

RBN Detection for Order of Finish

Hello!

I'm Eric Bayless

Data Science Student at
General Assembly

You can find me at:
ericbayless.github.io





1

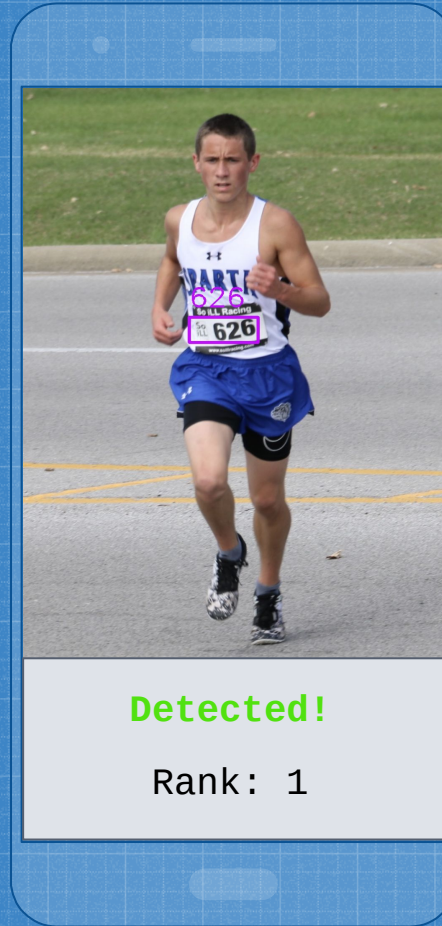
Problem Statement

Purpose and background



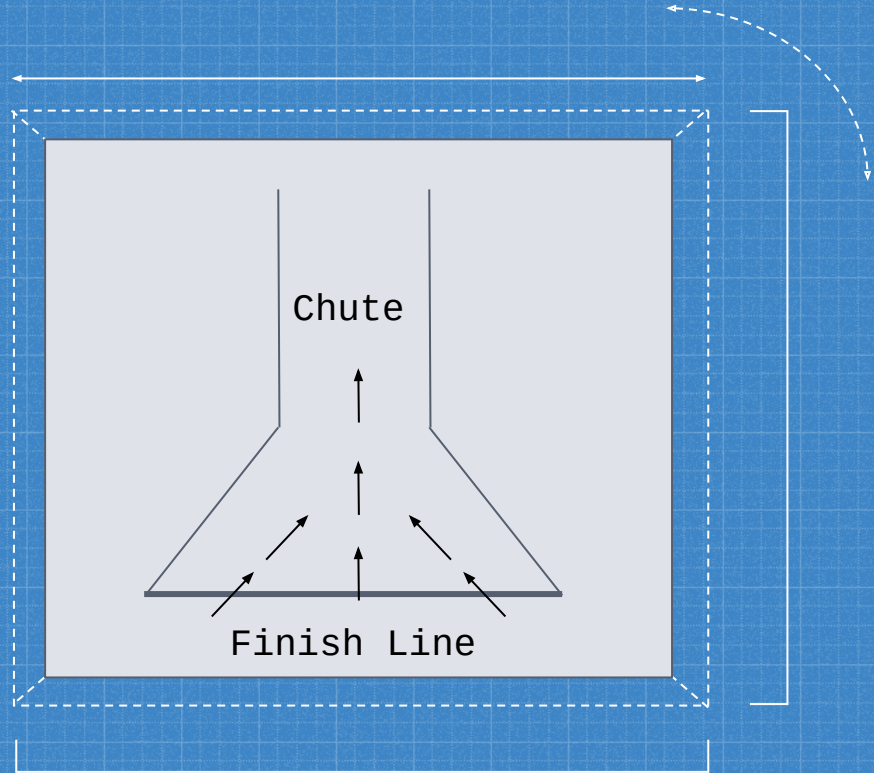
Proof Of Concept

Create a model that can detect race bib numbers in a manner that demonstrates its viability in a mobile app.



