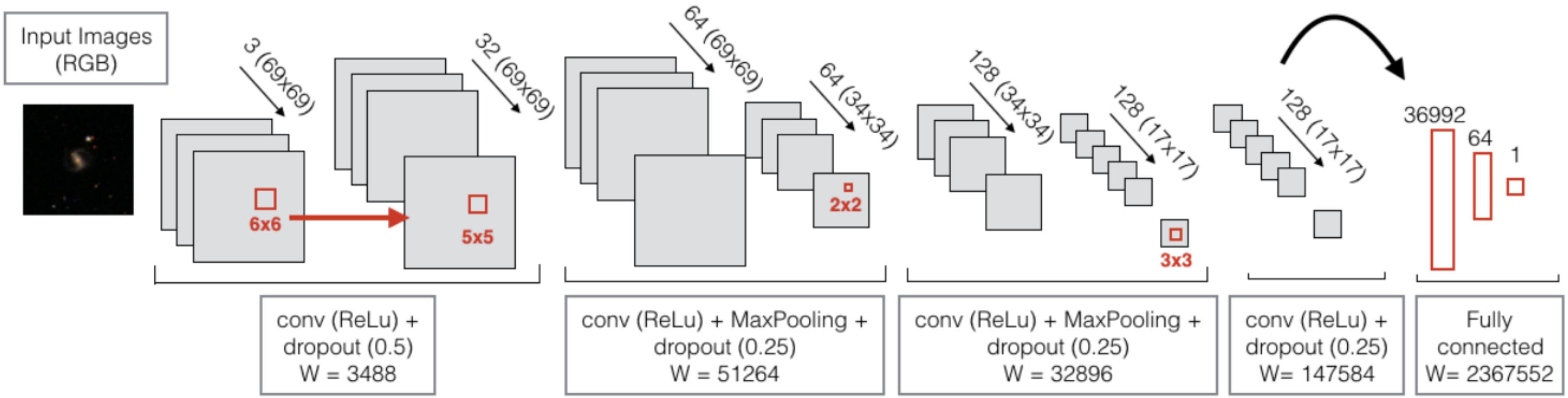
PART IV: BEYOND CLASSIFICATION: IMAGE2IMAGE NETWORKS

UP TO NOW CNNs MAP IMAGES (SIGNALS) INTO FLOATS



Dominguez-Sanchez+18

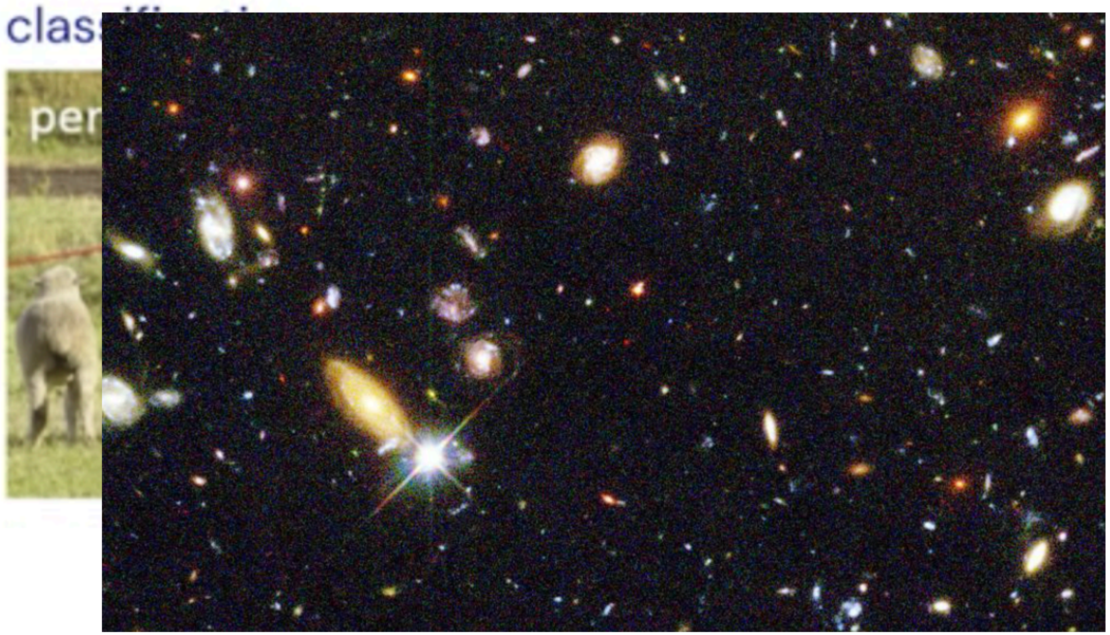
Classification has its limits ....

HOW DO I CLASSIFY THIS IMAGE?

Classification has its limits ....

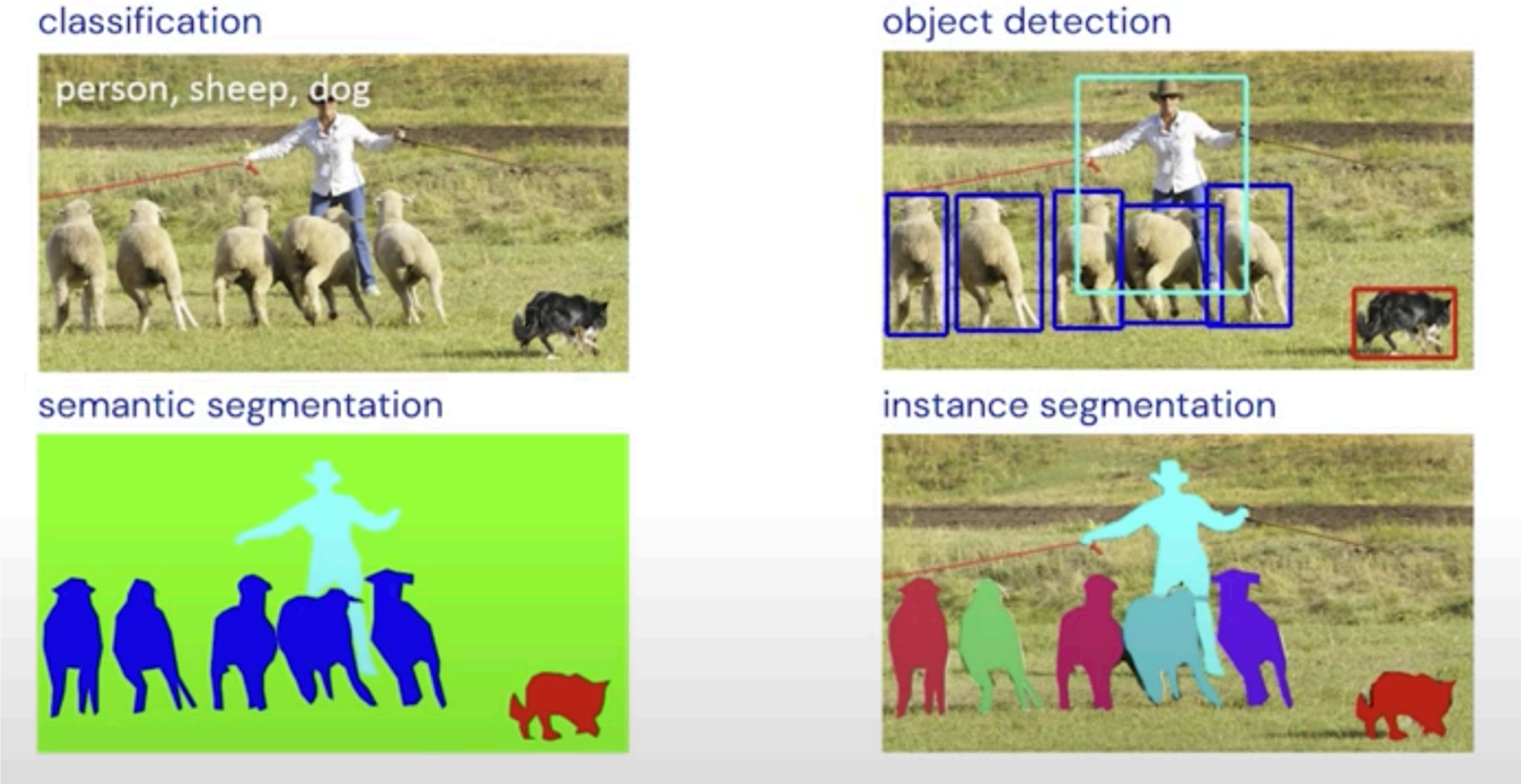
HOW DO I CLASSIFY THIS IMAGE?

