Machine Learning in Medical Imaging SS16

Project Presentation: U-Net Implementation

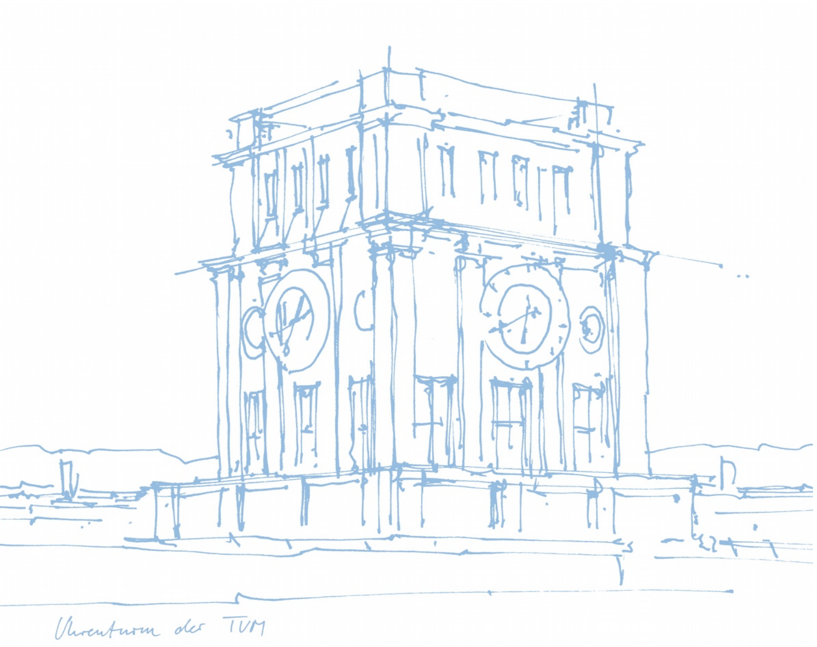
Claus Meschede, Leonard Rychly

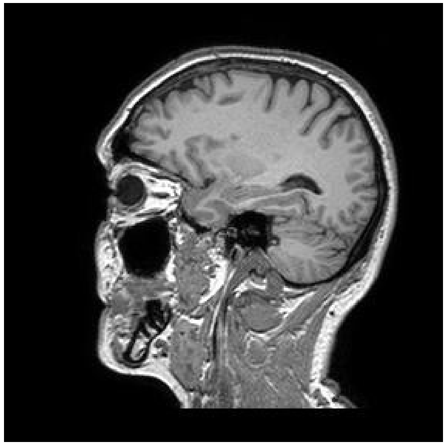
Technische Universität München

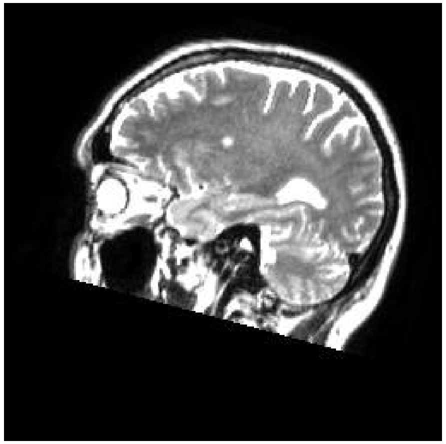
Fakultät für Informatik

Chair for Computer Aided Medical Procedures & Augmented Reality

München, July 18, 2016







Outline

I. Project description

