FAT
Farm Animal Tracking

Kemal Erdem
Marek Pokropiński

INTRODUCTION

Tracking and recognition problem for non-human subjects.

TABLE OF CONTENTS

1

OBJECTIVES

Why animal tracking either way? And what is our goal?

Health improvement and anomaly detection

- How much movement each animal has
- Which areas or facilities are visited and when
- Group animals
- Compare data in different time periods

2

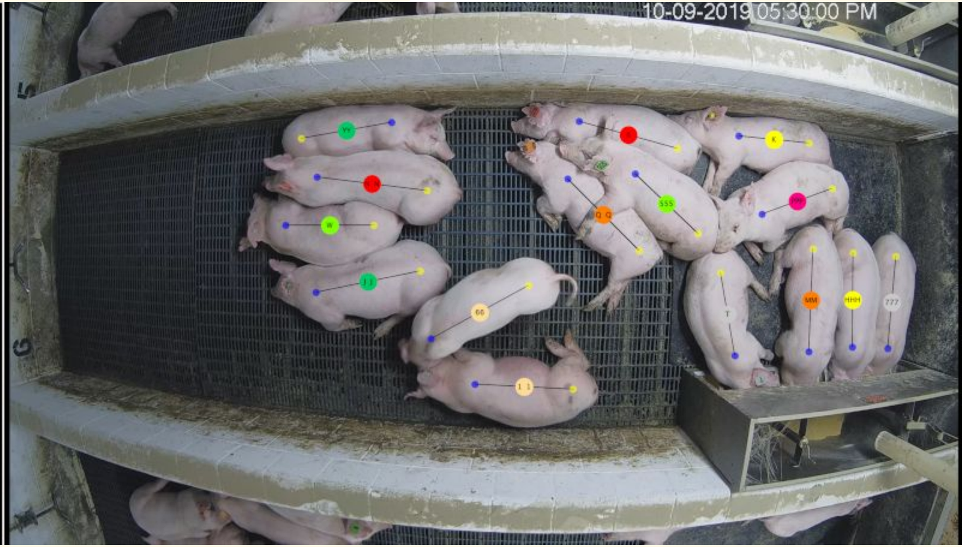
DATA AND MODELS

DATA GATHERING

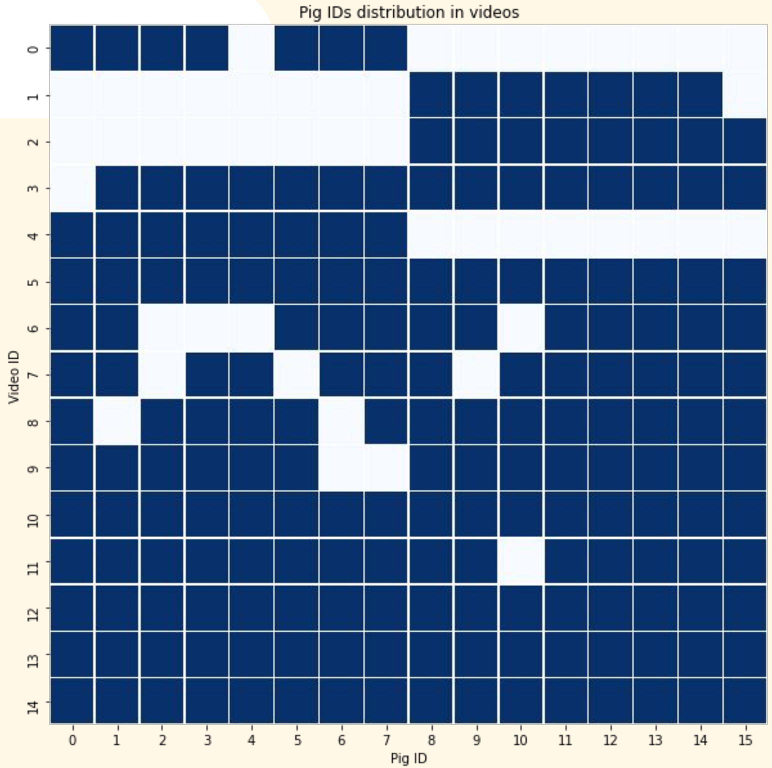
- 15 videos (5fps 1520x2688px)
- 135k frames
- Manually annotated points

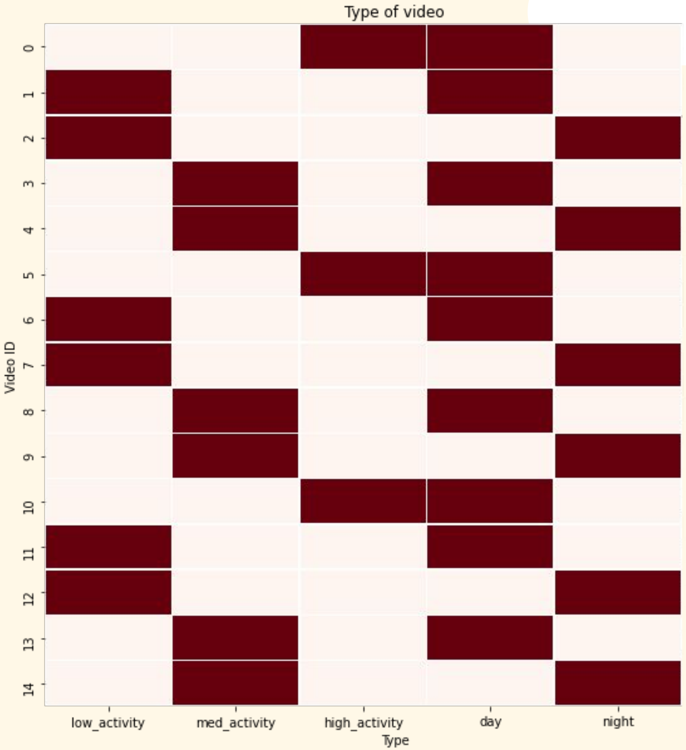
Perceptual Systems Research Group - University of Nebraska.

