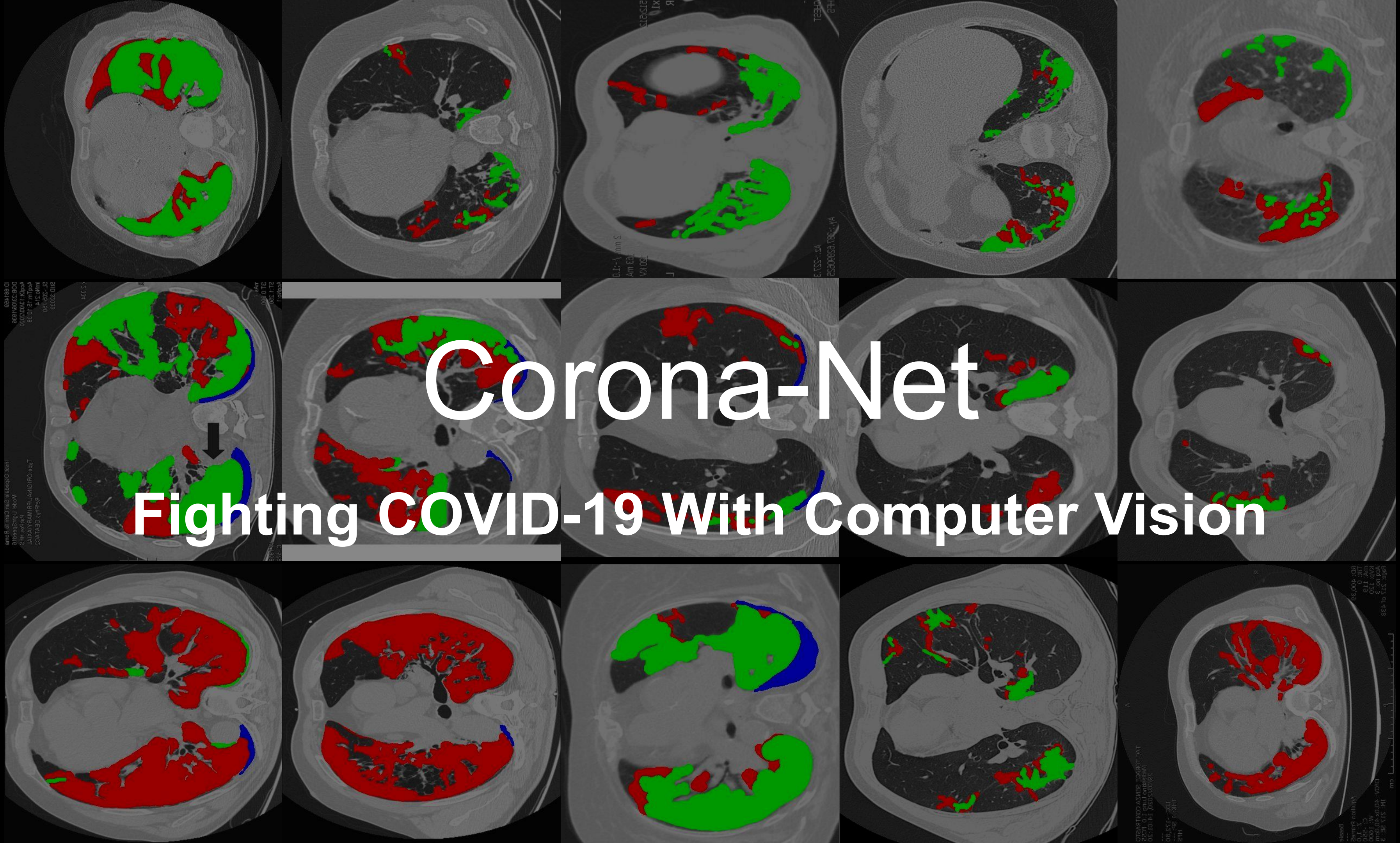
Corona-Net

Fighting COVID-19 With Computer Vision



1. <u>Self Intro</u>

- Choi Ching Lam
- 17 year old, Form 5 student at the Diocesan Girls' School
- Favourite languages: Python, Julia
- Into Machine Learning, aspires to become a researcher
- Email: ccl5a09@gmail.com
- https://github.com/chinglamchoi
- https://medium.com/@cchoi314