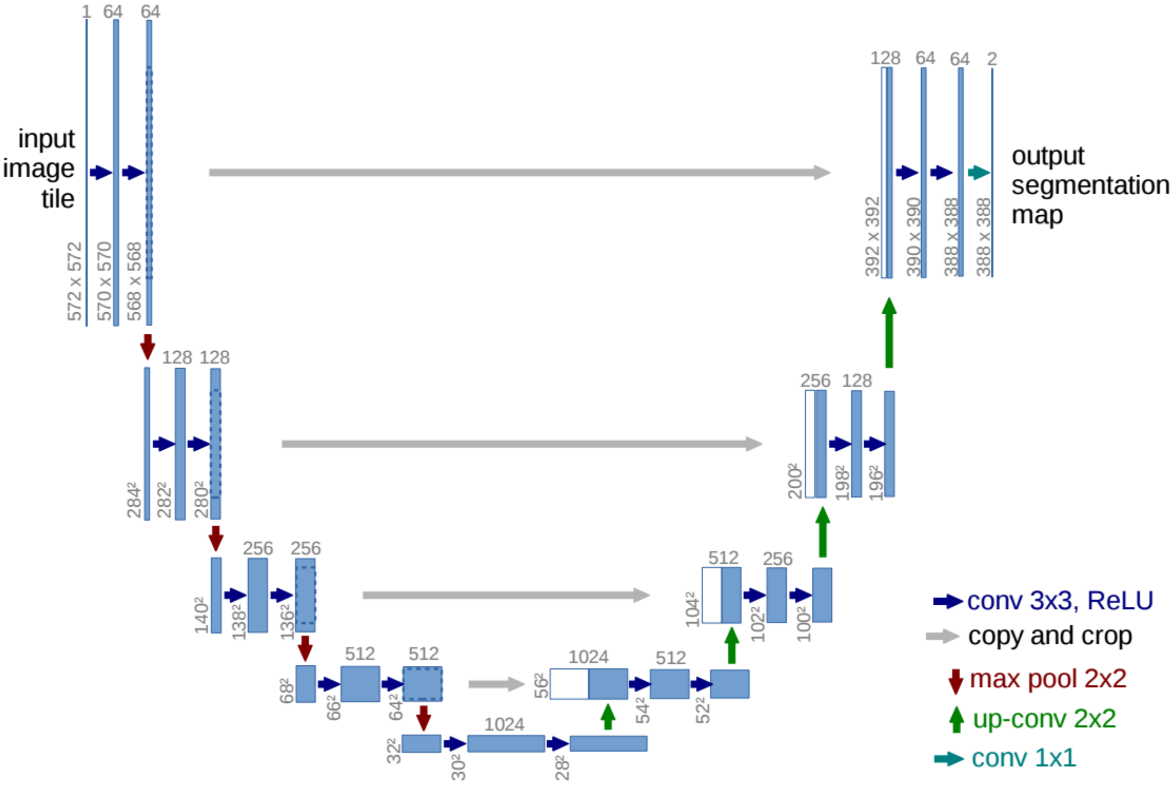
II. Issues

III. Results

IV. Summary

Project description – U-Net I

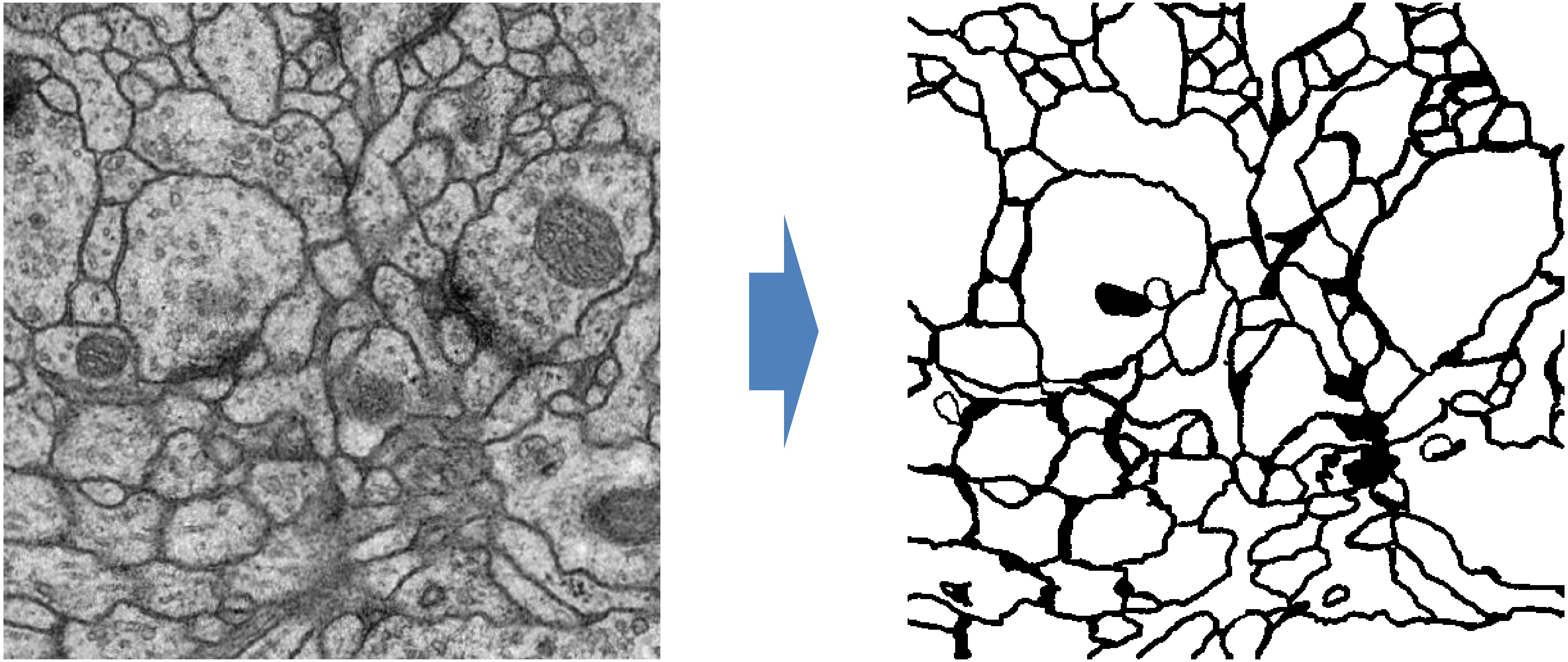
- Main feature: skip connections for reusing finer feature maps.
- Used for medical image segmentation.



Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation." Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015. Springer International Publishing, 2015. 234-241.

