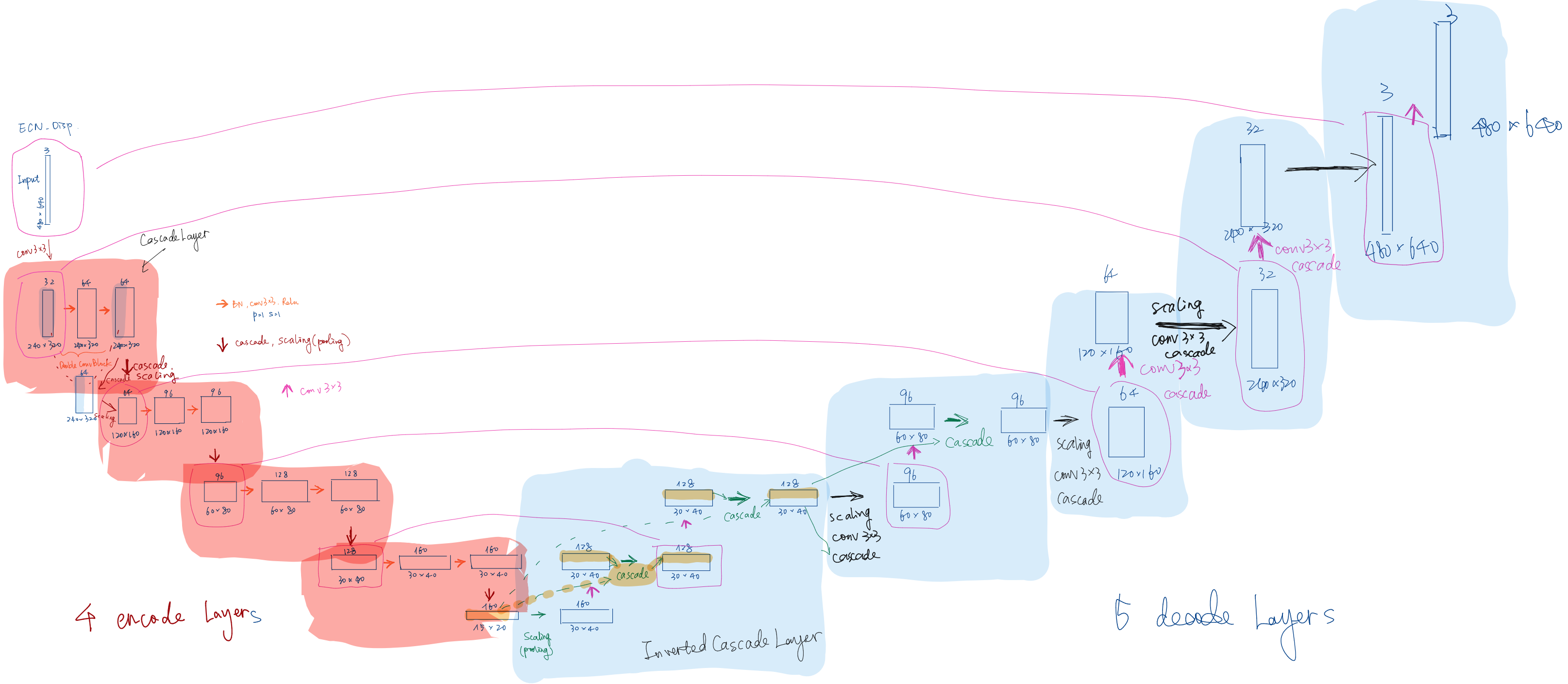
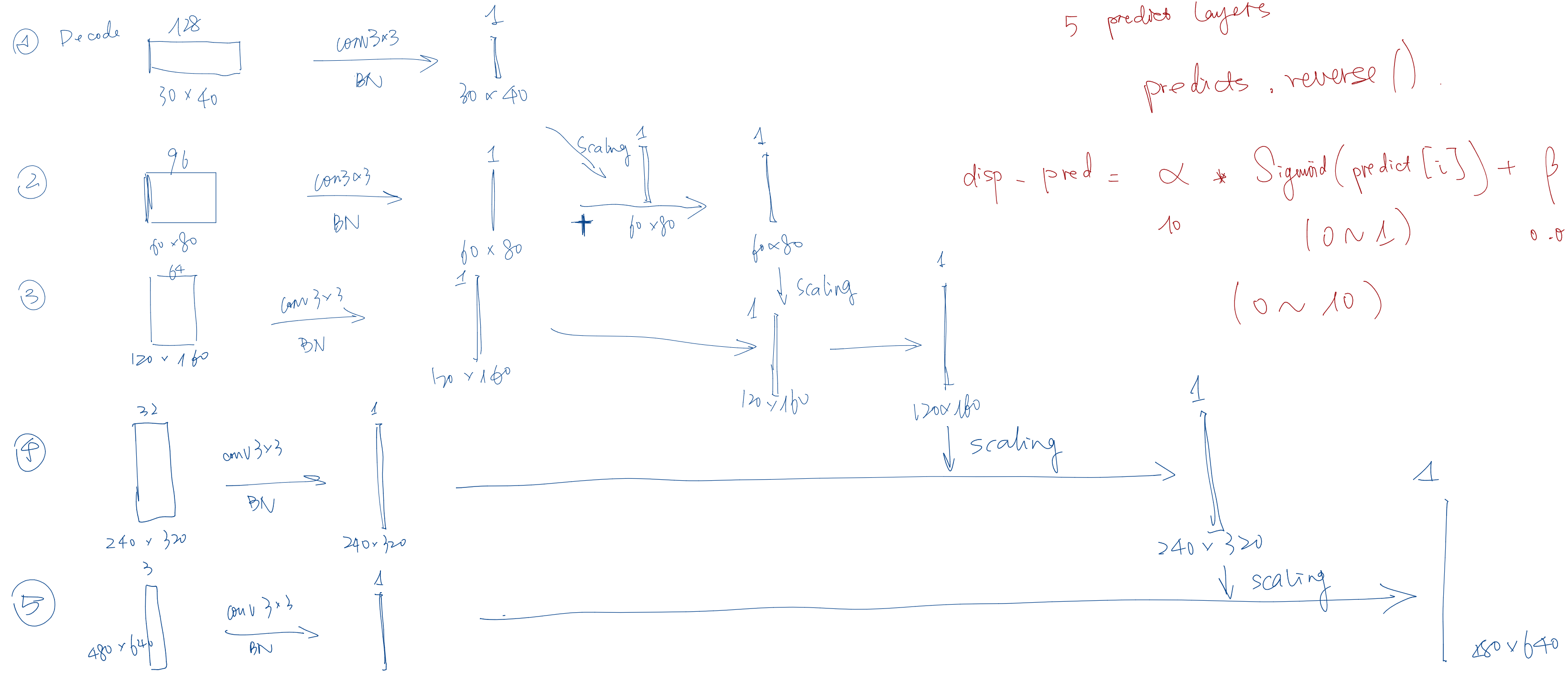

