Proofpoint

Small Object Detection Literature Review
Model Building

15th October 2020

Duke

Annotation Update

- Scraped 5 new images per company for a total of 2,250 new images.
- Annotated ~500 images so far this week with ~700 more by the end of the week.
- New images obtained using Google search terms like “Apple Inc. Logo”.
- More *false positives* than before but still have mostly *true positives*.
- Introduced a *generic logo* class to label logos we are unfamiliar with.

Duke

Challenge is Small Object Detection

Duke

Small object detection in the context of logos

- Size of images: **1440x2560** - 720x1280 with device scale factor of 2.0.
- Average size of logo is ~**188x331** i.e. **1.7%** of the image.
- Average size of very small logos is ~ **57x47** so **0.07%** of the image.
- We don’t expect to encounter too many issues with average sized logos but do expect to have issues with small social media logos.

Duke

Small Object Detection Overview

- Important and recent challenge in computer vision/object detection
- What is a “small object”
  - Physically smaller in the real world
  - Occupying areas less than 32x32 pixels in common object datasets (COCO)
- Challenges in detection
  - Often hard to distinguish from background or similar larger objects
  - Harder to accurately localize
  - Most state-of-the-art models are focused on larger object detection

Duke

Methods for Improving Small Object Detection

- Multi-scale feature learning
  - Features from different layers and scales
  - Pyramid feature map structures
- Data augmentation
  - Modifying existing images through rotation, cropping, flipping, etc to produce more samples of the class
- Training strategy
  - Only select ground truth boxes at specified size and resolution
- Context-based detection
  - Local → extract features from surrounding region/region proposals
  - Global → learning the context of the entire image
- GAN-based detection
  - Generator attempts to enhance poor representation of small objects
  - Discriminators attempts to recognize generated vs real

Duke

Performance

Table 5
Detection results on the MS-COCO test-dev dataset of some typical methods. "++" denotes applying inference strategy such as multi scale test, horizontal flip, etc. (in %).

Type		Method	Backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Multi-scale feature learning	Featurized image pyramids	SNIP [25]	DPN98 [84]	45.7	67.3	51.1	29.3	48.8	57.1
	Single feature map	Faster R-CNN [15]	VGG16	21.9	42.7	–	–	–	–
	Pyramidal feature hierarchy	SSD300 [16]	VGG16	23.2	41.2	23.4	5.3	23.2	39.6
		SSD512 [16]	VGG16	26.8	46.5	27.8	9.0	28.9	41.9
	Integrated features	ION [27]	VGG16	24.6	46.3	23.3	7.4	26.2	38.8
	Feature pyramid network	FPN [22]	ResNet101	36.2	59.1	39.0	18.2	39.0	48.2
	Variants of FPN	RefineDet512 [47]	ResNet101	36.4	57.5	39.5	16.6	39.9	51.4
	(including feature fusion and feature pyramid generation, multi-scaled fusion module, etc.)	RefineDet512++ [47]	ResNet101	41.8	62.9	45.7	25.6	45.1	54.1
		M2Det800 [53]	VGG16	41.0	59.7	45.0	22.1	46.5	53.8
		M2Det800++ [53]	VGG16	44.2	64.6	49.3	29.2	47.9	55.1
		FSSD300 [46]	VGG16	27.1	47.7	27.8	8.7	29.2	42.2
		FSSD512 [46]	VGG16	31.8	52.8	33.5	14.2	35.1	45.0
		DSSD321 [19]	ResNet101	28.0	46.1	29.2	7.4	28.1	47.6
		DSSD513 [19]	ResNet101	33.2	53.3	35.2	13.0	35.4	51.1
		MDSSD300 [21]	VGG16	26.8	46.0	27.7	10.8	–	–
		MDSSD512 [21]	VGG16	30.1	50.5	31.4	13.9	–	–
Data augmentation		Augmentation [24]	ResNet50	30.4	–	–	17.9	32.9	38.6
Training strategy		SNIP [25]	DPN98 [84]	45.7	67.3	51.1	29.3	48.8	57.1
		SNIPER [26]	ResNet101	46.1	67.0	51.6	29.6	48.9	58.1
		SAN [51]	R-FCN	36.3	59.6	–	16.7	40.5	55.5
Context-based detection	Local context	MPNet [40]	ResNet	33.2	51.9	36.3	13.6	37.2	47.8
		CoupleNet [45]	ResNet101	33.1	53.5	35.4	11.6	36.3	50.1
		CoupleNet++ [45]	ResNet101	34.4	54.8	37.2	13.4	38.1	50.8
	Global context	ION [27]	VGG16	24.6	46.3	23.3	7.4	26.2	38.8
		R-FCN++ [48]	R-FCN	42.3	63.8	–	25.2	46.1	54.2
		ORN [49]	ResNet50	30.5	50.2	32.4	–	–	–
		SIN [50]	VGG16	23.2	44.5	22.0	7.3	24.5	36.3
GAN-based detection		MTGAN [30]	ResNet101	41.4	63.2	45.4	24.7	44.2	52.6

