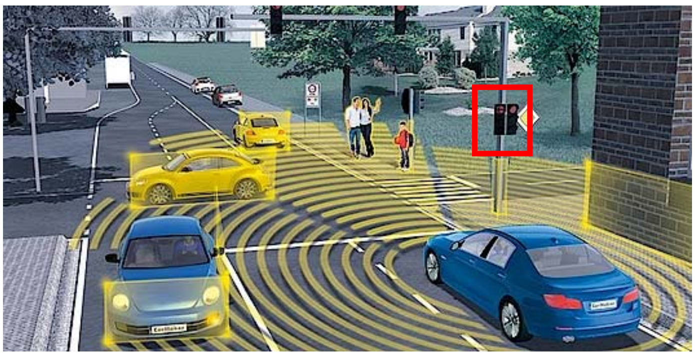
Accurate Traffic Light Detection using Deep Neural Network with Focal Regression Loss

Bak Gyeongmin

POSTECH

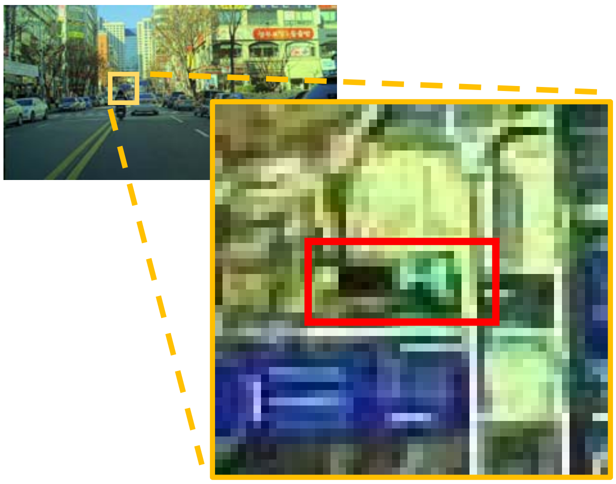
Goal

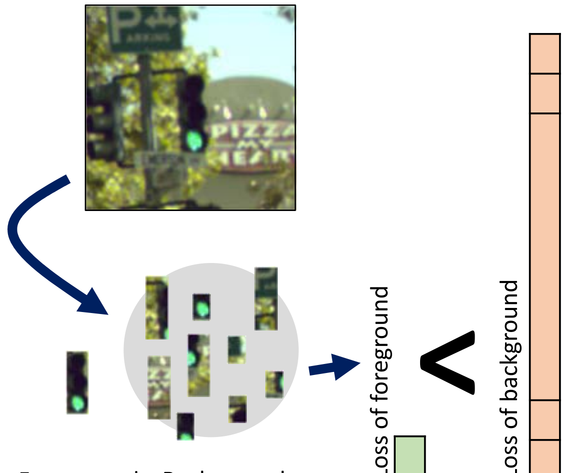
- Vision-based traffic light detection algorithm
  - Autonomous Vehicles (AV) and Advanced Driver Assistance Systems (ADAS) require ability to detect surrounding objects
  - Traffic light (TL) is one of most important elements to detect
    - A vehicle should be able to detect the traffic lights and take proper actions based on the signal of traffic lights
    - A vehicle can avoid traffic accidents



Problems to Detect TL

- Traffic light is too small
  - A TL at 50 m occupies only 12x4 px in 1280x720 an image
  - CNN easily lose of details of small object (TL)
- Numerous background examples dominate training procedure
  - *foreground:background = 1:12543* in our experiments