Classification has its limits ….



HOW DO I CLASSIFY THIS IMAGE?

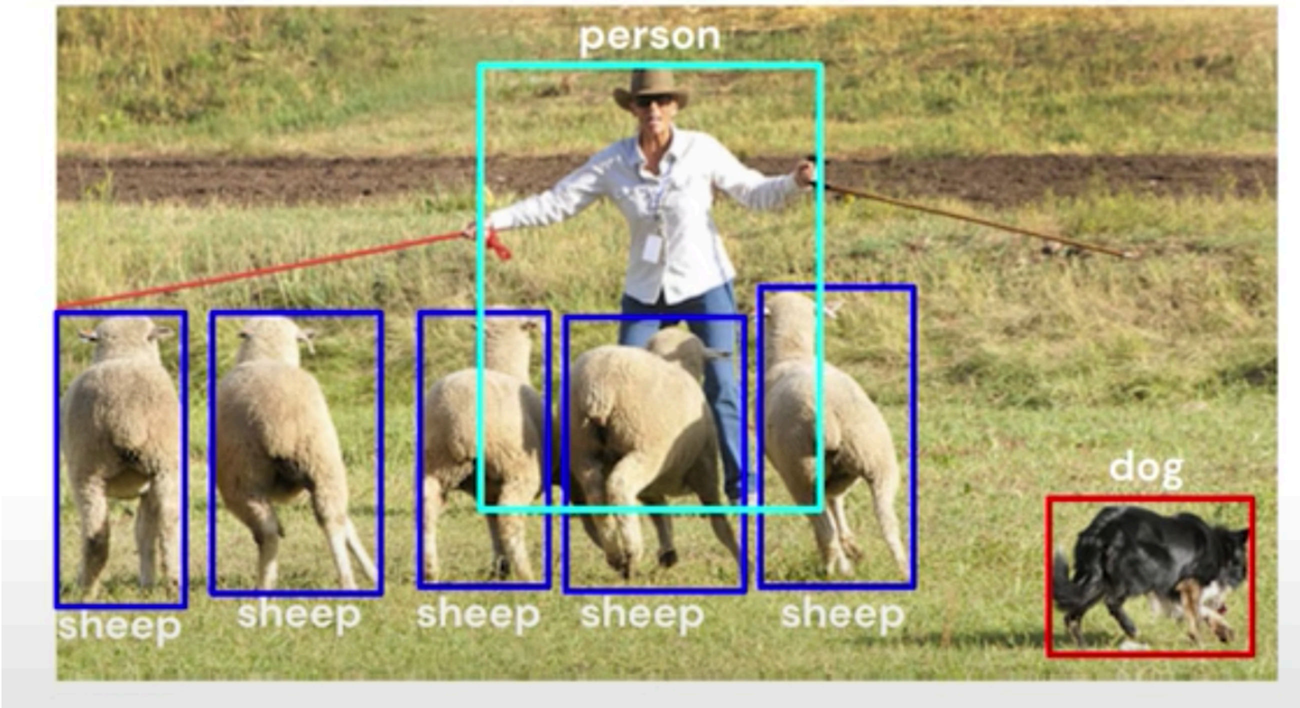
Going beyond classificaiton: increasing complexity



@deepmind (V. Patraucean)

Object detection

First task is to find a bounding box for every object. How we do that?



@deepmind (V. Patraucean)

Inputs

- RGB image $H \times W \times 3$

Targets

- Class label *one_hot* 0 0 0 1 0...
- Object bounding box

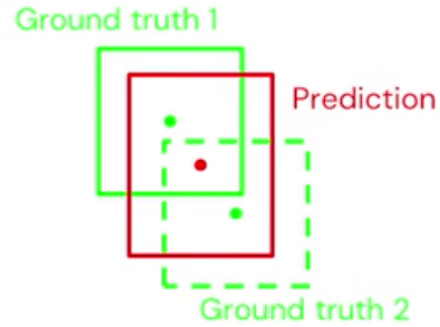
$$(x_c, y_c, h, w)$$

for all the objects present in the scene

@deepmind (V. Patraucean)

WHAT WOULD BE THE LOSS FUNCTION OF SUCH A PROBLEM?

$$\frac{1}{N}\sum_{i=1}^{N}(y_i - p_i)^2$$

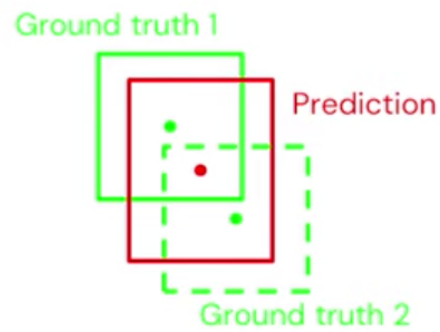


IT IS A REGRESSION WITH A SIMPLE QUADRATIC LOSS. WE TRY TO FIND THE BEST COORDINATES OF THE BOUNDING BOX.

@deepmind (V. Patraucean)

WHAT WOULD BE THE LOSS FUNCTION OF SUCH A PROBLEM?

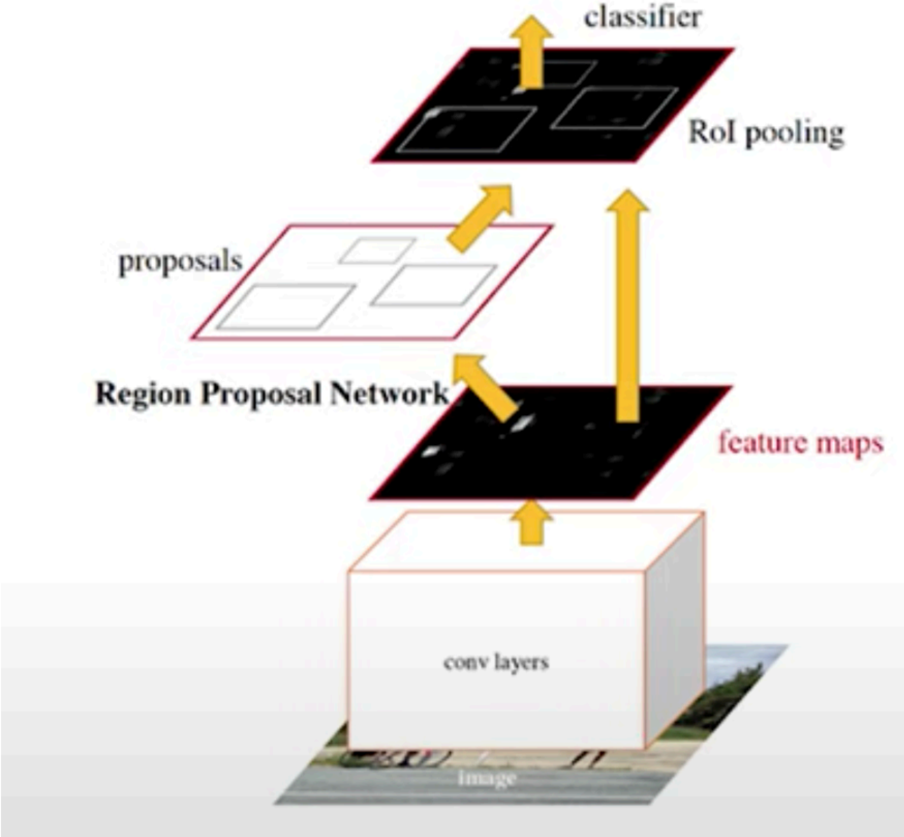
$$\frac{1}{N}\sum_{i=1}^{N}(y_i - p_i)^2$$



IT IS A REGRESSION WITH A SIMPLE QUADRATIC LOSS. WE TRY TO FIND THE BEST COORDINATES OF THE BOUNDING BOX.