Project description – U-Net II

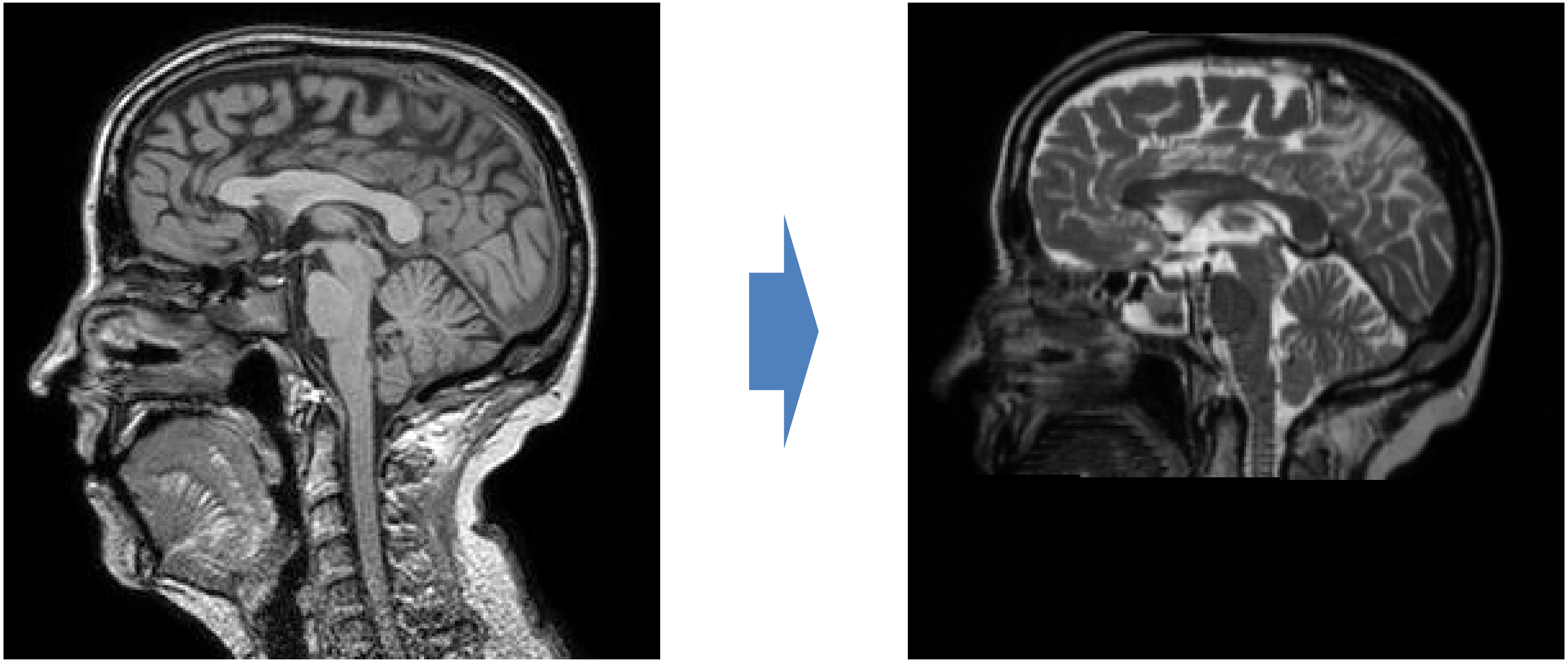
- Main feature: skip connections for reusing finer feature maps.
- Used for medical image segmentation.



Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation." Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015. Springer International Publishing, 2015. 234-241.

