RetinaNet for Road Damage Detection

1st Drake Svoboda
Wayne State University
Detroit, USA
drake.svoboda@wayne.edu

***Abstract*—We trained a RetinaNet object detector to compete in the *Road Damage Detection Challenge* hosted at the 2018 *IEEE Big Data Cup*. Our model achieved a 0.54 F-Measure on the test set withheld by the competition's organizers. Code can be found at github.com/deepditch/deep.lib.**

***Index Terms*—Object detection, Computer vision, CNN, Deep learning**

I. Introduction

We describe an adaptation of RetinaNet [4] for road damage detection. The network is comprised of a ResNet-34-FPN backbone and two subnetworks: a classification subnetwork and a bounding box regression subnetwork. We trained our model on the road damage dataset gathered in [6]. We evaluate the performance of our model on a validation and test set. We also evaluate the computational performance of our model.

II. Model Architecture

A. ResNet-FPN Backbone

An FPN with 4 levels, $\{P_4, P_5, P_6, P_7\}$, was built on top of a pretrained ResNet 34 base network. Each level P_l has 256 channels and a resolution 2^l lower than the input image [3]. The shared classification subnet and box regression subnet make predictions at each level of the FPN. This architecture differs slightly from [4] as P_3 was not used. We choose to remove P_3 since few ground truth bounding boxes having a width or height less than 32 pixels and removing P_3 improves computational performance. Fig. 1 and Fig. 2 show of ground truth bounding box width and height distributions respectively.

B. Anchor Boxes

Anchor boxes have areas $\{54^2, 108^2, 216^2, 432^2\}$ on levels P_4-P_7 respectively. These are smaller than the anchor boxes described in [4]. The authors of [4] used 600 pixel images; however, we chose to use 512 pixel images and scale the anchor boxes accordingly. Additionally, at each pyramid level anchors with ratios $\{20:3, 20:7, 20:13, 1:1, 13:20, 7:20\}$ were used; that is, each spatial location has $A = 6$ anchor boxes. These ratios were chosen to cover the distribution of bounding box aspect ratios in the dataset. Fig. 3 shows the bounding box aspect ratio distribution. Each anchor box is assigned a $K = 9$ one hot vector for the class label and a 4-vector for regression targets. Anchor boxes are assigned according to the assignment rules described in [4]. Specifically, anchor boxes are assigned to a ground truth object if the anchor box and the ground truth object have a Jaccard index (intersection over union) greater than or equal to .5. Anchor boxes are assigned to the background class if the Jaccard index is less than .4. Anchor boxes with a Jaccard index in $[.4, .5)$ were ignored during training.

C. Classification Subnet

The classification subnet has four 3×3 convolutional layers. Each convolutional layer is followed by a ReLU activation function. The classification subnet outputs a final 3×3 convolutional layer with $K * A$ channels followed by a sigmoid activation function. Each set of K activations predict the class label for an anchor box in the corresponding spatial location. This structure matches the structure described in [4].

D. Regression Subnet

The box regression subnet is identical to the classification subnet except for its output layer. The box regression subnet outputs a final 3×3 convolutional layer with $4 * A$ channels. Each set of 4 activations predict the offset from the an anchor box in the corresponding spatial location. Again, this structure matches the structure described in [4].

E. Initialization

The ResNet-34 base network was initialized with pretrained weights and biases. The parameters for the batch normalization layers of the ResNet-34 base network were not updated during training. We initialized additional convolutional layers in accordance with [4].

III. Training

80% of the total dataset was released by the organizers of the *Road Damage Detection* challenge. The remaining 20% was withheld for scoring the competition. Of the 80% released, we took a 9:1 random split for training and validation respectively. The model was trained for 63 epochs on the training set.

A. Focal Loss

We used focal loss as described in [4] with $\gamma = 2$ and $\alpha = .5$ on the output of the classification subnet. Loss was computed as the sum over all of the non-ignored anchor boxes divided by the number of anchor boxes assigned to a ground truth object. We used smooth L1 loss on the output of the regression subnet [1]. We calculated total loss as the sum of the classification loss and the regression loss.

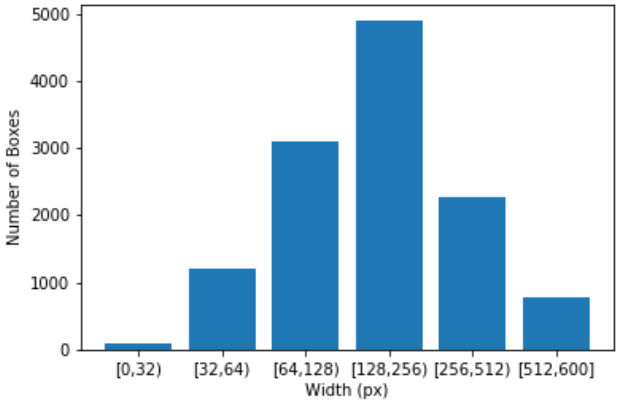

Fig. 1. Ground truth bounding box width distribution.