Frame Examples (with annotations)

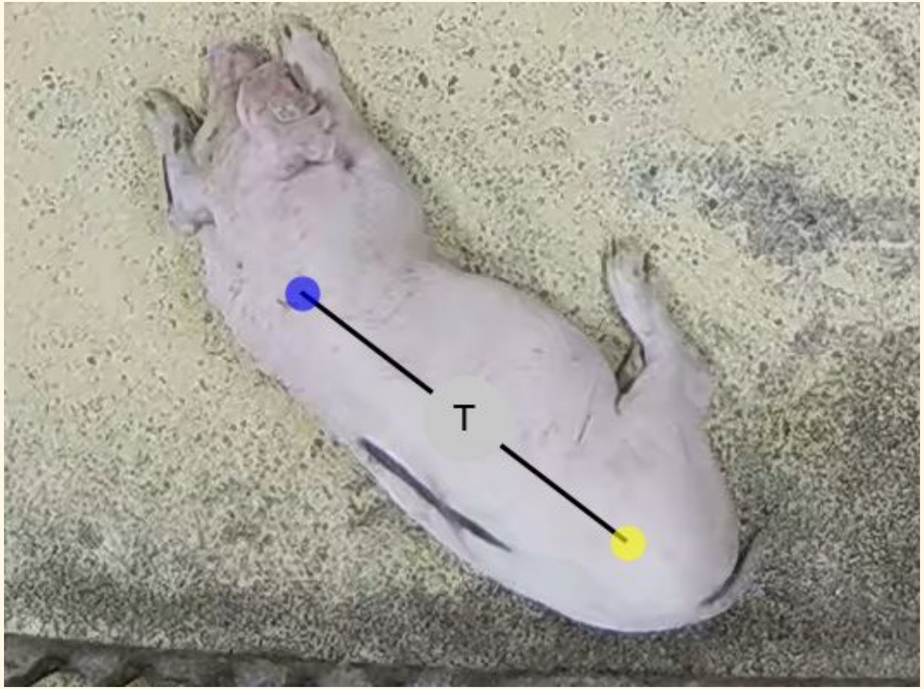


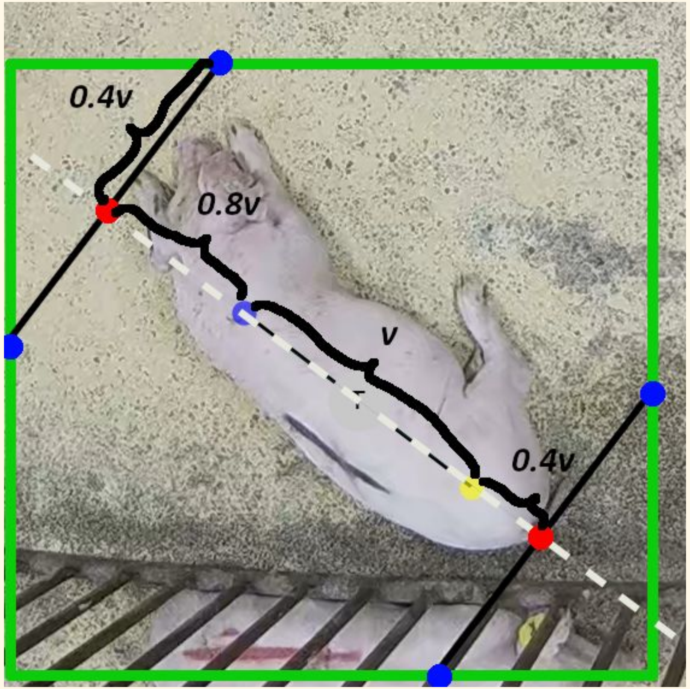
Data distribution





Annotations for detection





Cropping and Augmentations

Modular Approach

Detection

Find out where animals are on a frame and crop them

Recognition

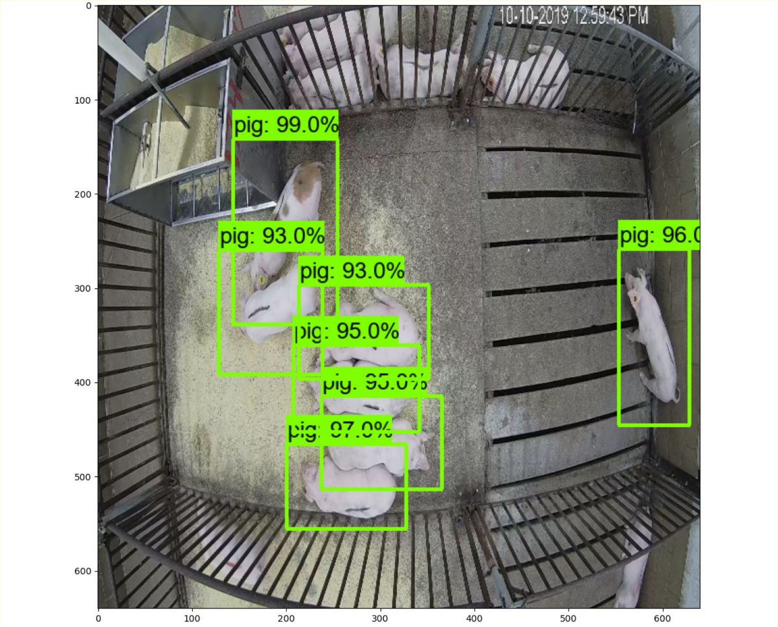
Perform recognition task and assign each subject to category based on appearance