Background-Cross Country

- Chute
 - Single file
- Order of finish
 - Collect tags
 - Scan barcode





2

Data Acquisition



Description, challenges,
and solutions

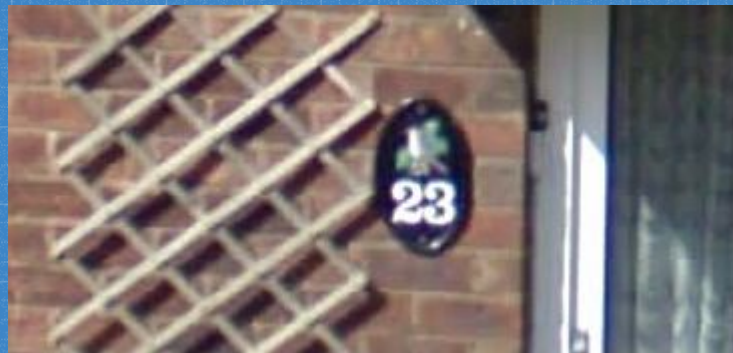
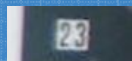
Race Bib Number (RBNR) Dataset

- 217 images containing 290 bibs
- Split into 3 sets from 3 races
- Mostly action shots



Street View House Number (SVHN) Dataset

- Over 600,000 images available
- Low resolution
- Has been used for bib number detection