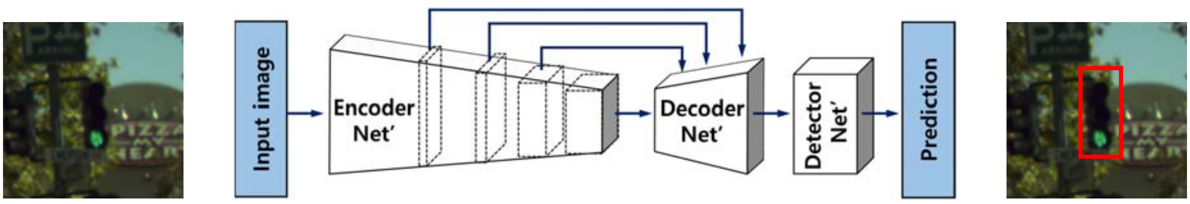


Proposed Methods

- Deconvolutional Deep Neural Network for TL detection
  - YOLOv2 based, Encoder-decoder hourglass structure
    - Preserve information of small TLs to end of the network
    - Improve detection accuracy of small TLs
  - Freestyle anchor box
    - defined by offsets, width, and height
    - Predict bounding box candidates more densely
- Focal Regression Loss
  - Reduce loss of easy examples
  - Prevent easy background examples to dominate training procedure
  - Used to train proposed TL detector

Deconvolutional Deep Neural Network for TL Detection

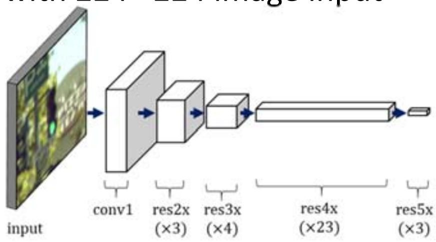
Overview



- Consists of 3 sub-networks
  - **Encoder network** : encode an input image to feature maps
  - **Decoder network** : refine encoder network’s output
  - **Detector network** :
    predict bounding boxes, confidences, class scores

Encoder Network

- Make an input image to feature maps
- Use ResNet-101 as Encoder network
  - Detailed structure with 224×224 image input

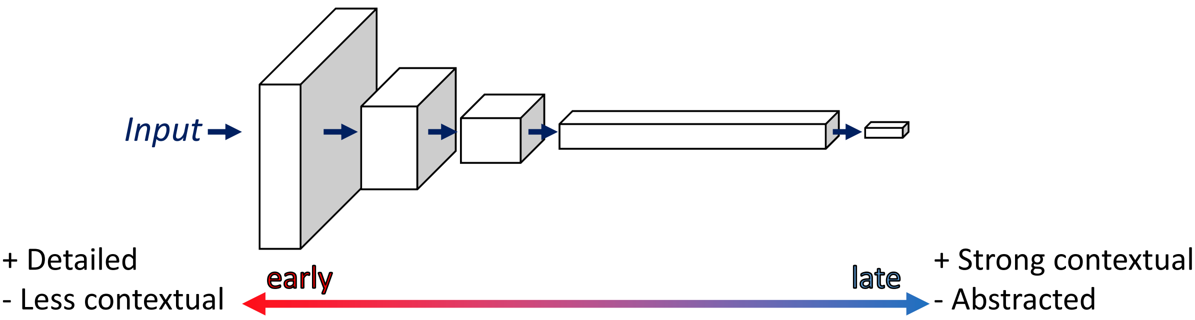


Layer name	Output size	Filter (kernel, #, stride)
conv1	112 × 112	7×7, 64, 2
pool1	56 × 56	3×3 max pool, -. 2
res2x	56×56	[(1×1, 64), (3×3, 64), (1×1, 256)] ×3
res3x	28×28	[(1×1, 128), (3×3, 128), (1×1, 512)] ×4
res4x	14×14	[(1×1, 256), (3×3, 256), (1×1, 1024)] ×23
res5x	7×7	[(1×1, 512), (3×3, 512), (1×1, 2048)] ×3

Feature Maps of Encoder Network

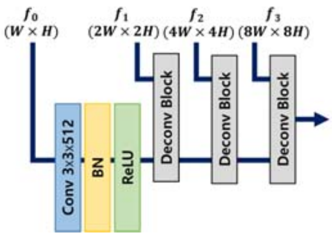
- Early feature map
  - A feature map from an early (= close to input) layer
  - Retain **detailed** information
  - Has less contextual information
- Late feature map
  - A feature map from a late (= far from input) layer
  - Has abstracted information
  - Has **strong contextual** information