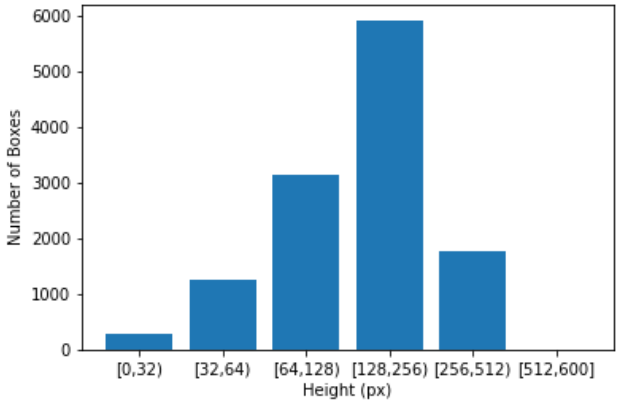

Fig. 2. Ground truth bounding box height distribution.

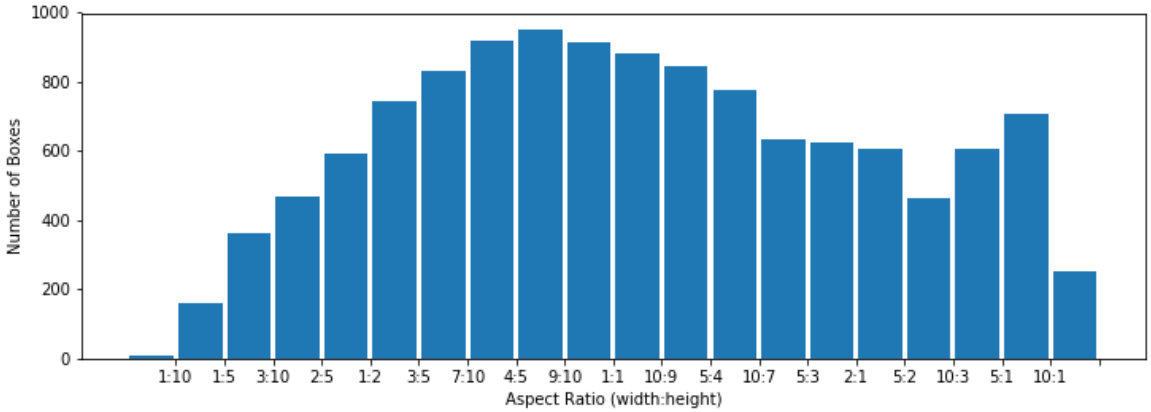

Fig. 3. Ground truth bounding aspect ratio distribution.

B. Optimization

During training, we used dropout regularization with $p = .2$ for each convolutional layer in the FPN and the subnets. The Adam optimizer [2] was used with an initial learning rate if .0001. The learning rate was decayed to .000001 with a cosine annealing as described in [5] with $T_{mult} = 2$.

For data augmentation, we randomly scaled images between 512 pixels and 600 pixels. After scaling, we took a random 512 pixel crop. Additionally, images were randomly horizontally flipped with a probability of .5.

IV. Evaluation

We applied non-maximum suppression to the outputs of our model for evaluation. Our model was evaluated against the *Road Damage Detection Challenge* test set and our own withheld validation set. The precision and recall for each class on our validation set is presented in Table 1. We used a Jaccard index of .5 or greater to determine positive matches. Inference takes roughly 10ms on an Nvidia Quadro P5000 GPU.

Our model achieved a 0.54 F-Measure on the test set withheld by the competition organizers.

TABLE I
Detection and classification results for each class

Class	D00	D01	D10	D11	D20	D40	D43	D44
Precision	0.48	0.66	0.33	0.16	0.69	0.19	0.77	0.74
Recall	0.58	0.76	0.23	0.15	0.72	0.30	0.74	0.76

References

[1] Ross B. Girshick. "Fast R-CNN". In: *CoRR* abs/1504.08083 (2015). arXiv: 1504 . 08083. URL: http://arxiv.org/abs/1504.08083.

[2] Diederik P. Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization". In: *CoRR* abs/1412.6980 (2014). arXiv: 1412.6980. URL: http://arxiv.org/abs/1412.6980.

[3] Tsung-Yi Lin et al. "Feature Pyramid Networks for Object Detection". In: *CoRR* abs/1612.03144 (2016). arXiv: 1612.03144. URL: http://arxiv.org/abs/1612.03144.

[4] Tsung-Yi Lin et al. "Focal Loss for Dense Object Detection". In: *CoRR* abs/1708.02002 (2017). arXiv: 1708.02002. URL: http://arxiv.org/abs/1708.02002.

[5] Ilya Loshchilov and Frank Hutter. “SGDR: Stochastic Gradient Descent with Restarts”. In: *CoRR* abs/1608.03983 (2016). arXiv: 1608 . 03983. URL: http://arxiv.org/abs/1608.03983.

[6] Hiroya Maeda et al. “Road Damage Detection Using Deep Neural Networks with Images Captured Through a Smartphone”. In: *CoRR* abs/1801.09454 (2018). arXiv: 1801.09454. URL: http://arxiv.org/abs/1801.09454.