Duke

Review Conclusions

- These methods can help improve performance of small object detection
- Input resolution is also very important
  - Easier to detect when resolution is higher
- More powerful CNN backbone still matters
- Data augmentation may be the easiest to implement for our models

Duke

Additional Small Object Detection Solutions

Duke

Small Object Detection Survey Paper - 2018

- Suggestions targeted toward improving detection of objects occupying less than 1% of an image.
- Oversample and Augment the small object samples.
- Improve the training scheme by modifying the loss function.
- Modify the existing model architecture or develop new innovations.

Duke

Oversampling and Data Augmentation

- Only about 50% of the images in MS COCO contain small objects while 80% contain large objects.
- Different oversampling ratios can be explored.
- Oversampling ratio can be chosen based on relative importance of small and large objects.
- Data augmentation can be used if the model is supposed to be invariant to object rotation. Does this apply to logos?

Focal Loss

- Focal loss is simply an addition to the cross entry loss function, optimizing it for dense object detection.
- Focal Loss can be used to penalize mistakes on harder samples more than easier samples.
- This leads to the model paying more attention to the harder samples (ones with small logos) and learning more from them.

Additions to Faster-RCNN

- Anchor Box Optimization - make anchor boxes smaller.
- Increase aspect ratios of the anchor boxes.
- Produce high resolution feature maps: upsampling small object image to a larger scale.

Duke

Object Detection in 20 Years: A Survey - 2019

- Large Receptive Fields: use dilated convolutional layers i.e. expand the convolutional filters and use sparse parameters.
- Adaptive Zoom-In: Adaptively enlarge small objects into large ones.
- Use GANs: Narrow the representations between small and large objects.

Duke

Model Building

Duke

Introduction

- We intend to build a YOLO v3 and Faster RCNN as our base models

- Start with Faster RCNN
  - Annotation format
  - **Screenshots with negative examples**

- We will now look into some potential GitHub repos...

Duke

Faster RCNN

- https://github.com/rbgirshick/py-faster-rcnn
  - Official repo by the authors
  - Python (not Matlab)
  - Deprecated

- https://github.com/jwyang/faster-rcnn.pytorch
  - Another implementation using Pytorch
  - Deprecated

Facebook Research - Detectron

- https://github.com/facebookresearch/Detectron (caffe2)
- https://github.com/facebookresearch/detectron2 (PyTorch)
  - A vast, comprehensive codebase for object detection
  - This would be a good starting point for Faster RCNN

- https://colab.research.google.com/drive/16jcaJoc6bCFAQ96jDe2HwtXj7BMD_-m5

Duke

Alternative

- https://github.com/tensorpack/tensorpack
  - Uses TensorFlow
  - Optimised for speedup
  - Not specific to Object Detection

Duke

YOLO v3

- https://github.com/ultralytics/yolov3 (Recommended)
  - Well documented - has a tutorial Colab notebook
  - PyTorch
  - Has specific notes on training custom data