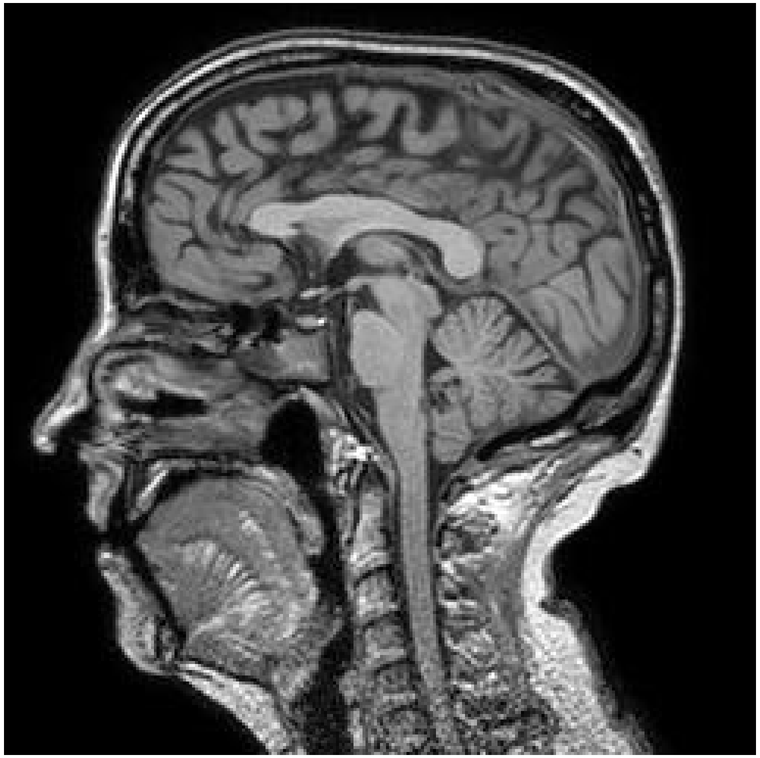
Project description – Idea

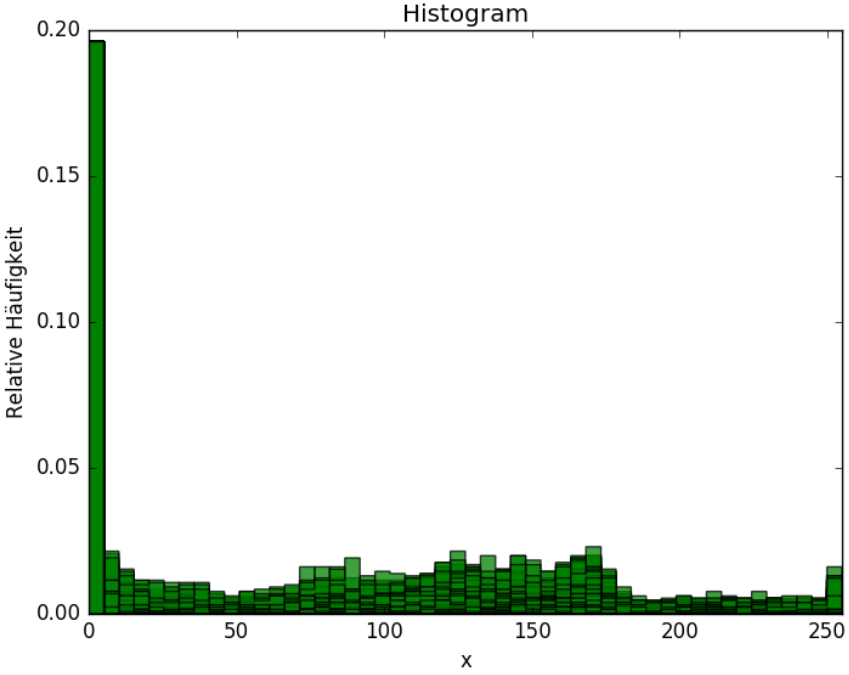
- Adapt U-Net architecture to predict one modality given another.
- Raw volumetric data is available (Brain MRI)