Tracking

Store each category position between frames and generate movement path

DETECTION

SSD ResNet50 FPN

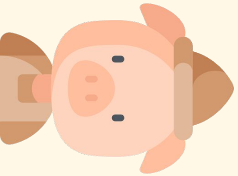


	mAP	mAP_{50}	mAP_{75}
SSD Resnet50 FPN	72.92	97.03	81.82

Table 1: mAP score

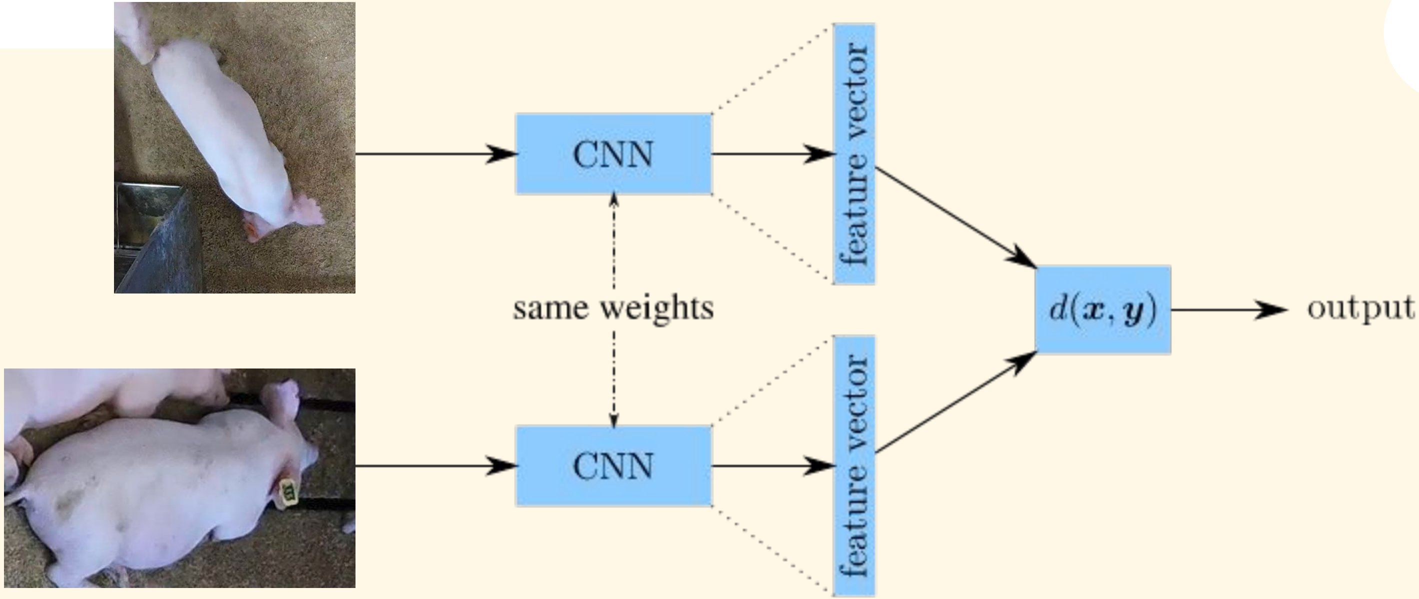
	mAR_1	mAR_{10}	mAR_{100}	mAR_{small}	mAR_{medium}	mAR_{large}
SSD Resnet50 FPN	7.04	60.14	78.73	27.69	69.33	84.63