- https://github.com/YunYang1994/TensorFlow2.0-Examples/tree/master/4-Object_Detection/YOLOV3
  - TensorFlow
- https://github.com/qqwweee/keras-yolo3
  - Keras

Quick Note: Final Report Format

```
# Whitesides Outline Template

**Working Title:**

**Authors:**

**Unmet Need:**

**Hypothesis/Goal/Aim:**

**Objectives with (short, one-sentence) justification:**
(Should serve to revisit/update your story board)

**Audience:**

**Style:** (e.g. academic paper, white paper, or public facing documents such as a blog)

**Abstract**
(Place holder; to be written only after rest of paper is complete and not in the outline)

## Introduction
(For now write only a brief introductory paragraph)

## (Section Title; can include methods, algorithm development, results, etc.)

**Purpose of section:**
(Mentally, set this up as either "Because of the previous section, we now do/study/examine/explore..." or "Despite the previous section's results, we must also consider...")

**Short list of studies/results to be done for section:**

**Figures/tables:**
(Actual mock-ups, hand-drawn/cartoons are ok to start)

## (Follow with as many sections as needed with appropriate titles)

## Conclusion/Discussion
(Place holder)
```

Duke

Next Steps

- Annotate ~700 images.
- Will try to have a preliminary model for Faster R-CNN by next week (Present to David).
- Will try to have a preliminary model YOLO v3 by the next Proofpoint meeting.

Other logo sources - email, documents etc.

Look into tackling open-set problem (one or two ways to tackle this)

Duke

References

1. Tong, Kang & Wu, Yiquan & Zhou, Fei. (2020). “Recent advances in small object detection based on deep learning: A review.” Image and Vision Computing. 97:103910.
2. Wang, J., He, Z., & Li, H. (2018). Small object detection survey paper.
3. Zou, Z., Shi, Z., Guo, Y., & Ye, J. (2019). Object detection in 20 years: A survey. *arXiv preprint arXiv:1905.05055*.