Required to detect small objects

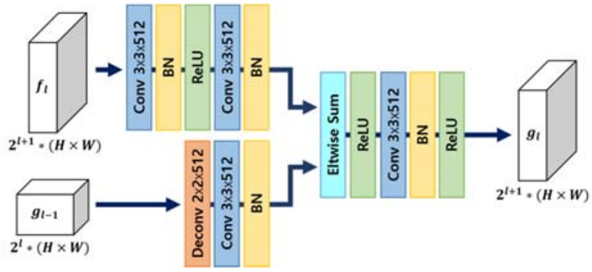


Decoder Network

- Combines encoder network's feature maps
  - Upsamples late feature maps to match the resolution with early feature maps by using deconvolutional layer.
  - Combines upsampled late feature maps and early feature maps.
- The **final result feature map** contains **details** as well as **strong contextual information**.
- f_0=res5c, f_1=res4b22, f_2=res3b3, f_3=res2c



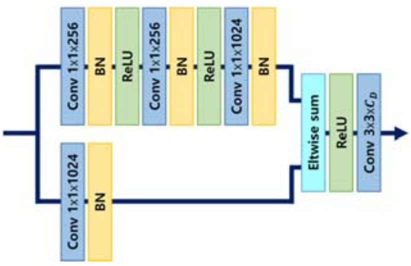
Decoder network



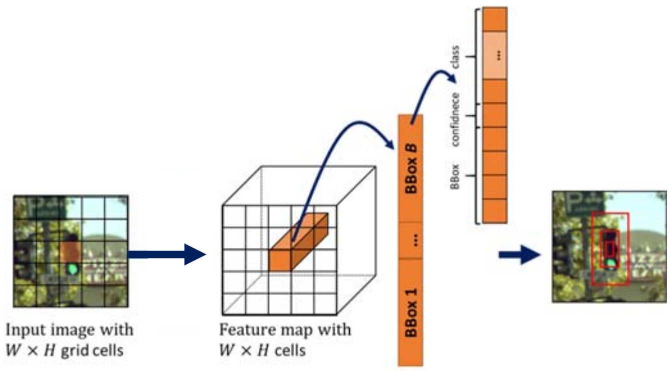
Deconv block

Detector Network

- Detect traffic lights from decoder network's result
- Configuration of final result is based on YOLOv2
  - When final feature map has spatial resolution of $W \times H$, then the input image is divided into $W \times \mathrm{H}$ grid cells.
  - Each grid cell corresponds to each cell of the final feature map
  - Bounding box predictions are spatially based on anchor boxes



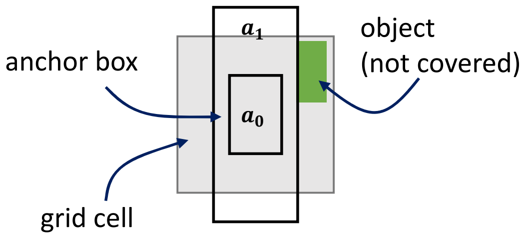
Detector network

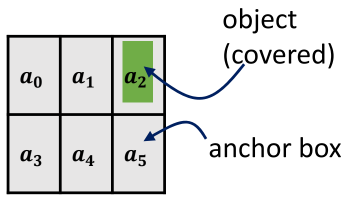


Final result configuration

Anchor Box Definition

- Existing anchor box (in YOLOv2)
  - Defined by width and height, then located at center of a grid cell
  - Anchor box for a small traffic light is much smaller than a grid cell
  - **<u>Can not cover whole area of a grid cell with a small anchor box</u>**
- **<u>Freestyle anchor box</u>**
  - Defined by offsets, width, and height ($\boldsymbol{a} = \left(o^x, o^y, a^w, a^h\right)$)
  - Can be located at arbitrary location in a grid cell
  - **<u>Cover whole area of a grid cell with small anchor boxes</u>**





Freestyle Anchor Box Definition

- j-th anchor box :

$$\boldsymbol{a_j} = (o_j^x, o_j^y, a_j^w, a_j^h)$$

- j-th anchor box placed at i-th grid cell :

