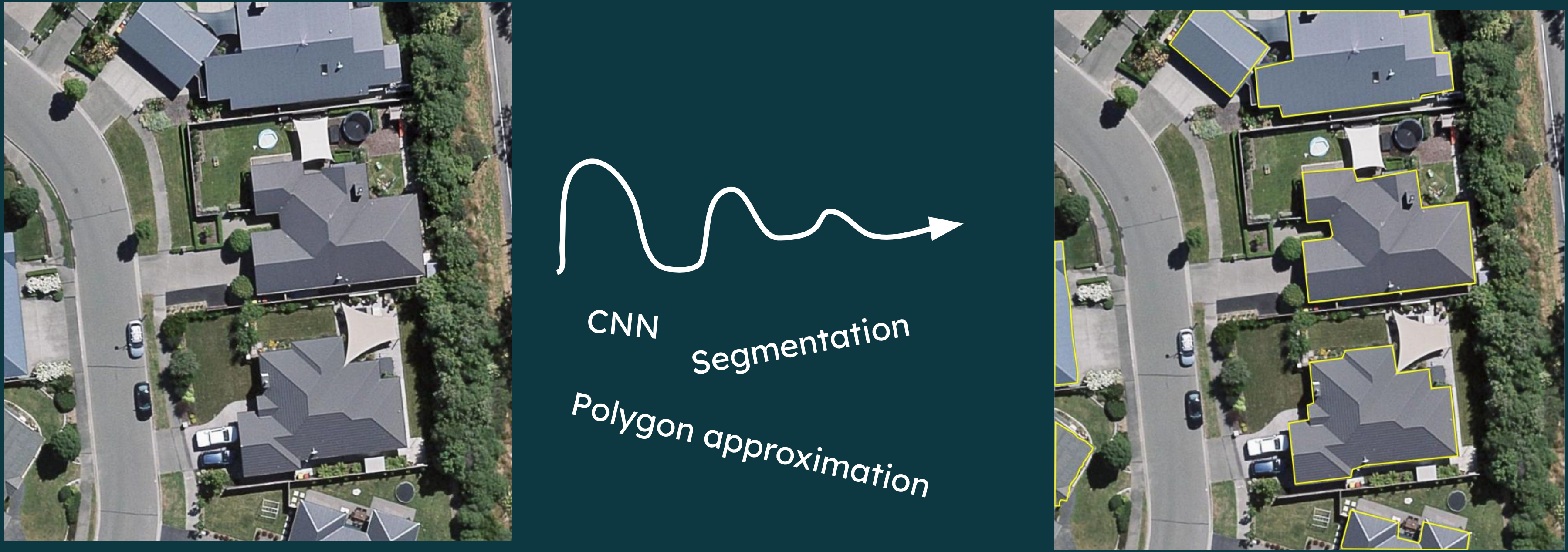
Roof Plan Polygon Extraction from High-resolution Aerial Images

Author: Ondřej Kafka
Supervisor: Marcus Klang

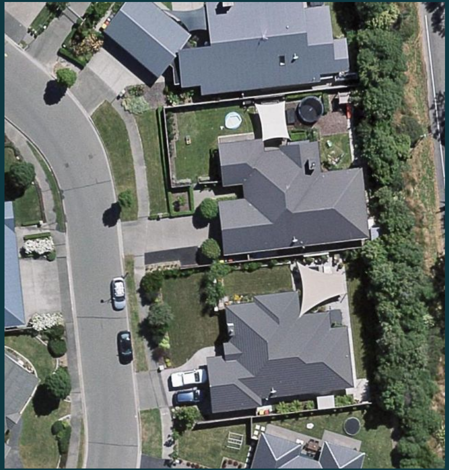
Task description

The objective is to arrive at as precise roof plan polygons as possible.

