.75)	4.98 (39.35)

Neural Network Segmentation - Deep Dive U-Net

Results

Automated heart segmentation

Detection rate 100%

Max detection error is 1 frame

Ventricular dimension estimation is robust over multiple cardiovascular cycles.

Reported human performance is between 5% to 12% on fractional shortenings.

Current heart beat	diastolic frame; systolic frame	heart rate [*bps*]	minor fractional shortening [%]	major fractional shortening [%]
1.	2; 6 (2; 7)	1.50 (1.50)	41.33 (47.29)	52.18 (38.80)
2.	11; 15 (11; 16)	1.67 (1.67)	40.87 (30.90)	57.58 (38.69)
3.	19; 23 (19; 24)	1.58 (1.58)	28.57 (48.92)	53.46 (34.57)
4.	28; 31 (28; 32)	1.46 (1.67)	29.37 (43.21)	47.16 (28.68)
5.	36; 42 (36; 40)	1.69 (1.50)	42.56 (45.43)	60.73 (41.05)
6.	45; 49 (45; 49)	1.52 (1.50)	39.36 (42.56)	58.66 (31.75)
7.	53; 59 (54; 58)	1.82 (1.67)	33.81 (45.71)	61.26 (34.37)
8.	62; 65 (62; 66)	1.58 (1.52)	38.61 (47.74)	58.32 (49.29)
9.	70; 74 (70; 76)	1.52 (1.58)	50.33 (43.99)	53.06 (37.22)
10.	78; 84 (79; 84)	1.58 (1.50)	44.15 (49.85)	55.30 (31.20)
11.	87; 92 (88; 93)	1.58 (1.67)	39.93 (45.48)	43.54 (39.53)
12.	96; 100 (96; 101)	1.58 (1.67)	44.27 (47.76)	58.00 (44.15)
13.	104; 109 (104; 109)	1.58 (1.58)	47.96 (42.90)	59.27 (41.57)
14.	112; 118 (113; 117)	1.58 (1.58)	38.71 (46.30)	64.93 (20.96)
15.	121; 126 (122; 125)	1.58 (1.52)	41.75 (44.24)	55.37 (22.83)
16.	130; 134 (130; 135)	1.58 (1.58)	20.39 (52.80)	59.46 (37.15)
17.	138; 143 (139; 143)	1.67 (1.52)	43.92 (48.11)	52.80 (37.61)
18.	146; 151 (147; 153)	1.50 (1.50)	35.01 (36.93)	56.59 (35.89)
mean		1.59 (1.57)	38.93 (45.01)	55.98 (35.85)
std		0.08 (0.07)	7.10 (4.65)	4.98 (6.62)
median		1.59 (1.58)	38.93 (45.59)	55.98 (37.19)
lower quartile		1.53 (1.50)	7.10 (43.40)	4.98 (32.41)
upper quartile		1.58 (1.65)	7.10 (47.75)	4.98 (39.35)