Table 2: mAR score



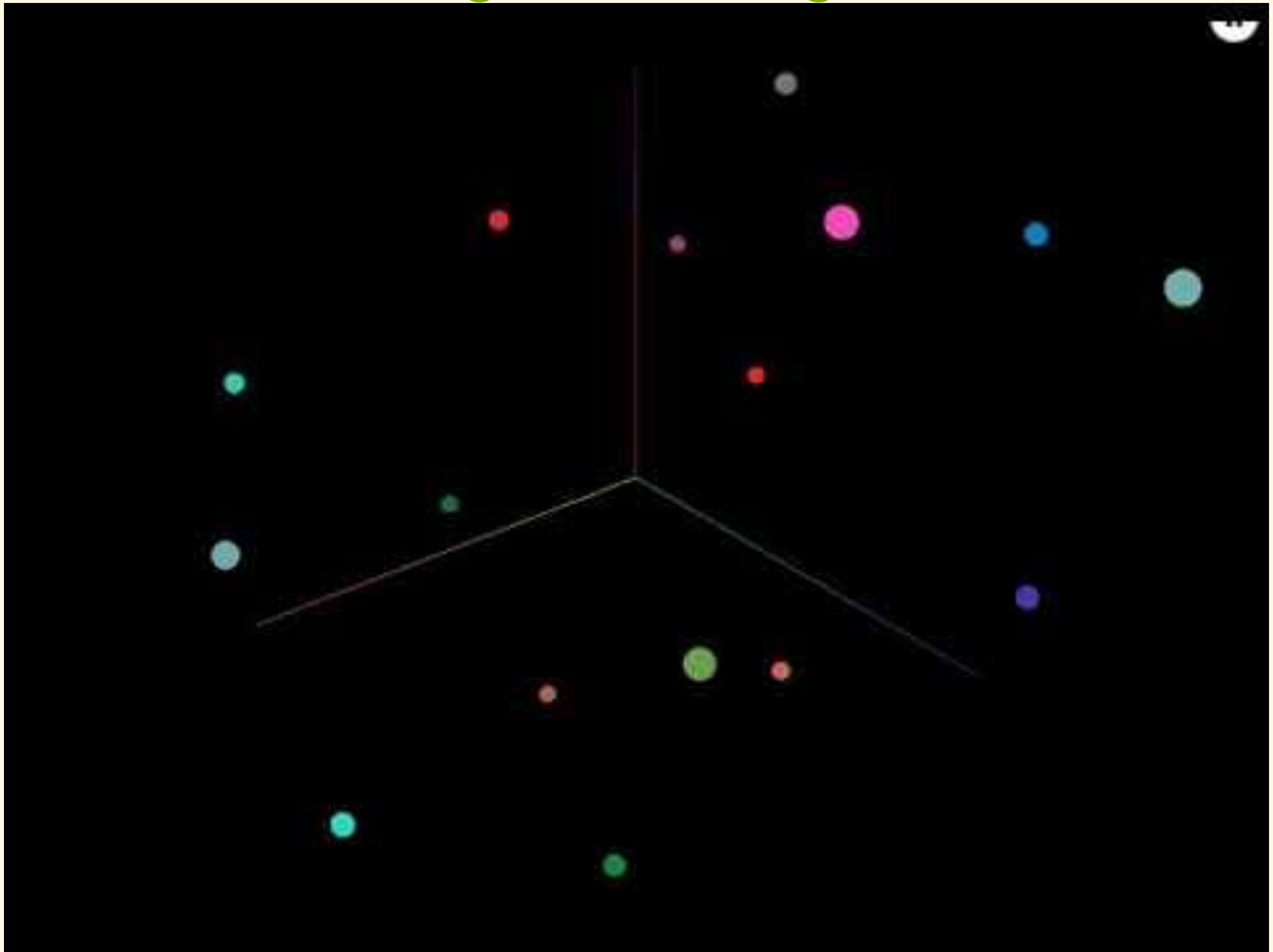
Recognition

Siamese Network

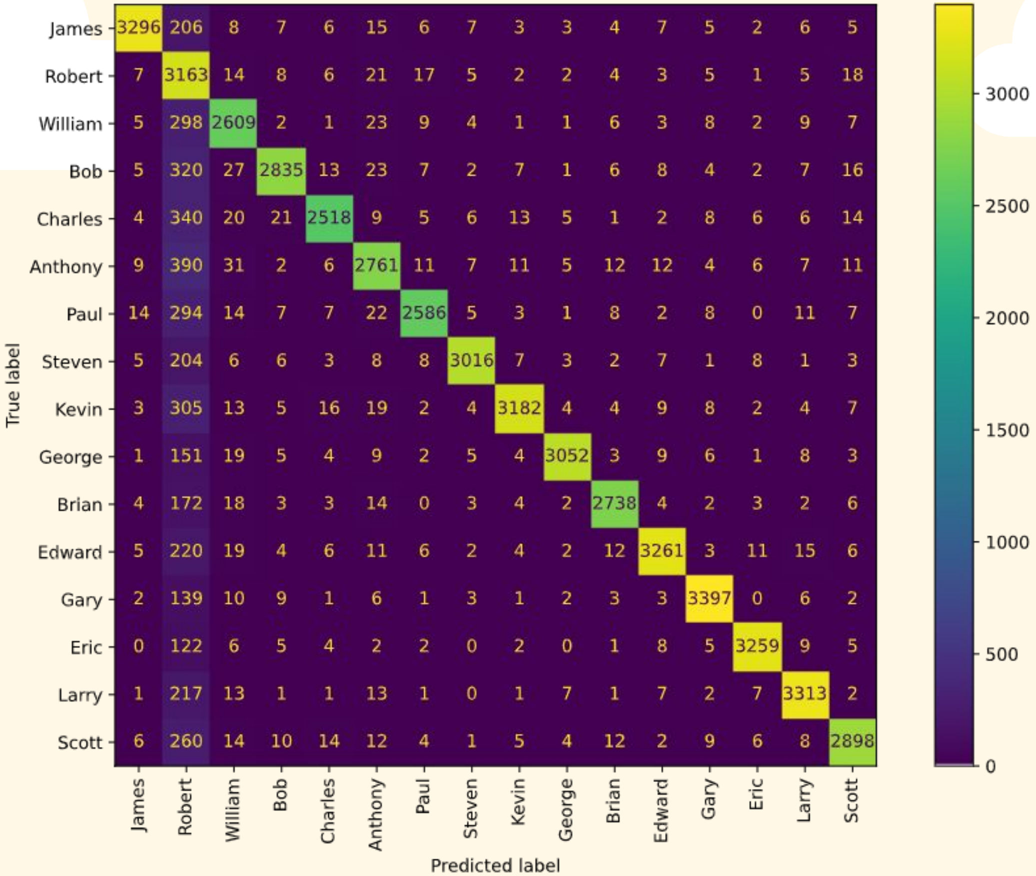


Embedding Space

Avg. Embeddings



Scores



True label
James 3296 206 8 7 6 15 6 7 3 3 4 7 5 2 6 5
Robert 7 3163 14 8 6 21 17 5 2 2 4 3 5 1 5 18
William 5 298 2609 2 1 23 9 4 1 1 6 3 8 2 9 7
Bob 5 320 27 2835 13 23 7 2 7 1 6 8 4 2 7 16
Charles 4 340 20 21 2518 9 5 6 13 5 1 2 8 6 6 14
Anthony 9 390 31 2 6 2761 11 7 11 5 12 12 4 6 7 11
Paul 14 294 14 7 7 22 2586 5 3 1 8 2 8 0 11 7
Steven 5 204 6 6 3 8 8 3016 7 3 2 7 1 8 1 3
Kevin 3 305 13 5 16 19 2 4 3182 4 4 9 8 2 4 7
George 1 151 19 5 4 9 2 5 4 3052 3 9 6 1 8 3
Brian 4 172 18 3 3 14 0 3 4 2 2738 4 2 3 2 6
Edward 5 220 19 4 6 11 6 2 4 2 12 3261 3 11 15 6
Gary 2 139 10 9 1 6 1 3 1 2 3 3 3397 0 6 2
Eric 0 122 6 5 4 2 2 0 2 0 1 8 5 3259 9 5
Larry 1 217 13 1 1 13 1 0 1 7 1 7 2 7 3313 2
Scott 6 260 14 10 14 12 4 1 5 4 12 2 9 6 8 2898
James Robert William Bob Charles Anthony Paul Steven Kevin George Brian Edward Gary Eric Larry Scott
Predicted label
3000
2500
2000
1500
1000
500
0

Scores

	precision	recall	f1-score	support
James	0.98	0.92	0.95	3586
Robert	0.47	0.96	0.63	3281
William	0.92	0.87	0.90	2988
Bob	0.97	0.86	0.91	3283
Charles	0.97	0.85	0.90	2978
Anthony	0.93	0.84	0.88	3285
Paul	0.97	0.87	0.91	2989
Steven	0.98	0.92	0.95	3288
Kevin	0.98	0.89	0.93	3587
George	0.99	0.93	0.96	3282
Brian	0.97	0.92	0.94	2978
Edward	0.97	0.91	0.94	3587
Gary	0.98	0.95	0.96	3585
Eric	0.98	0.95	0.97	3430
Larry	0.97	0.92	0.95	3587
Scott	0.96	0.89	0.92	3265
accuracy			0.90	52979
macro avg	0.94	0.90	0.91	52979
weighted avg	0.94	0.90	0.91	52979

Tracking

Default Tracker

Tracking is based on position of new detections with relation to previous frame

Position similarity

- Euclidean distance
- Detections assigned in order from closest to farthest

Kalman Tracker

Takes into account similarity of position and of appearance. Uses kalman filter for new position prediction

Position similarity

- Mahalanobis distance
- Between predicted distribution and new detection

Appearance similarity

- Embeddings from siamese network
- Cosine distance
- Minimum distance between cropped image of new detection and k images from previous frames

Average Embedding Tracker

Takes into account similarity of position and of appearance. Appearance similarity based on example images supplied before tracking.