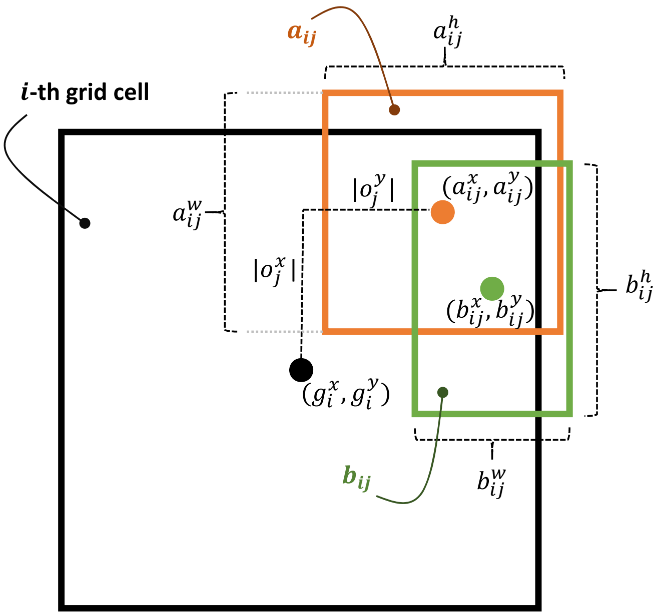
$$\boldsymbol{a_{ij}} = \left(g_i^x + o_j^x, g_i^y + o_j^y, a_j^w, a_j^h\right) = \left(a_{ij}^x, a_{ij}^y, a_{ij}^w, a_{ij}^h\right)$$

- A bounding box whose center falls in $\boldsymbol{a_{ij}}$:

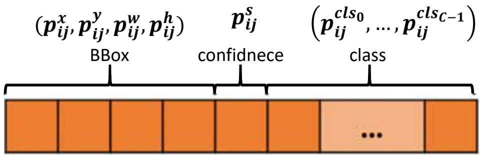
$$\boldsymbol{b_{ij}} = \left(b_{ij}^x, b_{ij}^y, b_{ij}^w, b_{ij}^h\right)$$

- Relative representation form of $\boldsymbol{b_{ij}}$ to $\boldsymbol{a_{ij}}$:

$$\begin{aligned} \boldsymbol{t_{ij}^{bb}} &= \left(t_{ij}^x, t_{ij}^y, t_{ij}^w, t_{ij}^h\right), \\ t_{ij}^x &= (b_{ij}^x - a_{ij}^x)/a_{ij}^w + 0.5, \\ t_{ij}^y &= (b_{ij}^x - a_{ij}^y)/a_{ij}^h + 0.5, \\ t_{ij}^w &= \ln(b_{ij}^w/a_{ij}^w), \\ t_{ij}^h &= \ln(b_{ij}^h/a_{ij}^h). \end{aligned}$$



Prediction Interpretation



- $(\boldsymbol{p_{ij}^x}, \boldsymbol{p_{ij}^y}, \boldsymbol{p_{ij}^w}, \boldsymbol{p_{ij}^h})$
  - Prediction for bounding box coordinates
  - Correspond to $t_{ij}^{bbox} = \left(t_{ij}^x, t_{ij}^y, t_{ij}^w, t_{ij}^h\right)$
  - Absolute form of predicted bounding box $\bar{b}_{ij}$:

    $\boldsymbol{\bar{b}_{ij}} = \left(\bar{b}_{ij}^x, \bar{b}_{ij}^y, \bar{b}_{ij}^w, \bar{b}_{ij}^h\right)$

    $\bar{b}_{ij}^x = a_{ij}^w\left(\sigma\left(p_{ij}^x\right) - 0.5\right) + a_{ij}^x$

    $\bar{b}_{ij}^y = a_{ij}^h\left(\sigma\left(p_{ij}^y\right) - 0.5\right) + a_{ij}^y$

    $\bar{b}_{ij}^w = a_{ij}^w e^{p_{ij}^w}$

    $\bar{b}_{ij}^h = a_{ij}^h e^{p_{ij}^h}$

- $\boldsymbol{p_{ij}^s}$
  - Confidence of the bounding box
  - $\sigma\left(p_{ij}^s\right) = \Pr(object|i,j) * IOU(\bar{b}_{ij}, b_{ij})$

