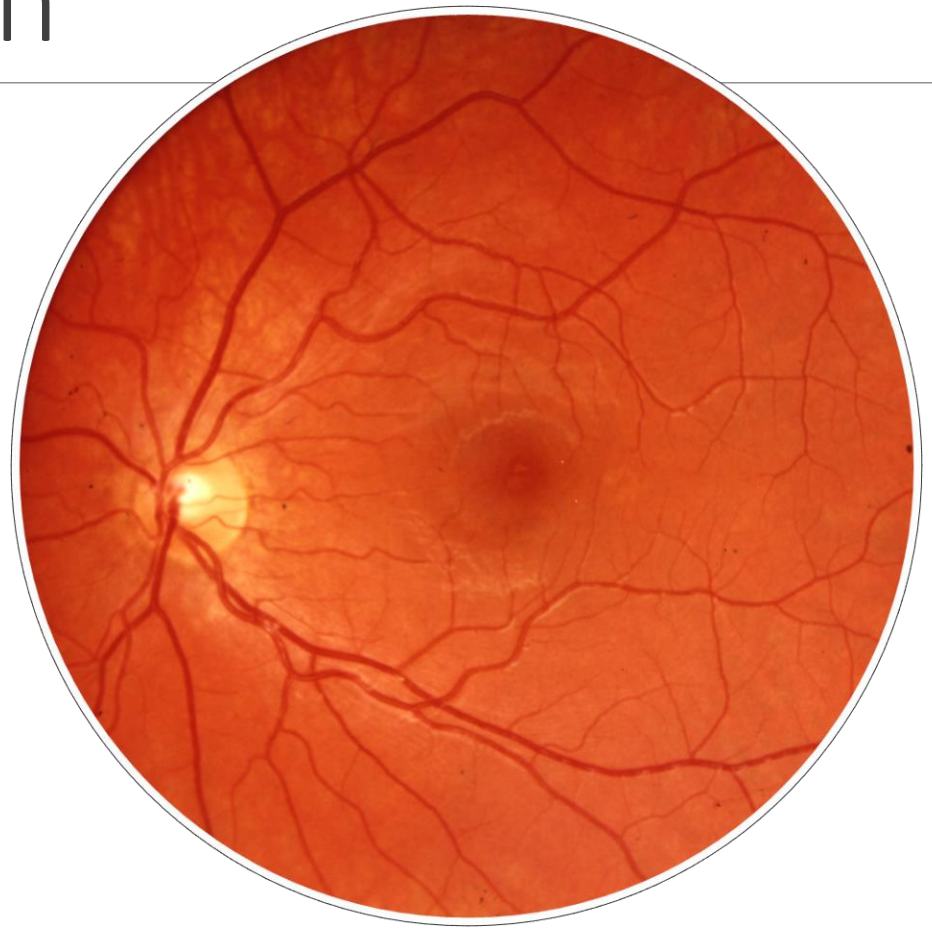
Detecting Blood Vessels in Retinal Images

SADHIRA WAGISWARA

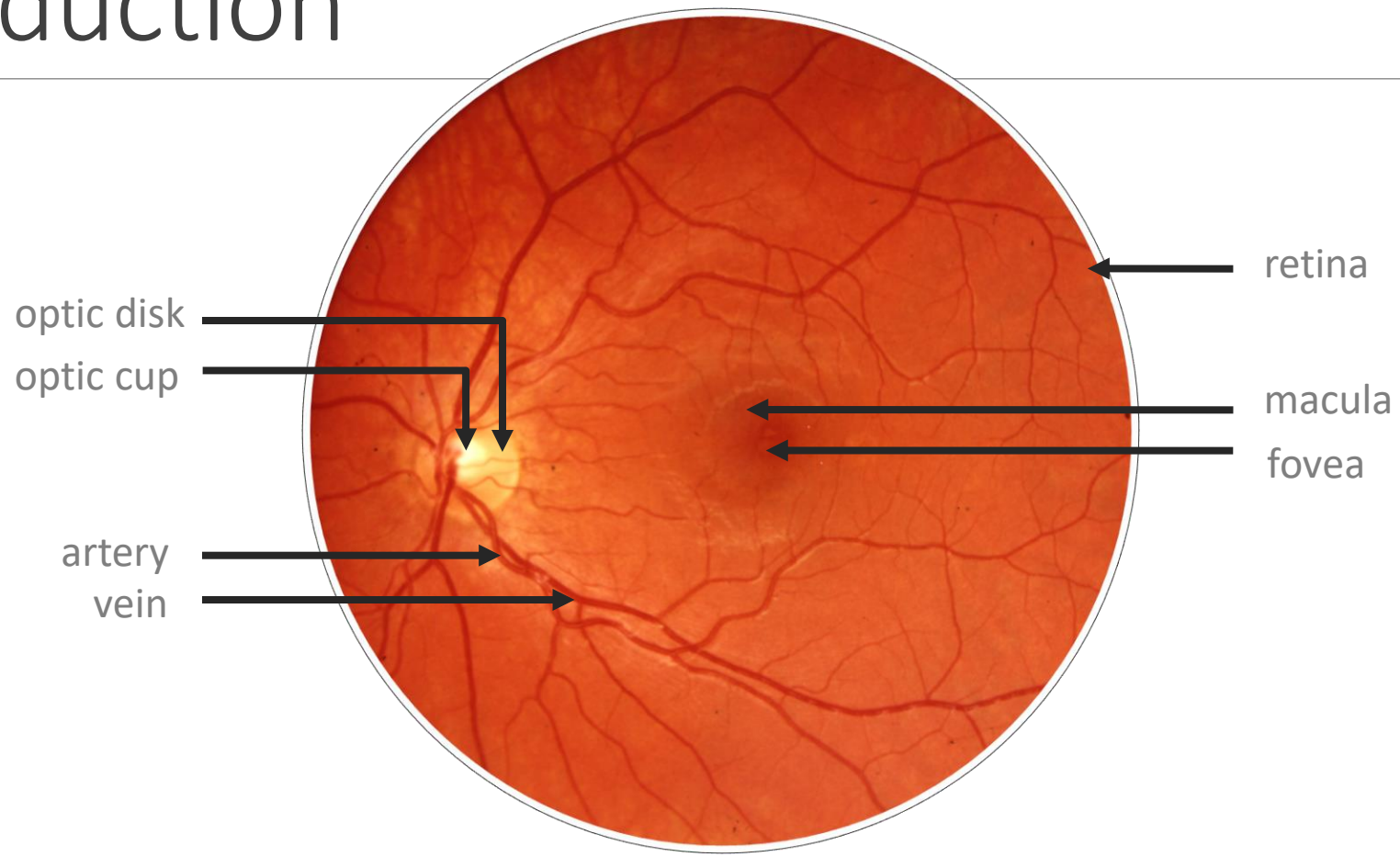
Introduction



fundus photography

Source: Ohiolion

Introduction



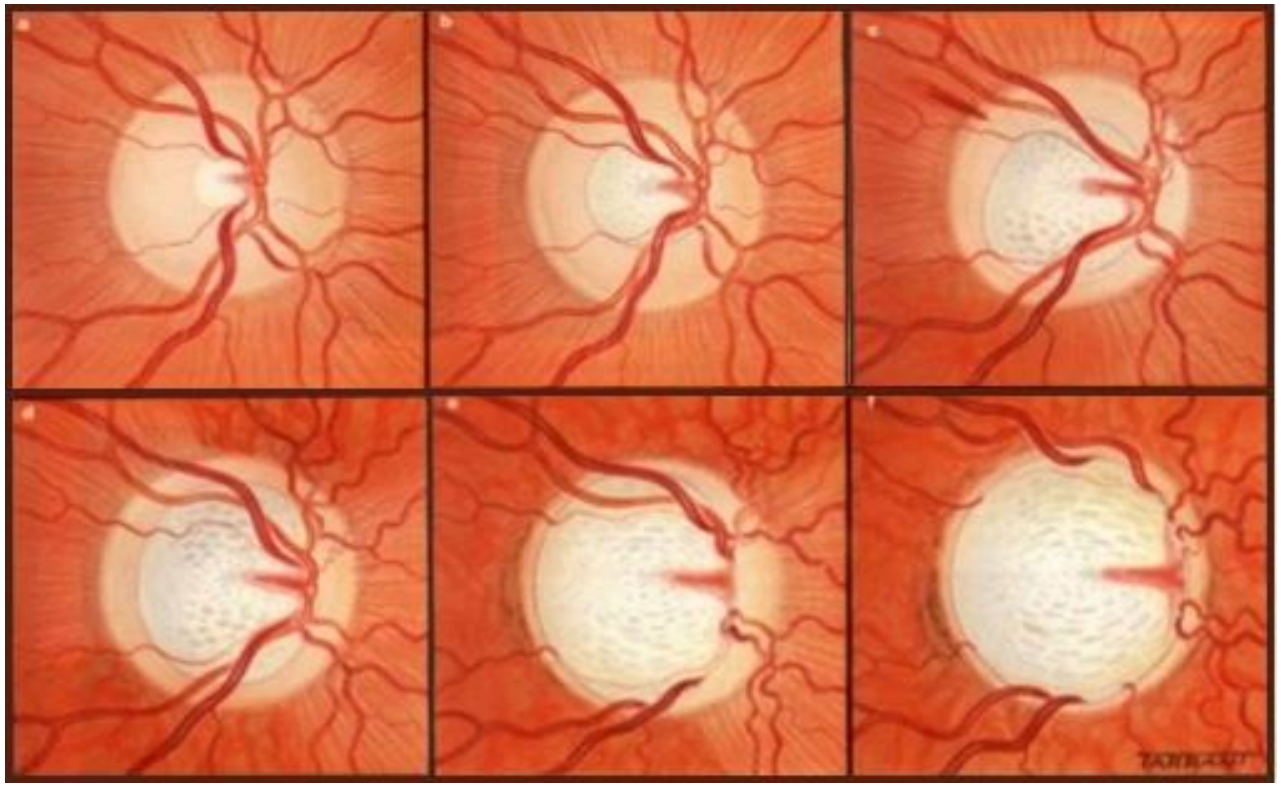
fundus photography

Source: Ohiolion

Glaucoma

- Increased intraocular pressure
- Damage to the optic nerve
- Potential loss of vision

- Clear progressive symptoms due to optic cupping

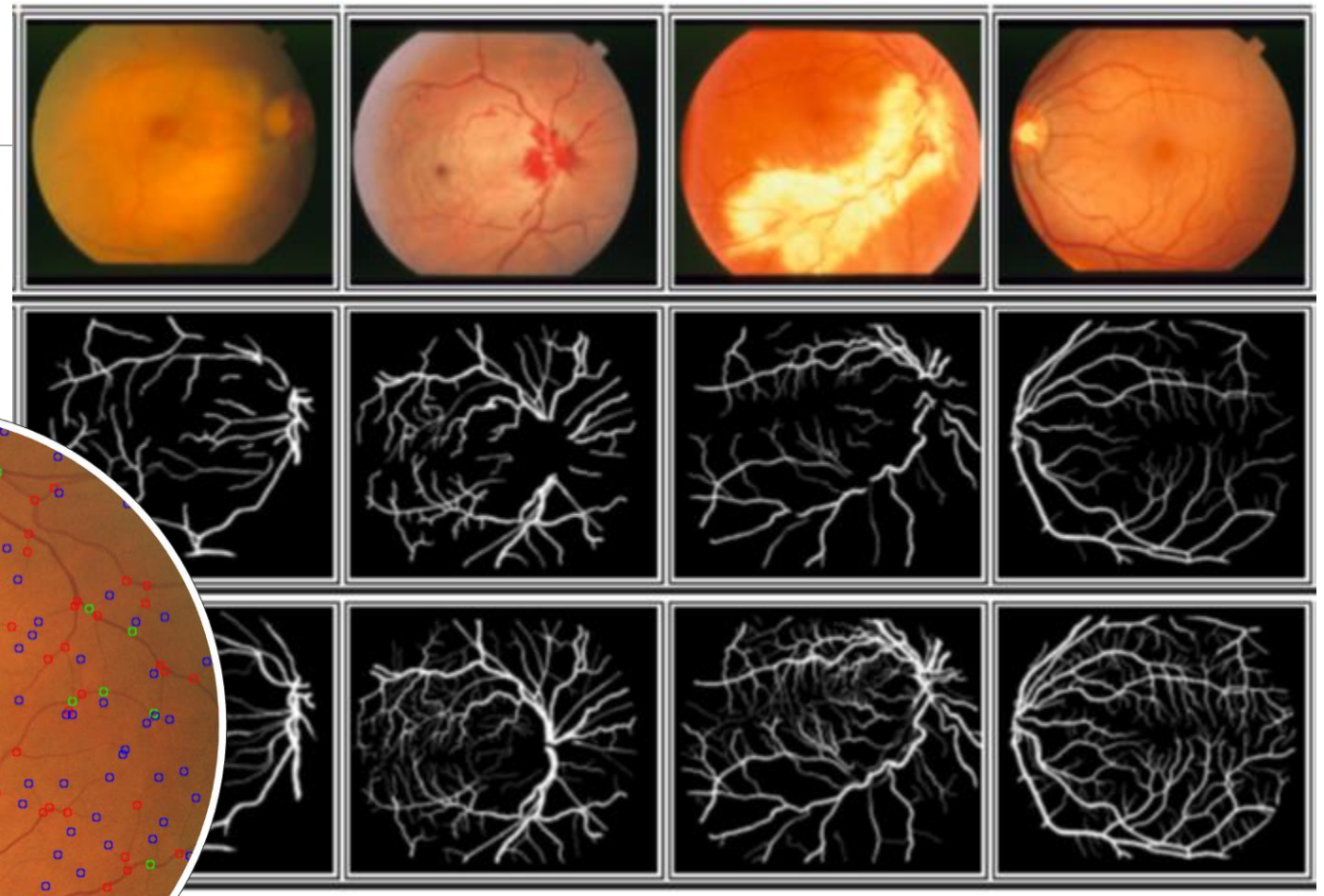


glaucoma progression

Source: Duplessie

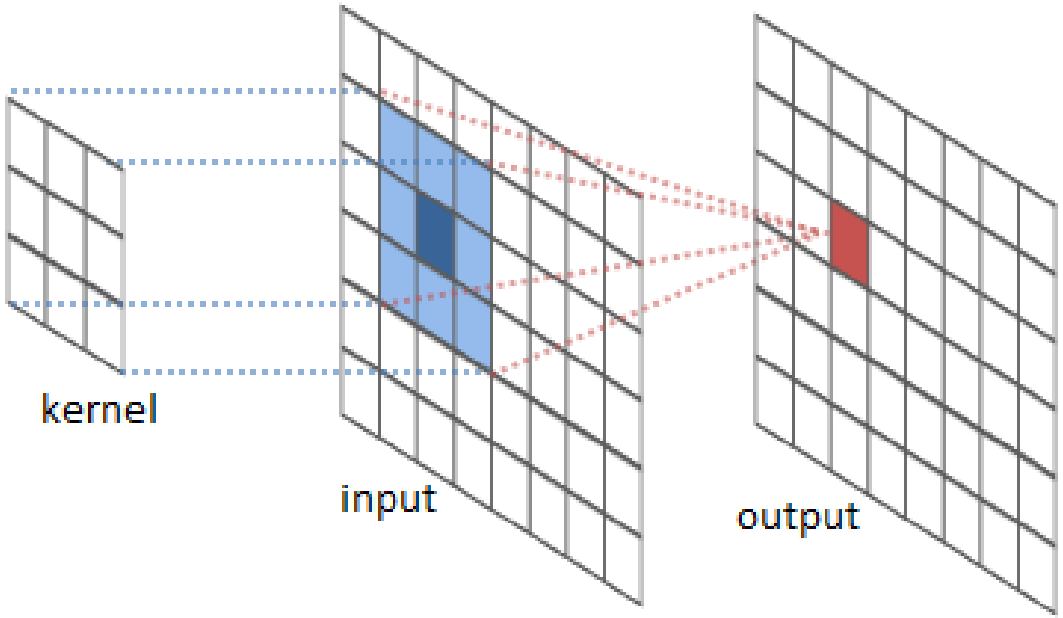
Aim

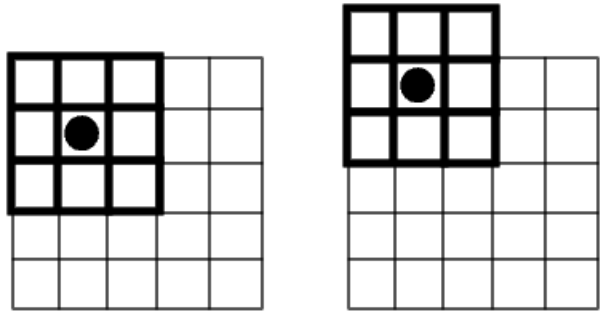
- Semantic Segmentation
- Bifurcation Detection