Neural Network Segmentation - Deep Dive U-Net

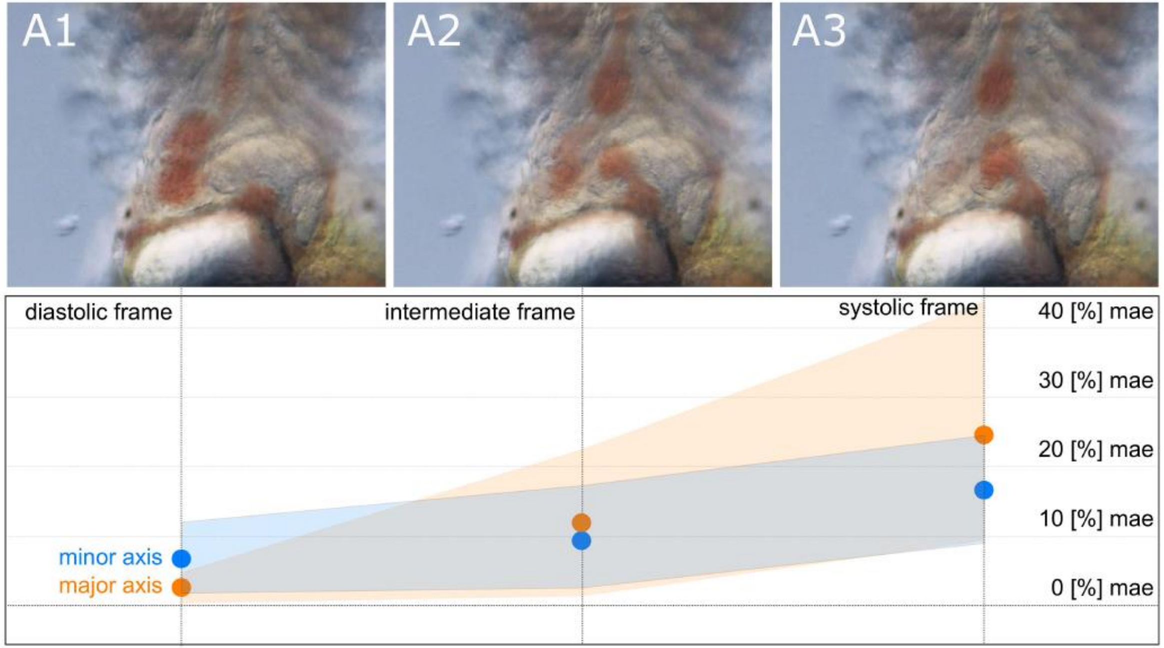
Results

Automated heart segmentation

Detection rate 100%

Max detection error is 1 frame

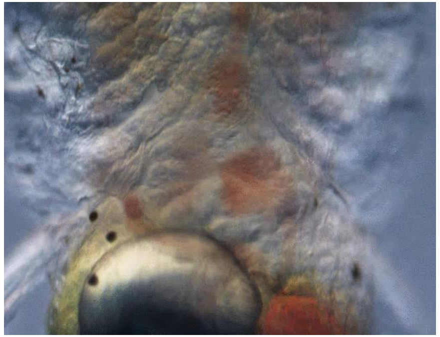
Ventricular dimension estimation is robust

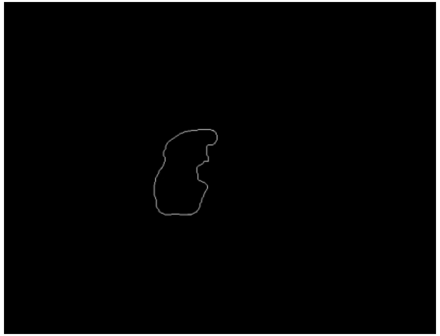


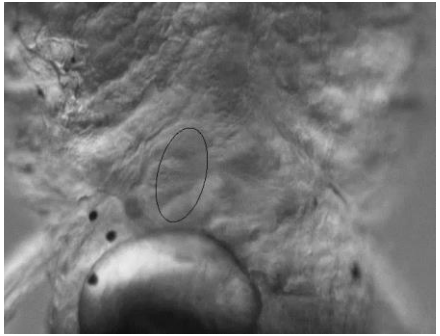
Errors are largest for systolic frames, this is when humans also fail to label precisely.

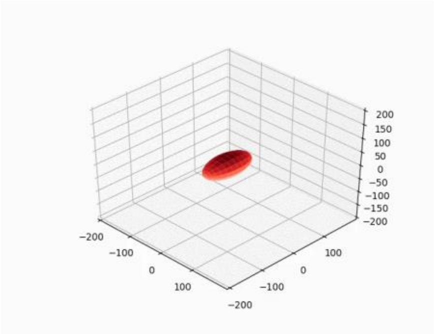
Results

Data-driven end-to-end automated heart segmentation from ventral view and lateral view.
Robust estimation of ventricular dimensions.









Neural Network Segmentation - Take it from here

Codebase

https://osf.io/snb6p/

Estimation of further ventricular dimensions seems feasible

Want to get your hands dirty?

Let me know.

Journal Paper

Tutorial

Machine Learning Methods for Automated Quantification of Ventricular Dimensions

Contributors: Mark Schutera, Steffen Just, Jakob Gierten, Ralf Mikut, Markus Reischl, Christian Pylatiuk

Date created: 2019-03-24 02:53 AM | Last Updated: 2019-05-29 11:47 PM

Create DOI

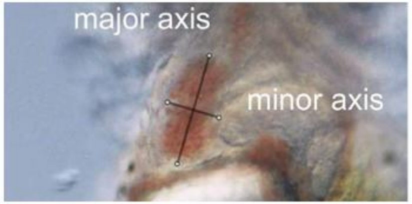
Category: Project

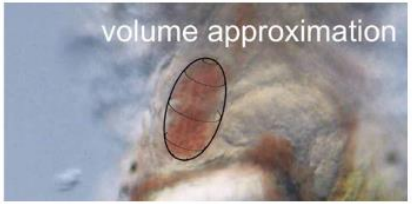
Description:

Machine learning methods for automated quantification of ventricular dimensions (for further details see Wiki bellow or take a closer look on our paper).