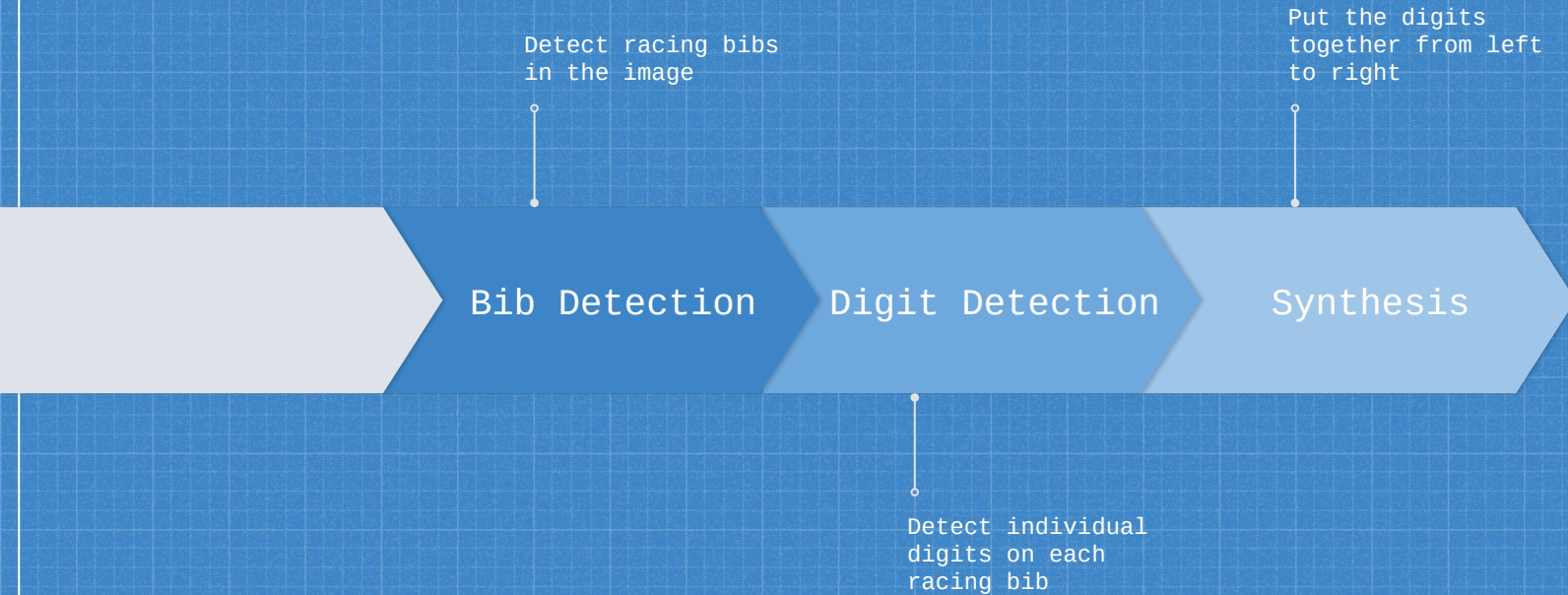
3

Model Selection



What and why

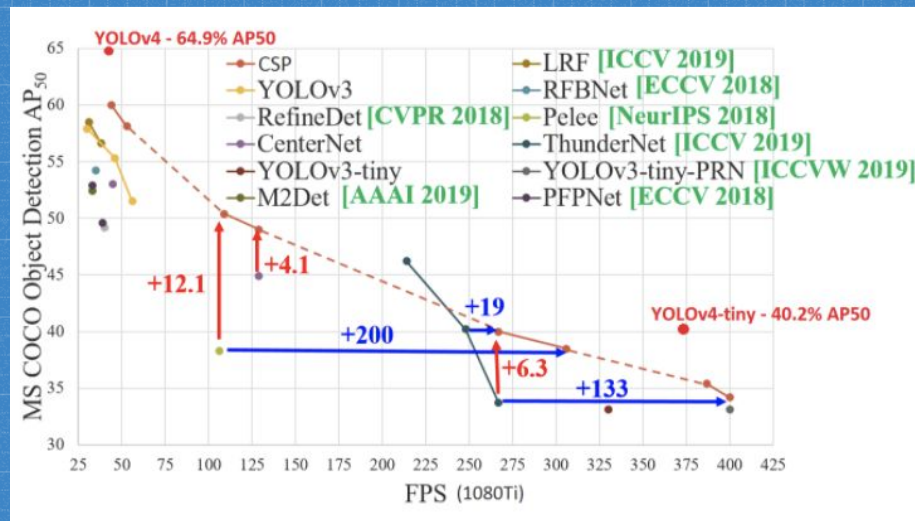
Process



YOU CAN ALSO SPLIT YOUR CONTENT

Why?

- Relatively small footprint
- 8X as fast as YOLOv4 with $\frac{2}{3}$ the performance on MS COCO
- Even less of a performance hit with fewer classes



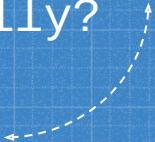


4

Training Results



How did the models do
individually?



Bib Detection

- Initial results limited by size of dataset
- Image augmentation used to create 5,088 images from initial 127
- Initial mAP@0.5 = 76.03%
- Final mAP@0.5 = 94.42%

Digit Detection - SVHN

```
calculation mAP (mean average precision)...  
13068  
detections_count = 95055, unique_truth_count = 26032  
class_id = 0, name = 0, ap = 86.72%      (TP = 1512, FP = 304)  
class_id = 1, name = 1, ap = 74.05%      (TP = 3789, FP = 884)  
class_id = 2, name = 2, ap = 86.74%      (TP = 3485, FP = 425)  
class_id = 3, name = 3, ap = 82.89%      (TP = 2200, FP = 364)  
class_id = 4, name = 4, ap = 84.72%      (TP = 2040, FP = 297)  
class_id = 5, name = 5, ap = 86.79%      (TP = 1958, FP = 256)  
class_id = 6, name = 6, ap = 85.75%      (TP = 1625, FP = 276)  
class_id = 7, name = 7, ap = 83.91%      (TP = 1608, FP = 258)  
class_id = 8, name = 8, ap = 85.98%      (TP = 1395, FP = 214)  
class_id = 9, name = 9, ap = 83.61%      (TP = 1324, FP = 286)  
  
for conf_thresh = 0.25, precision = 0.85, recall = 0.80, F1-score = 0.83  
for conf_thresh = 0.25, TP = 20936, FP = 3564, FN = 5096, average IoU = 62.52 %  
  
IoU threshold = 50 %, used Area-Under-Curve for each unique Recall  
mean average precision (mAP@0.50) = 0.841158, or 84.12 %  
Total Detection Time: 61 Seconds
```


Digit Detection - Race Bibs

- Cropped all bibs from RBNR dataset
- Only complete match counted
- Accuracy = 67.59%



