Summer Workshop 2020

Deep Learning to Count and Track Strawberries

Raymond Kirk, Grzegorz Cielniak

L-CAS

NIAB EMR

Berry Gardens

UNIVERSITY OF LINCOLN

Background: Why do we need to count?

Problem

Predicting yield has a lot of value, one UK grower group lost £1M in two weeks

Predicting timing and yield of strawberries extremely difficult; affected by variety and environment

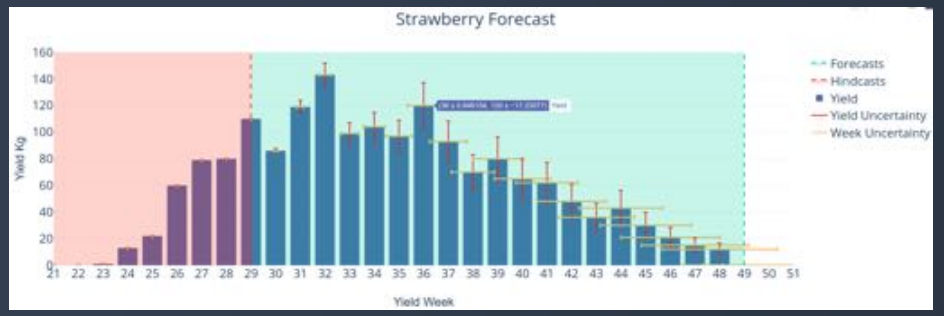
Highly perishable crop, can't be stored for long periods

Methods

Currently done by using a combination of:

Historical crop profiles, flower counts, meteorological data and 'sense checking' by field supervisor

These methods are inaccurate: only 7 / 25 growers got within 15% from a UK grower group



Background: PhD Research to Date

Adaptation from people: Bayesian Tracker

Problem Statements

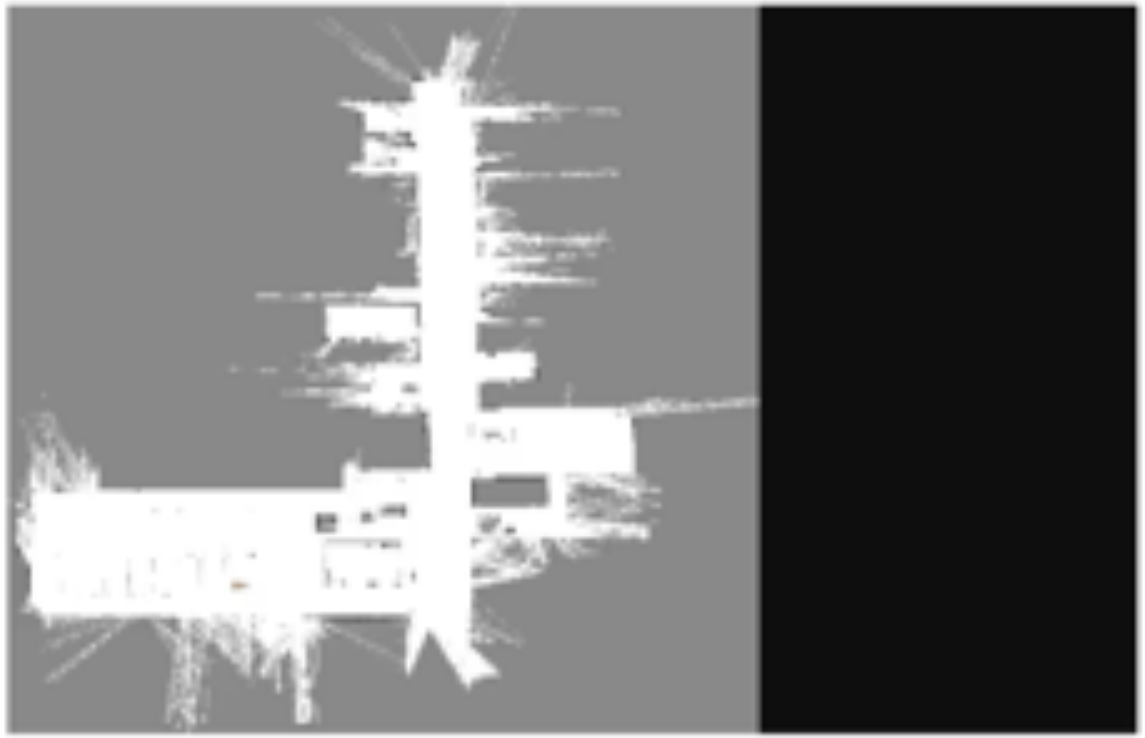
Are detections (positional information) enough to track?