Position similarity

- Euclidean distance
- Between previous frame and new detection

Appearance similarity

- Embeddings from siamese network
- Euclidean distance
- Distance between embedding of cropped image of new detection and average of embeddings of example images

Tracking process

Detect objects
SSD ResNet50 FPN

Crop images
Use bboxes to extract animal images

Recognize objects
Generate embeddings and use them to recognize classes

Store path
Store path and make corrections if necessary

3

RESULTS ANALYSIS

Evaluation

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

Full Tracking

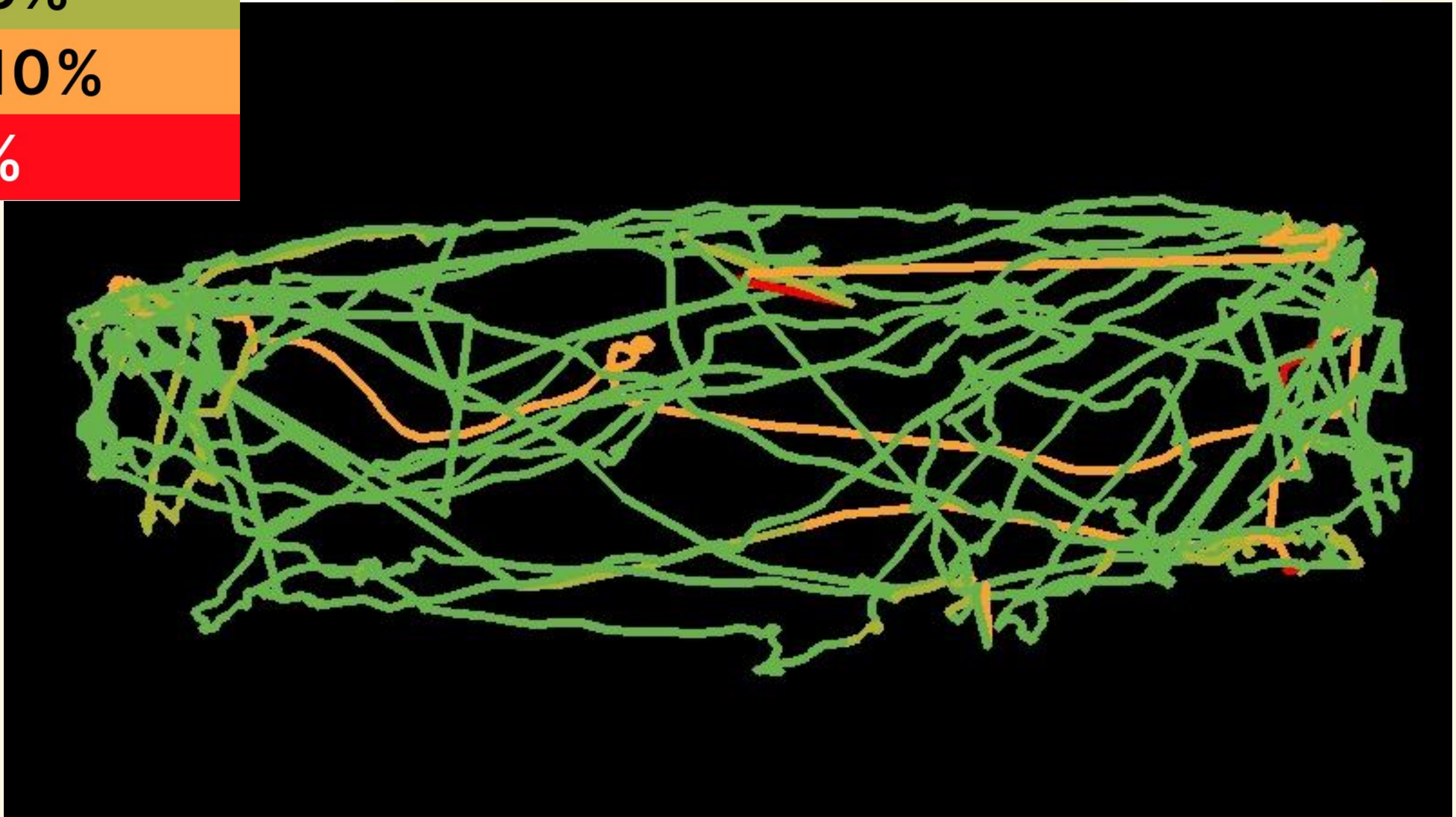
Keep one track even if class is not correct

Partial Tracking

Evaluate path in intervals (some number of frames), allows to evaluate initial mistakes.

Track Scores - video 11 (Eric)