IT BECOMES MESSY QUITE RAPIDLY…

EXAMPLE: FASTER R-CNN

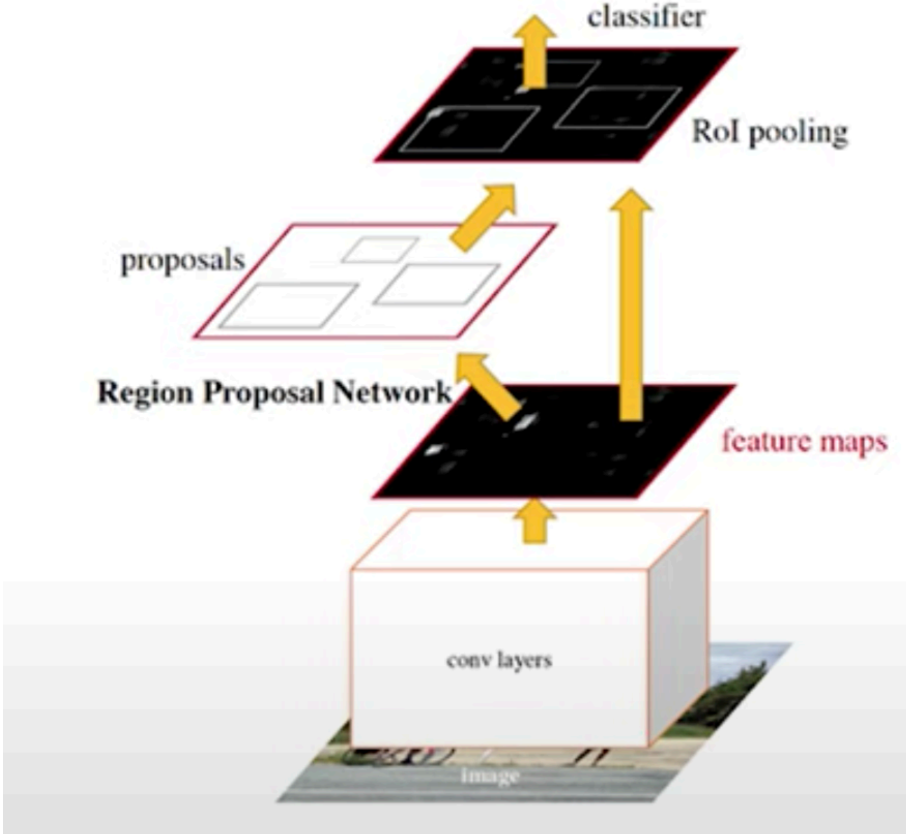


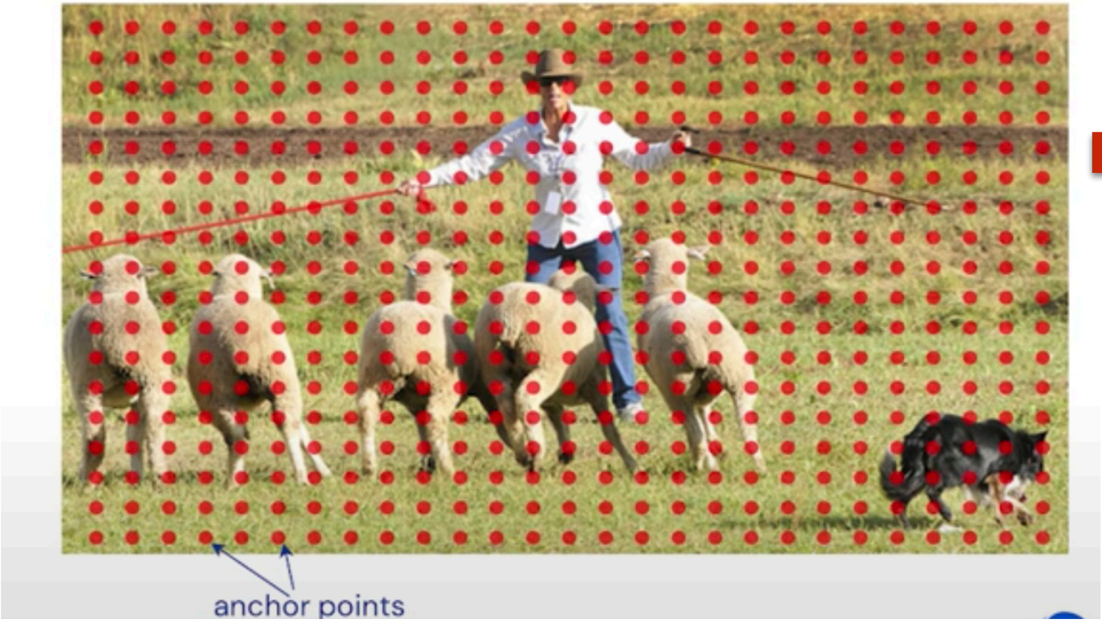
We divide the task in 2 steps:

1. Identify bounding box candidates (CLASSIFICATION)

2. Classify and Refine (REGRESSION)

Ren+16


classifier
RoI pooling
proposals
Region Proposal Network
feature maps
conv layers
image



Discretize Bounding Box Space

Choose n candidates per position

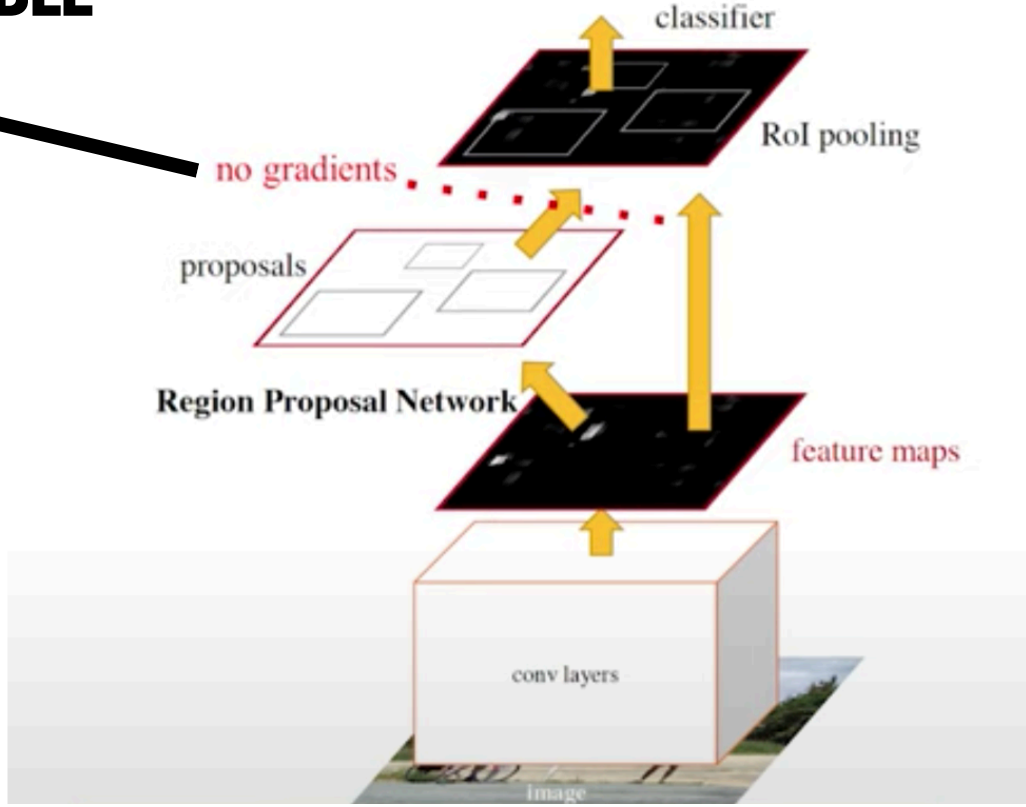
Predict objectness score (classifcation)

Sort and keep top K

@deepmind (V. Patraucean)

THIS IS NOT DIFFERENTIABLE

(FIXED)