How can we deal with overlapping/heavy occlusion/high density cases?

Is the Bayesian tracker suitable to track small heavily overlapping strawberries?

Ripe Strawberry Count: 9 (7)

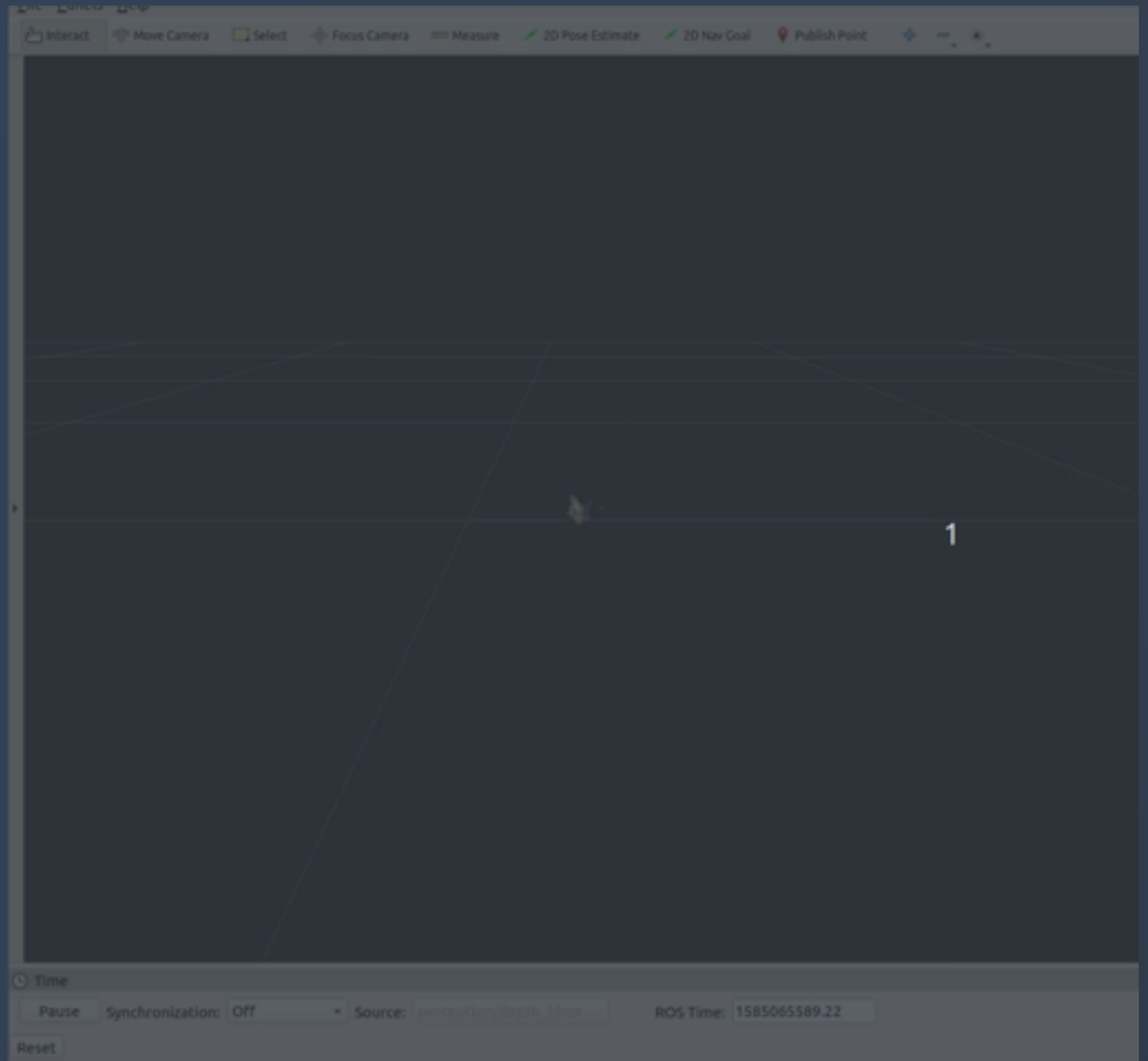
Map



Camera

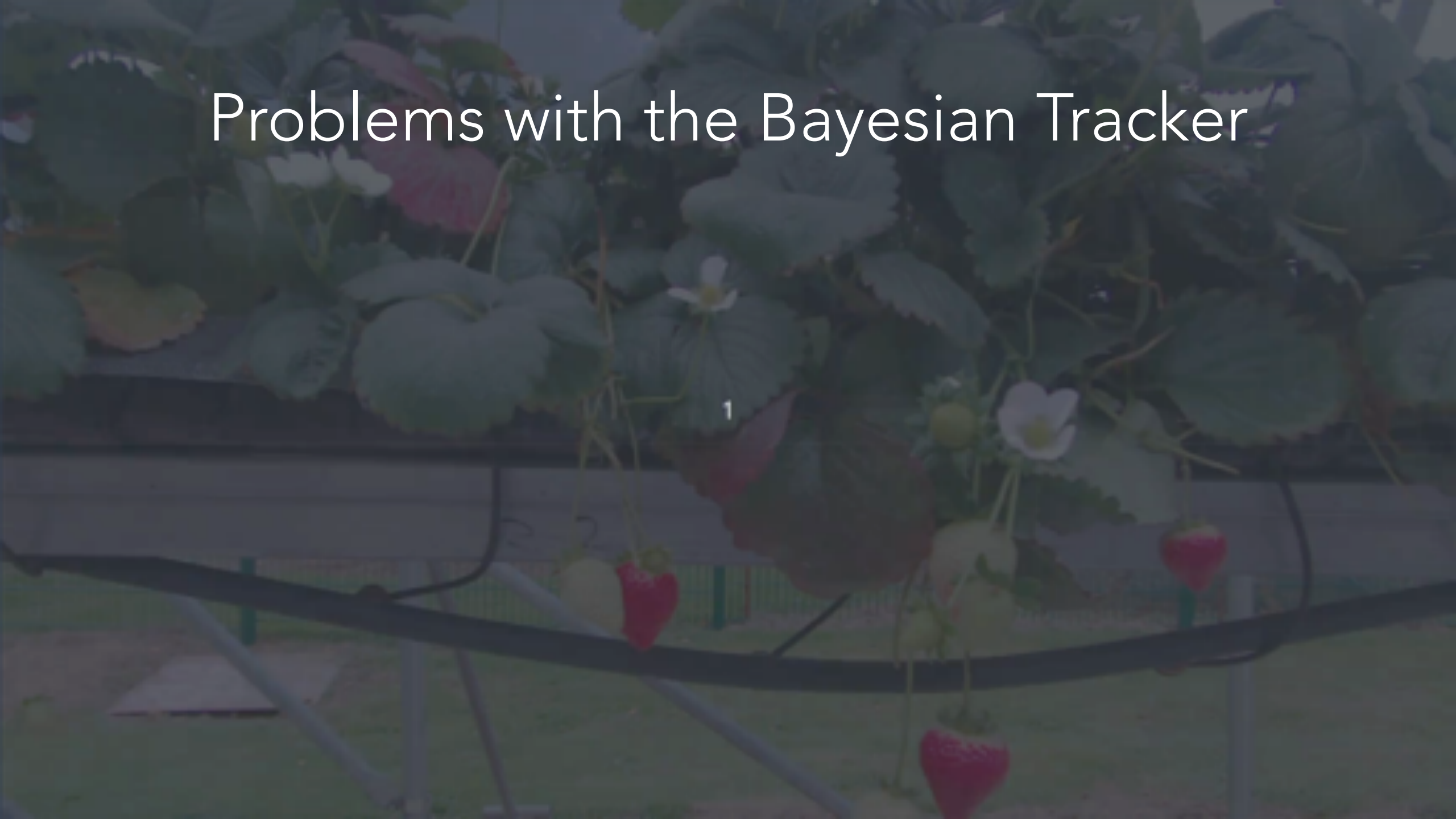
BBC Countryfile – Screen Recording of Strawberry Counting Bayesian (70 Actual Berries)