<2%
2%-5%
5%-10%
>10%



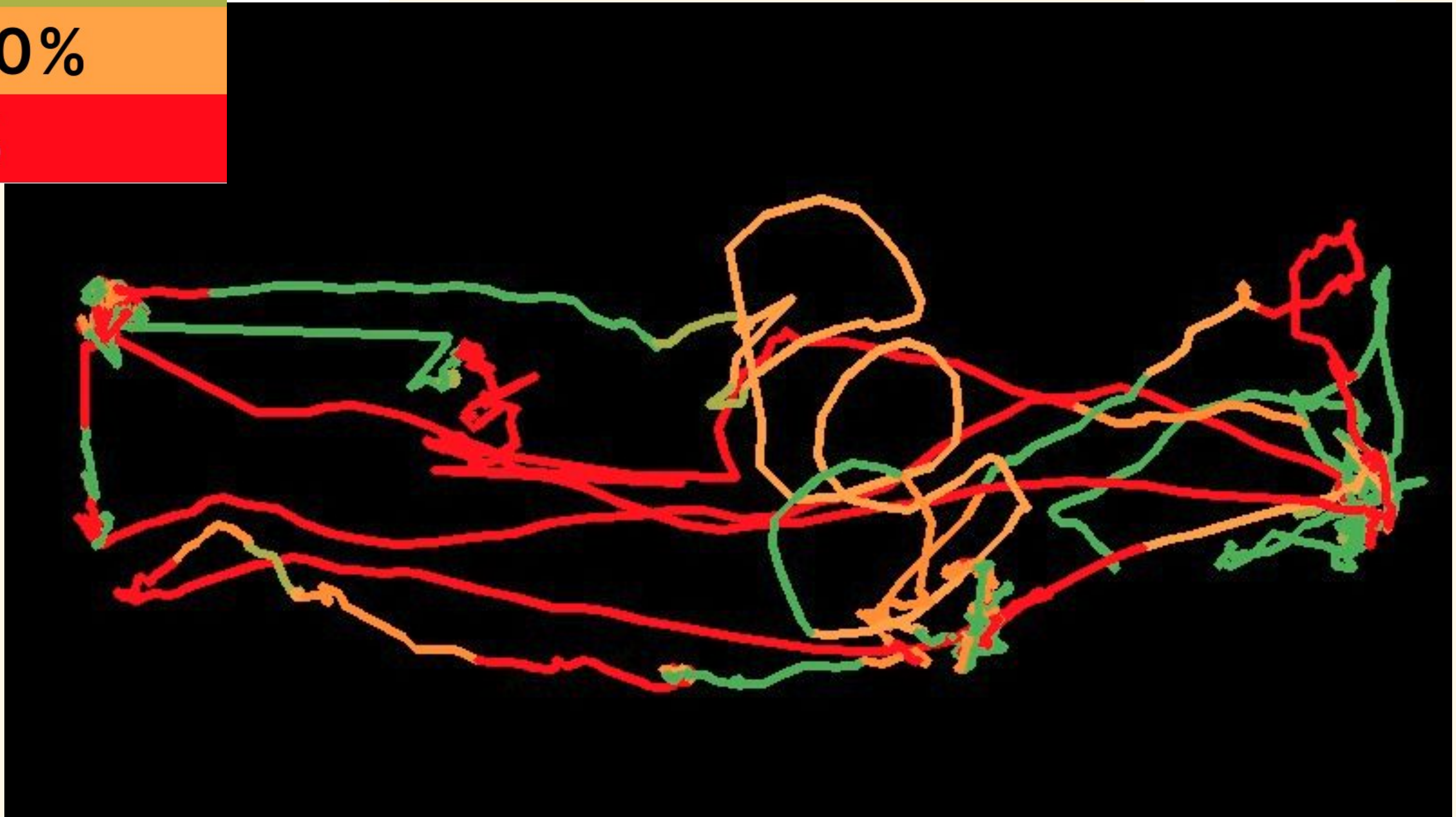
<2%

2%-5%

5%-10%

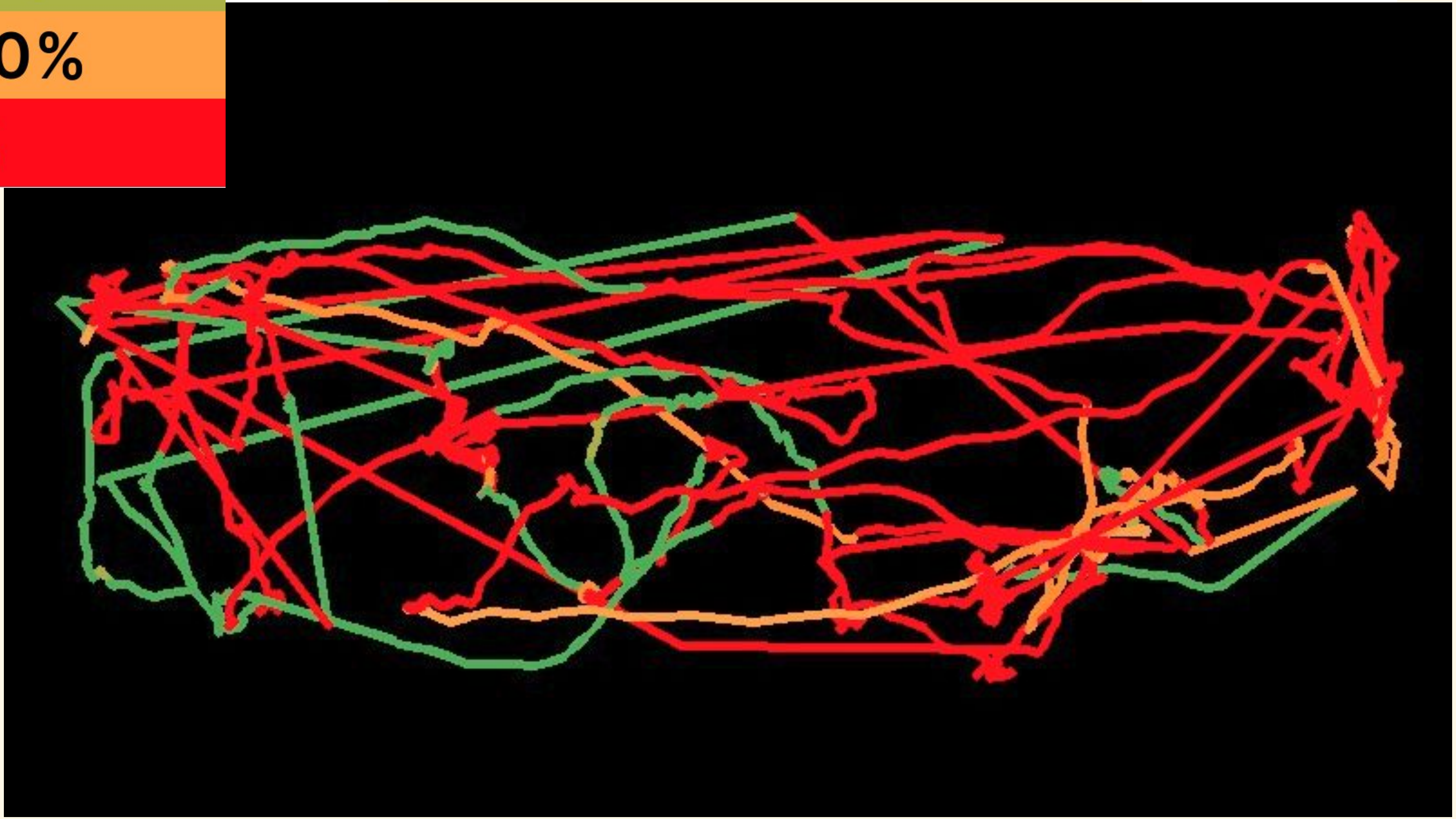
>10%

Track Scores - video 11 (Anthony)