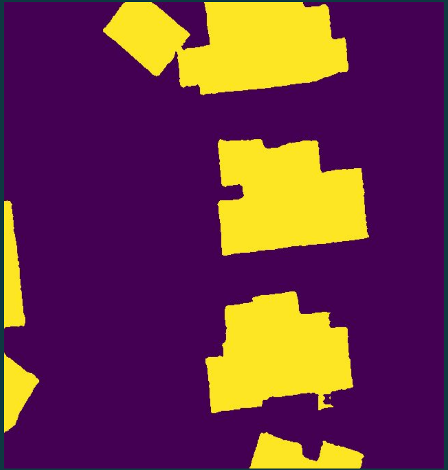


Example pipeline

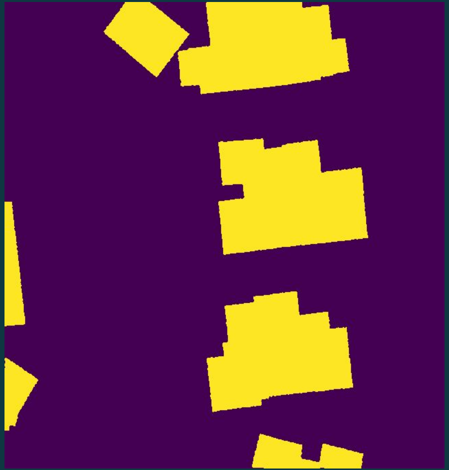
Input image



Segmentation mask



Polygonized mask

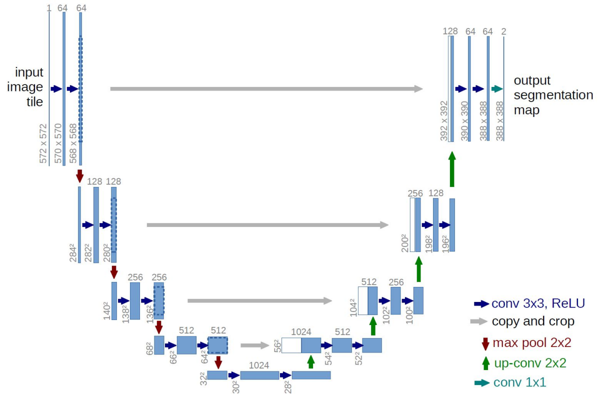


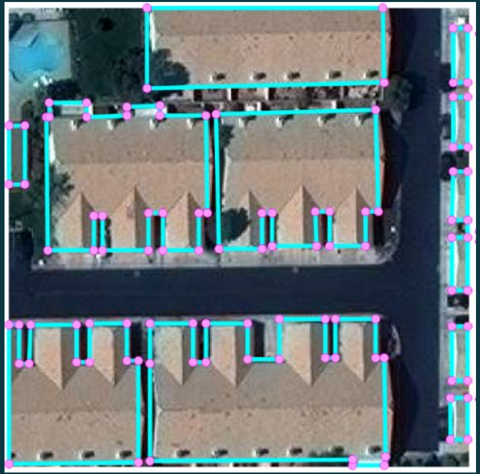
Extracted polygon

Related work

There's already some effort put into this, but no ultimately good solution. The approaches are:

- U-Net-style architectures
- These trained with special losses
- Special models with special losses
- Whatever and regularization
- End-to-end polygon predictors



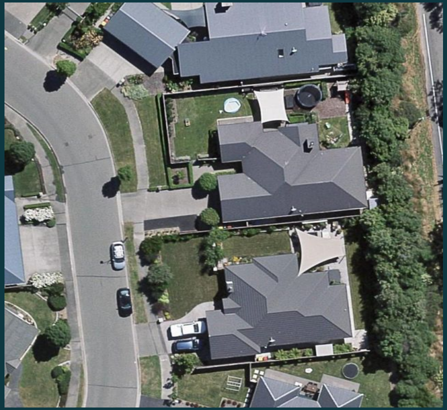


The approach also depends on the scale.

Datasets

Three main datasets:

- Christchurch - New Zealand
- INRIA - More cities
- Our dataset - Czechia



Methods

From the previous work can be concluded the following:

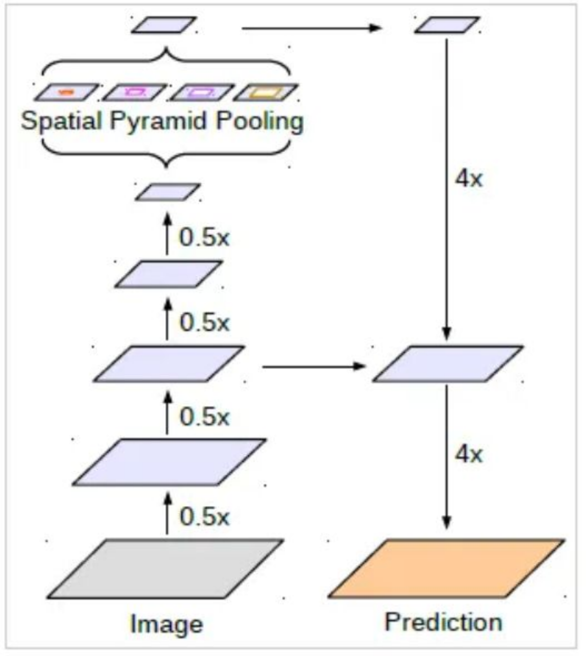
- DeeplabV3+ > U-Net
- BCE Loss not the best for borders

Thus the first approaches are based on DLV3+

trained with some specific losses.