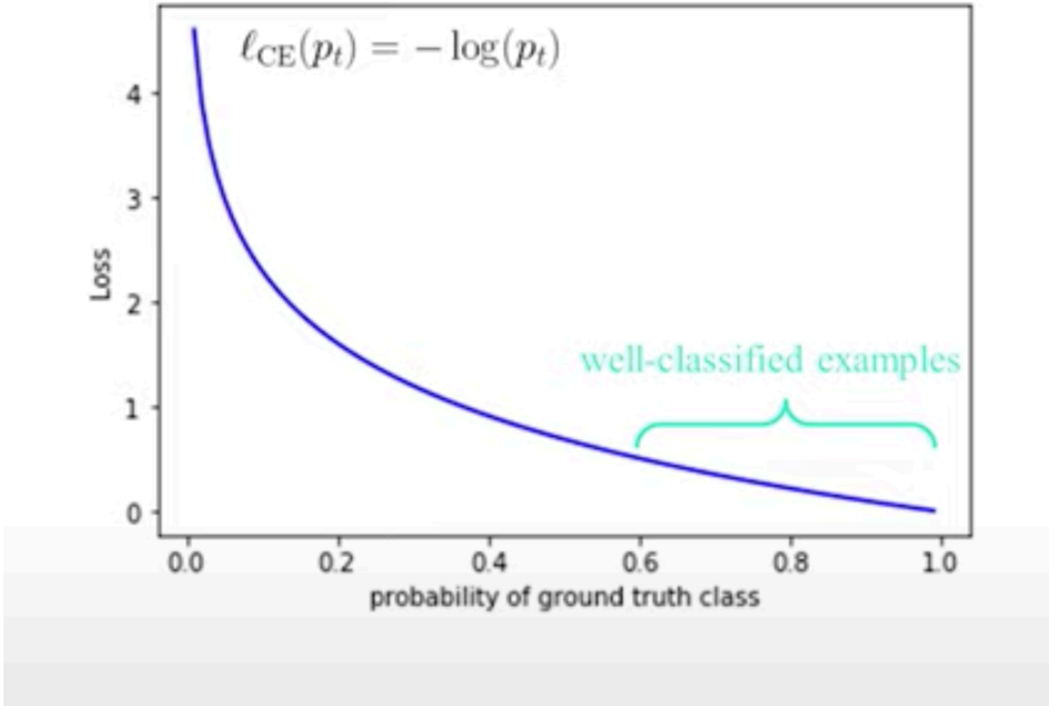
WAYS OF MAKING THIS DIFFERENTIABLE:

Spatial Transformer Networks (Jaderberg+18)

WHY NOT DOING IT ONE STAGE?

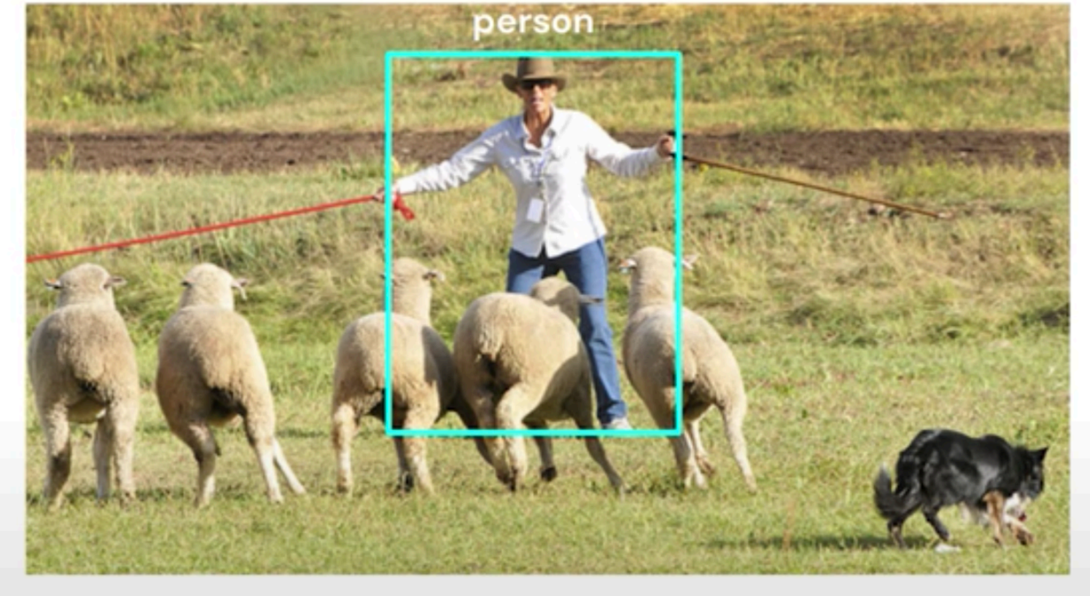
MOST OF THE CANDIDATES ARE BACKGROUND, EASY TO IDENTIFY

THE LOSS OF THE MANY EASY EXAMPLES DOMINATES OVER THE RARE USEFUL ONES



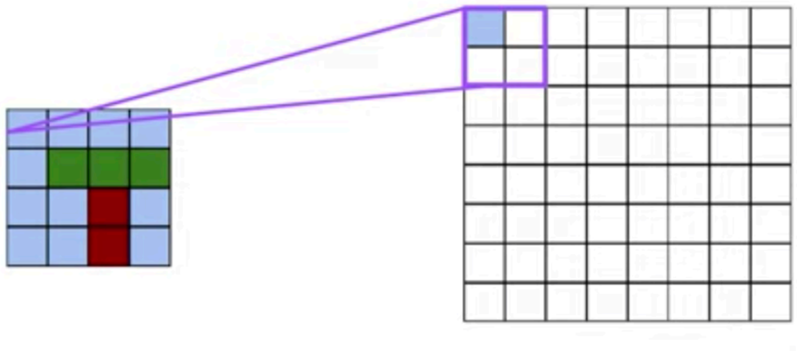
@deepmind (V. Patraucean)

LET'S GO A STEP FURTHER INTO SEMANTIC SEGMENTATION

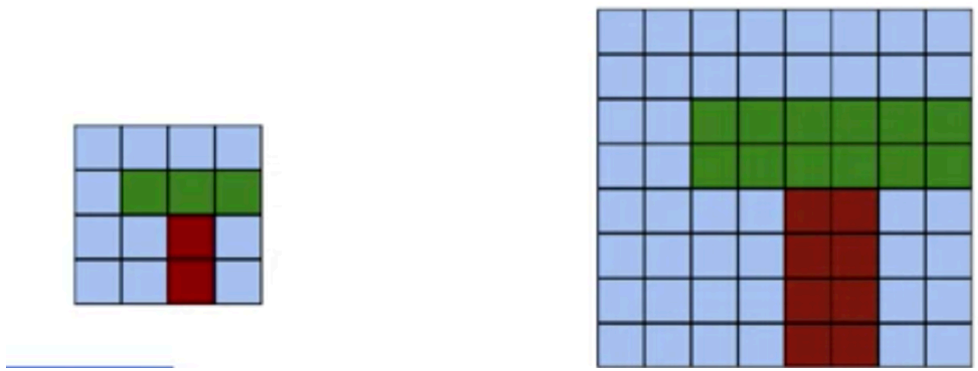


BOUNDING BOXES ARE NOT ALWAYS GOOD REPRESENTATIONS

UNPOOLING OPERATION (INVERSE OF POOLING)