License: GNU General Public License (GPL) 3.0

Wiki





See demonstration of our algorithm and framework on the test set data:
https://youtu.be/i5bX_XbwXq0

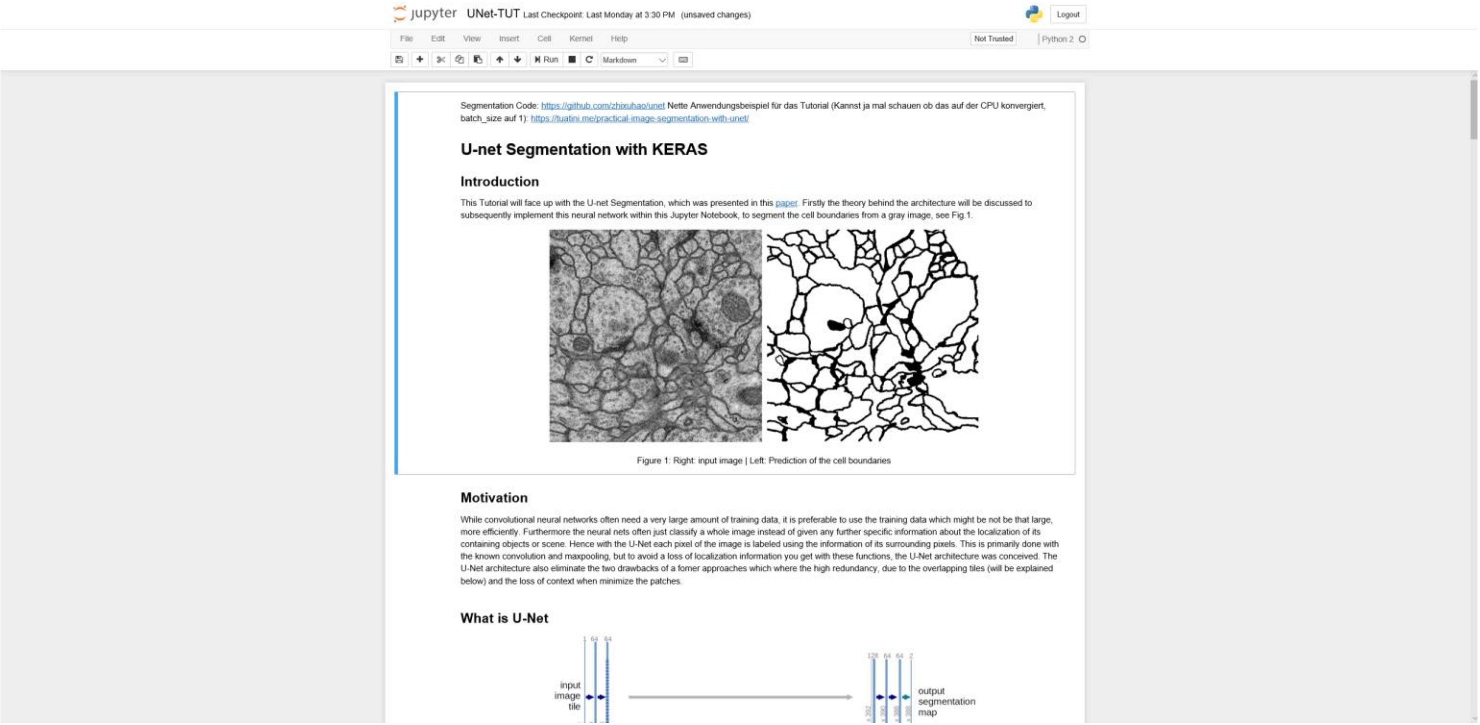
The medaka (Oryzias latipes) and the zebrafish (Danio rerio) are used as a model organism for a variety of subjects in biomedical research the here presented work aims to study the potential of automated ventricular dimension estimation through heart segmentation in medaka. For more on this, it's t...

Read More

Neural Network Segmentation - Take it from here

Codebase
Journal Paper

Tutorial
Kudos to
Hendrik Vogt



References

[6] Lex Fridman. MIT 6.S094: Deep Learning for Self-Driving Cars https://selfdrivingcars.mit.edu/, 2019. Zugriff: 20.06.2019.

[7] ^ Sørensen, T. (1948). "A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on Danish commons". *Kongelige Danske Videnskabernes Selskab*. **5** (4): 1–34.

[8] A guide to convolution arithmetic for deep learning. V. Dumoulin, and F. Visin. (2016) arxiv:1603.07285.

[9] Deep learning. Yann LeCun, Yoshua Bengio & Geoffrey Hinton Nature volume521, pages436–444 (2015)

[10] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. 1, 2, 3, 5

[11] A. Bréhéret. Pixel annotation tool. *GitHub Repository*, 2017.

[12] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. *Medical Image Computing and Computer-Assisted Intervention*, 9351:234–241, 2015.

[13] T. Hoage, Y. Ding, and X. Xu. Quantifying cardiac functions in embryonic and adult zebrafish. *Cardiovascular Development*, 843:11–20, 2012.

Thanks for your time
Questions?

Contact!

mark.schutera@kit.edu