Sources: STARE, Retinacheck

Convolution

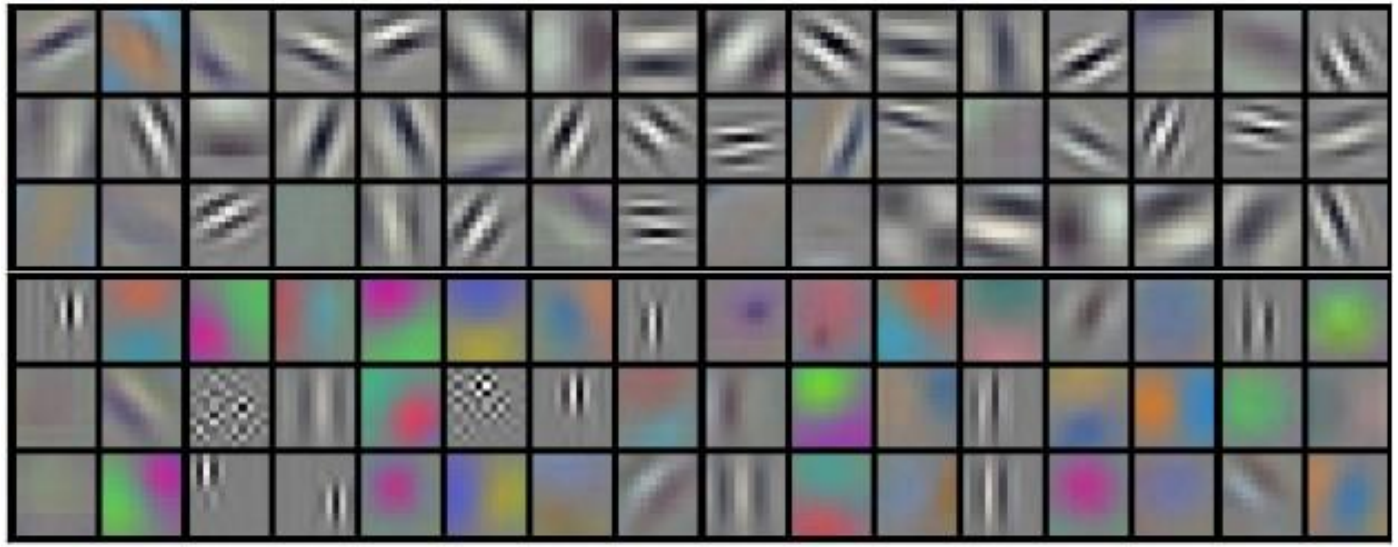




zero padding

Source: Intel Labs

Filters



11 x 11 x 3

horizontal

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

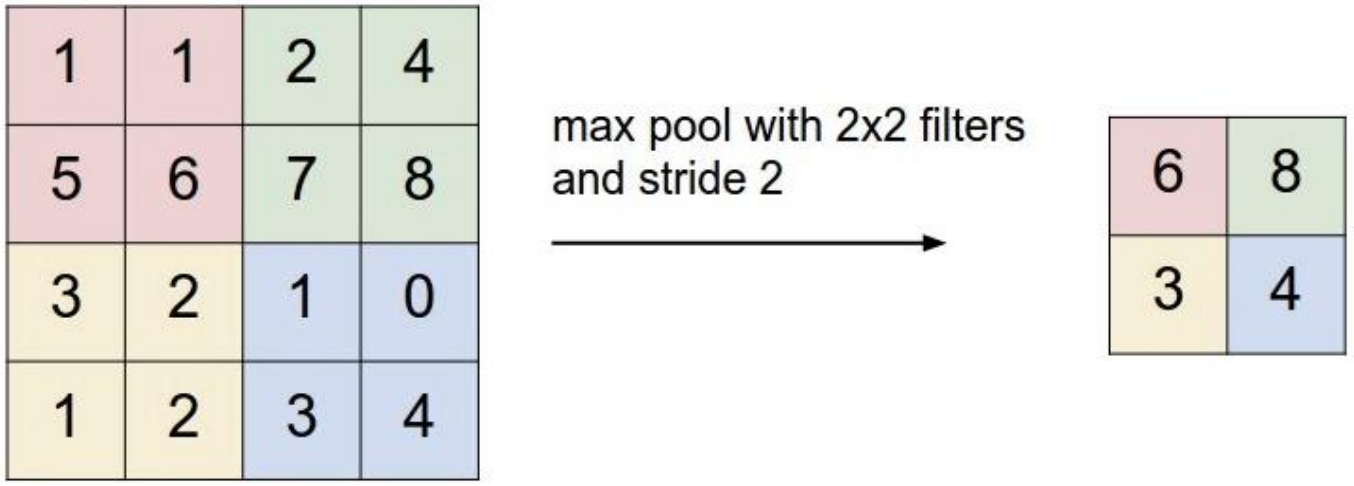
vertical

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

3 x 3 x 1

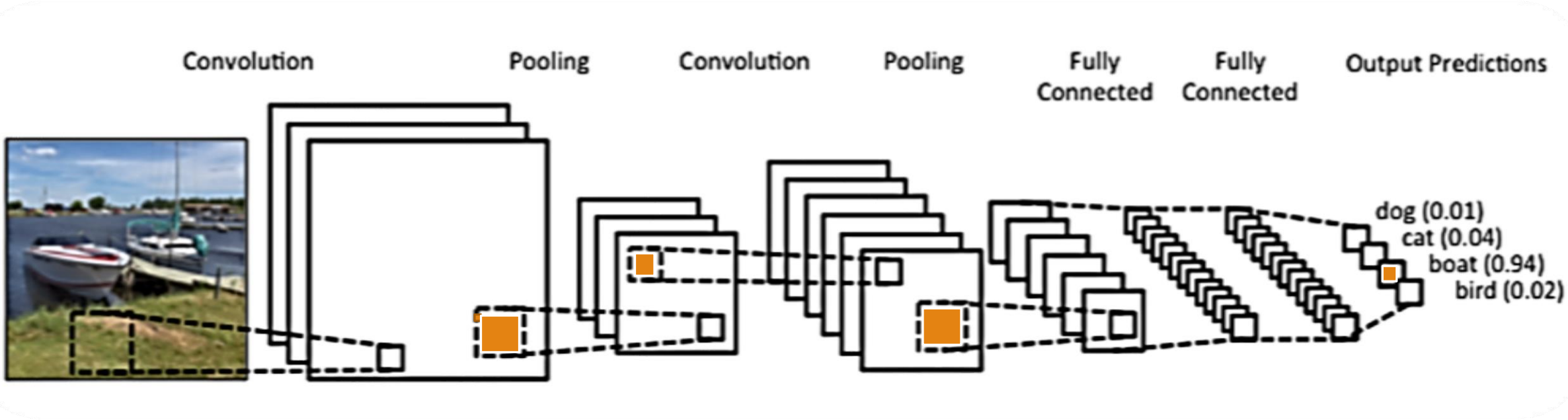
Source: Krizhevsky

Max Pooling



Source: Intel Labs

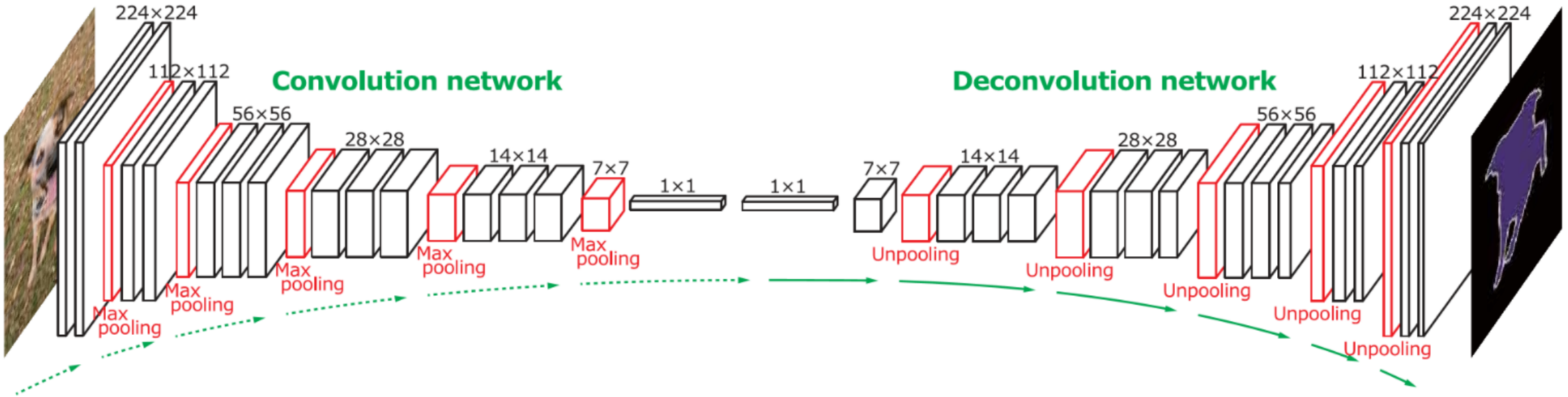
Convolutional Neural Networks



Key concepts: Hidden Layers, Convolution, Pooling, Deconvolution, Scale Invariance