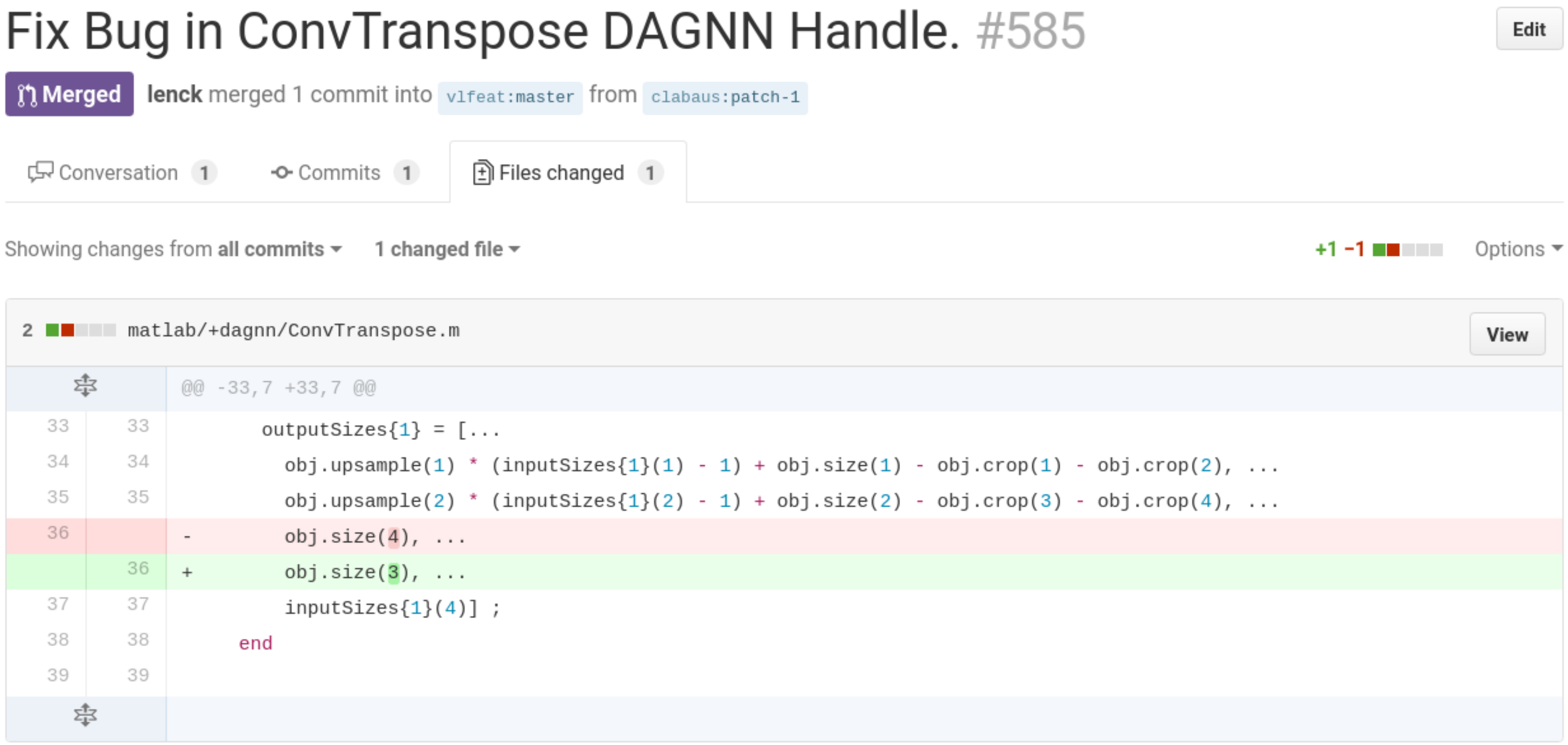
Project description – Image Generation

- Generate 2D Images from raw volumetric data.
- Normalize gray scale images.





Issues – Transposed Convolution Bug



Fix Bug in ConvTranspose DAGNN Handle. #585

Merged **lenck** merged 1 commit into `vlfeat:master` from `clabaus:patch-1`

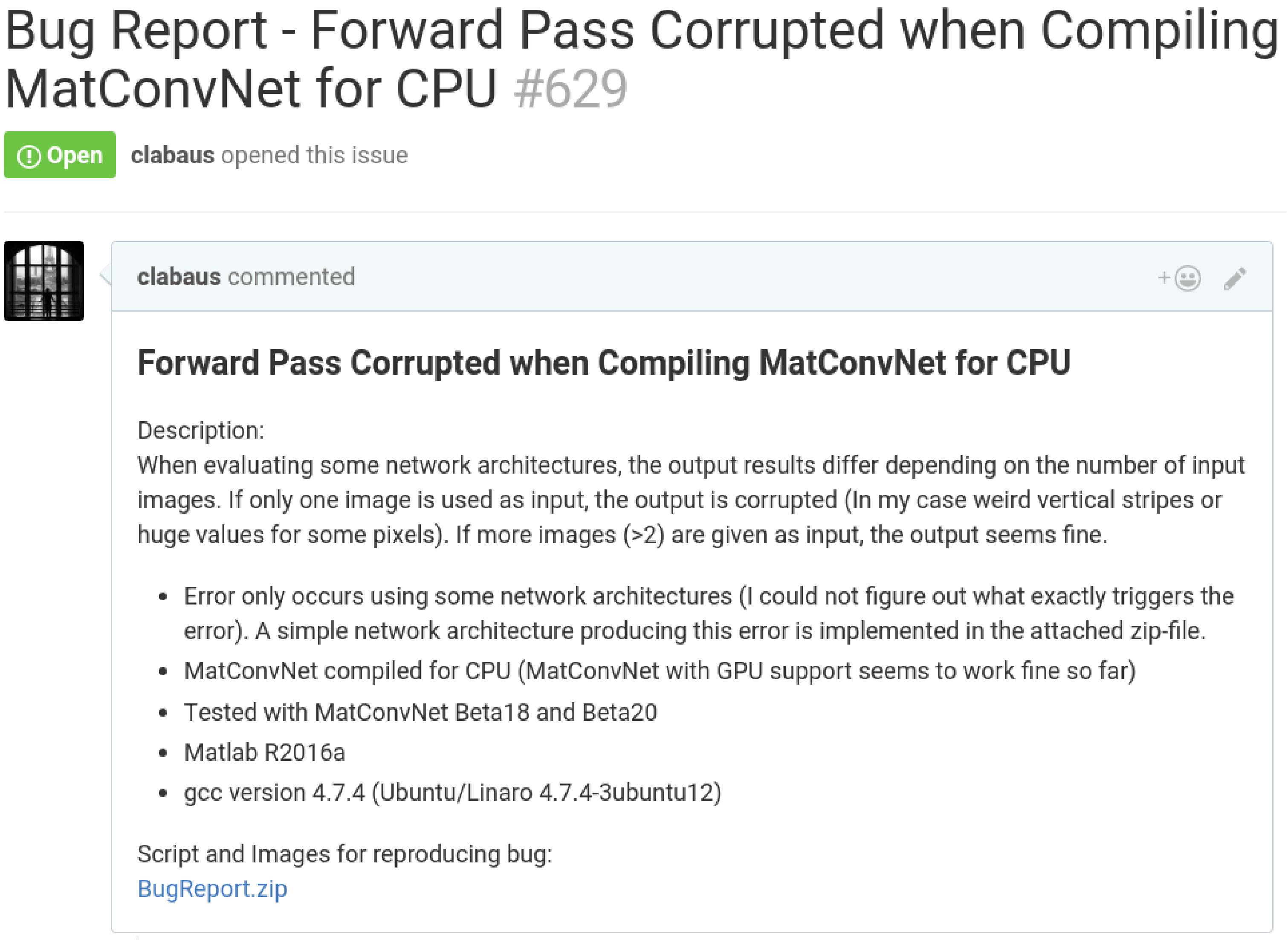
Showing changes from **all commits** ▾ **1 changed file** ▾ +1 −1 🟩🟥