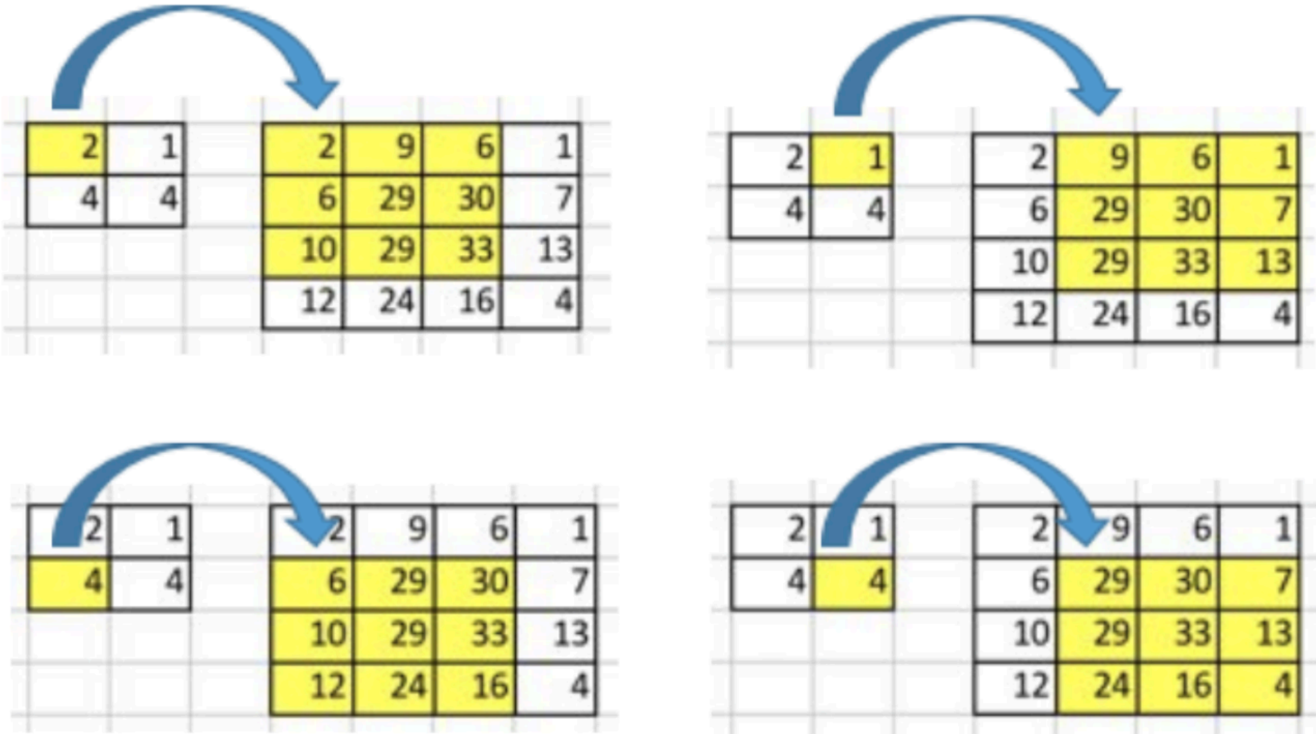
COPY PIXELS IN A GIVEN WINDOW



GENERATES LARGER IMAGES FROM SMALLER ONES

TRANSPOSED CONVOLUTION

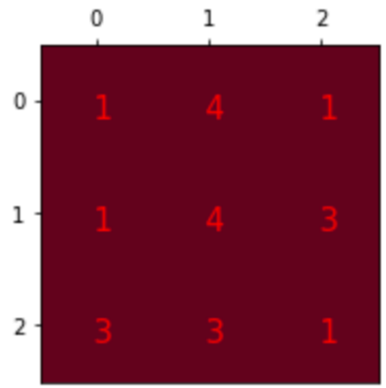
ALLOWS TO INCREASE THE SIZE



Going Backward of Convolution

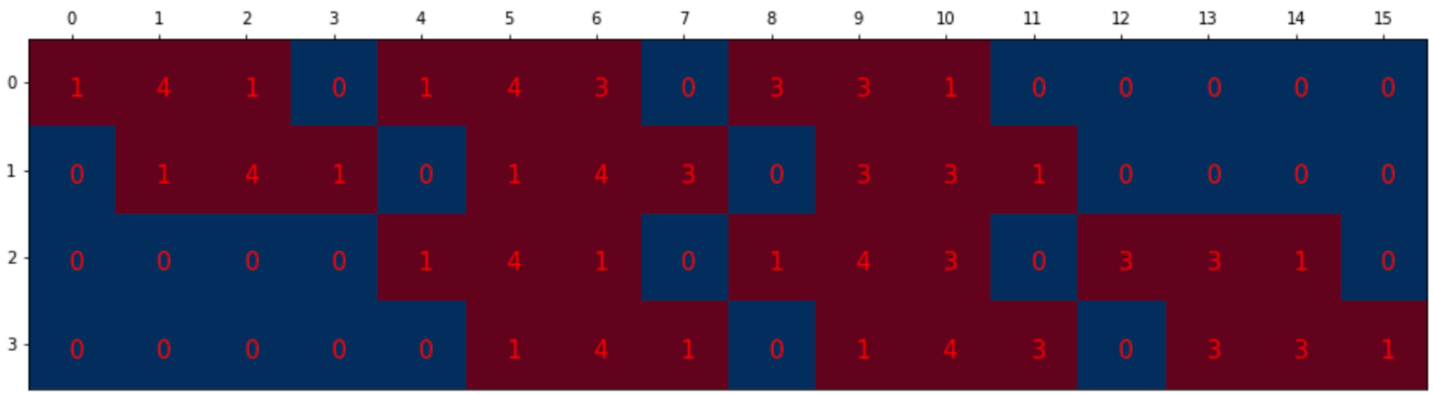
EXAMPLE TAKEN FROM HERE

CONVOLUTION MATRIX



Kernel (3, 3)

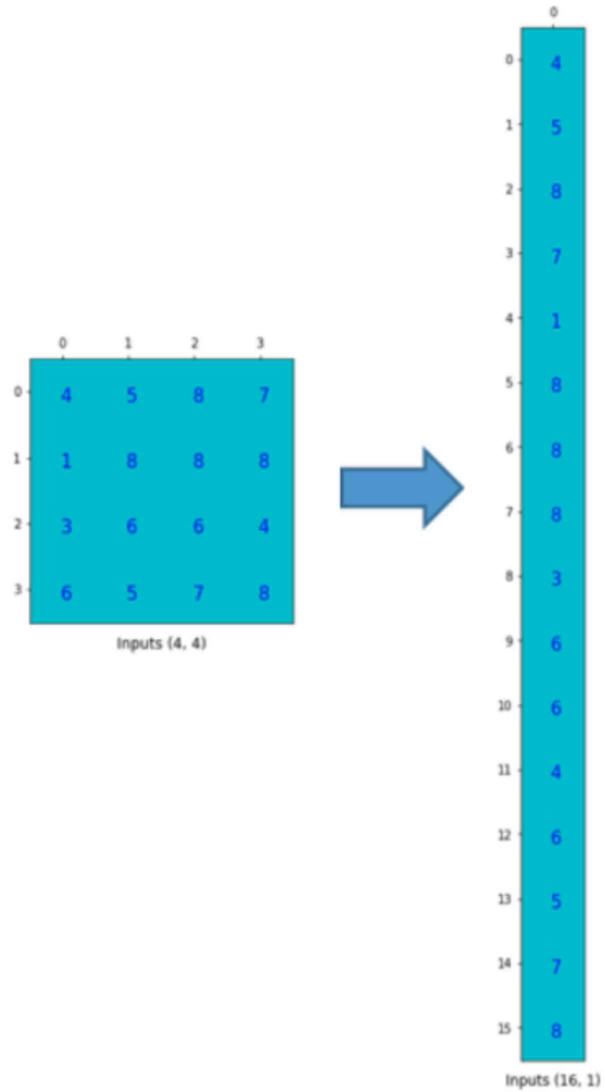
THE KERNEL CAN BE ARRANGED IN FORM OF A MATRIX:



Convolution Matrix (4, 16)