Sources: Wildml

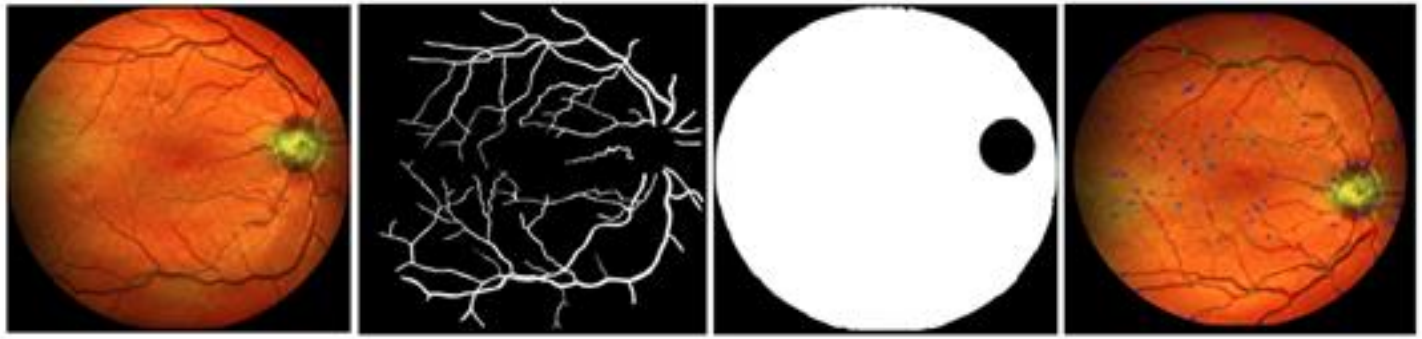
Deconvolution

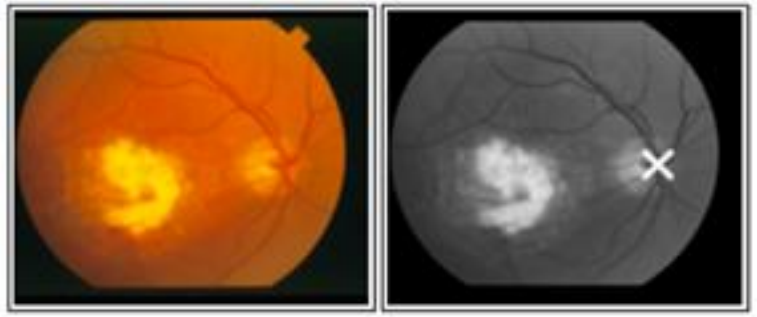


Source: Noh

Software

Fundus Datasets





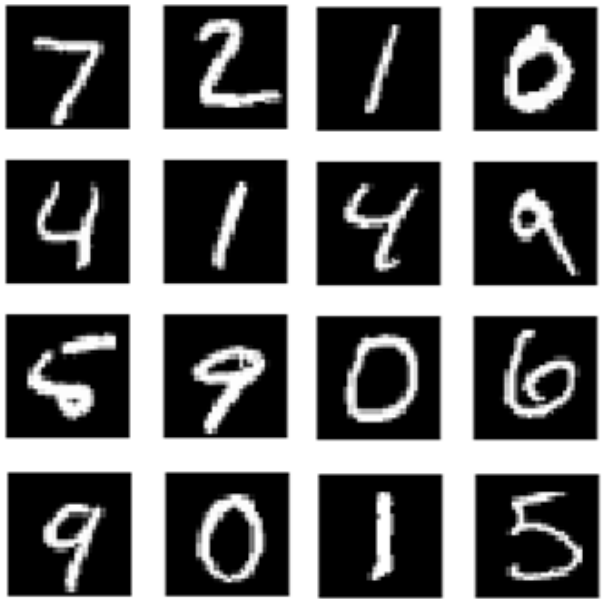
Source: IOSTAR, DRIVE, STARE, RC-SLO

MNIST Dataset

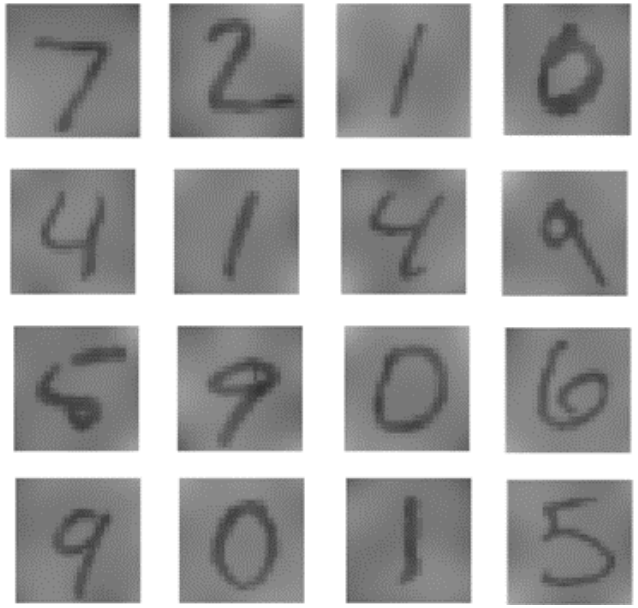
28 x 28

Source: Lecun

Augmented MNIST Dataset Input



MNIST original

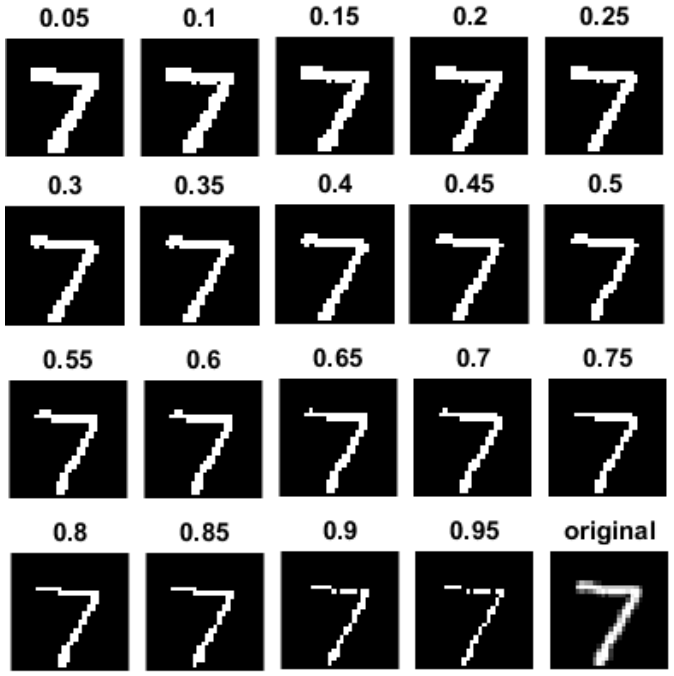


MNIST augmented

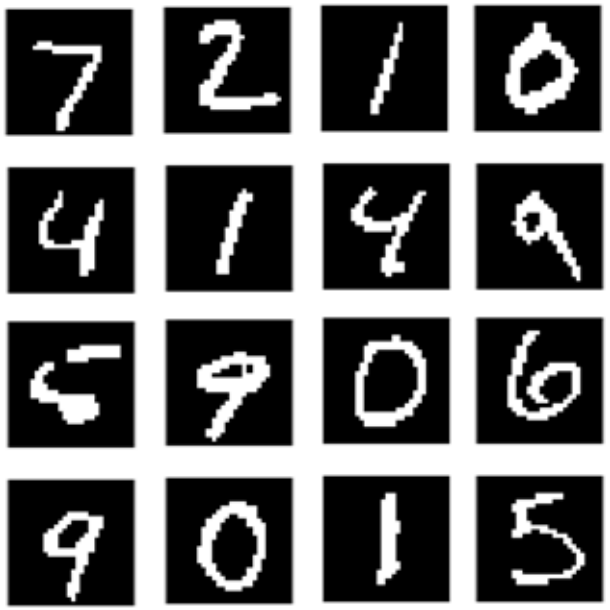
black and white fundus

Source: Hijazi

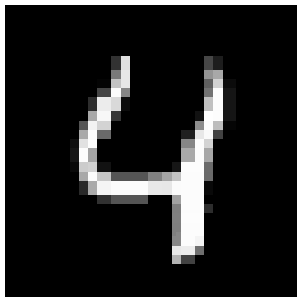
Segmentation Ground Truth



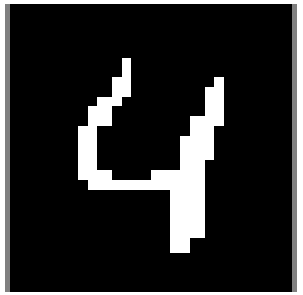
thresholding



segmentation ground truth
0.48 threshold

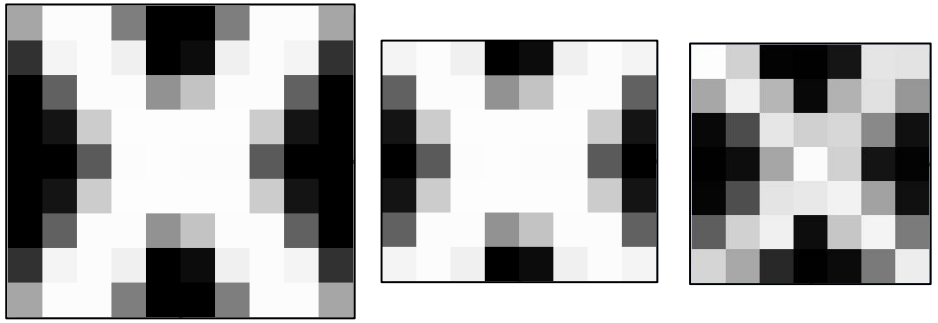


MNIST

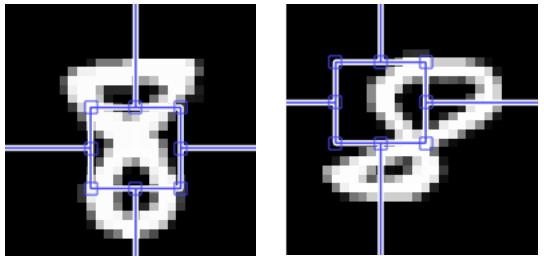


ground truth

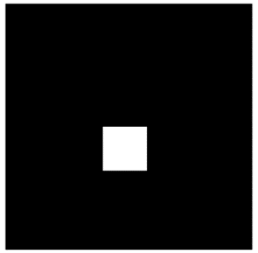
Detection Ground Truth



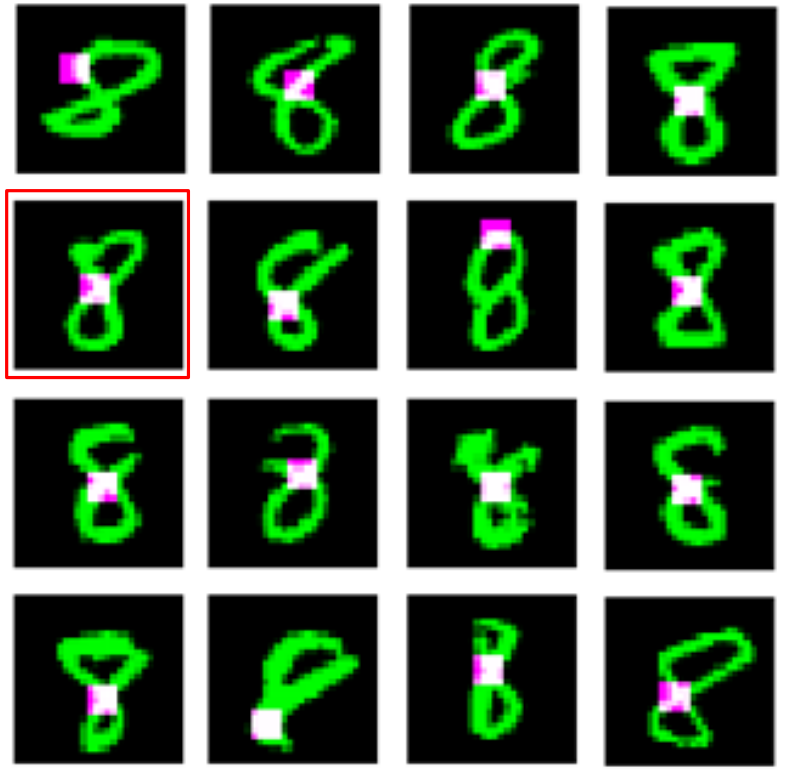
Masks for digit 8 crossing point



cross correlation

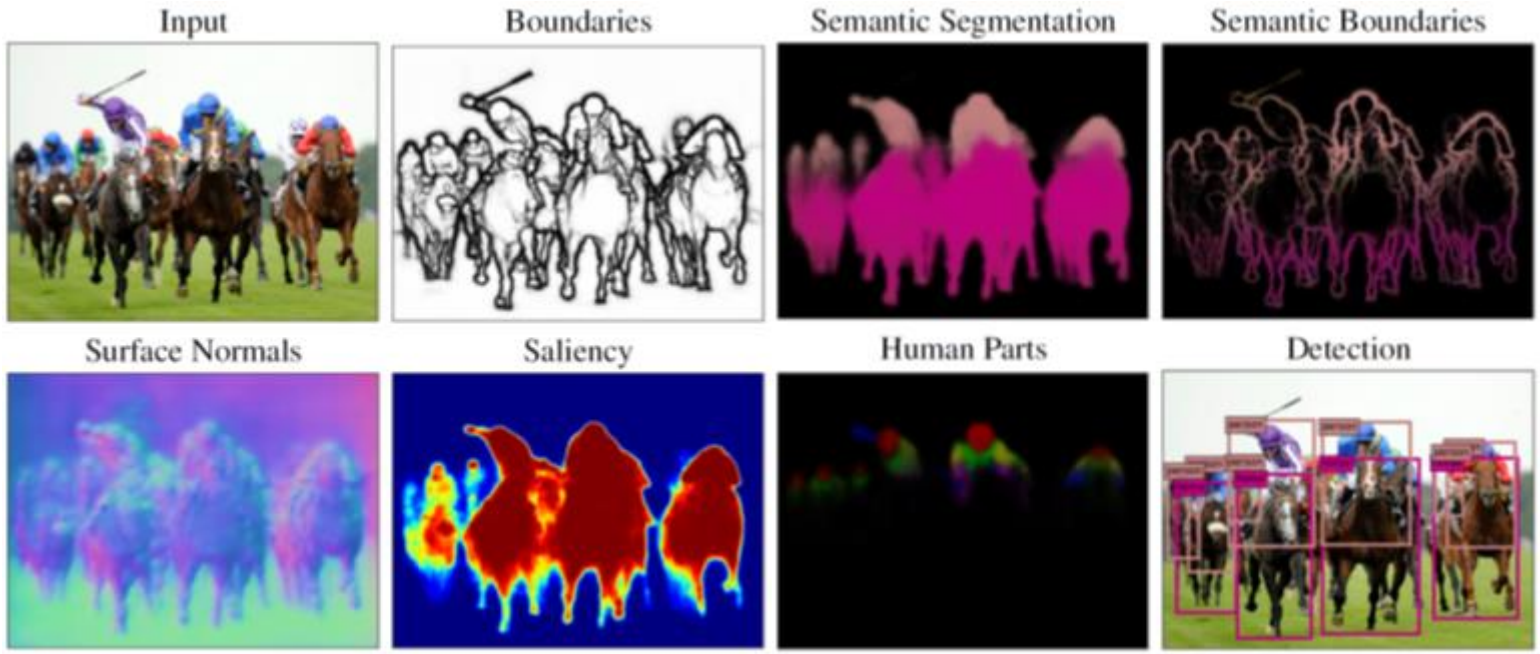


ground truth

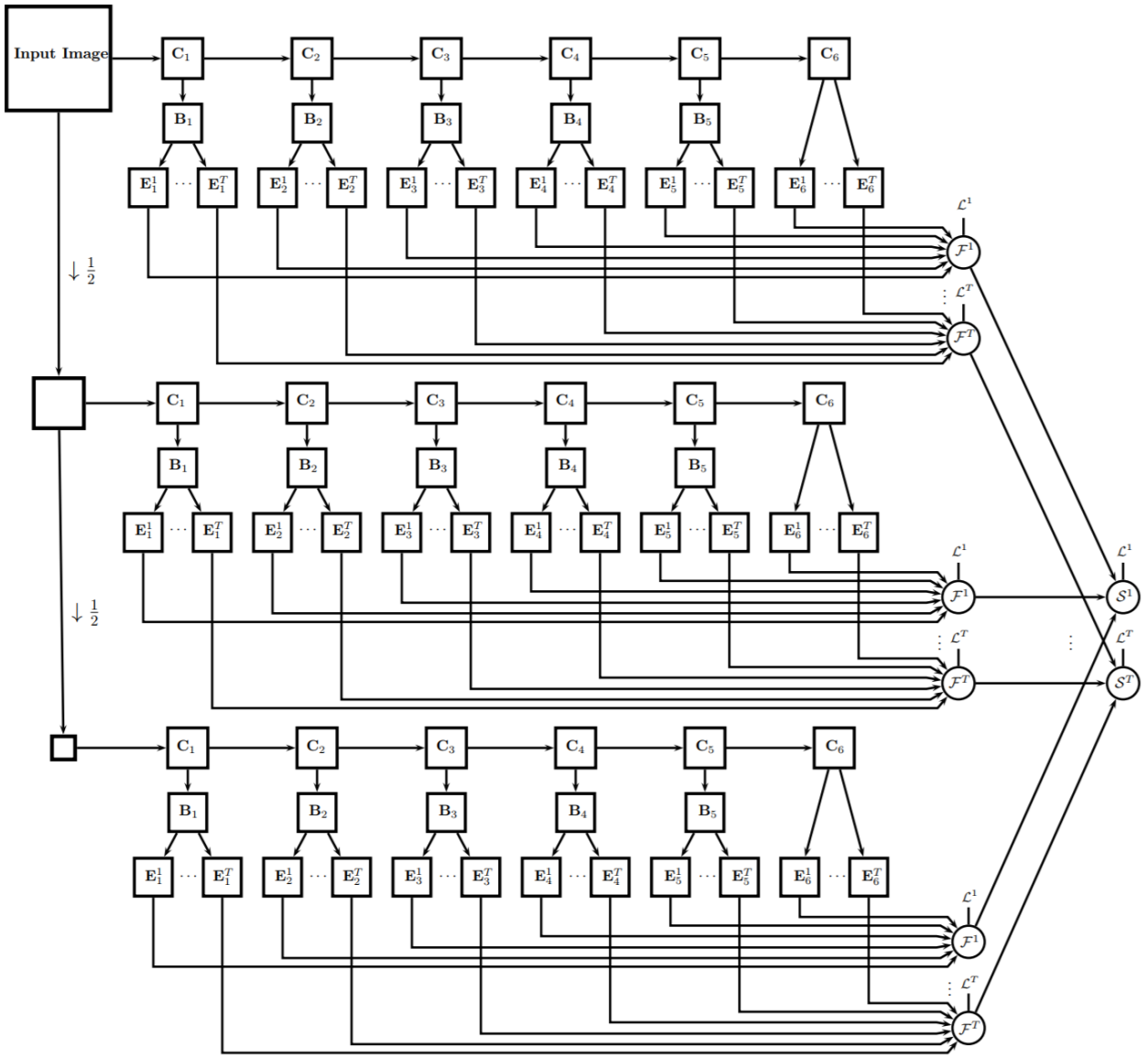


MNIST overlayed by ground truth

Reference: Ubernet



Source: Kokkinos

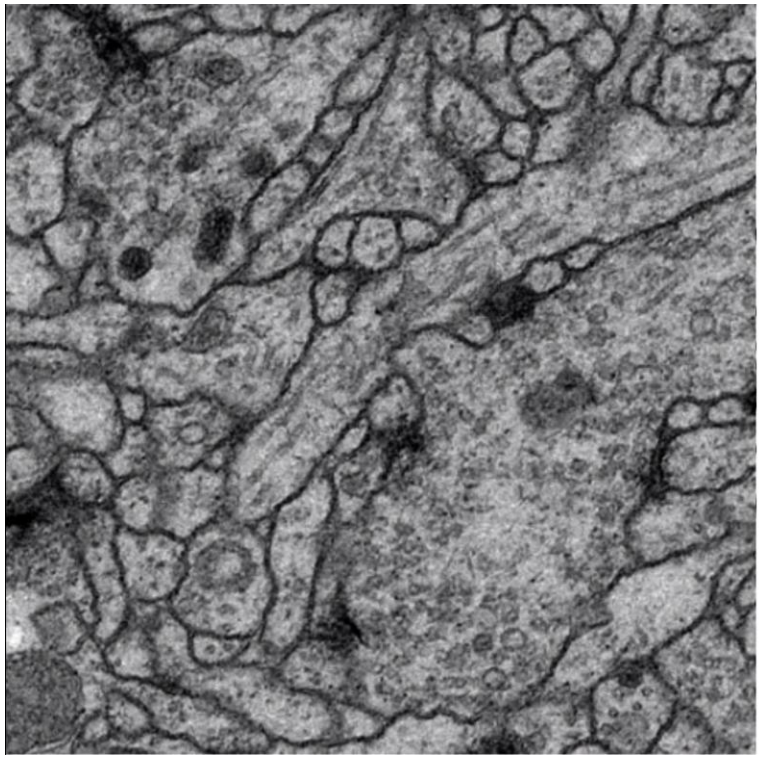


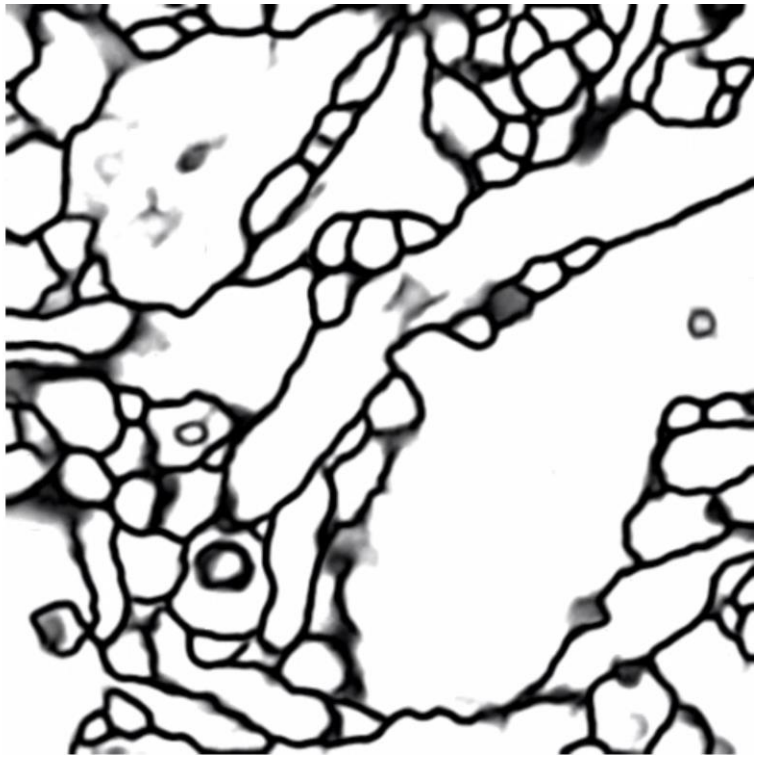
Key concept:

Branching from a common trunk into multiple tasks

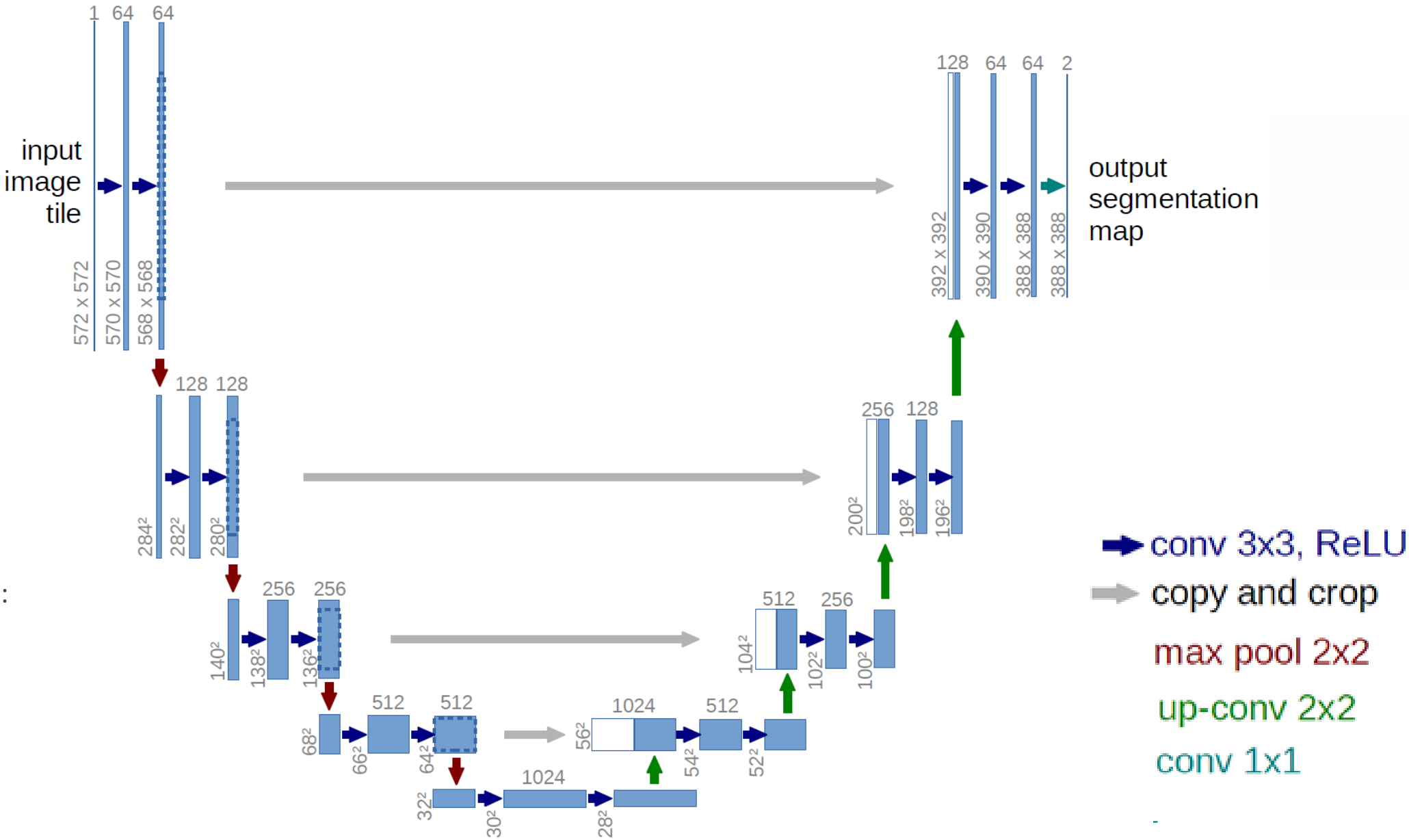
Source: Kokkinos

Reference: U-net





Source: Ronneberger et al, 2015



Key concept:

Streamlining of architecture into:

- Convolution
- Pooling
- Upsampling
- Skip layers

Source: Ronneberger et al, 2015

Architecture Iterations

segmentation

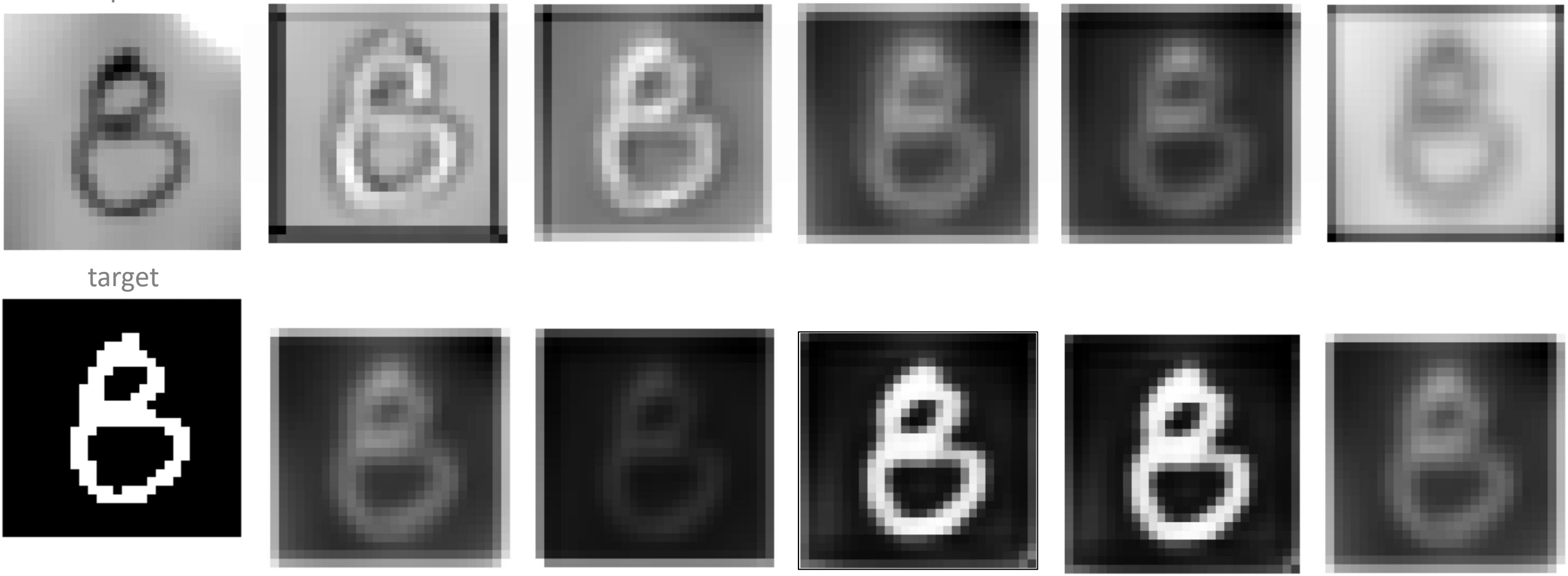
Network	Architecture
1v1	relu, linear
1v2	relu, sig
1v3	relu, pool, up, relu, sig
1v4	relu, pool, up, relu, relu, sig
1v5	relu, relu, sig
1v6	relu, pool, up, relu, sig, sig
1v7	**relu, pool, up, relu, sig, norm, sig**
1v8	relu, pool, up, relu, sig, norm, sig, norm, sig

detection

Network	Architecture
2v1	relu, sig
2v2	relu, pool, up, relu, sig
2v3	relu, pool, relu, pool, relu, up, relu, up, relu, sig
2v4	relu x2, pool, relu x2, pool, relu x2, up, relu x2, up, relu x2, sig
2v5	**relu x3, pool, relu x3, pool, relu x3, up, relu x3, up, relu x3, sig**
2v6	relu x4, pool, relu x4, pool, relu x4, up, relu x4, up, relu x4, sig
2v7	relu x3, pool, relu x3, pool, relu x3, up, relu x3, up, relu x3, sig, norm, sig

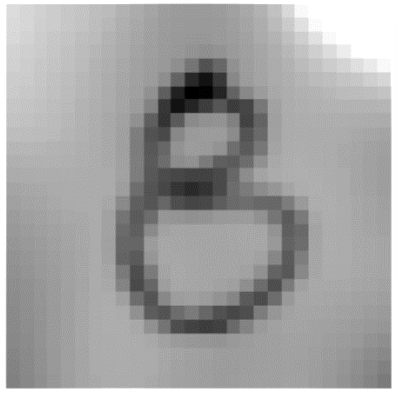
input

segmentation predictions

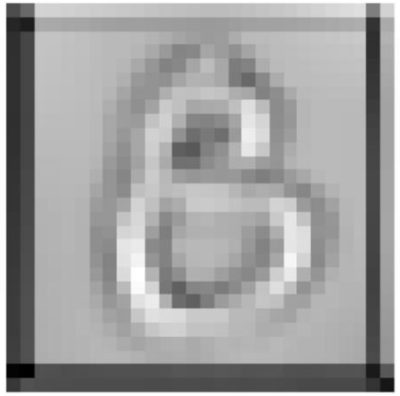


target

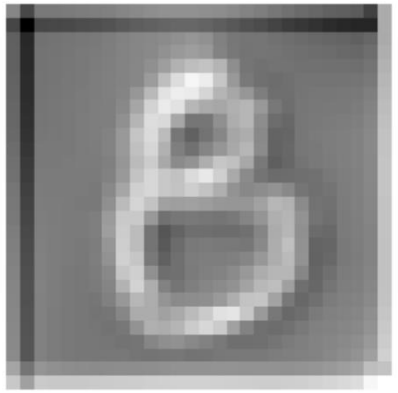
input



1v1

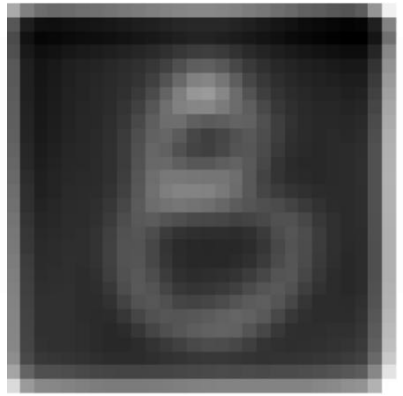


1v2

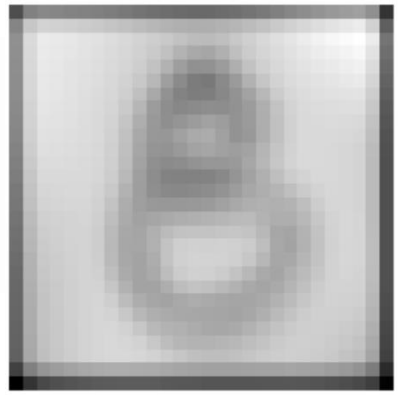


1v3

1v4

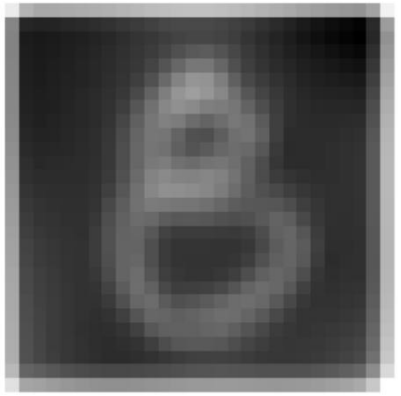


weighted



target

1v5

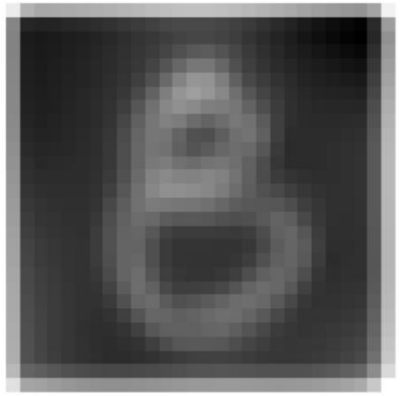


1v6

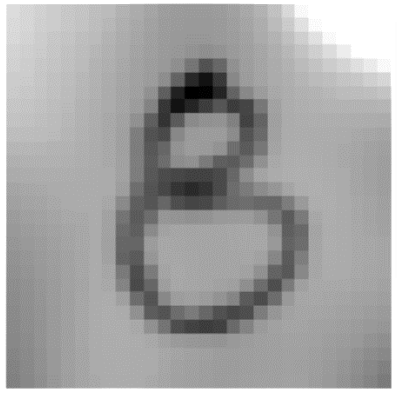
1v7

1v8

inverse weighted

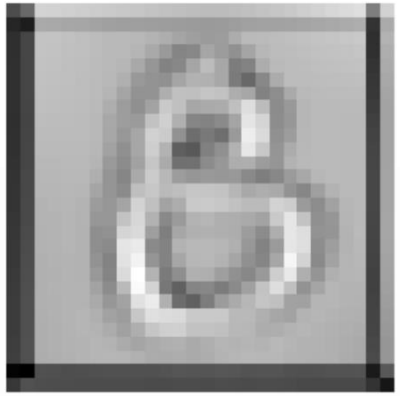


input



target

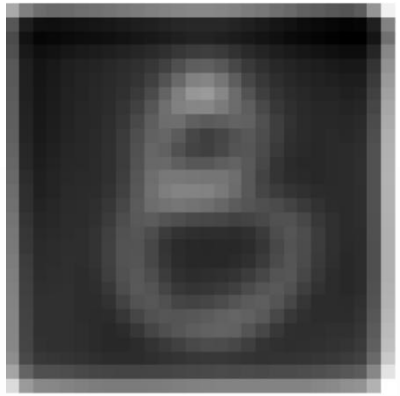
1v1



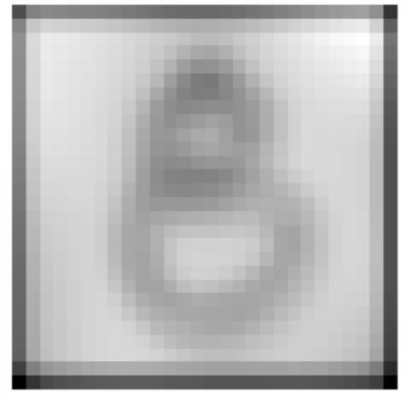
1v2

1v3

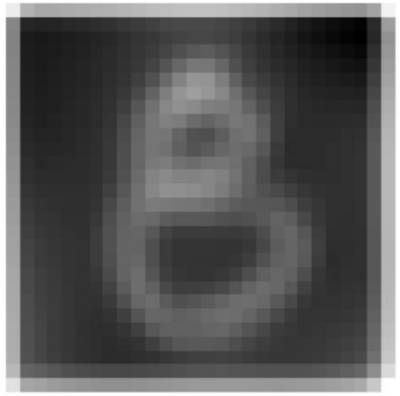
1v4



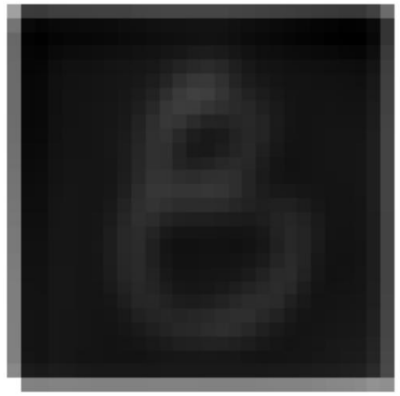
weighted



1v5



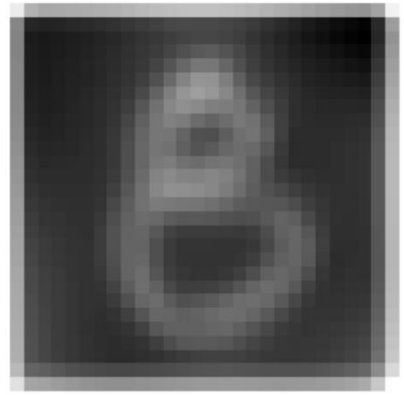
1v6



1v7

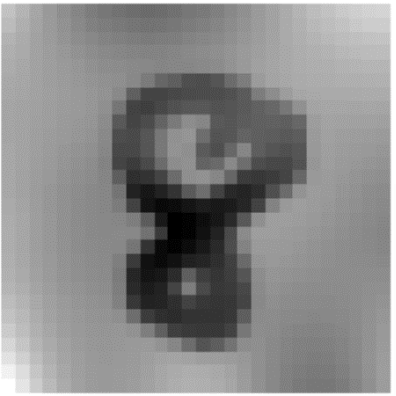
1v8

inverse weighted

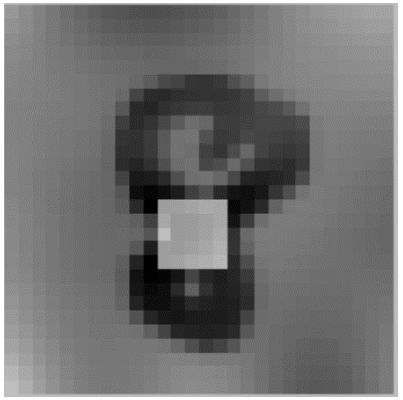


1v1	relu, linear	**1v5**	relu, relu, sig
1v2	relu, sig	**1v6**	relu, pool, up, relu, sig, sig
1v3	relu, pool, up, relu, sig	**1v7**	relu, pool, up, relu, sig, norm, sig
1v4	relu, pool, up, relu, relu, sig	**1v8**	relu, pool, up, relu, sig, norm, sig, norm, sig