2 🟩🟥 matlab/+dagnn/ConvTranspose.m

```
@@ -33,7 +33,7 @@
     outputSizes{1} = [...
      obj.upsample(1) * (inputSizes{1}(1) - 1) + obj.size(1) - obj.crop(1) - obj.crop(2), ...
      obj.upsample(2) * (inputSizes{1}(2) - 1) + obj.size(2) - obj.crop(3) - obj.crop(4), ...
-     obj.size(4), ...
+     obj.size(3), ...
      inputSizes{1}(4)] ;
    end

```

Issues – Forward Pass Bug



Bug Report - Forward Pass Corrupted when Compiling MatConvNet for CPU #629

Open **clabaus** opened this issue

clabaus commented

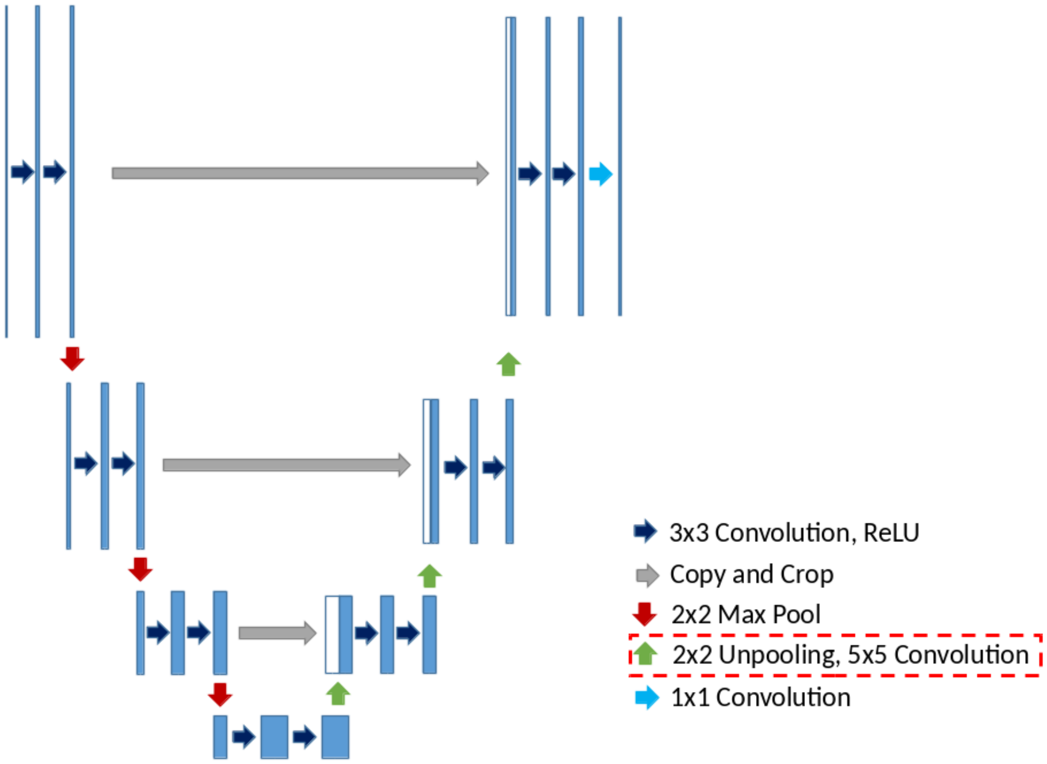
Forward Pass Corrupted when Compiling MatConvNet for CPU

Description:
When evaluating some network architectures, the output results differ depending on the number of input images. If only one image is used as input, the output is corrupted (In my case weird vertical stripes or huge values for some pixels). If more images (>2) are given as input, the output seems fine.