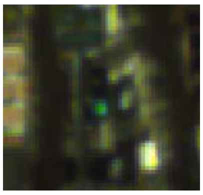
- $\left(\boldsymbol{p_{ij}^{c_0}}, \ldots, \boldsymbol{p_{ij}^{c_{C-1}}}\right)$
  - Class probabilities of the bounding box

Focal Regression Loss

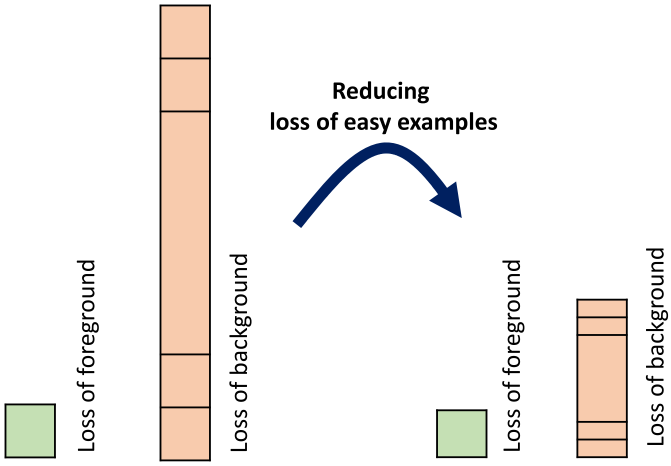
Balancing Loss of Foregrounds and Backgrounds

- Main idea : reducing loss of easy examples
  - Most of background examples are easy examples
  - Reducing loss of easy examples to balance between loss of foreground and loss of backgrounds
  - Focus on hard examples

Easy foreground and backgrounds

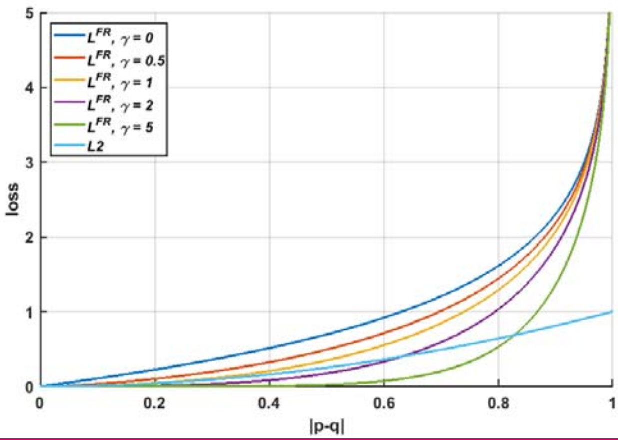


Hard foreground



Focal Regression Loss

- **Focal Regression Loss**
  - $\mathcal{L}^{FR}(p,q) = -|p-q|^{\gamma}\log(1-|p-q|)$
    - $p \in [0,1]$: regressed value
    - $q \in [0,1]$: regression target
    - $\gamma \geq 0$: focusing parameter
    - $|p-q|^{\gamma}$: modulating factor



- The ratio of the $\mathcal{L}^{FR}$ for $|p-q| = 0.8$ over $\mathcal{L}^{FR}$ for $|p-q| = 0.2$

γ	$\frac{\mathcal{L}^{FR}(0.8,0)}{\mathcal{L}^{FR}(0.2,0)}$
0	7.21
2	115.40
5	7385.67

Loss for Proposed TL Detector

- The loss for proposed TL detector
  - $\mathcal{L} = \mathcal{L}_{obj} + \lambda_{bb}\mathcal{L}_{bb} + \lambda_{class}\mathcal{L}_{class}$
  - Based on YOLOv2
  - Weighted sum of 3 sub-losses

- $\mathcal{L}_{obj}$: Loss for confidence regression
  - Original YOLOv2 uses L2 loss, but we **substitute L2 loss to focal regression loss.**
  - $\mathbb{1}_{ij}$: foreground indication function
    - $\mathbb{1}_{ij}$=1 where a_{ij} has target object, $\mathbb{1}_{ij}$=0 for otherwise.