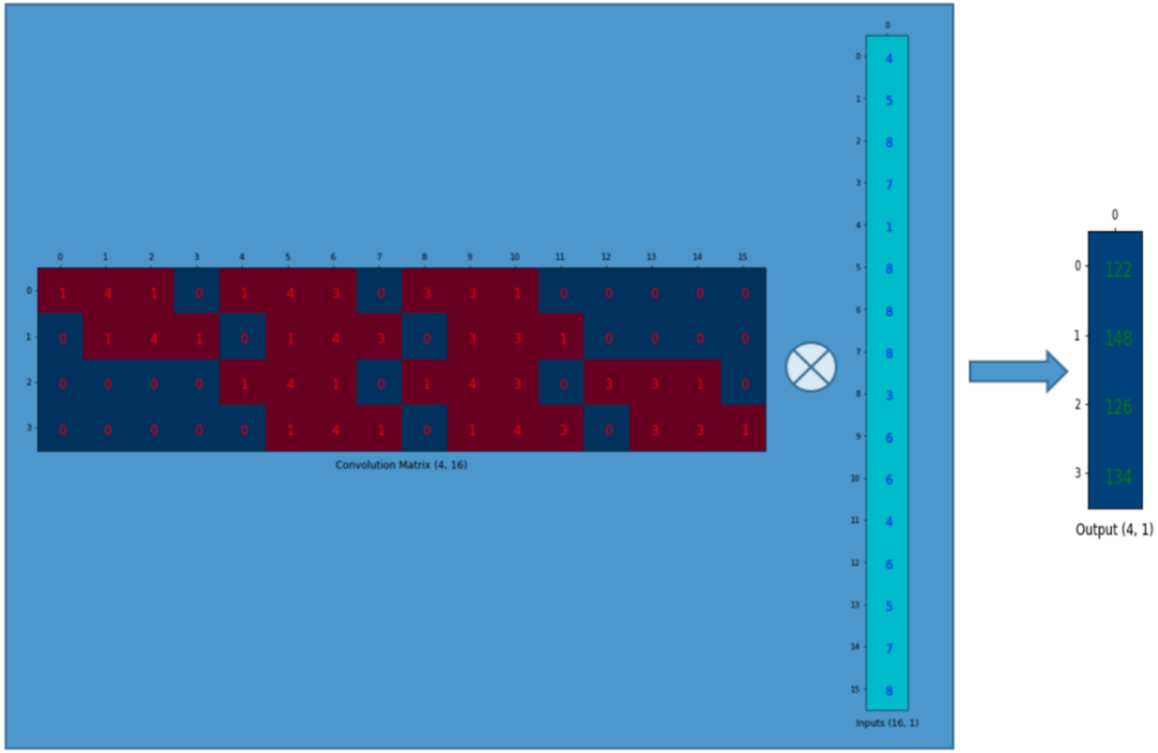
<u>EXAMPLE TAKEN FROM HERE</u>

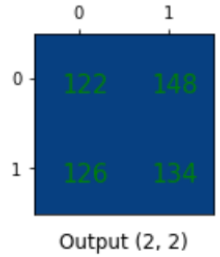
THE INPUT IS FLATTENED INTO A COLUMN VECTOR



EXAMPLE TAKEN FROM HERE

THE CONVOLUTION IS TRANSFORMED INTO A PRODUCT OF MATRICES

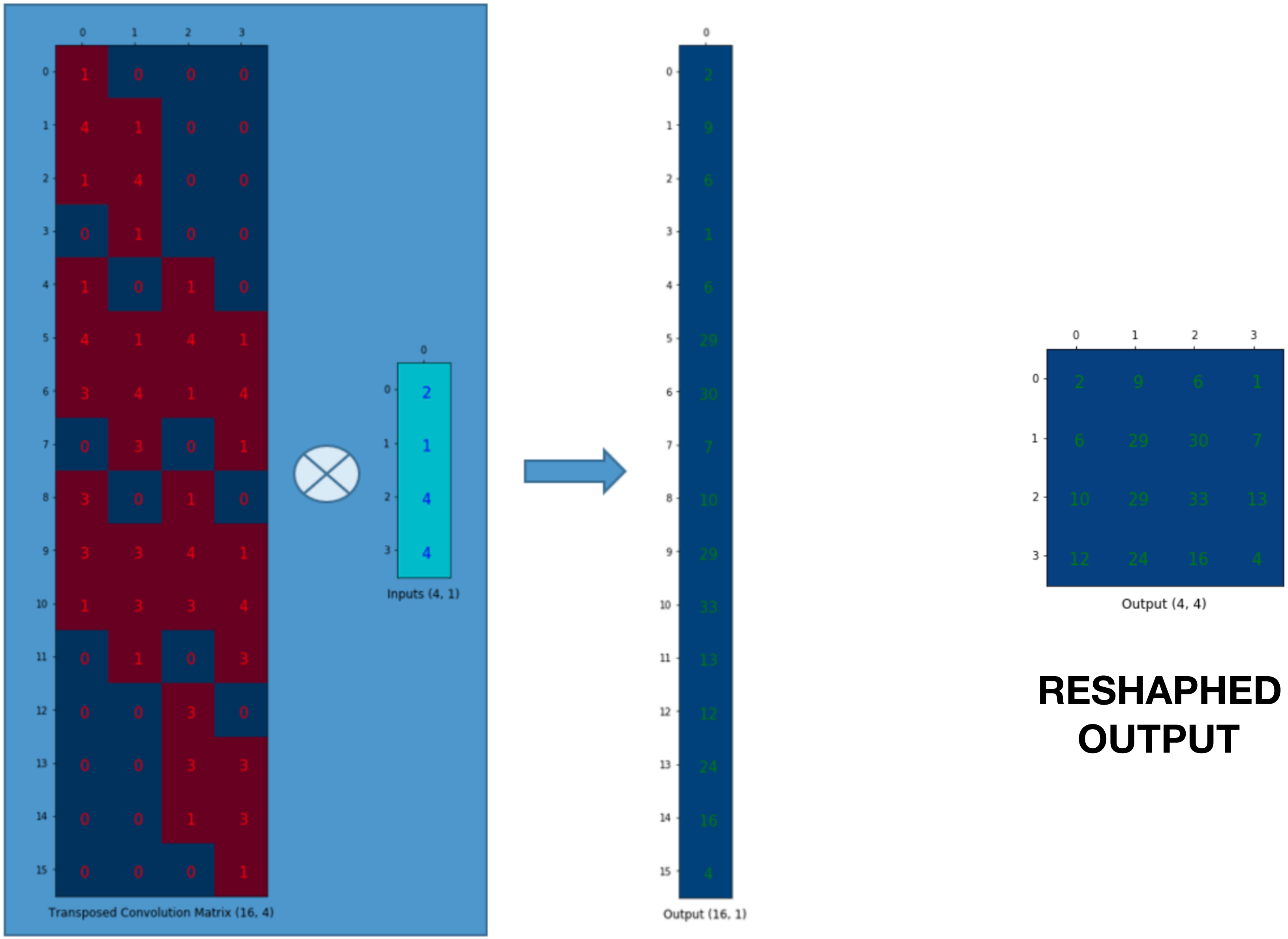




RESHAPED OUTPUT

EXAMPLE TAKEN FROM HERE

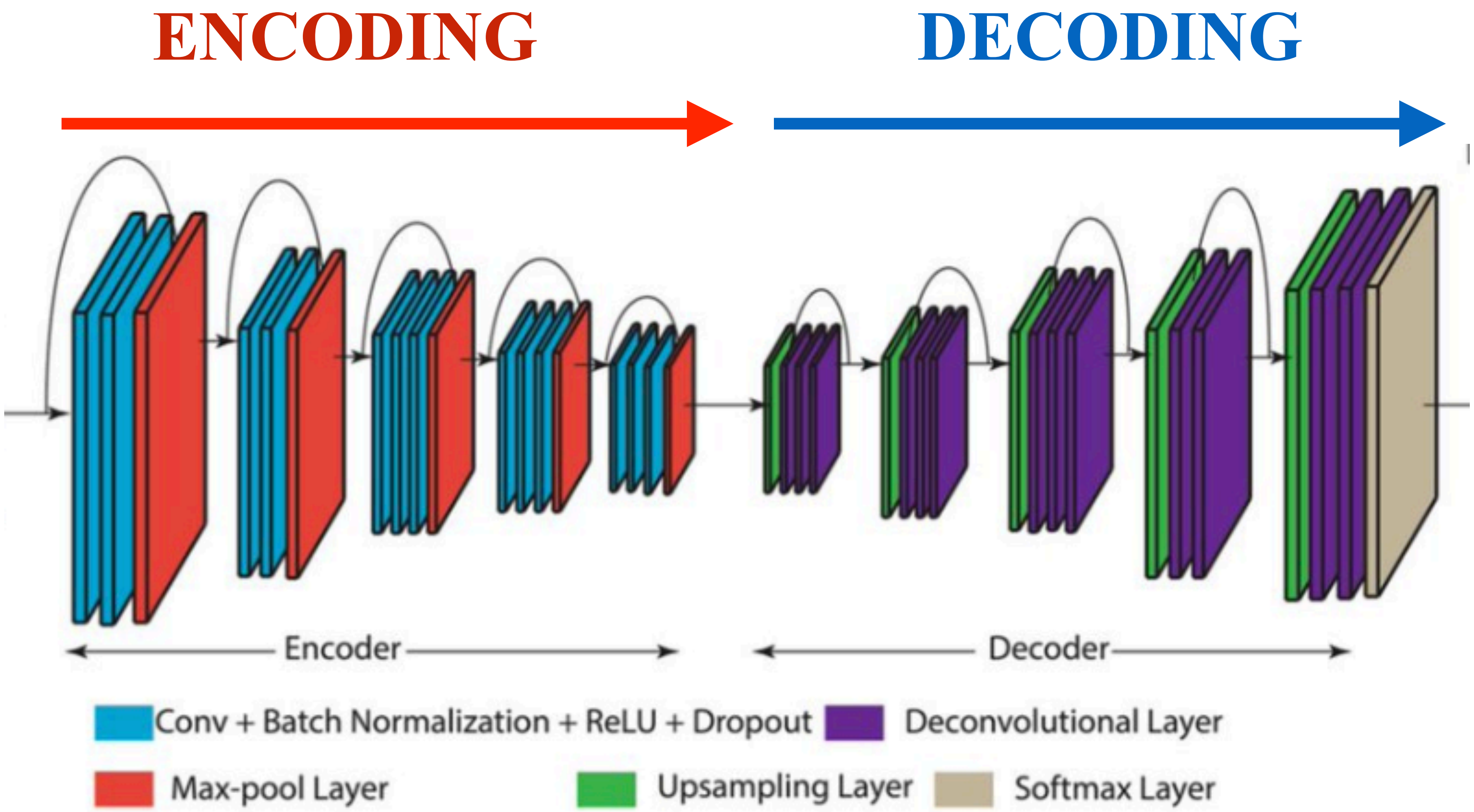
THE TRANSPOSED CONVOLUTION IS THE INVERSE OPERATION