- Error only occurs using some network architectures (I could not figure out what exactly triggers the error). A simple network architecture producing this error is implemented in the attached zip-file.
- MatConvNet compiled for CPU (MatConvNet with GPU support seems to work fine so far)
- Tested with MatConvNet Beta18 and Beta20
- Matlab R2016a
- gcc version 4.7.4 (Ubuntu/Linaro 4.7.4-3ubuntu12)

Script and Images for reproducing bug:
BugReport.zip

Results – U-Net with Unpooling I



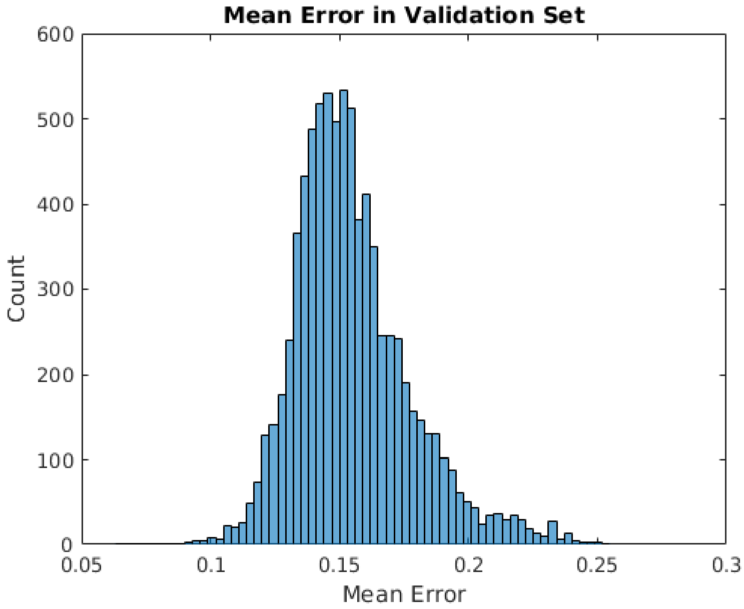
➡ 3x3 Convolution, ReLU

⇨ Copy and Crop

⬇ 2x2 Max Pool

⬆ 2x2 Unpooling, 5x5 Convolution

➡ 1x1 Convolution

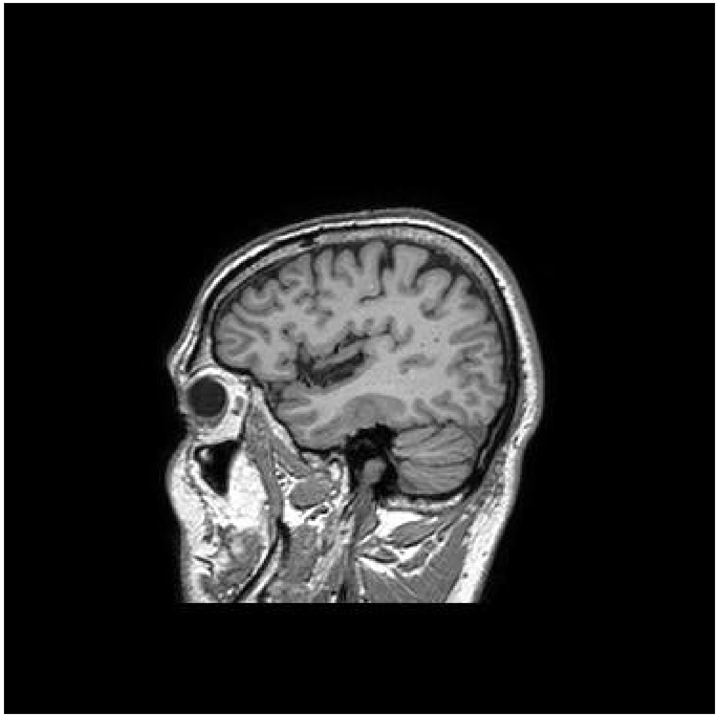


Training parameters:

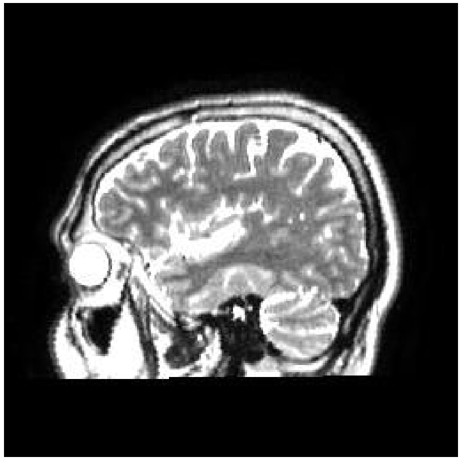
• # of images:	8000
• # of epochs:	22
• Learning rate:	1e-7
• Weight decay:	0.01
• Momentum:	0.9