Duke

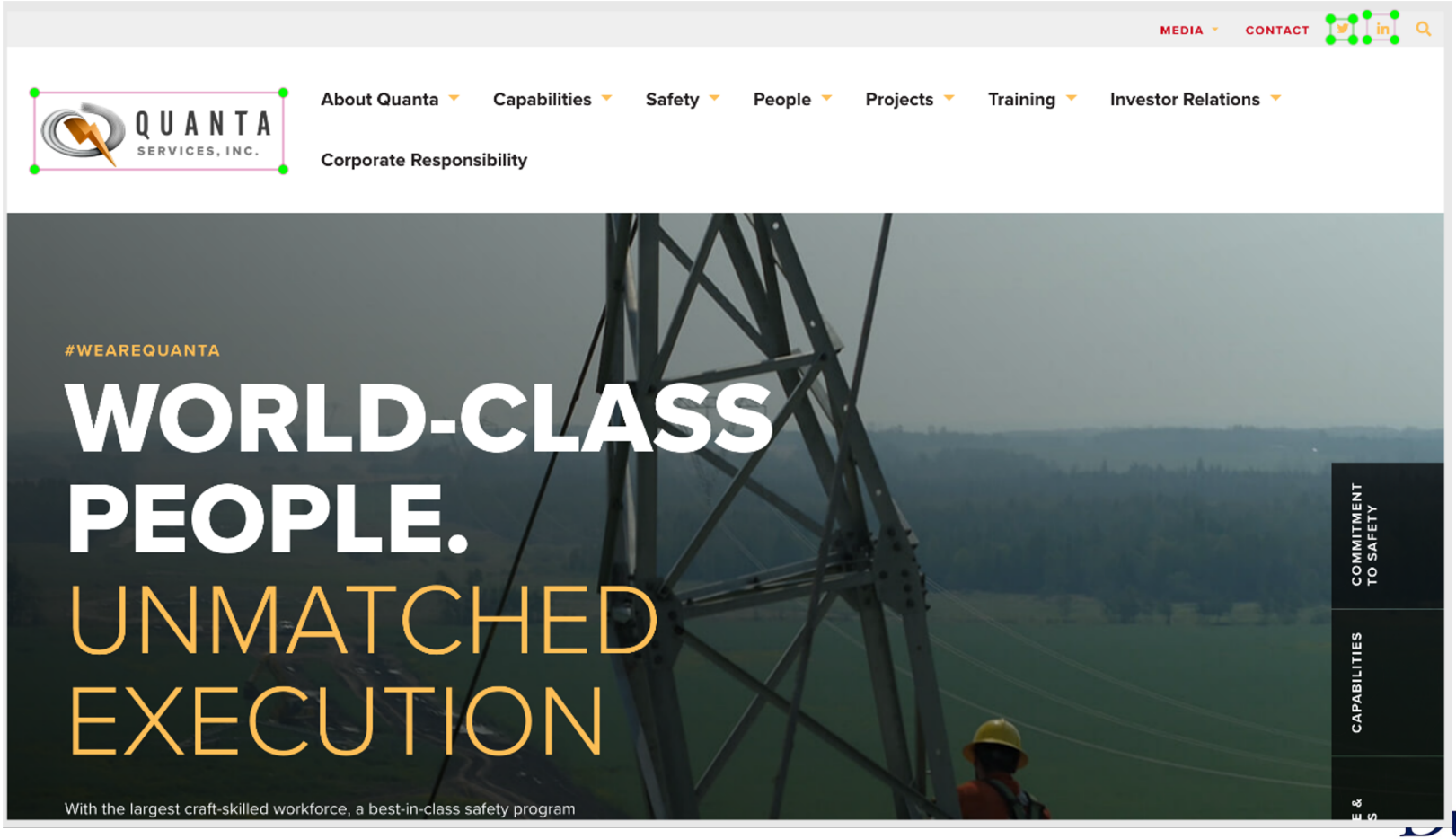
Appendix

Duke

Appendix Contents

- Image 1: Small logos.
- Image 2: Small logos + Logos with text.
- Image 3: Partially obstructed logos.

Duke


MEDIA
CONTACT
QUANTA
SERVICES, INC.
About Quanta
Capabilities
Safety
People
Projects
Training
Investor Relations
Corporate Responsibility
#WEAREQUANTA
WORLD-CLASS
PEOPLE.
UNMATCHED
EXECUTION
With the largest craft-skilled workforce, a best-in-class safety program
COMMITMENT
TO SAFETY
CAPABILITIES

Duke

Qualcomm-Logo.8.15

08 January, 2016 at 8:39 AM

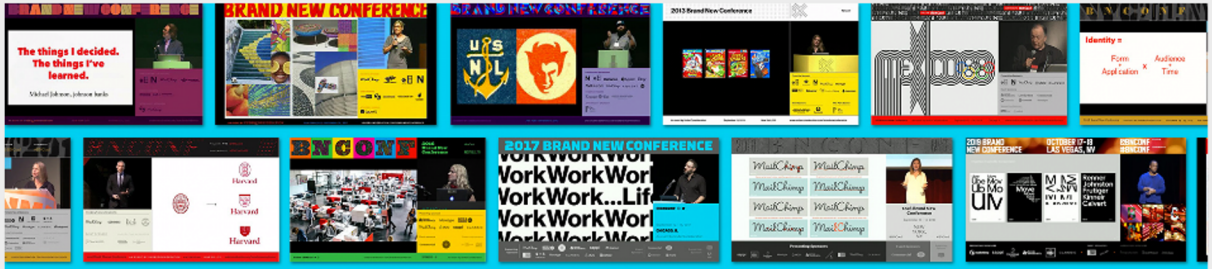
Image

Duke

Brand New

Search | Submit Tips/Work | Log in

Browse | About | SUBSCRIBE



Brand New Conference Videos / 148 presentations available / Buy singles or bundles / Old years free!

Share ›

Facebook | Twitter | Email | Pinterest | More

Image

New Name and Logo for Raytheon Technologies

Spotted

Raytheon

Raytheon

Duke