$$\mathcal{L}_{obj} = \lambda_{obj}\sum_i\sum_j \mathbb{1}_{ij}\mathcal{L}^{FR}\left(\sigma(p_{ij}^s), IOU(\bar{b}_{ij}, b_{ij})\right) + \lambda_{noobj}\sum_i\sum_j (1-\mathbb{1}_{ij})\mathcal{L}^{FR}(\sigma(p_{ij}^s), 0)$$

Sub-Loss

- $\mathcal{L}_{bb}$: Loss for bounding box prediction
  - Same as original YOLOv2's.
  - Foreground : regress to target
  - Background : regress to corresponding anchor box
    - (0.5, 0.5, 0, 0) is the anchor box which is relatively represented by itself

$$\mathcal{L}_{bb} = \sum_i\sum_j \mathbb{1}_{ij}\left[\left(\sigma(p_{ij}^x) - t_{ij}^x\right)^2 + \left(\sigma\left(p_{ij}^y\right) - t_{ij}^x\right)^2 + \left(p_{ij}^w - t_{ij}^w\right)^2 + \left(p_{ij}^h - t_{ij}^h\right)^2\right]$$
$$+\sum_i\sum_j (1-\mathbb{1}_{ij})\left[\left(\sigma(p_{ij}^x) - 0.5\right)^2 + \left(\sigma\left(p_{ij}^y\right) - 0.5\right)^2 + \left(p_{ij}^w - 0\right)^2 + \left(p_{ij}^h - 0\right)^2\right]$$

- $\mathcal{L}_{class}$: loss for classification
  - Use softmax with focal loss
  - $\bar{p}_{ij}^{cls_t}$: probability for class t

$$\bar{p}_{ij}^{c_t} = \frac{e^{p_{ij}^{c_t}}}{\sum_k e^{p_{ij}^{c_k}}}$$

$$\mathcal{L}_{class} = -\sum_i\sum_j 1_{ij}\left(1-\bar{p}_{ij}^{c_t}\right)^{\gamma}\ln p_{ij}^{c_t}$$

Experiment

Datasets

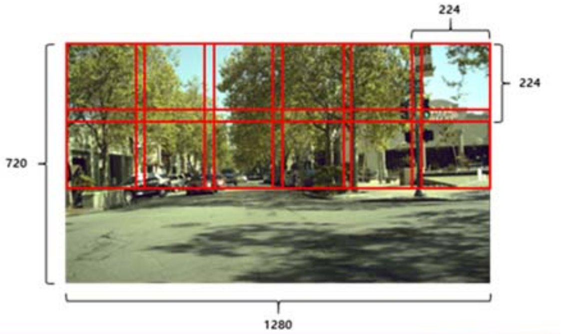
- Bosch Small Traffic Lights Dataset
  - 1280x720 images
  - training set : 5093 images, 10765 annotated TLs
  - test set : 8334 consecutive images, 13486 annotated TLs
  - 4 classes : red, yellow, green, off
  - Specialized for **small TL**
    - Even annotated to 4x8 px TLs
    - Median width of TLs : 8.5px
- Lisa Traffic Lights Dataset
  - 1280x960 images
  - training set : 13 day clips (14034 images), 97910 annotated TLs
  - test set : 4060 images
  - 5 classes : stop, stopLeft, warning, warningLeft, go
  - Rather inaccurate and inconsistent
  - Median with of TLs : 22px

Experiment for Bosch dataset

- Evaluated 5 models

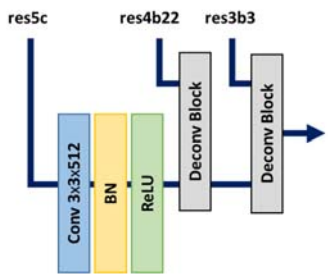
model	Description
model A	2 deconv blocks
model A+	2 deconv blocks, focal regression loss
model B	3 deconv blocks
model B+	3 deconv blocks, focal regression loss