Trained using Adam optimizer, lr scheduler,

fraction of NZ dataset



Proposed losses

Region oriented losses

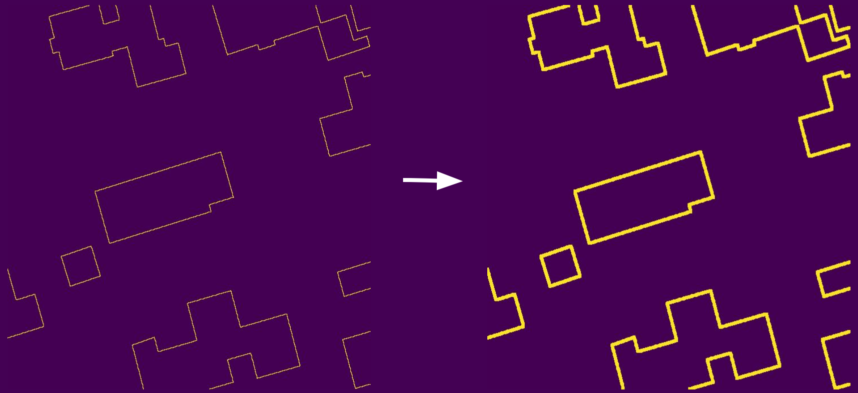
- Binary Cross-entropy loss
- Binary Dice loss
- Focal loss

$$L_{BD}(y, \hat{y}) = 1 - \frac{2\sum_i y_i \hat{y}_i + s}{\sum_i (y_i + \hat{y}_i) + s}$$

$$L_F(y, \hat{y}) = -\sum_i (1 - \hat{y}_i)^{\gamma} \log(\hat{y}_i)$$

Differentiable boundary extraction, extension and weights

- $Y_b = \mathrm{maxPool}(1 - Y) - (1 - Y)$
- $Y_{be} = \mathrm{maxPool}(Y_b, \theta)$
- $w = \mathrm{GaussianBlur}(Y_b) + \alpha$

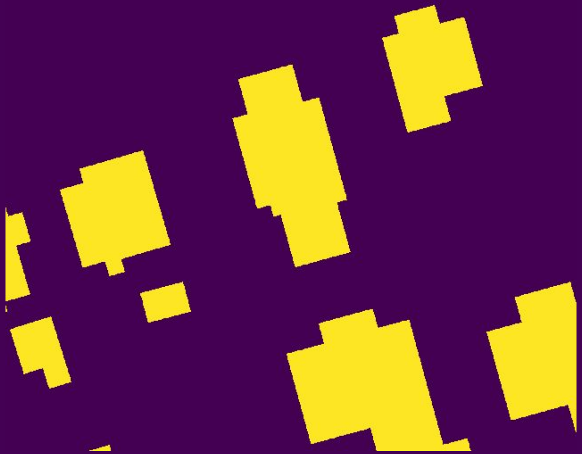


First comparison

metric	iou	dice	dice-b1	dice-b3	dice-b5
bce	**91.00**	**95.23**	27.14	58.10	70.66
bce-enc	90.92	95.18	26.94	57.87	70.47
bdc	90.84	95.14	26.17	57.13	69.98
foc	90.61	95.01	25.97	56.65	69.43
wbce-a08	90.60	95.00	25.85	56.61	69.53
wbce-a05	86.66	92.74	17.85	44.08	58.35
wbce-a02-enc	90.50	94.94	24.72	55.81	69.11
bl-bce-a084	90.92	95.18	27.74	**59.47**	**71.84**
bl-bdc-a089	90.66	95.03	**27.80**	59.38	71.67
bl-bdc-a06	90.91	95.17	27.61	59.23	71.66
bl-bdc-a08-enc	90.94	95.18	27.63	59.31	71.74
bl-bdc-a084	90.94	95.18	27.63	59.31	71.74