2. Background

- Inspired by doctors from Wuhan on TV
- Inspired by Johns Hopkins University’s (Center for Systems Science and Engineering (CSSE)) COVID-19 Dashboard
- Relevant to previous work on brain tumour boundary resection for lower grade glioma

3. Technologies Used

- Language: Python with NumPy library
- AI library: PyTorch
- Image processing libraries: Albumentations, Torchvision, Scikit-image, Matplotlib

Python with NumPy

- Speed: dynamically typed
- NumPy: parallelism & vectorisation
- NumPy: better support for matrices & tensors & operations
- Easy to prototype with, elegant syntax
- Powerful libraries

What to use for Image Processing?

- Matplotlib vs. Scikit-image vs. Torchvision vs. Albumentations
- Matplotlib: General purpose
- Scikit-image: Advanced algorithms
- Torchvision: Tight integration with PyTorch
- Albumentations: Biomedical Imaging

Why PyTorch?

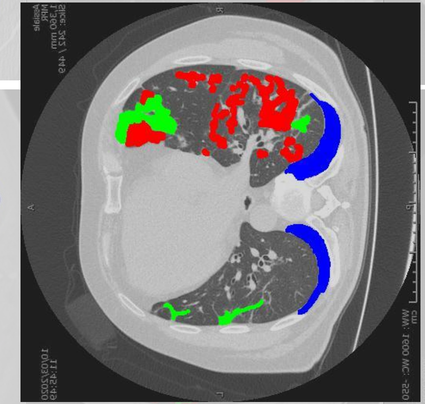
- More research / academia support
- Better customisation ability
- Similar to NumPy
- Dynamism e.g. Dynamic computation graphs

4. Data & Augmentation

- COVID-19 CT segmentation dataset
  - http://medicalsegmentation.com/covid19/
- COVID-19 image data collection
  - https://github.com/ieee8023/covid-chestxray-dataset

Challenge: Dataset size

- 100 CT scan slices taken from > 40 patients with COVID-19.
- Mask-labelled data by expert radiologist:
  - 3 symptoms: ground glass (red), consolidation (green), pleural effusion (blue)

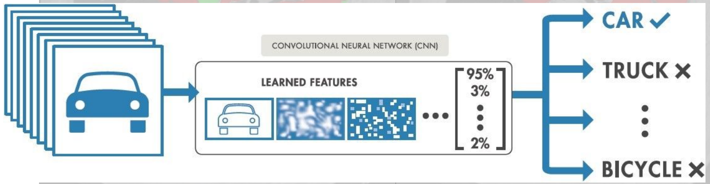


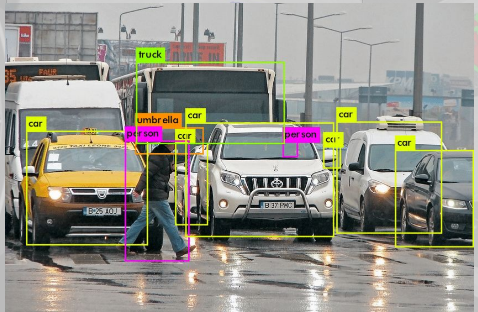
Data & Augmentation

- Train-test split of 80-20 images
- Augment 80 → 1920 via:
    - Random rotations → rotational invariance (Torchvision)
    - Random cropping → shift invariance (Torchvision)
    - Normalisation → grey value invariance (Torchvision)
    - Elastic Transformations & Scale shift → model human tissue (Albumenations)

5. Model Architecture

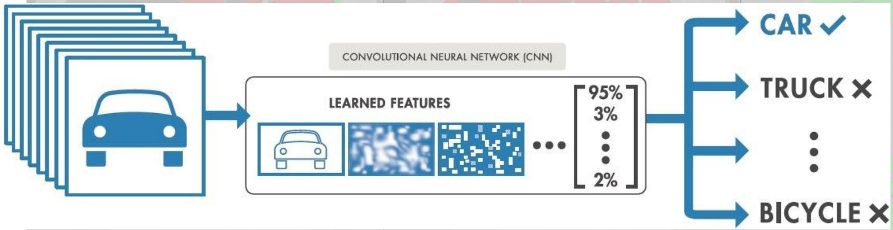
- For multi-label segmentation
- Classification vs. detection vs. segmentation
- Classification: Input image → output class label
- Detection: Input image → output bounding box & class label
- Segmentation: Input image → output image mask