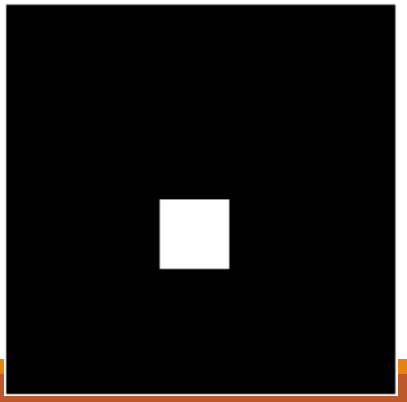
input



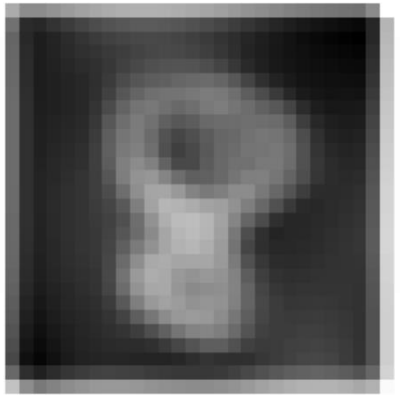
target over input

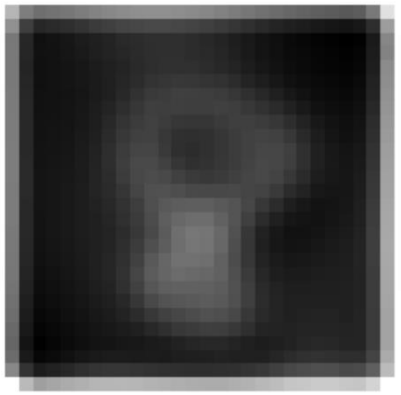


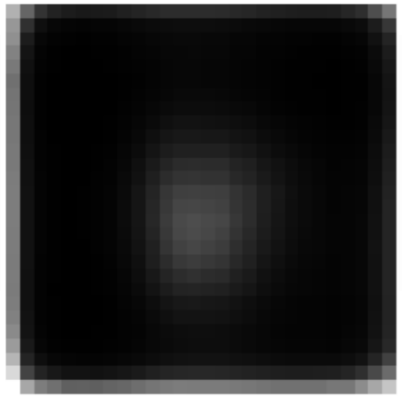
target

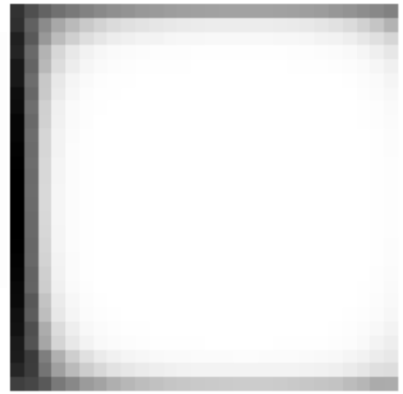


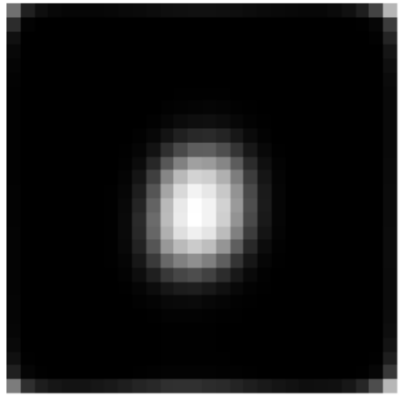
detection predictions

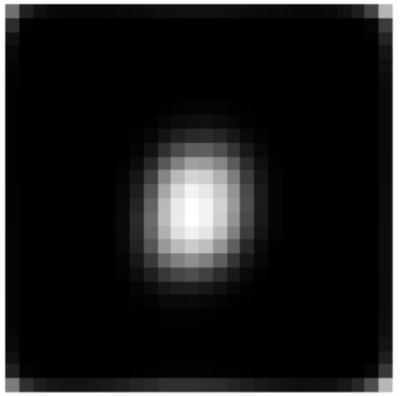


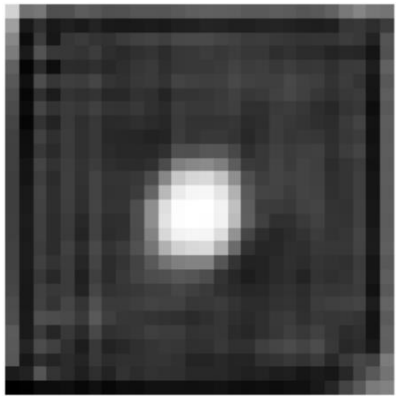


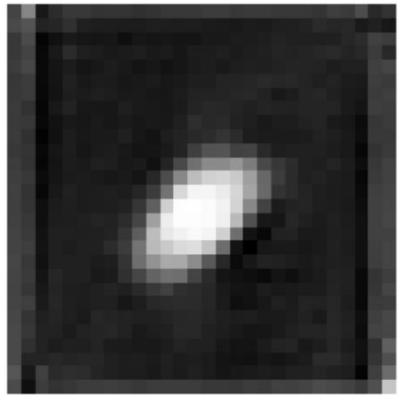


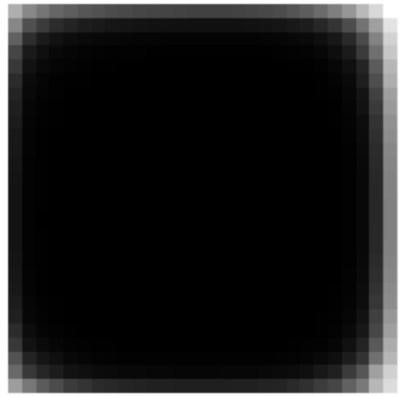




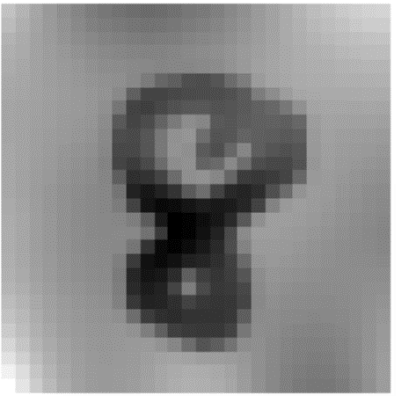




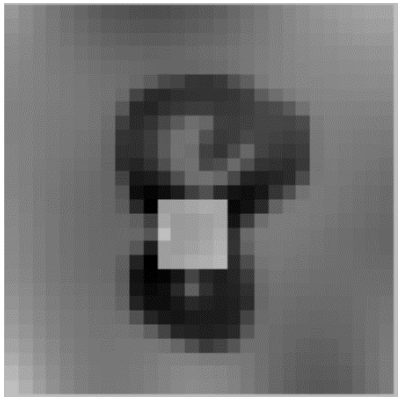




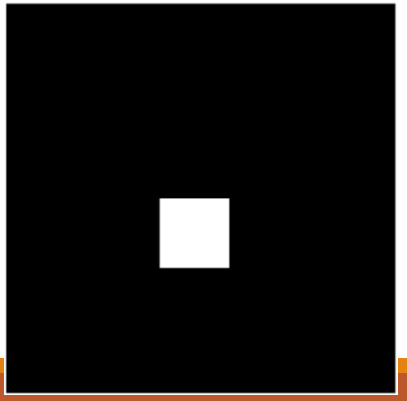
input



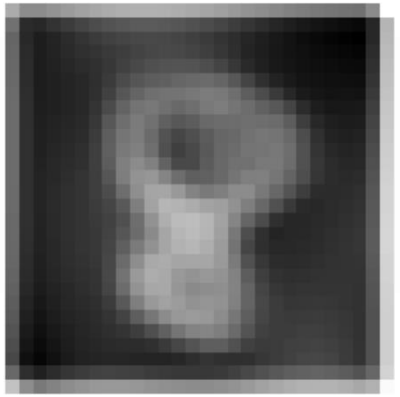
target over input



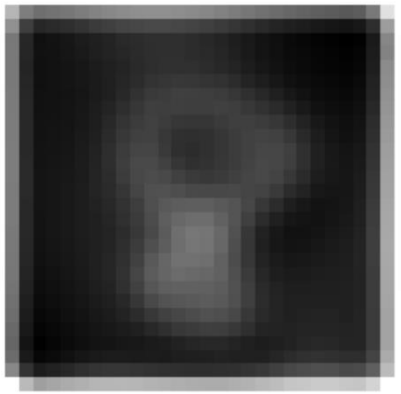
target



2v1

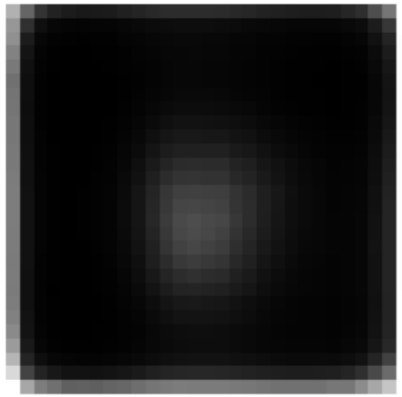


2v2

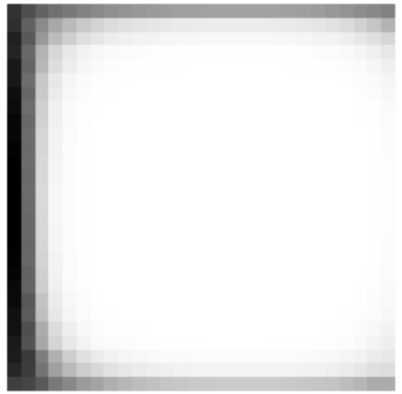


2v3

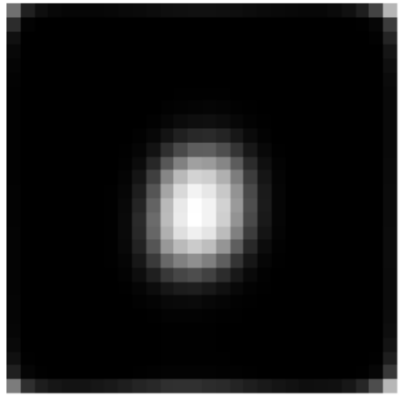
2v4



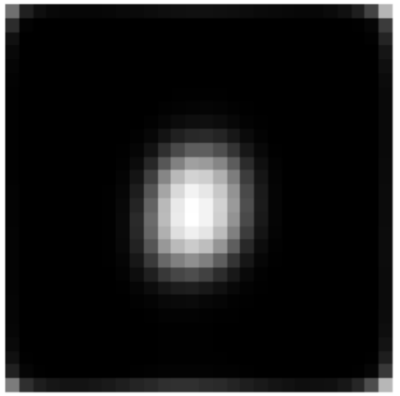
weighted



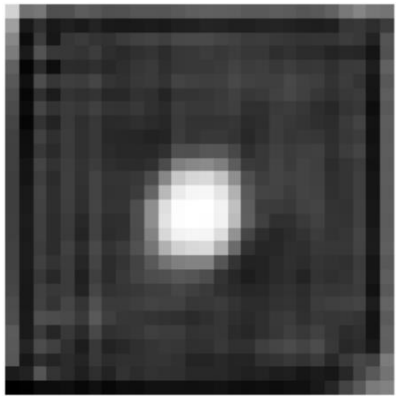
2v5



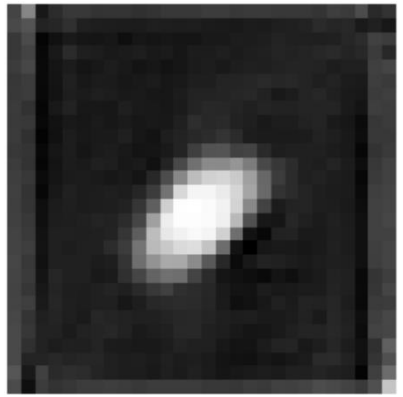
2v6



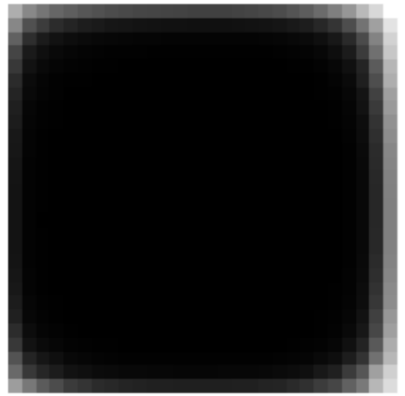
2v7



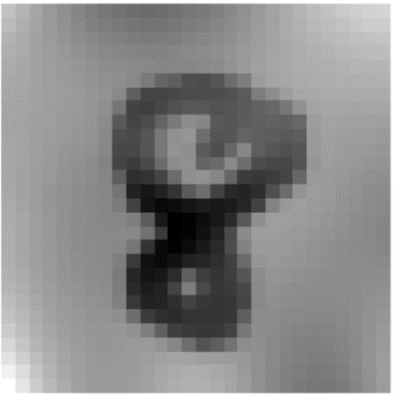
2v8



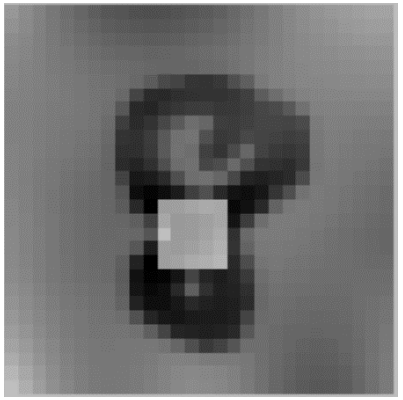
inverse weighted



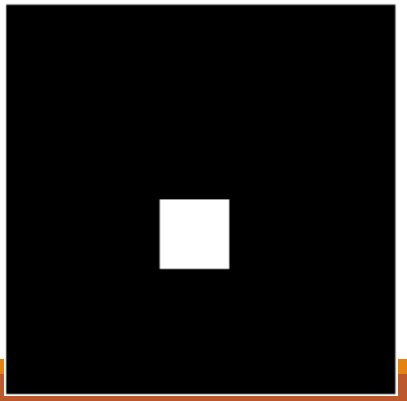
input



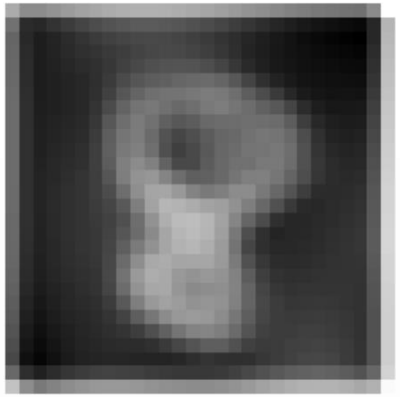
target over input



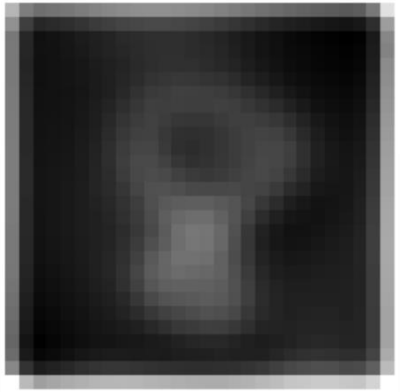
target



2v1

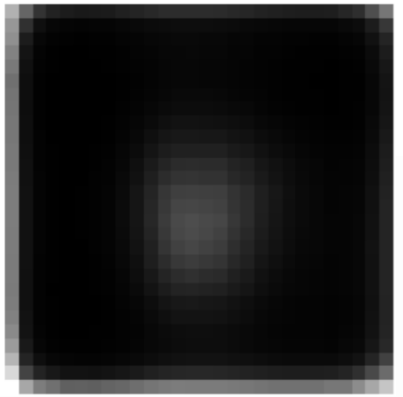


2v2

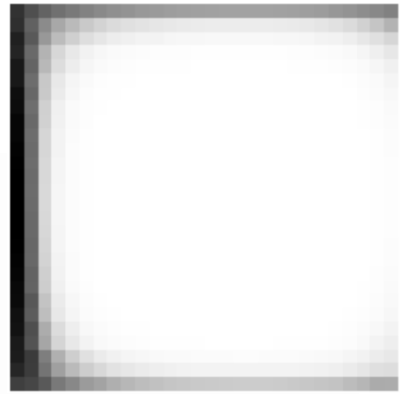


2v3

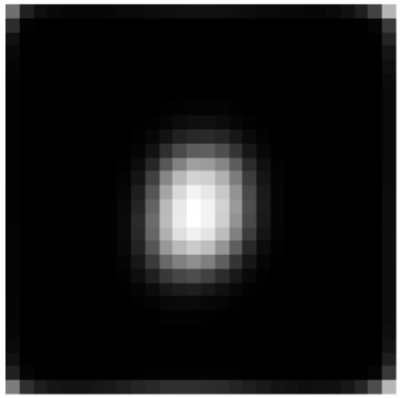
2v4



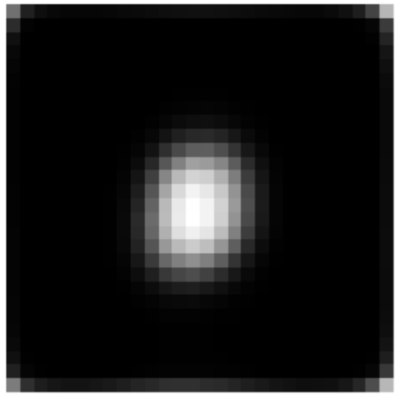
weighted



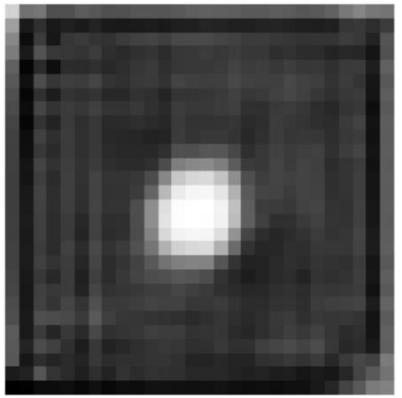
2v5



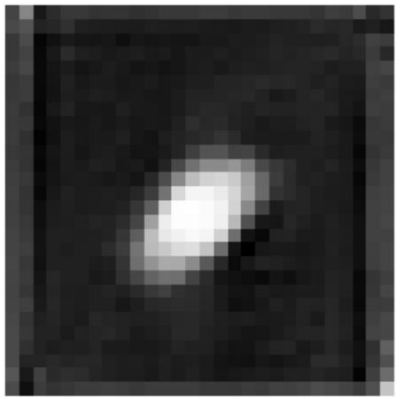
2v6



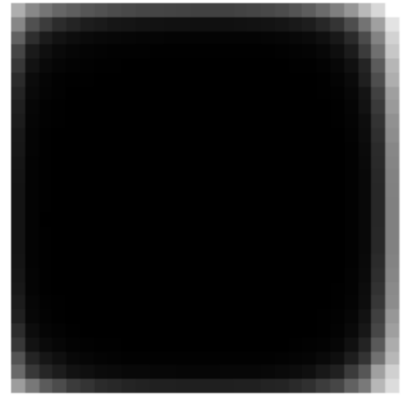
2v7



2v8

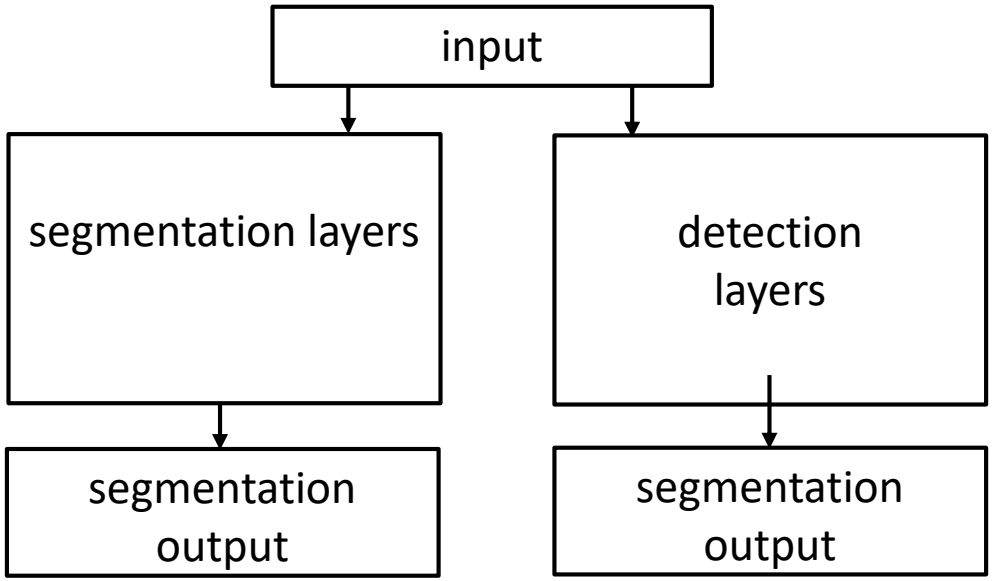


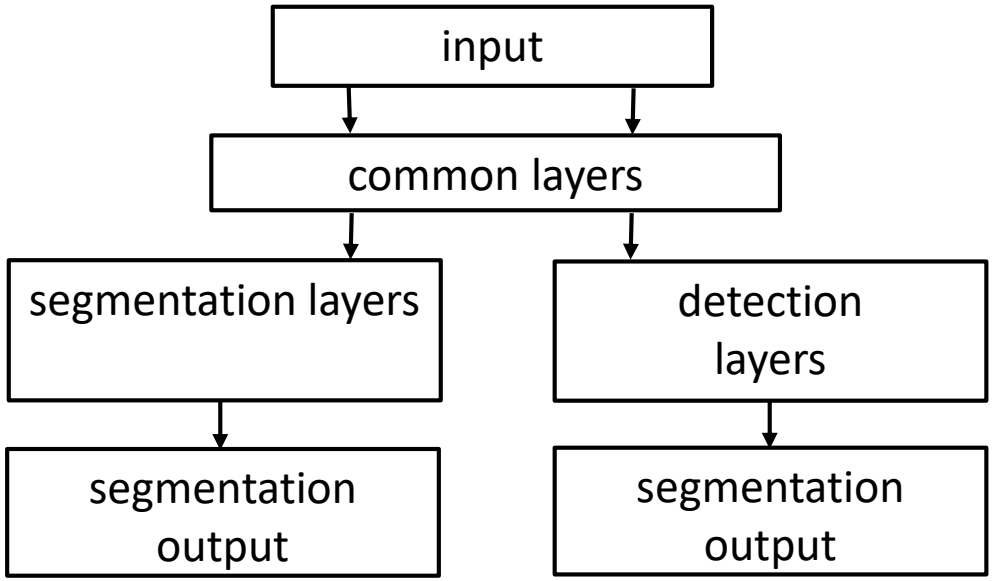
inverse weighted



2v1	relu, sig	**2v5**	relu x2, pool, relu x2, pool, relu x2, up, relu x2, up, relu x2, sig
2v2	relu, pool, up, relu, sig	**2v6**	**relu x3, pool, relu x3, pool, relu x3, up, relu x3, up, relu x3, sig**
2v3	relu, pool, relu, pool, relu, up, relu, up, relu, sig	**2v7**	relu x4, pool, relu x4, pool, relu x4, up, relu x4, up, relu x4, sig
2v4	relu x2, pool, relu x2, pool, relu x2, up, relu x2, up, relu x2, sig	**2v8**	relu x3, pool, relu x3, pool, relu x3, up, relu x3, up, relu x3, sig, norm, sig

Branching


input
segmentation layers
detection layers
segmentation output
segmentation output


input
common layers
segmentation layers
detection layers
segmentation output
segmentation output

Branched Training Iteration

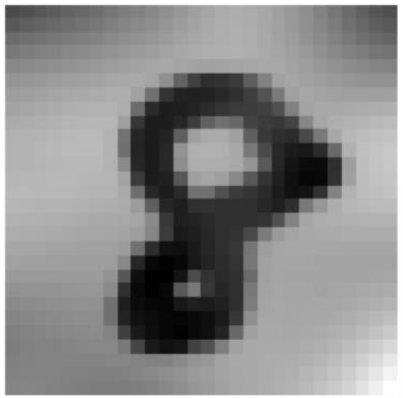
Architecture

Task	Architecture
segmentation	relu, pool, up, relu, sig, norm, sig
detection	relu x3, pool, relu x3, pool, relu x3, up, relu x3, up, relu x3, sig
common segmentation detection	relu, pool, > up, relu, sig, norm, sig > relu x3, pool, relu x3, up, relu x3, up, relu x3, sig