RESHAPED OUTPUT

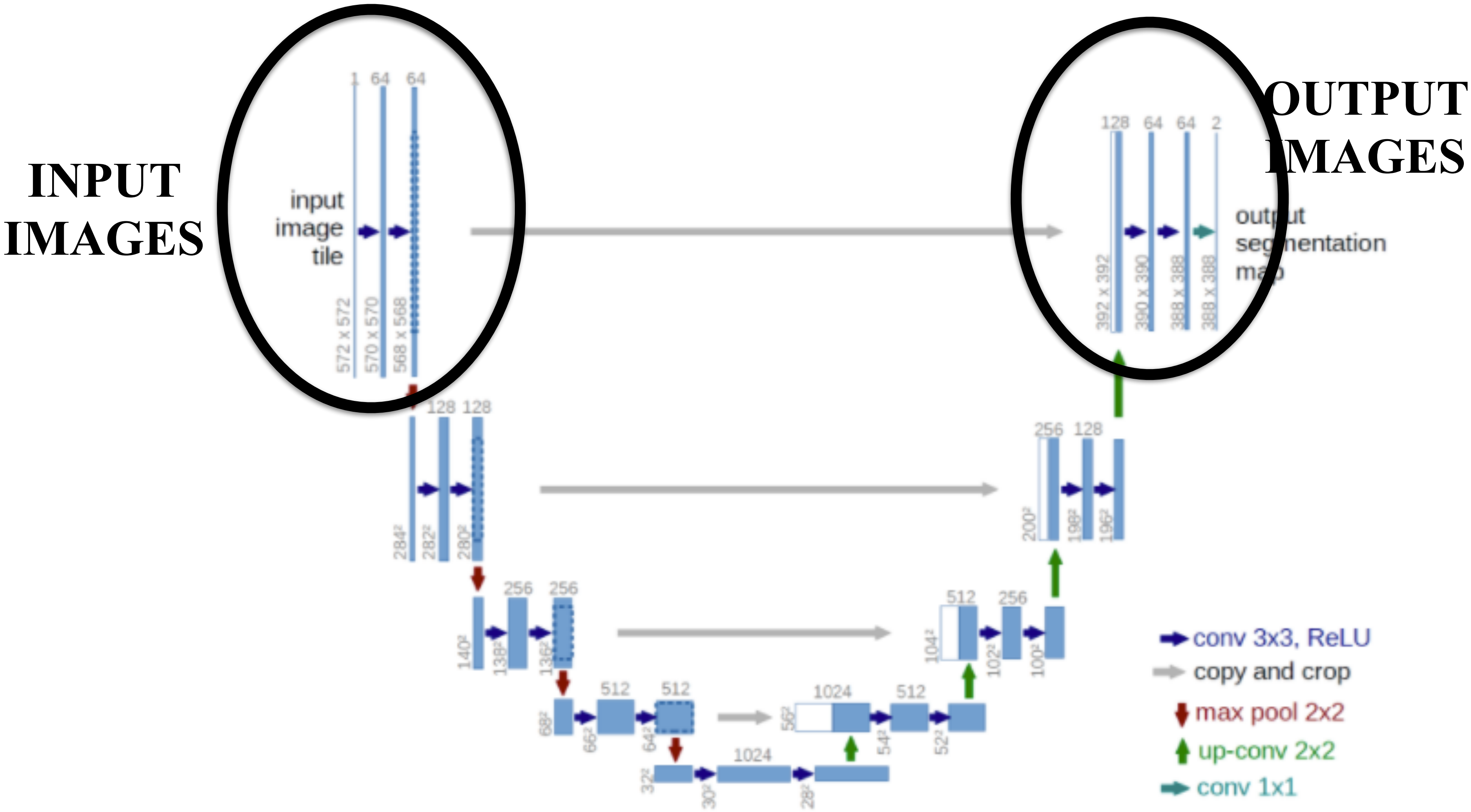
EXAMPLE TAKEN FROM HERE

ENCODER-DECODERS GO FROM IMAGE 2 IMAGE



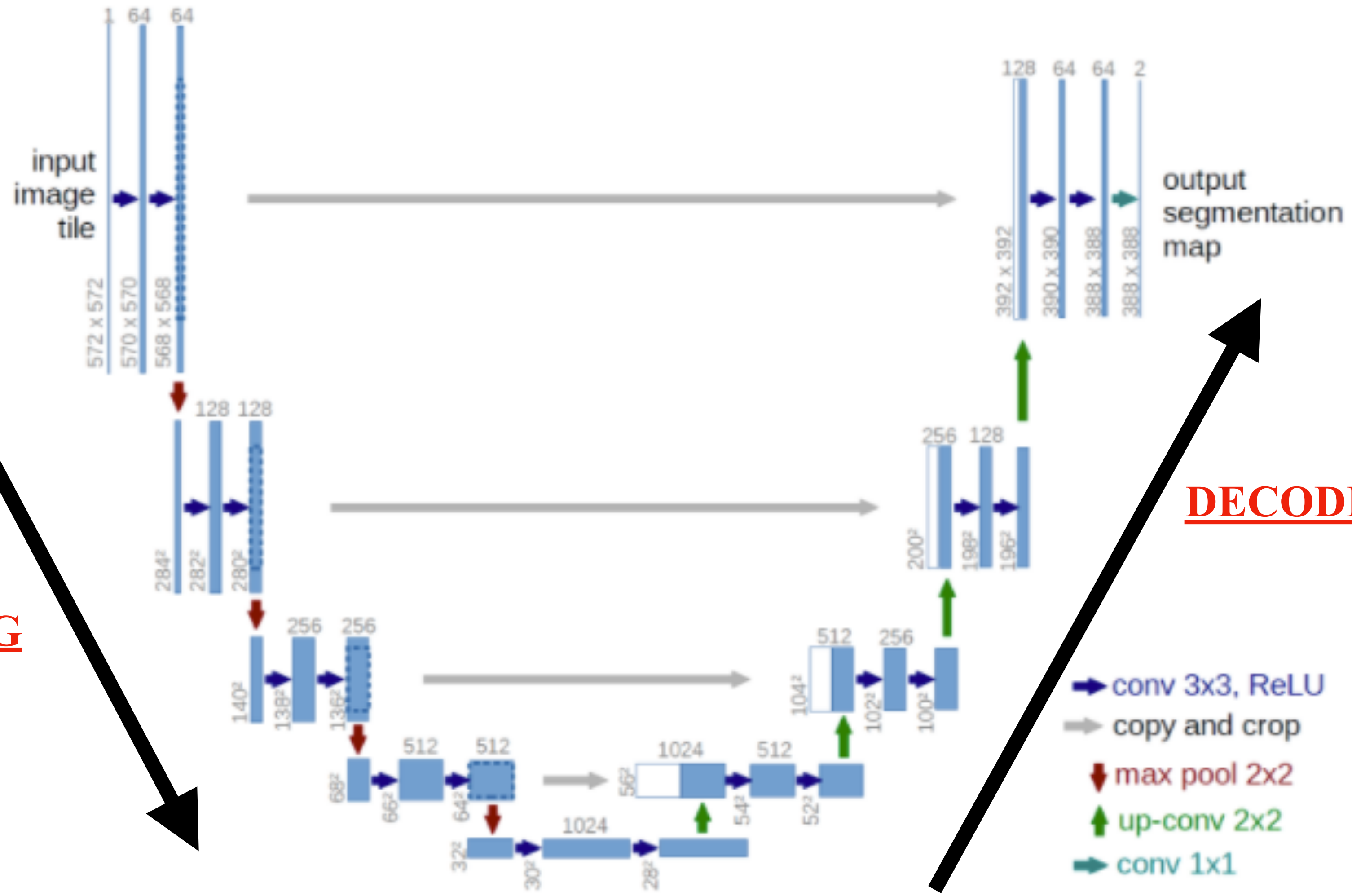
WE CALL THIS FULLY CONVOLUTIONAL NEURAL NETWORKS