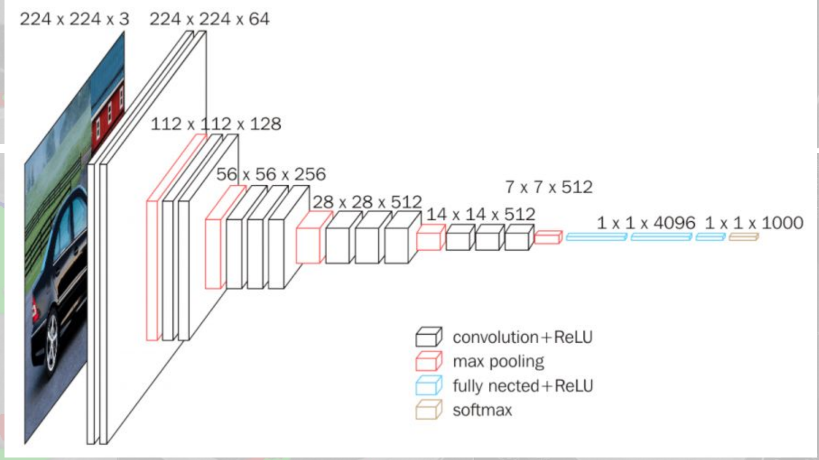




Classification

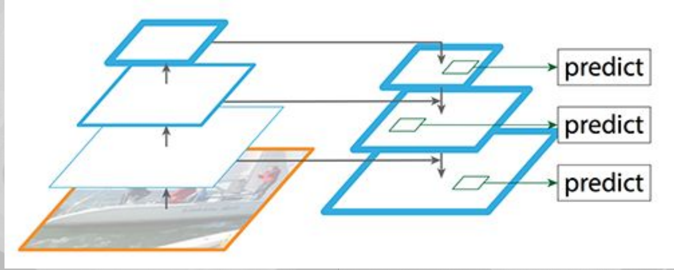
- Input image → output class label (FC layer)
- Can use vanilla Convolutional Neural Networks
- Deep CNNs: accuracy saturation & degradation problem
    - Residual Networks
    - Feature Pyramid Networks





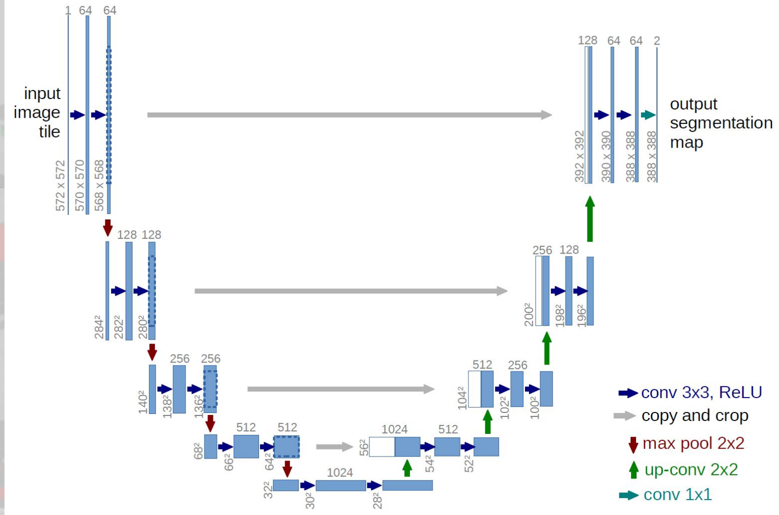
Classification

- ResNets: Shortcut connections
  - Relieves pressure from added deep layers when identity mapping
- FPNs: lateral, top-down connections
  - Fuses feature maps at different scales
  - Each feature map retains local & global information