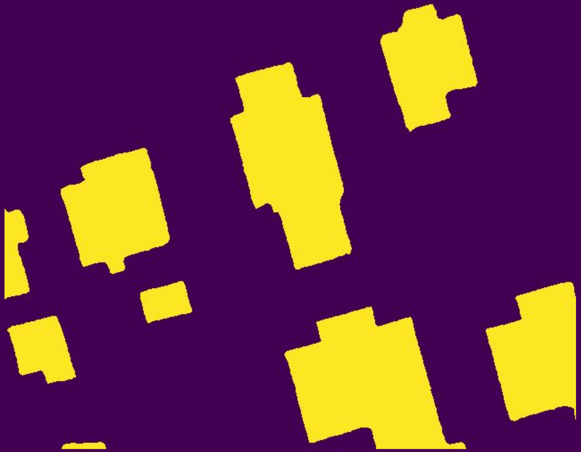
Result examples

Label



BCE loss

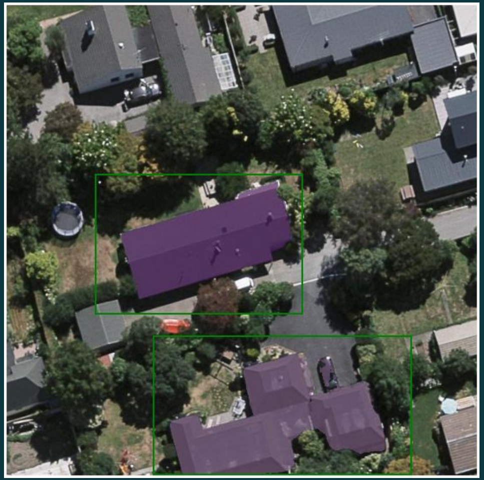
Combo loss

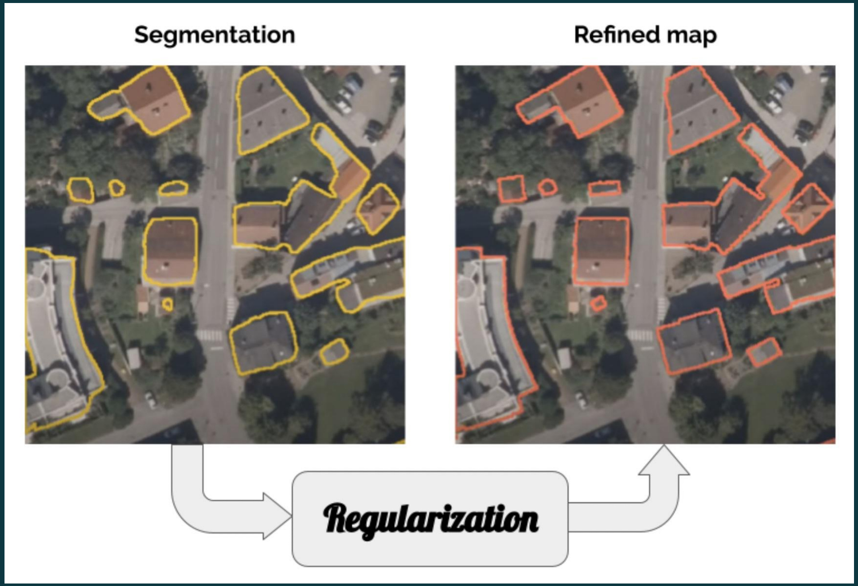


Regularization and polygon extraction

The segmentation mask have still quite far from polygons, so I used:

- Regularization
- SAM
- Polygon extraction





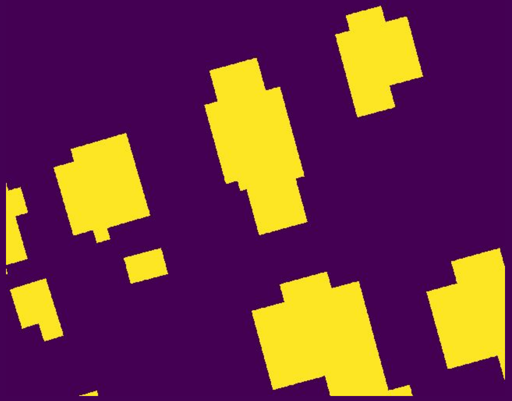
Second comparison

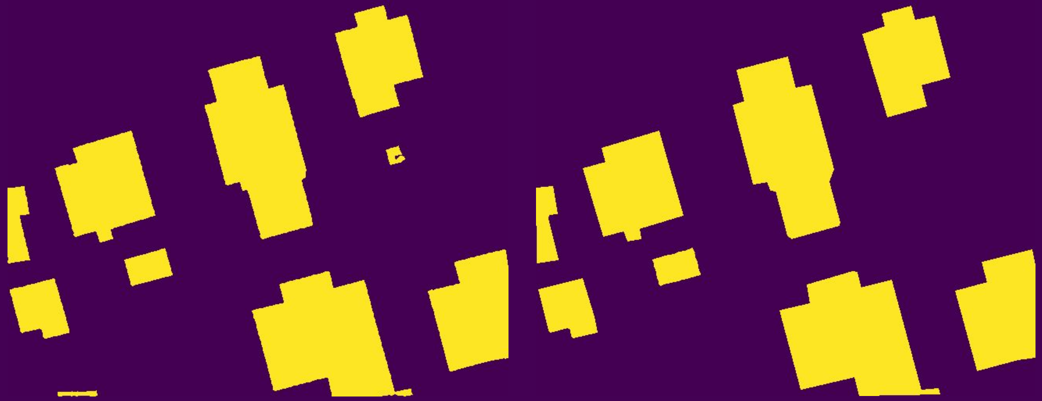
metric	iou	dice	dice-b1	dice-b3	dice-b5
bce-r	**90.32**	**94.86**	22.54	53.35	67.44
bdc-r	90.16	94.77	22.01	52.51	66.83
bl-bce-a084-r	90.27	94.83	**23.13**	**54.36**	**68.36**
bl-bdc-a089-r	90.02	94.68	23.12	54.27	68.20
wbce-a08-rp	89.68	94.50	20.50	50.45	65.21
bce-rp	**90.14**	**94.76**	21.88	52.34	66.63
bdc-rp	89.99	94.68	21.26	51.42	65.97
bl-bce-a084-rp	90.09	94.73	22.28	**53.16**	**67.44**
bl-bd-a089-rp	89.85	94.58	**22.29**	53.06	67.27
bce-rp	**90.29**	**94.83**	24.47	54.89	68.15
bl-bce-a084-rp	90.20	94.77	24.92	**56.17**	**69.36**
bl-bdc-a089-rp	89.89	94.58	**24.94**	56.12	69.22

Result examples

We can see:

- Worse model (regpoly)
- Better model (reg + regpoly + poly)
- Label

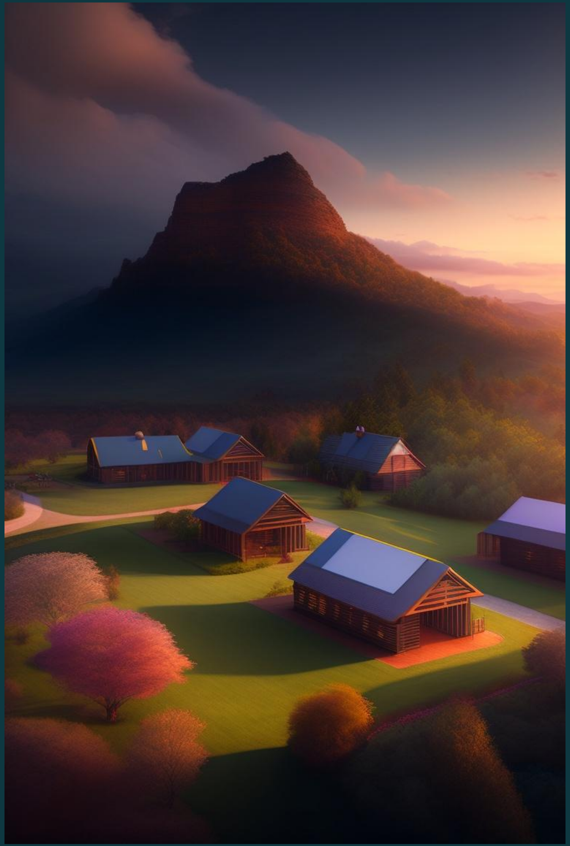




Future work

As there is still a lot to do, this are further steps:

- Add PolyWorld to comparison
- Add vertex-distance metric
- Finetune the regularization model
- Finetune Polyworld model
- Finetune all the models on our data



Conclusion & discussion

Conclusions

- BCE is indeed not the best loss for boundaries
- Regularization helps a lot
- SAM is not the best model

Thank you for your attention!