Training

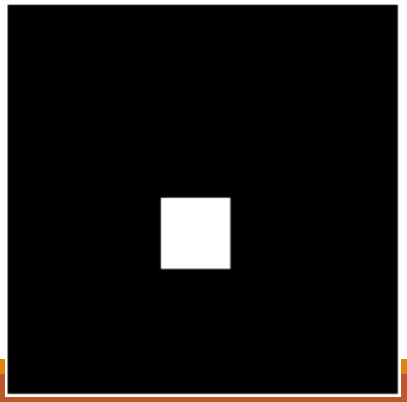
Training	Epochs	Target
3v1	15	both
3v2	15	both
3v3	30	both
3v4	15 each, 75 total	both, seg, det, seg, det

input

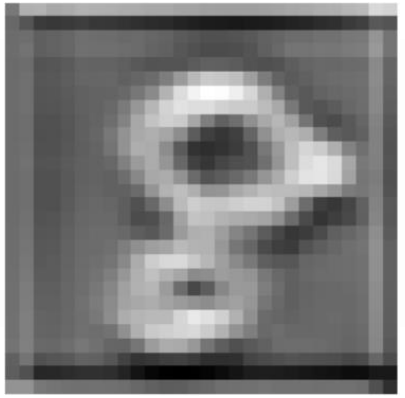


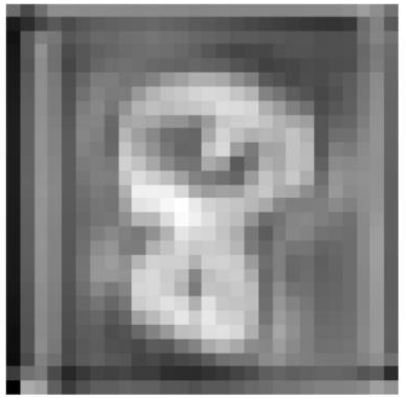
segmentation

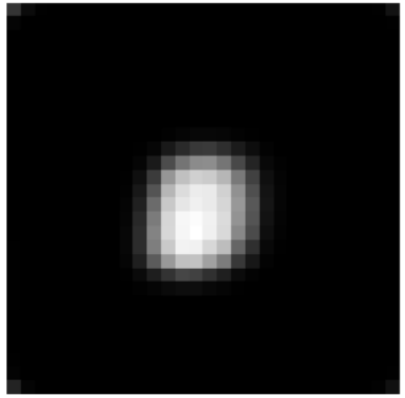
detection

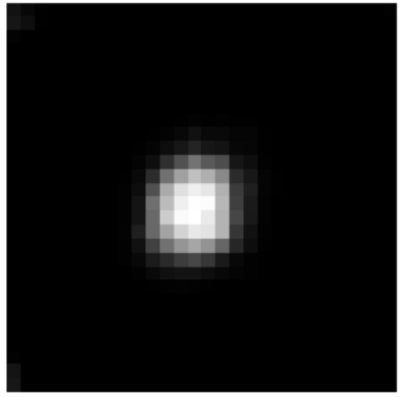


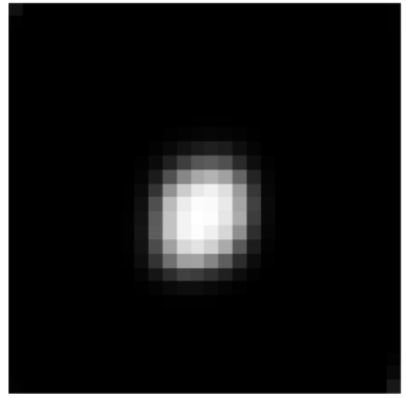
branched predictions

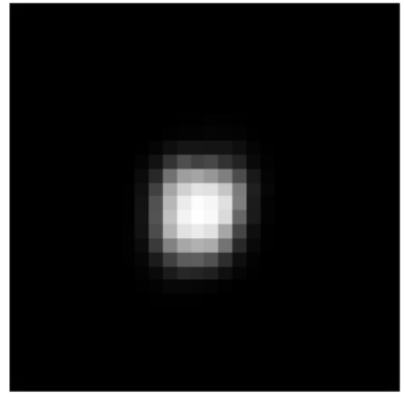




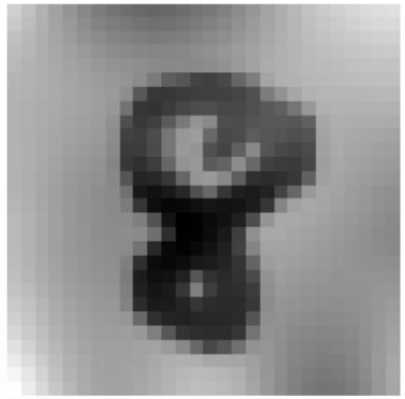






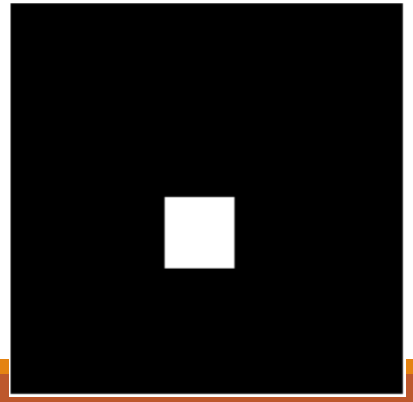


input

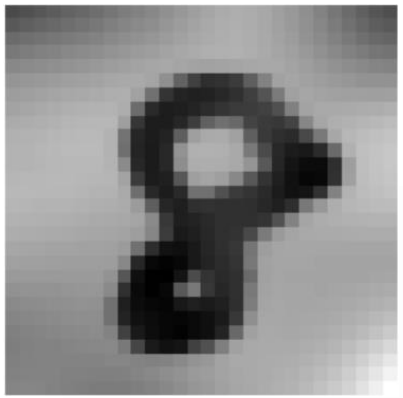


segmentation

detection

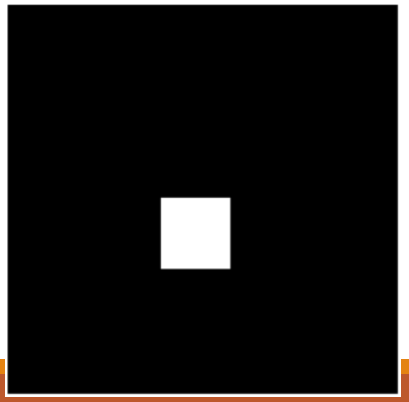


input

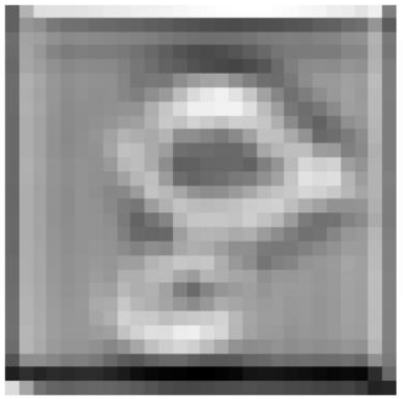


segmentation

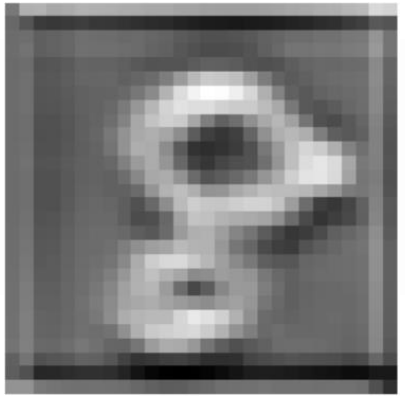
detection



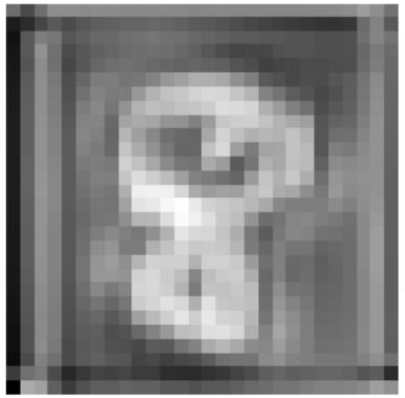
branched predictions



3v1

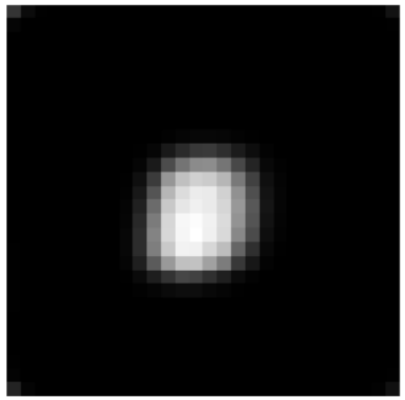


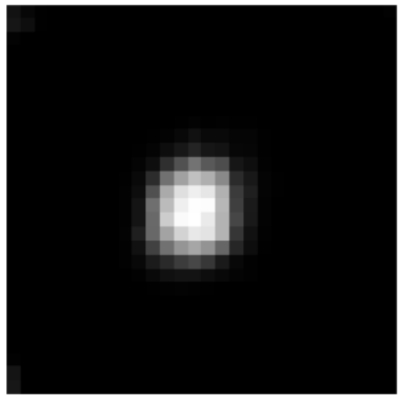
3v2

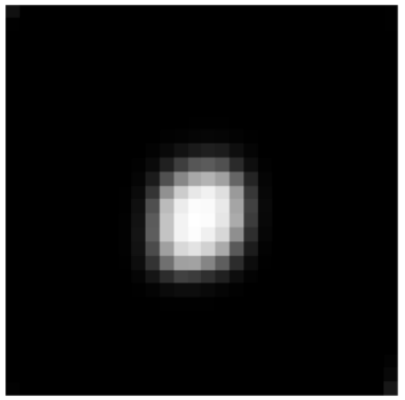


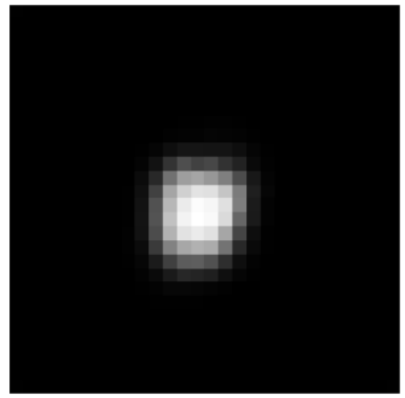
3v3

3v4

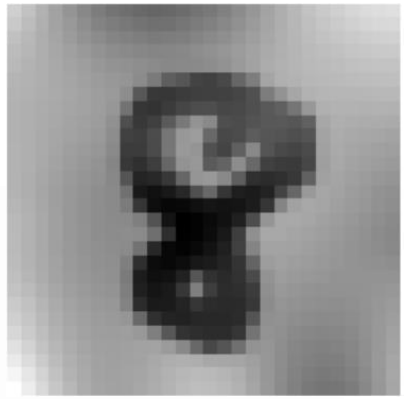






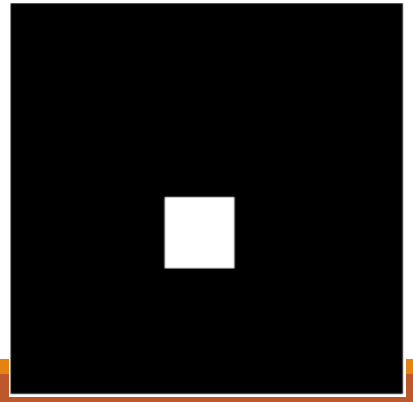


input

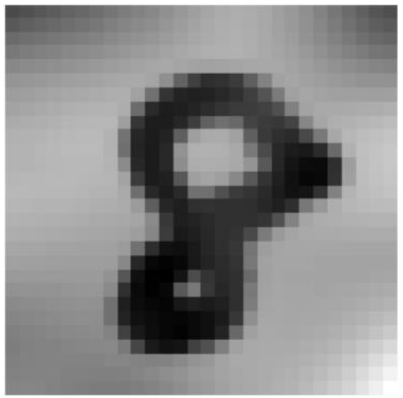


segmentation

detection

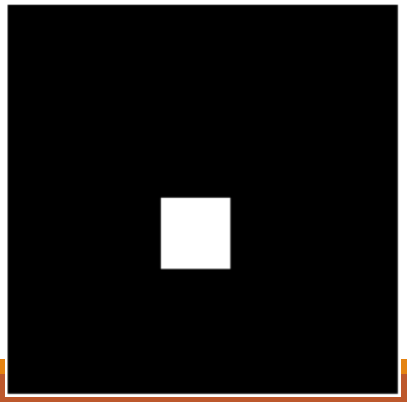


input

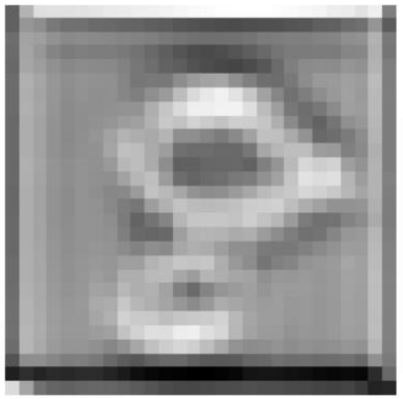


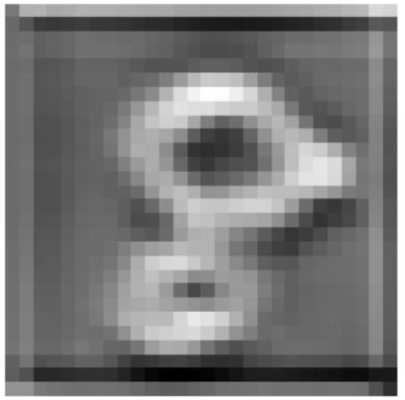
segmentation

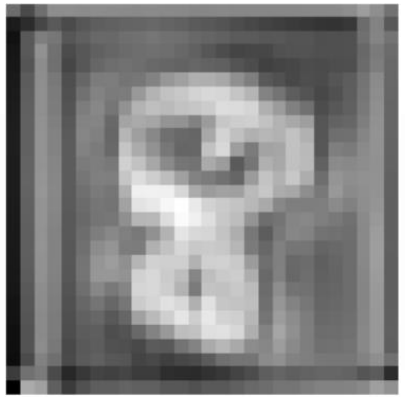
detection



branched predictions





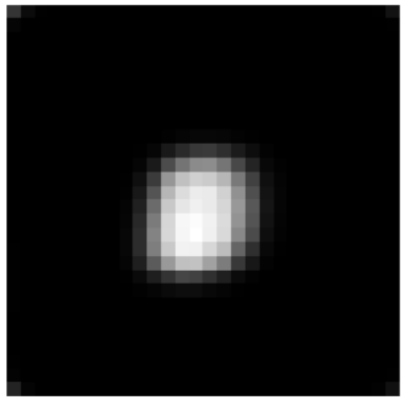


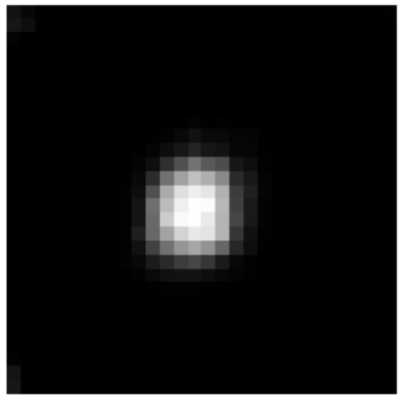
3v1

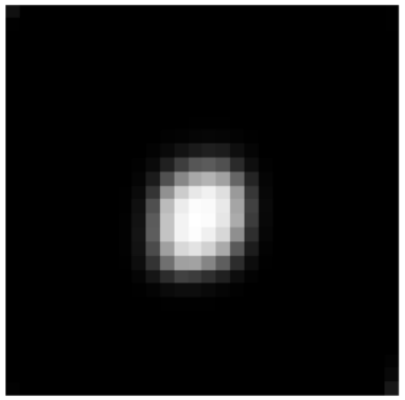
3v2

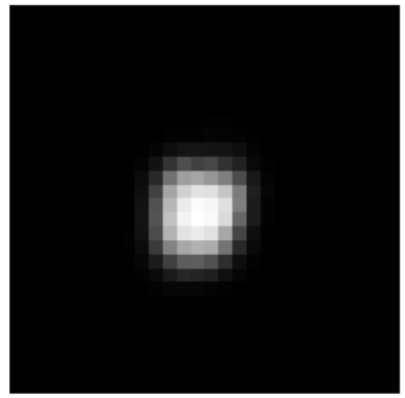
3v3

3v4

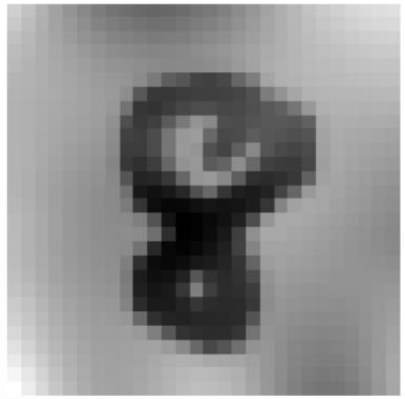






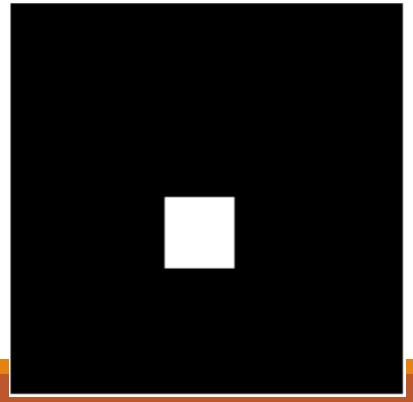


input



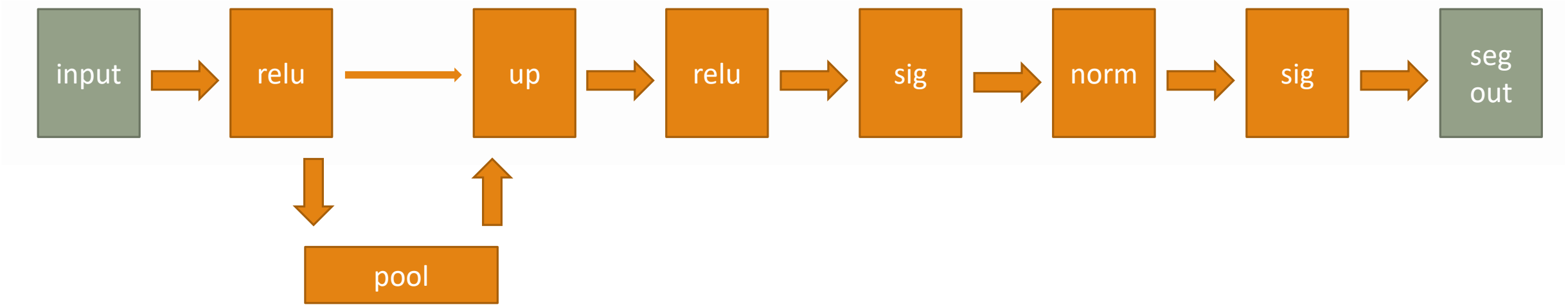
segmentation

detection

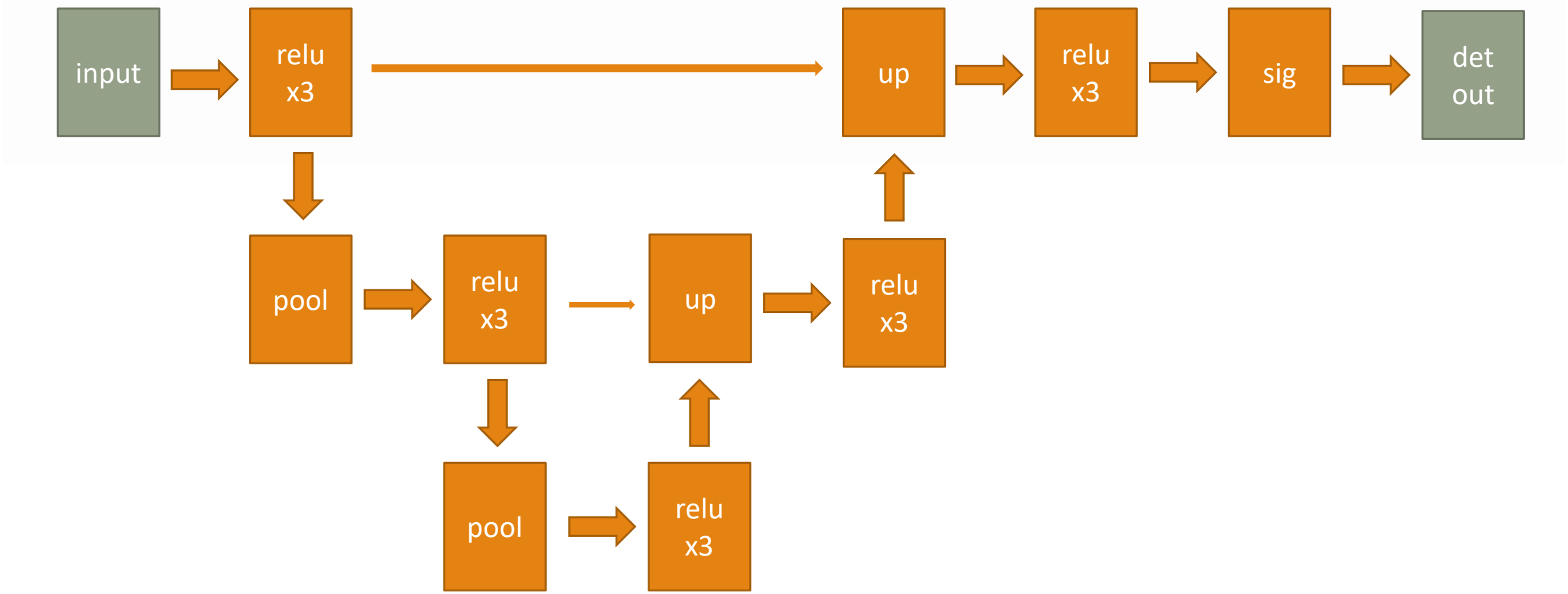


Training	Epochs	Target
3v1	15	both
3v2	15	both
3v3	30	both
3v4	15 each, 75 total	both, seg, det, seg, det