ENCODING-DECODING TO EXTRACT IMAGE FEATURES: U-NET



Ronnerberger+15

ENCODING-DECODING TO EXTRACT IMAGE FEATURES: THE U-NET

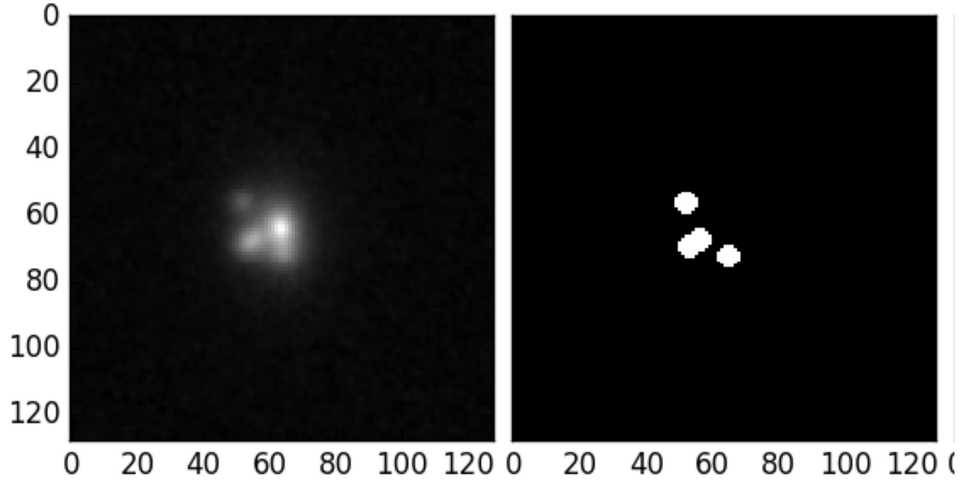


ENCODING

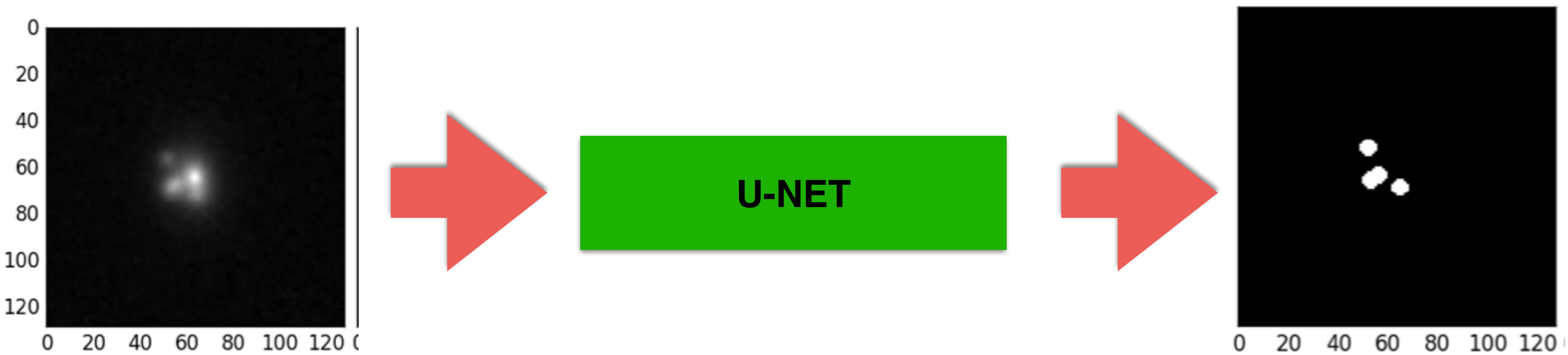
DECODING

Ronnerberger+15

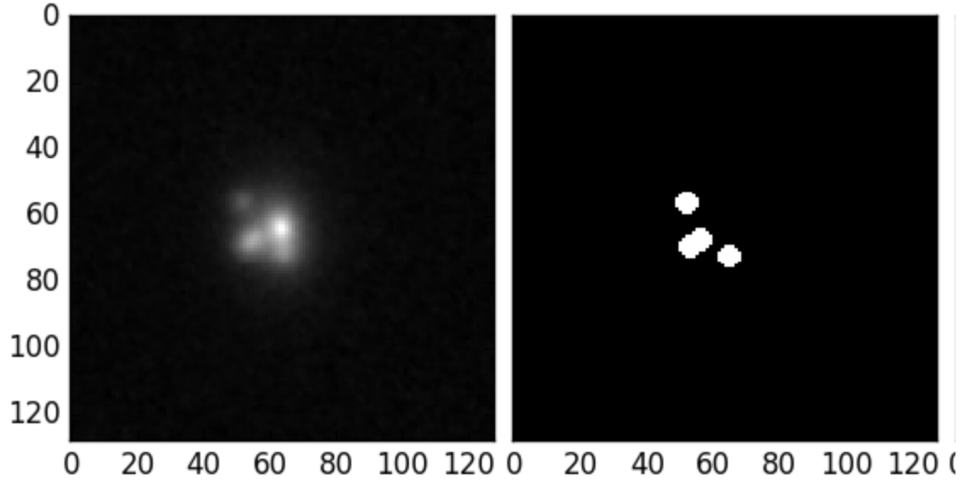
VERY SIMPLE SERSIC ANALYTIC SIMULATIONS + UNRESOLVED CLUMPS



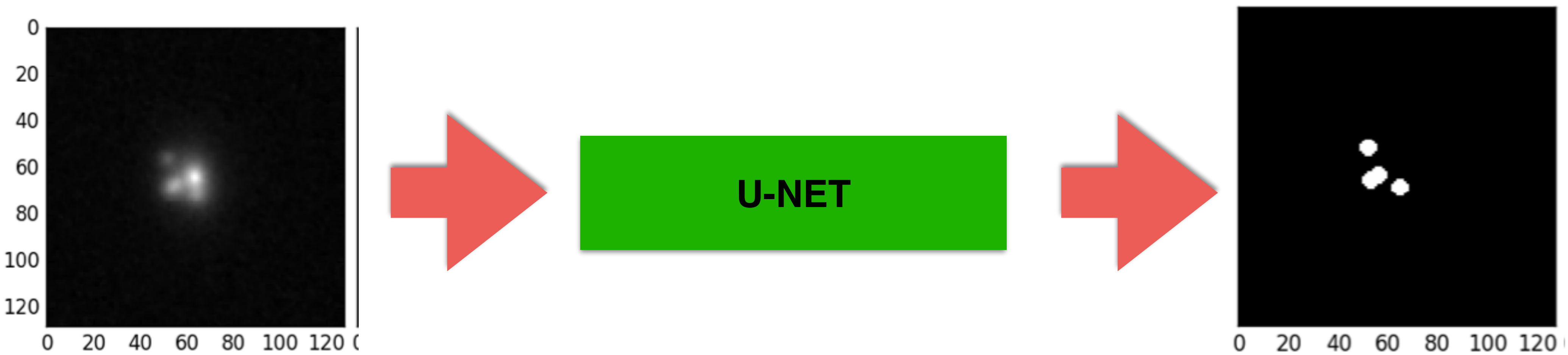
CLUMP POSITION IS KNOWN



VERY SIMPLE SERSIC ANALYTIC SIMULATIONS + UNRESOLVED CLUMPS



CLUMP POSITION IS KNOWN



WHAT LOSS FUNCTION?