Fully Convolutional Networks

- Used in Corona-Net
- Encoder-decoder network



- Learns convolutional filter directly not function
- U-Net: FCN for biomedical imaging with symmetrical upsampling & downsampling paths, SOTA

6. Code

- https://github.com/chinglamchoi/Corona-Net
- Binary & Multi-class segmentation
- Loss: Binary Cross Entropy with logits
- Evaluation metrics

$$Dice = \frac{2|A \cap B|}{|A| + |B|}$$

  - Dice Loss
  - Rand Loss

$$RI = \frac{a+d}{\binom{n}{2}}, \quad RE = 1 - RI$$

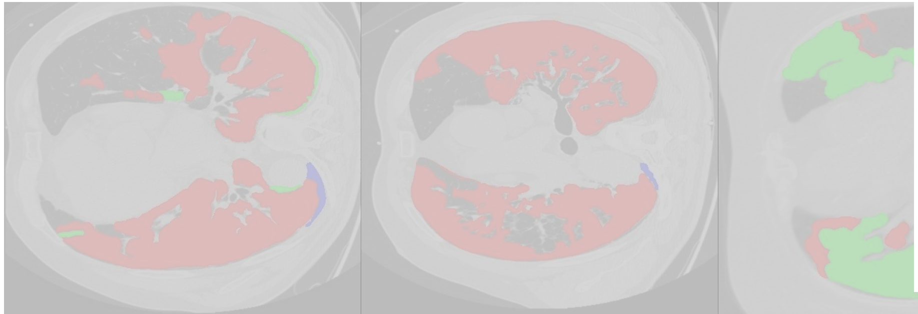
Where a denotes pixels with labels in agreement with the ground truth,
d denotes pixels with labels in disagreement with the ground truth,
n denotes the total pixels in each segmentation mask. (5)

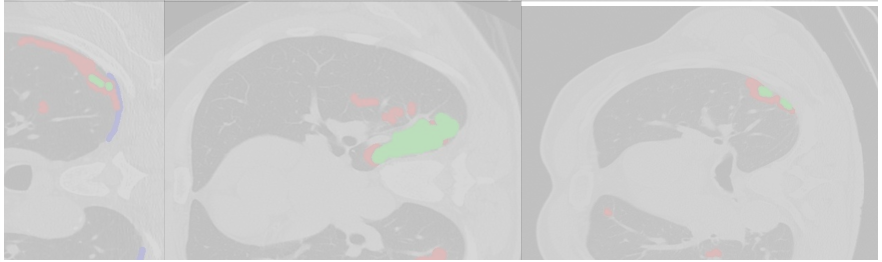
7. Results

- https://github.com/chinglamchoi/Corona-Net

1. Binary Segmentation

Dice Coefficient	Rand Loss	Optimiser	Learning Rate
0.5641	0.2167	Adam	1e-02
0.7068	0.1421	Adam	1e-03
0.7434	0.1347	Adam	1e-04
0.4745	0.1591	Adam	1e-05