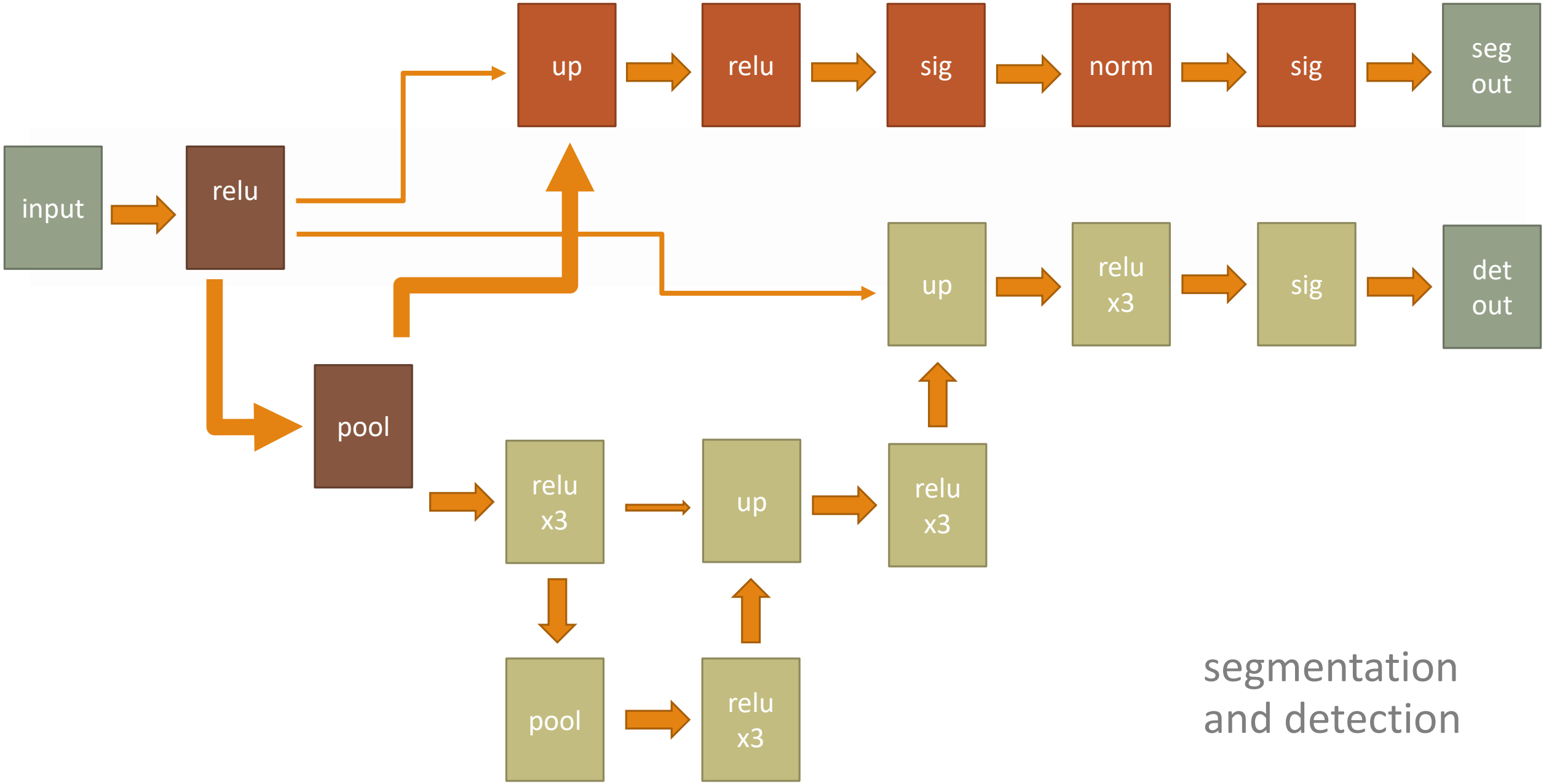
Final Architectures



segmentation

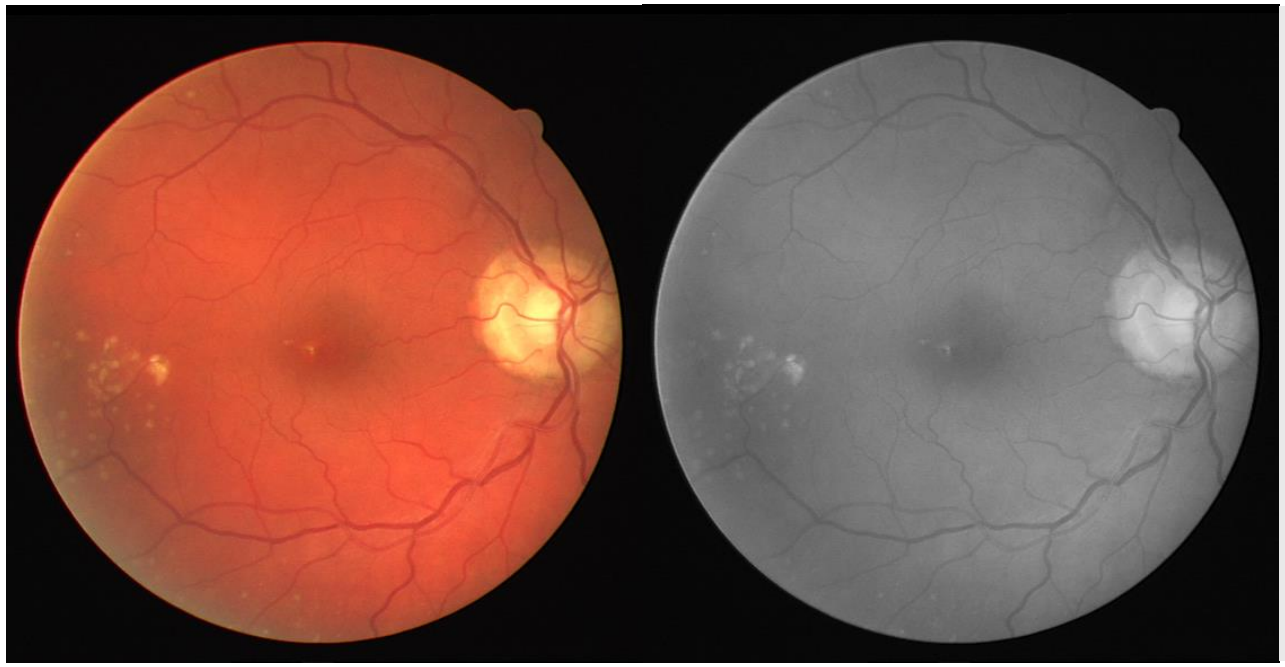

input
relu x3
up
relu x3
sig
det out
pool
relu x3
up
relu x3
pool
relu x3

detection



segmentation and detection

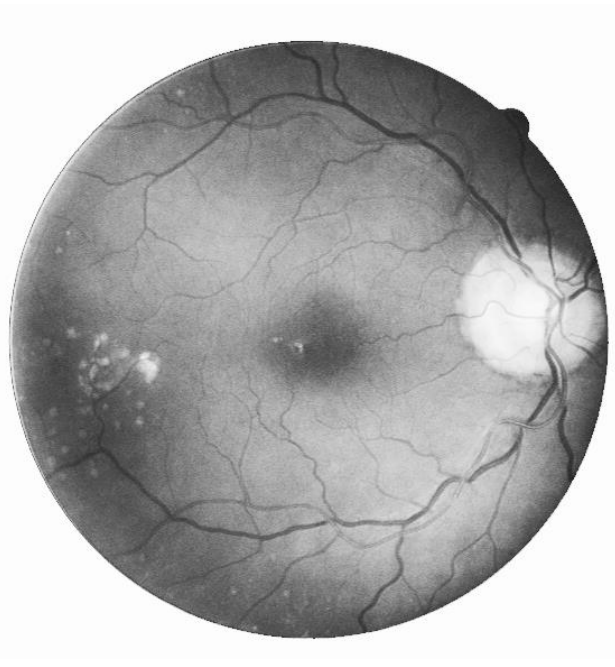
Fundus Dataset



original

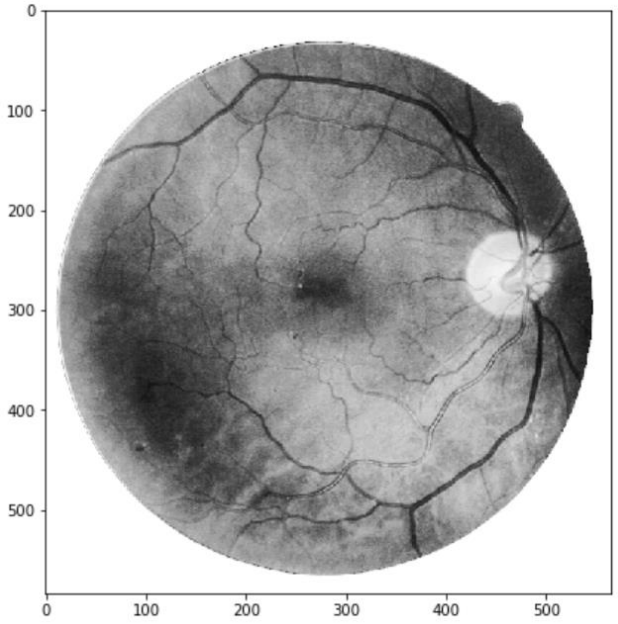
black and white
v1

lightened background
v2

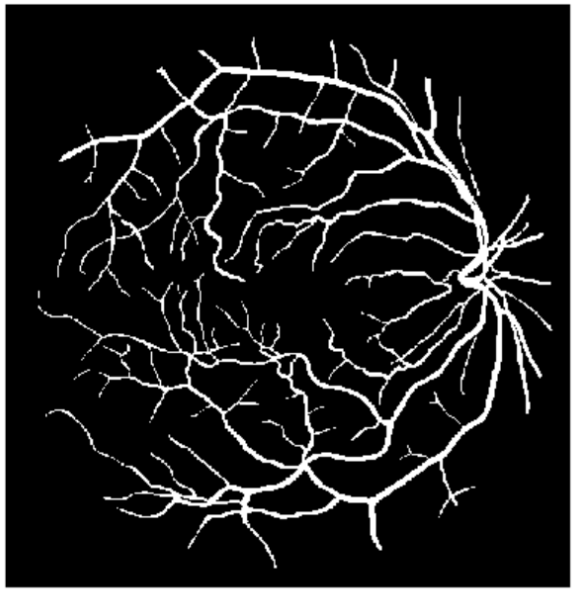


increased contrast
v3

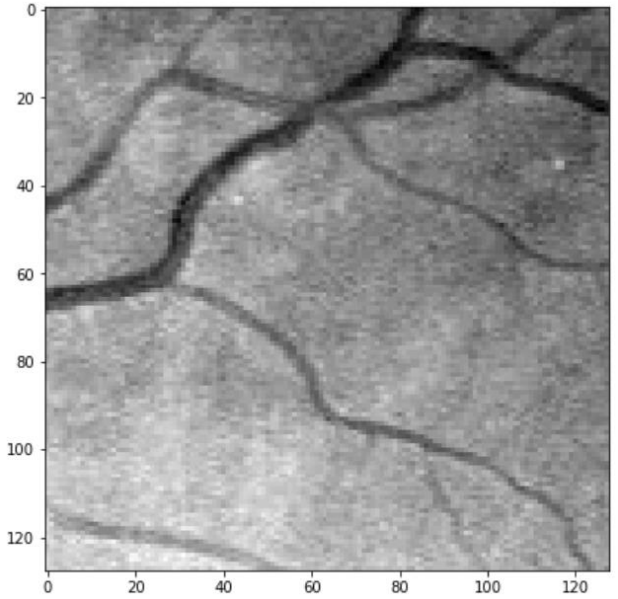
Fundus Training



original size
584L x 565



original segmentation

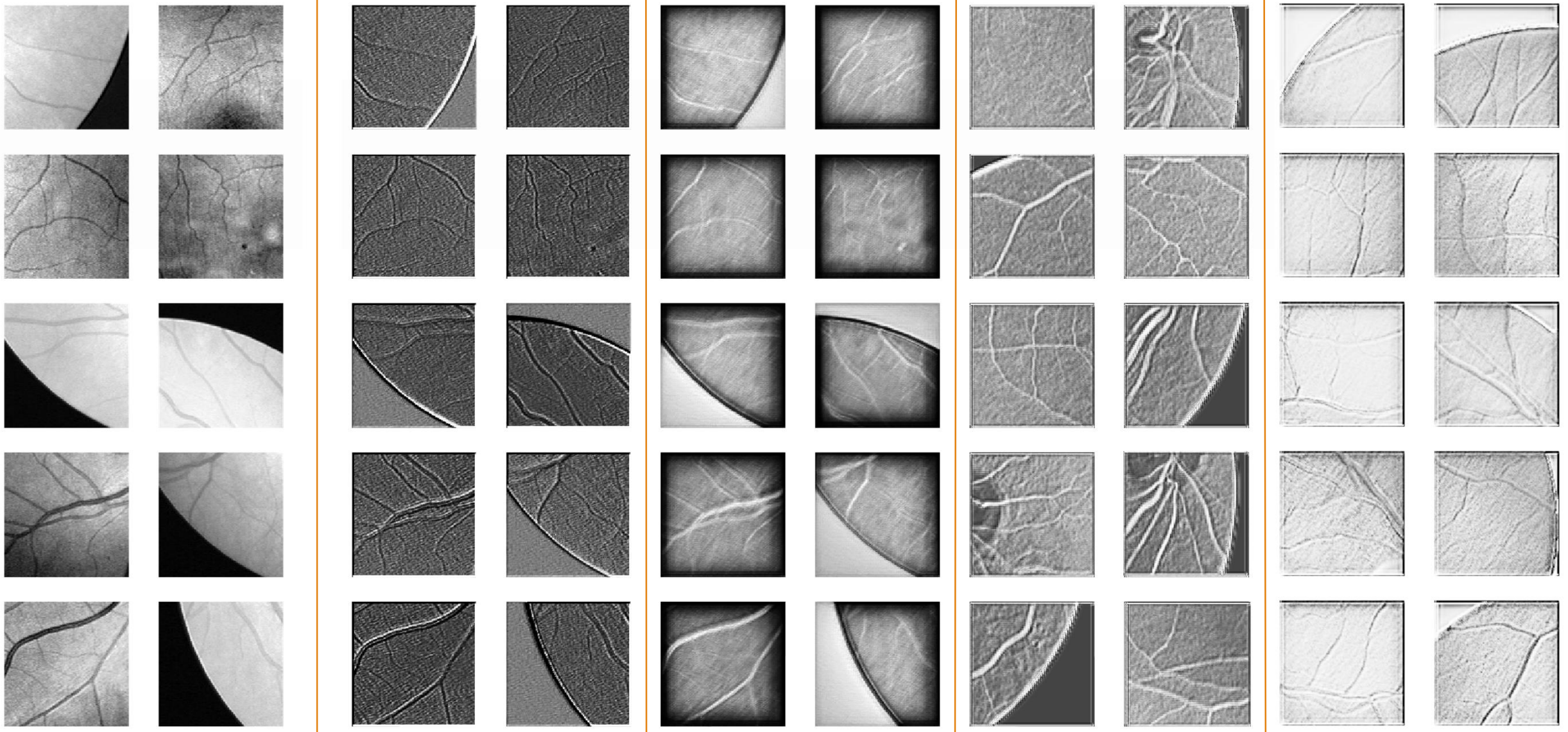


crop
128 x 128

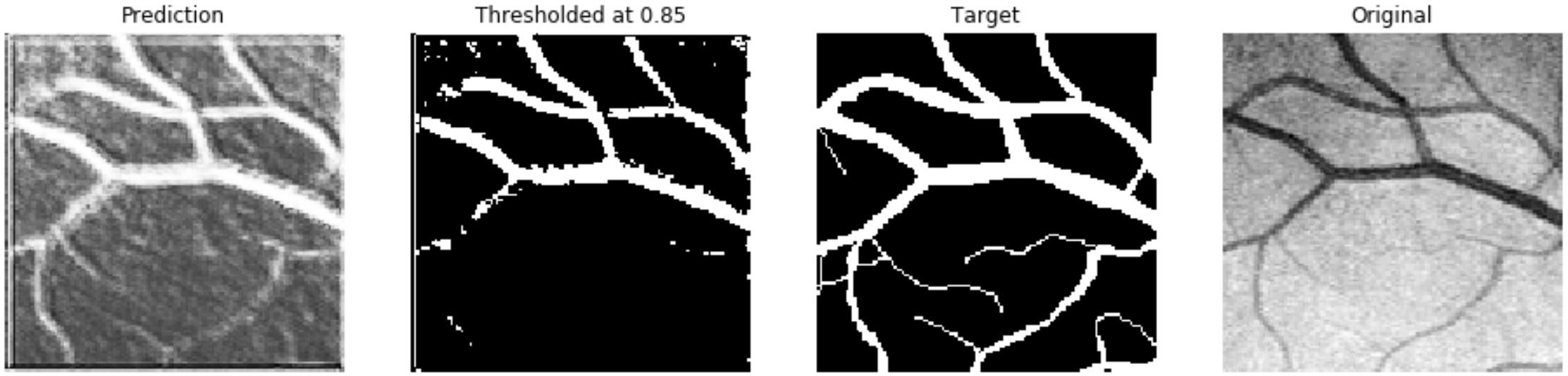
segmentation crop

input

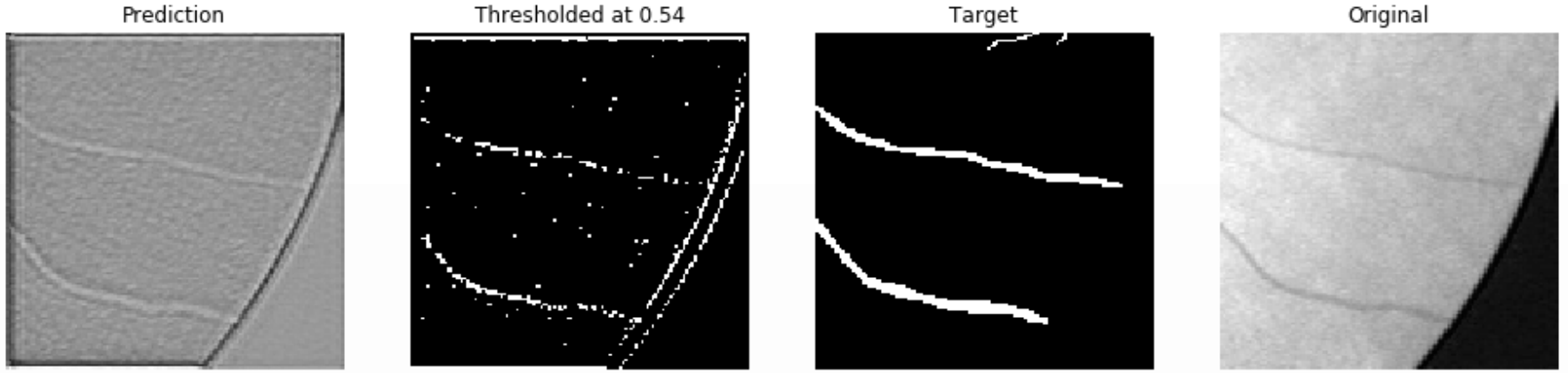
various predictions



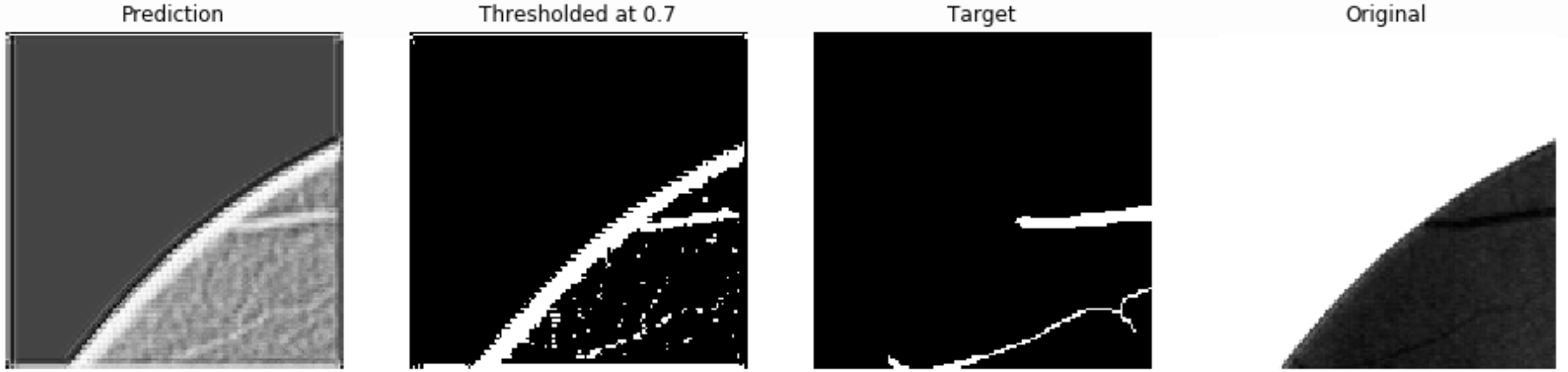
Thresholding



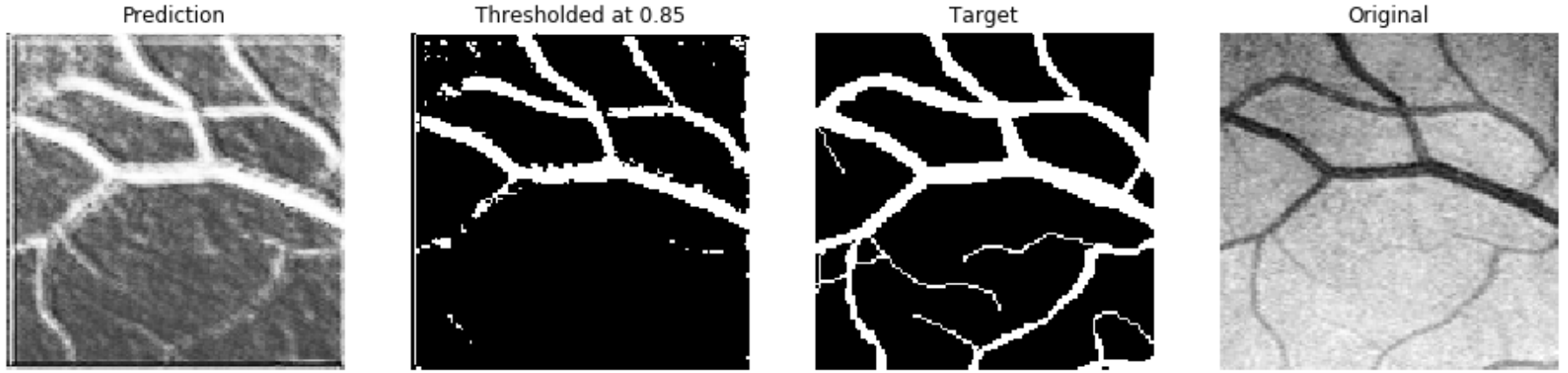
v1


Prediction
Thresholded at 0.54
Target