ECN - Disp .
3
Input
480 × 640
conv3x3
Cascade Layer
32
64
64
240 × 320
240×320
240×320
Double Conv Block
64
240×320
cascade
cascade
scaling
scaling
64
96
96
120×160
120×160
120×160
BN , conv3x3 . Relu
p=1 s=1
cascade , scaling(pooling)
Con v 3×3
96
128
128
60×80
60× 80
60× 80
128
160
160
30 × 40
30×40
30×40
160
15 × 20
Scaling
(pooling)
160
30×40
128
30 × 40
cascade
128
30 × 40
128
30 × 40
cascade
128
30 × 40
Inverted Cascade Layer
scaling
conv 3x3
cascade
96
60 × 80
96
60 × 80
Cascade
96
60 × 80
scaling
conv 3×3
cascade
64
120 × 160
64
120 × 160
conv3x3
cascade
scaling
conv 3× 3
cascade
32
240 × 320
32
240 × 320
conv3×3
cascade
3
480 × 640
3
480 × 640
4 encode Layers
5 decode Layers

Channel : [3 , 32 , 64 , 96 . 128 . 160]

Predict_map :



5 predict layers

predicts . reverse ( ) .

$$disp\_pred = \underset{10}{\alpha} * \underset{(0\sim1)}{Sigmoid(predict[i])} + \underset{0.01}{\beta}.$$

(0 ~ 10)

len = 5

depth = $\frac{1}{disp}$ ← normalisé disparities ← output

master中
没有调用 loss_6 ?

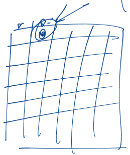
①

arg d (depth - loss_ weight)
Defaut : w6 = 0.5 (w6 = 0 , loss_6 = 0)

(gt , depth)

Loss _ depth (loss_6) = $\frac{\text{sum(abs\_ rel) + sum(thresh error)}}{\text{nb\_ pixel} \times \text{5 layer}}$

pred : [b, 1 , 480 , 640)



$\begin{pmatrix} dx^2 & dxdy \\ dydx & dy^2 \end{pmatrix}$

arg . s (smoth_ loss_ weight)
Defaut w3 = 1.0

②

smooth . loss (depth) (loss_ 3) = $\| \Delta \text{ depth } \|_1$

norm 1

smoothness penalty on

$\underline{\underline{2^{nd} \text{ gradient}}}$ of depth

$\left(\sqrt{dx^2} + \sqrt{dxdy} + \sqrt{dydx} + \sqrt{dy^2} \right)$