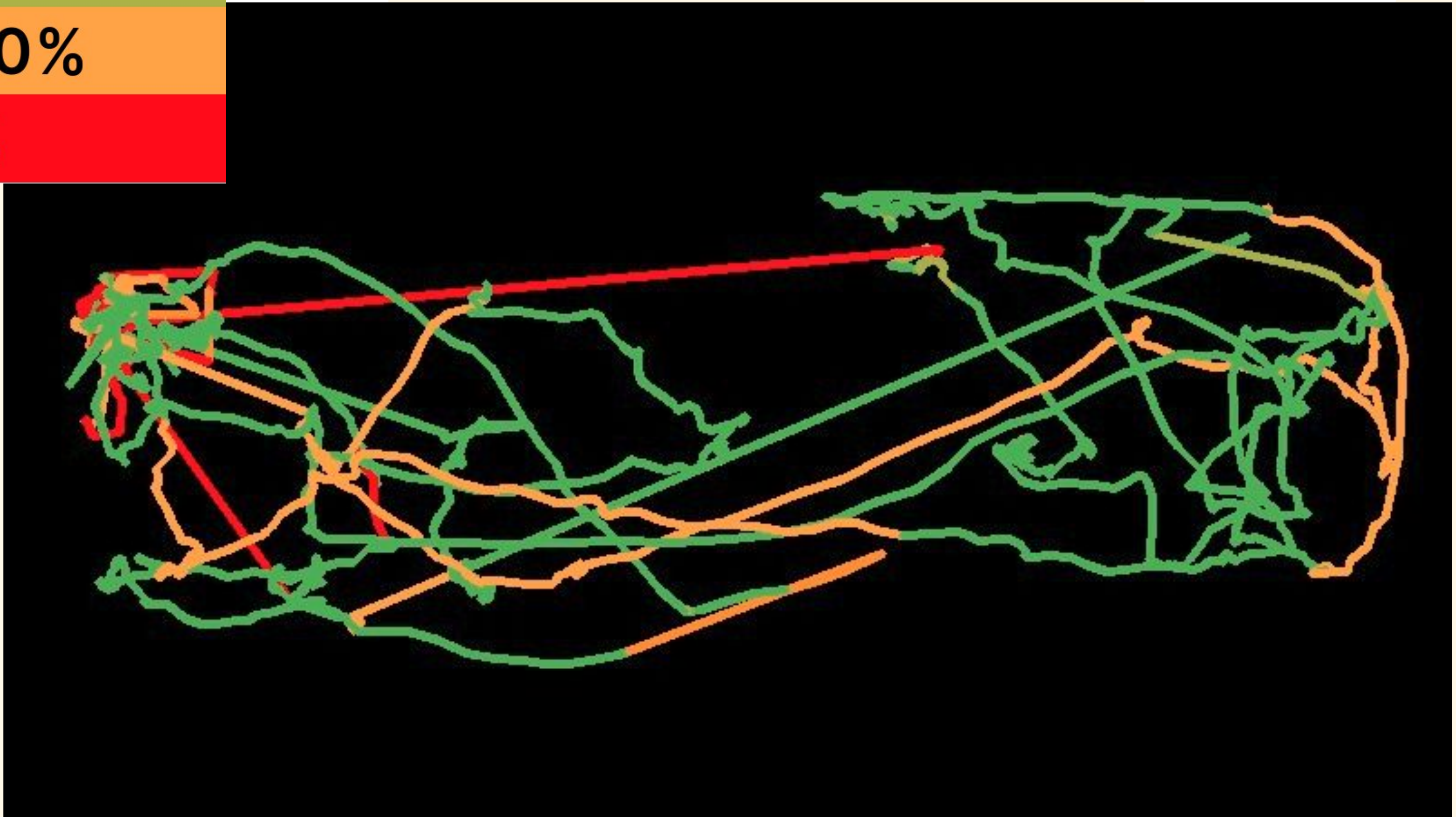
Track Scores - video 11 (George)

<2%
2%-5%
5%-10%
>10%



Track Scores - video 11 (Robert)

<2%
2%-5%
5%-10%
>10%



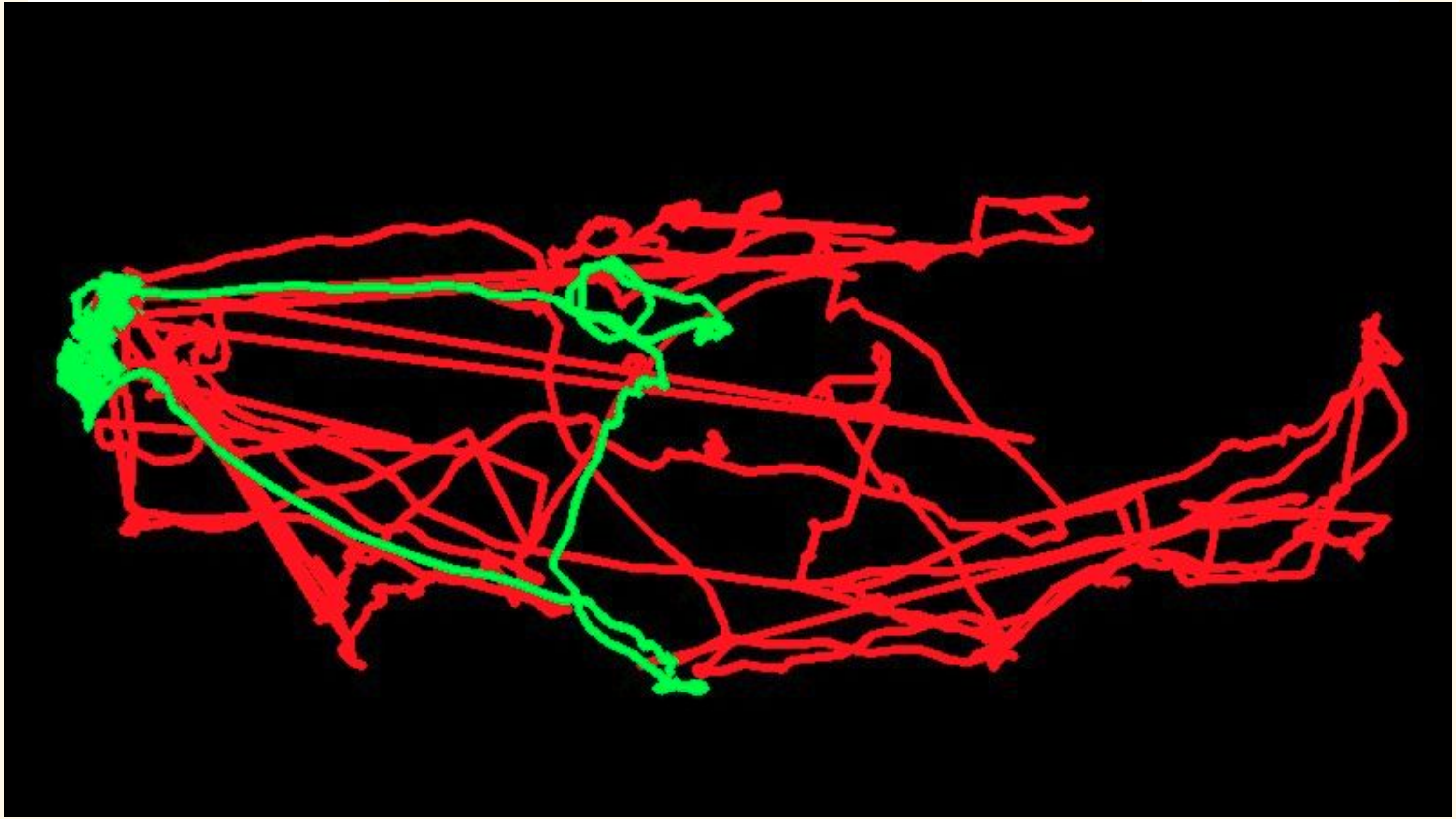
Partial Tracking

Class	MSE
Eric	1.77
Edward	1.85
Bob	2.46
Paul	2.59
Gary	3.20
Larry	3.27
Scott	3.43
Kevin	4.65
Robert	5.98
Brian	6.87
William	7.75
Charles	10.22
Steven	15.94
James	19.18
Anthony	19.34
George	39.90