Results – U-Net with Unpooling II

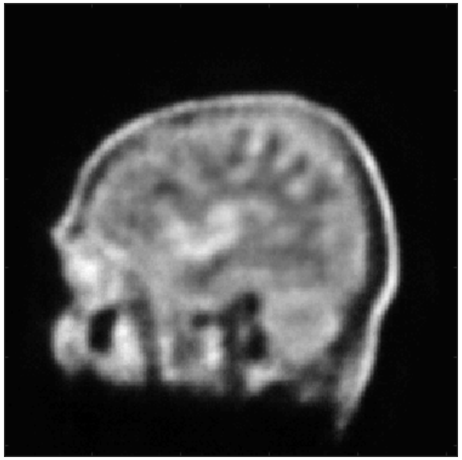
Test Image



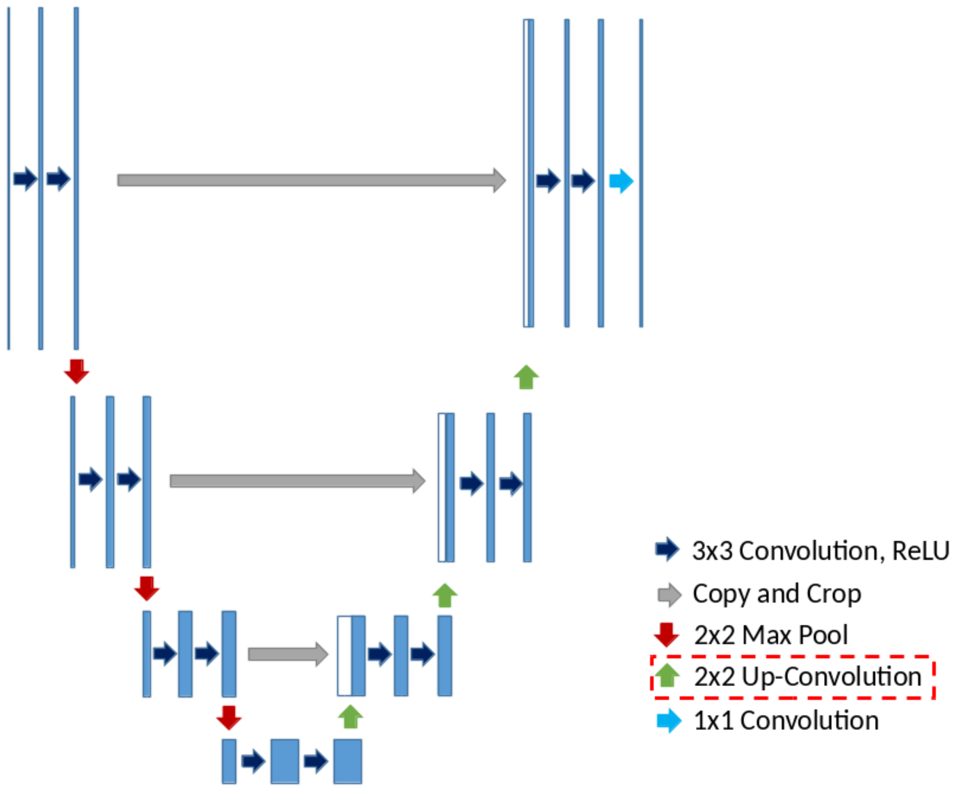
Target Image



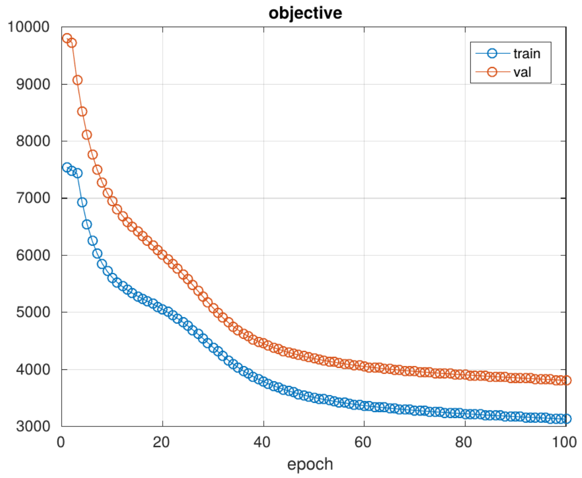
Output Image



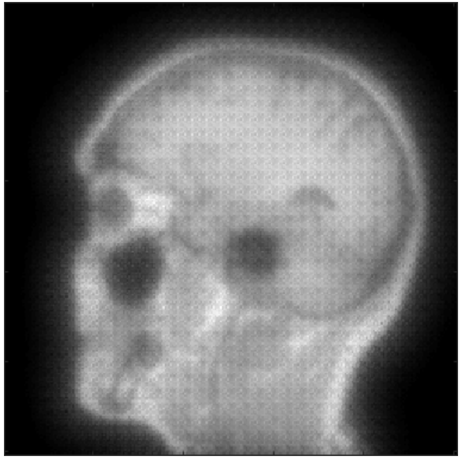
Results – U-Net with Up-Convolutions



# of images: 15



Example Output



Not trained with all available images yet!

Summary I

What we have learned:

- Using state-of-the-art software in development can be cumbersome:
  - → Importance of systematic troubleshooting
- MatConvNet Framework
- Implementing and training CNNs
- Optimize learning parameters

Summary II

What we have learned:

- Using state-of-the-art software in development can be cumbersome:
  - → Importance of systematic troubleshooting
- MatConvNet Framework
- Implementing and training CNNs
- Optimize learning parameters

Conclusion:

- U-Net capable of roughly predicting the target modality.
- Further testing needed for performance evaluation …
  - Larger training set
  - Up-Convolutions ↔ Unpooling
  - Increasing depth of U-Net
  - Convolutions using padding
  - Adapt input images to output shape

→ Compare performance of U-Net with other CNNs