Original

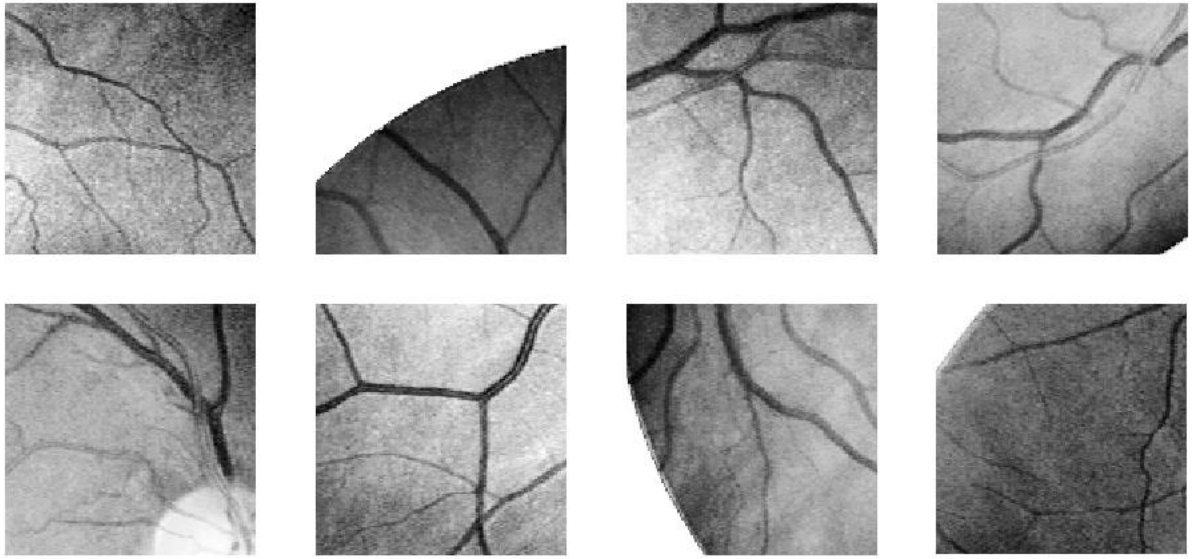
v2


Prediction
Thresholded at 0.7
Target
Original

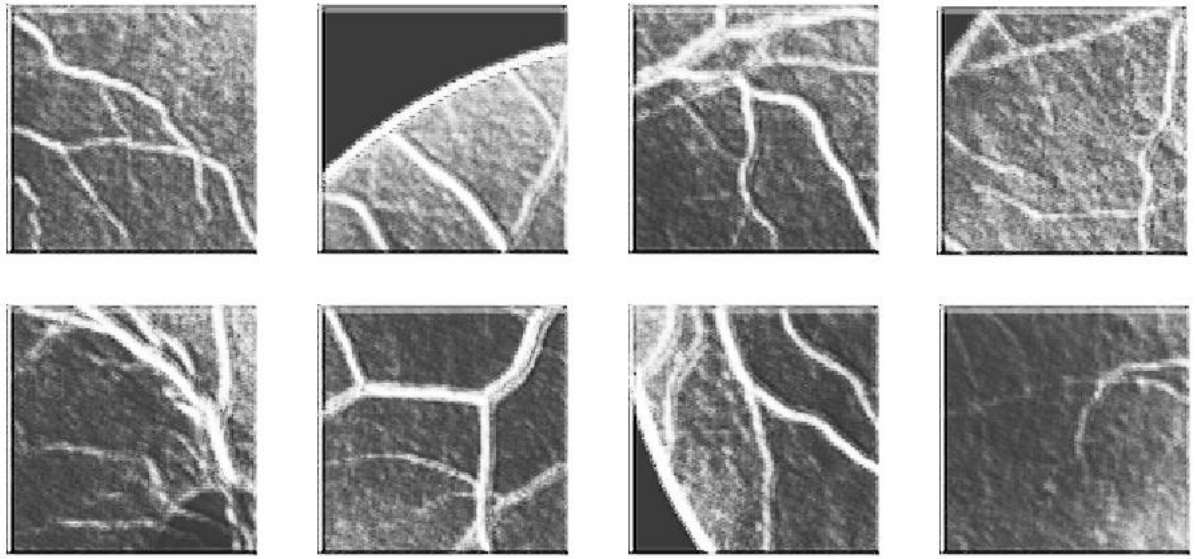
v3

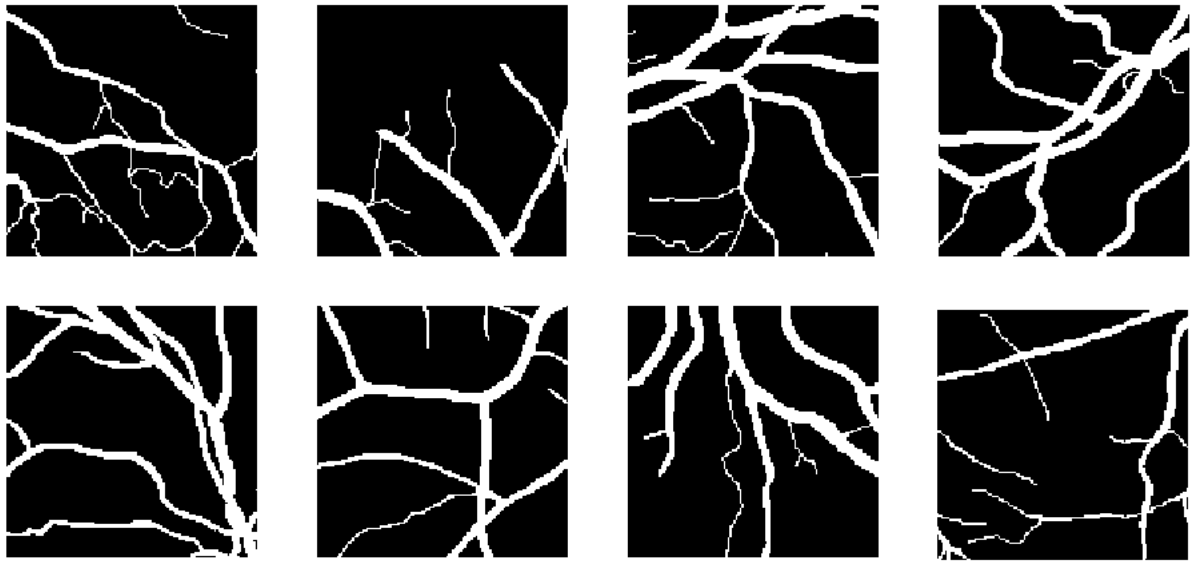

Prediction
Thresholded at 0.85
Target
Original

input



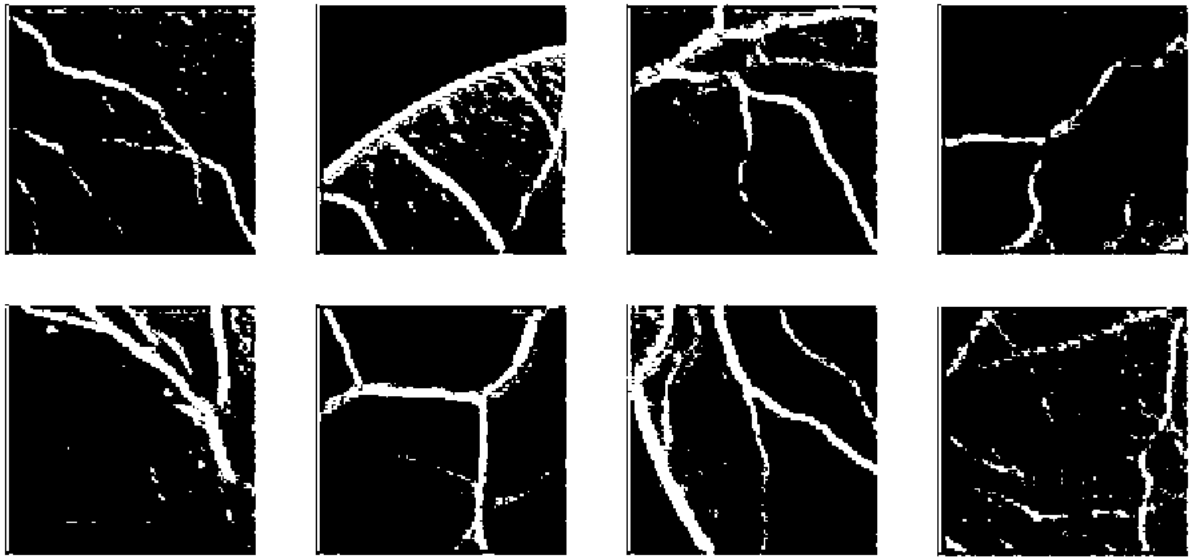
prediction





ground truth

thresholded prediction



Future Development

Augmented MNIST dataset

- refine ground truth
  - thinner segmentation masks
  - smaller detection masks
- train with partial ground truth

Fundus dataset

- test with full-sized fundus input
- train with RGB colour input

Expanded fundus dataset

- create detection masks from coordinates
- train with detection masks
- train branched network for segmentation and detection
- train with detection coordinates
- train branched network for all three outputs

Thank You

To

Dr Anil Bharath,

Antonia Creswell,

Cher Bachar,

and the audience.