Full Tracking

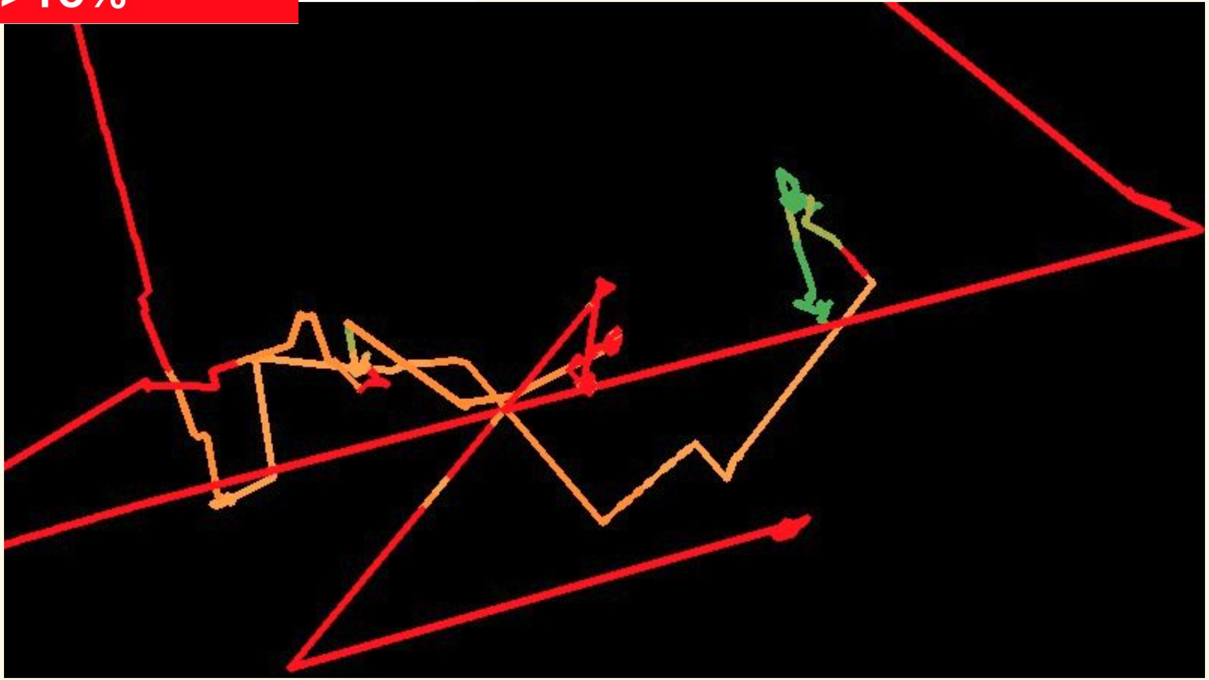
Class	MSE
Edward	19.02
Paul	30.55
Bob	31.92
Steven	32.68
Larry	33.36
George	33.74
Gary	34.44
Kevin	35.46
William	35.50
Robert	36.77
Charles	37.10
Scott	38.61
Anthony	40.16
Brian	44.02
James	51.79
Eric	51.91

Full Tracking issue



<2%
2%-5%
5%-10%
>10%

Tracking with Kalman



Full Tracking

Class	MSE
Bob	5.48
Scott	27.89
George	28.27
Edward	32.80
Paul	32.84
Kevin	33.16
Steven	34.96
Charles	36.24
Anthony	36.79
Larry	38.04
William	38.68
Robert	38.75
Brian	48.99
Eric	50.07
Gary	71.77
James	604.48

4 CONCLUSIONS

Siamese Networks are hard to train

Even harder to select feature extractor for them

Tracking position is different from the bbox center

Augmentation not always helps

Sometimes augmented data introduces unnecessary noise

Evaluate base on the problem

Define what exactly is important not what is a popular measurement

Hard to find error in a very large dataset

Make staff modular

It's easier to evaluate and use when it's made of blocks

BIBLIOGRAPHICAL REFERENCES

- [1] Navaneeth Bodla et al. Soft-NMS – Improving Object Detection With One Line of Code. 2017. arXiv: 1704.04503 [cs.CV].
- [2] Kaiming He et al. Deep Residual Learning for Image Recognition. 2015. arXiv: 1512.03385 [cs.CV].
- [3] Jonathan Huang et al. Speed/accuracy trade-offs for modern convolutional object detectors. 2017. arXiv: 1611.10012 [cs.CV].
- [4] Gregory R. Koch. “Siamese Neural Networks for One-Shot Image Recognition”. In: (2015).
- [5] T. Lin et al. “Feature Pyramid Networks for Object Detection”. In: (July 2017), pp. 936–944. Issn: 1063-6919. doi: 10.1109/CVPR.2017.106.
- [6] Tsung-Yi Lin et al. Microsoft COCO: Common Objects in Context. 2020.
- [7] Leland McInnes, John Healy, and James Melville. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. 2020. arXiv: 1802.03426
- [8] Y. Pang et al. “Efficient Featurized Image Pyramid Network for Single Shot Detector”. In: (2019), pp. 7328–7336. doi: 10 . 1109 / CVPR . 2019 . 00751.
- [9] Perceptual Systems Research Group - University of Nebraska. url: http://psrg.unl.edu/Projects/Details/12-Animal-Tracking
- [10] Florian Schroff, D. Kalenichenko, and J. Philbin. “FaceNet: A unified embedding for face recognition and clustering”. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)(2015), pp. 815–823.

THANKS!

Do you have any questions?

Kemal Erdem
Marek Pokropiński