5

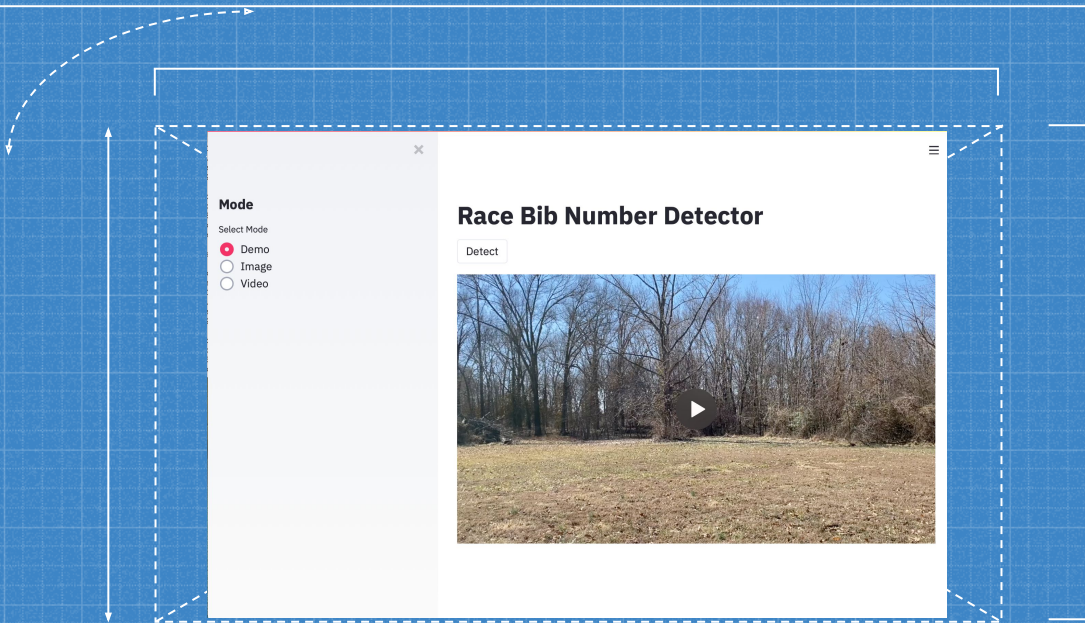
End to End Detection



Score and Demo

Digit Detection - Race Bibs

- Used set 3 of RBNR dataset
- Only complete match counted
- Accuracy = 38.05%



Demo

The proof is in the pudding



6

Conclusions

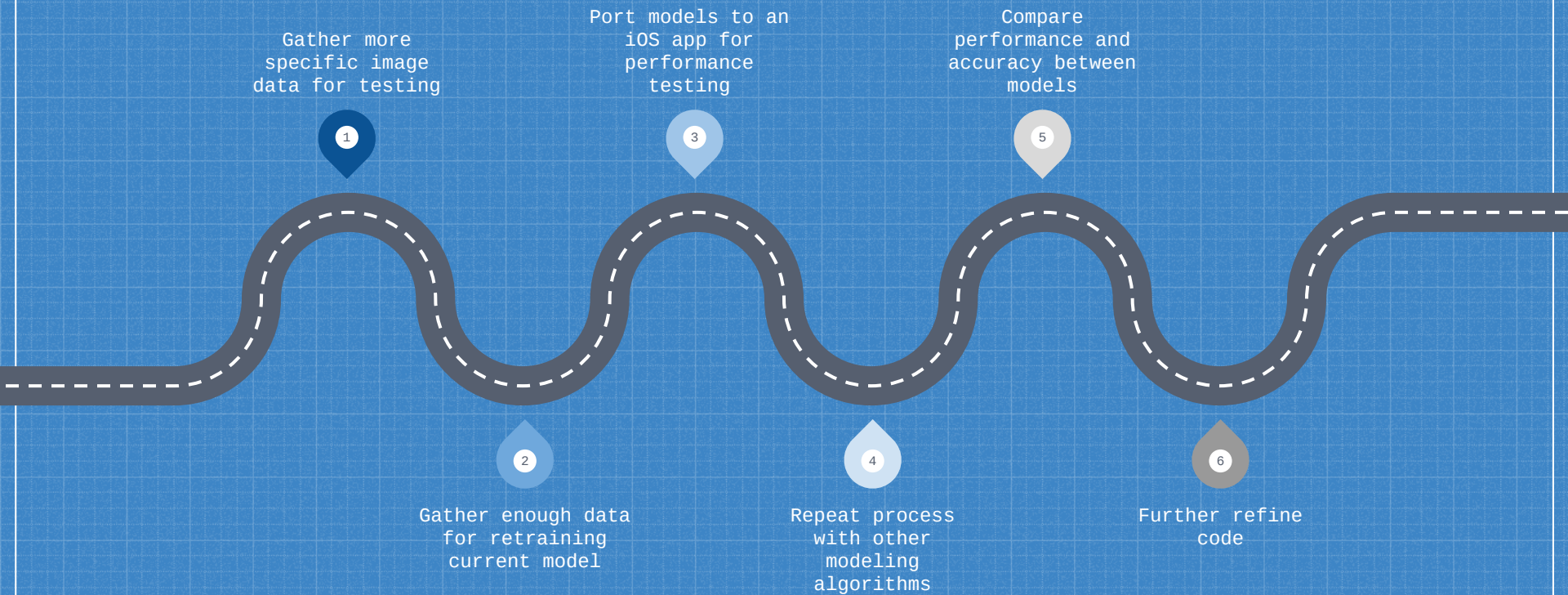


Does it work?

Summary

- Accuracy scores limited by data
- Demo and live webcam detection indicate viability
- Further work is needed to validate these assumptions

Next steps



Thanks!

ANY QUESTIONS?

You can find me at:
`ericbayless.github.io`

Resources

- <https://people.csail.mit.edu/talidekel/RBNR.html>
- <http://ufldl.stanford.edu/housenumbers/>
- https://www.researchgate.net/publication/335234017_Racing_Bib_Number_Recognition_Using_Deep_Learning
- <https://blog.roboflow.com/train-yolov4-tiny-on-custom-data-lighting-fast-detection/>
- <https://www.slidescarnival.com/valentine-free-presentation-template/234>