- Input
  - Extract 224×224 sized patches from original image

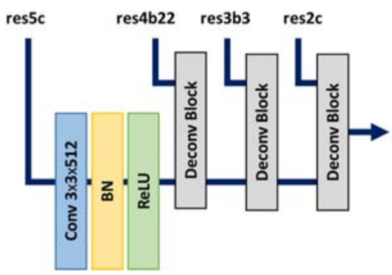


Model Details

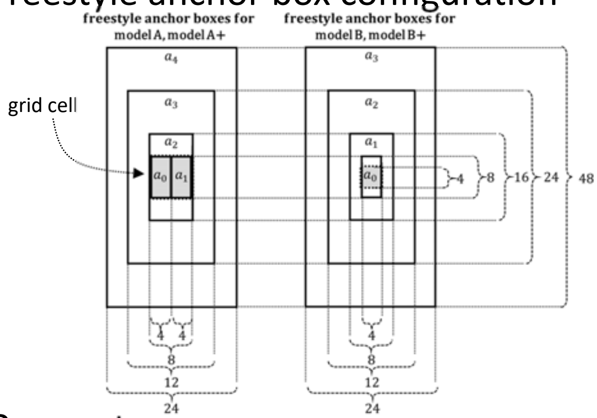
- Decoder network
  - **model A, model A+**



  - **model B, model B+**



- Freestyle anchor box configuration

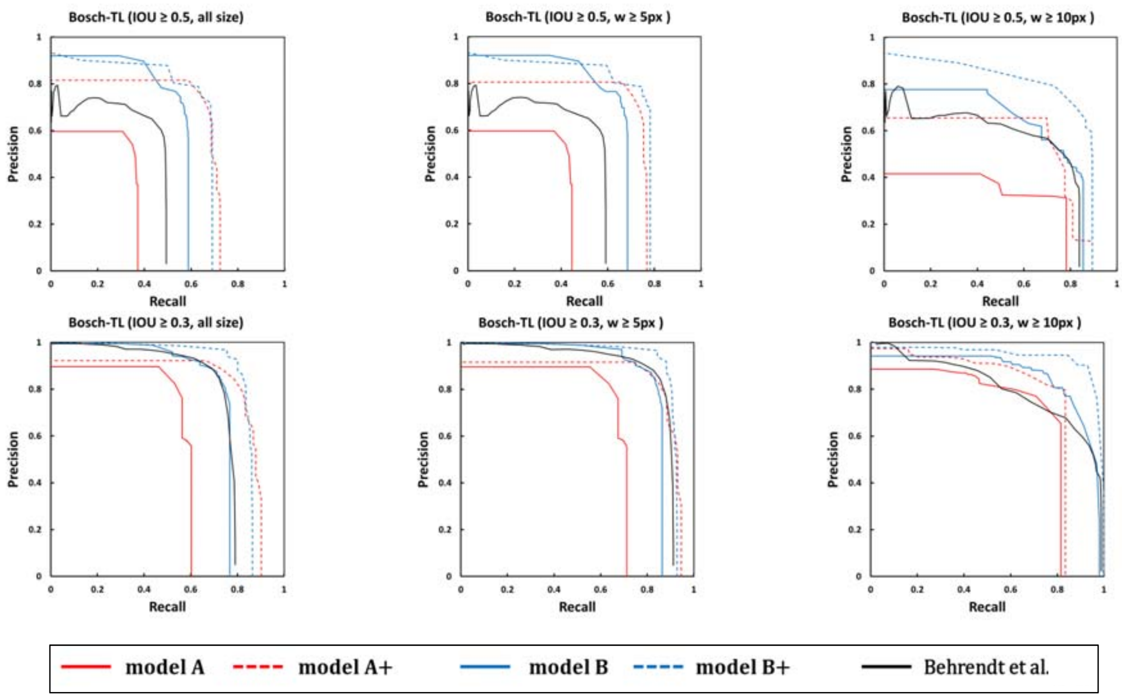


- Parameters
  - **model A, model B**
    - $\lambda_{obj} = 50, \lambda_{noobj} = 1,$ $\lambda_{bbox} = 1, \lambda_{class} = 10, \gamma = 2$
  - **model A+, model B+**
    - $\lambda_{obj} = 30, \lambda_{noobj} = 1,$ $\lambda_{bbox} = 1, \lambda_{class} = 10, \gamma = 2$