2. Multi-Class Segmentation

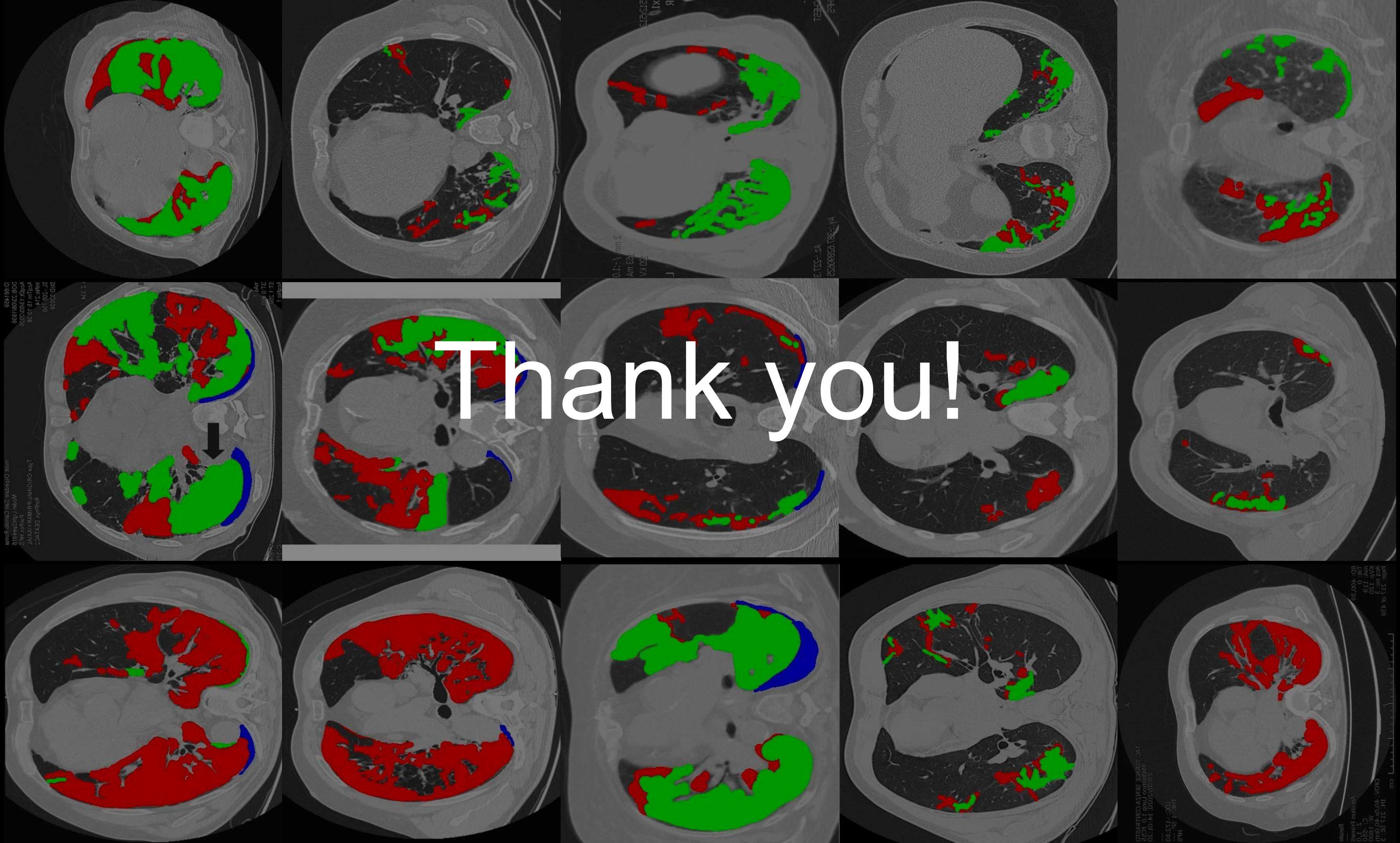
Dice Coefficient	Rand Loss	Optimiser	Learning Rate
0.5160	0.2490	Adam	1e-02
0.5900	0.2114	Adam	1e-03
0.6160	0.1985	Adam	1e-04
0.5001	0.2565	Adam	1e-05

8. Future Development

- Weakly-supervised segmentation
- Using Global Average Pooling & Object Region Mining
- Further training with larger datasets
- Use hard negative mining / focal loss

9. References

- COVID-19 Dashboard by the Center for Systems Science and Engineering (CSSE) at Johns Hopkins University (JHU). (n.d.). Retrieved from https://coronavirus.jhu.edu/map.html
- MedicalSegmentation.com. (n.d.). COVID-19 CT segmentation dataset. Retrieved from http://medicalsegmentation.com/covid19/
- Joseph Paul Cohen and Paul Morrison and Lan Dao COVID-19 image data collection, arXiv:2003.11597, 2020 https://github.com/ieee8023/covid-chestxray-dataset
- K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proceedings of the IEEE CVPR, 2016, pp. 770–778.
- Tsung-Yi Lin, Piotr Dollar, Ross Girshick, Kaiming He, ´ Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 2117–2125, 2017.
- J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in Proceedings of IEEE Conference on CVPR, 2015.
- O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In MICCAI. Springer.


Thank you!