Training Details

- 2 step training
  - Step1
    - Extract patches at random position
    - Foreground : Background = 3 : 1
    - Use stochastic gradient descent algorithm (SGD), train with 10^5 iterations
    - Learning rate : started at 10^{-7}, decreased by 10^{-1} per 2×10^4 iterations
    - It is to stabilize model quickly.
  - Step2
    - Extract patches at fixed position like test process
    - Use SGD, train with 10^5 iterations
    - Learning rate : started at 10^{-7}, decreased by 10^{-1} per 2×10^4 iterations
    - It reduce false alarms

Detection Result (Bosch)



Detection Result

- Experiment for TL detectors on Bosch-TL

Area-under-the-curve (AUC) for each model

Model	$IOU \geq 0.5$			$IOU \geq 0.3$		
	all	$w \geq 5px$	$w \geq 10px$	all	$w \geq 5px$	$w \geq 10px$
Behrendt et al.	0.3267	0.3916	0.5087	0.7019	0.8209	0.7844
model A	0.2175	0.2612	0.2965	0.5215	0.6193	0.6805
model A+	0.5692	0.6073	0.5192	0.796	0.8351	0.7607
model B	0.5130	0.5995	0.5907	0.7354	0.8357	0.8588
model B+	**0.5973**	**0.6806**	**0.7518**	**0.8376**	**0.9039**	**0.9442**

Detection Result

- Experiment for TL detectors on Bosch-TL

mAP for each model

Model	$IOU \geq 0.5$			$IOU \geq 0.3$		
	all	$w \geq 5px$	$w \geq 10px$	all	$w \geq 5px$	$w \geq 10px$
Faster R-CNN	0.53	-	-	-	-	-
Behrendt et al.	0.4*	-	-	-	-	-
Pon et al.	0.46	-	-	-	-	-
model A	0.2010	0.2414	0.3246	0.4802	0.5702	0.6793
model A+	**0.5681**	0.6131	0.5112	0.7816	0.8253	0.7463
model B	0.5021	0.5865	0.5676	0.6850	0.7736	0.8115
model B+	0.5641	**0.6418**	**0.7289**	**0.7871**	**0.8499**	**0.9058**

* : estimated by Pon et al.
bold : largest mAP

DEMO

Conclusion

- The proposed TL detector is evaluated on two public TL detection benchmark datasets, then it shows higher mAP and AUC than existing TL detection methods.
- The proposed focal regression loss improves detection accuracy of the TL detector.

References

- K. Behrendt, L. Novak, and R. Botros, "A deep learning approach to traffic lights: Detection, tracking, and classication", in Robotics and Automation (ICRA), 2017 IEEE International Conference on. IEEE, 2017, pp. 1370-1377.
- M. B. Jensen, K. Nasrollahi, and T. B. Moeslund, "Evaluating state-of-the-art object detector on challenging traffic light data", in Computer Vision and Pattern Recognition Workshops (CVPRW), 2017 IEEE Conference on. IEEE, 2017, pp. 882-888.
- M. P. Philipsen, M. B. Jensen, A. Mgelmose, T. B. Moeslund, and M. M. Trivedi, "Tracffic light detection: A learning algorithm and evaluations on challenging dataset", in intelligent transportation systems (ITSC), 2015 IEEE 18th international conference on. IEEE, 2015, pp. 2341-2345.
- J. Redmon and A. Farhadi, "Yolo9000: Better, faster, stronger", in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 2017, pp. 6517-6525.
- V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation", IEEE transactions on pattern analysis and machine intelligence, vol. 39, no. 12, pp. 2481-2495, 2017.
- K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition", in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770-778.
- T. Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection", in 2017 IEEE International Conference on Computer Vision (ICCV), Oct 2017, pp. 2999-3007.