Bayesian Tracker Formulation

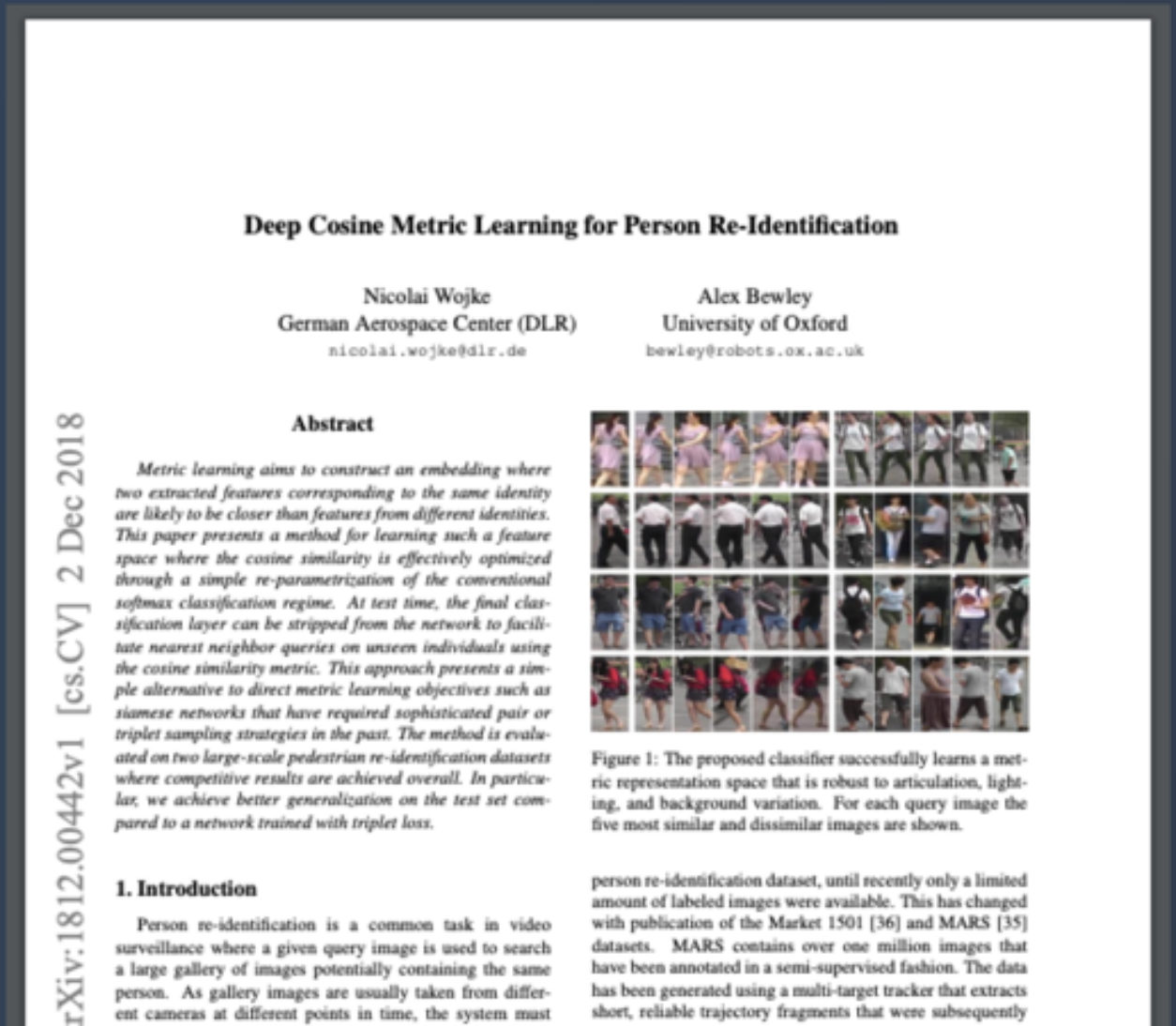


1. Kalman filters with state space (x, v_x, y, v_y, z, v_z) (EKF, UKF, Particle filter) of detections and tracks.
2. Nearest neighbour data association matrix where $\mathrm{n}(i, j)$ contains the pair with the best similarity measure from distance metric $d(i, j)$ of Kalman predictions and current detections.
3. Data are considered candidates if they fall within gating region and 95% chi-squared distribution. Otherwise new-tracks are created from unmatched sequences from the data association step falling in this region.

Problems with the Bayesian Tracker



Tracking through Re-Identification



Deep Cosine Metric Learning for Person Re-Identification

Nicolai Wojke
German Aerospace Center (DLR)
nicolai.wojke@dlr.de

Alex Bewley
University of Oxford
bewley@robots.ox.ac.uk

Abstract

Metric learning aims to construct an embedding where two extracted features corresponding to the same identity are likely to be closer than features from different identities. This paper presents a method for learning such a feature space where the cosine similarity is effectively optimized through a simple re-parametrization of the conventional softmax classification regime. At test time, the final classification layer can be stripped from the network to facilitate nearest neighbor queries on unseen individuals using the cosine similarity metric. This approach presents a simple alternative to direct metric learning objectives such as siamese networks that have required sophisticated pair or triplet sampling strategies in the past. The method is evaluated on two large-scale pedestrian re-identification datasets where competitive results are achieved overall. In particular, we achieve better generalization on the test set compared to a network trained with triplet loss.

1. Introduction

Person re-identification is a common task in video surveillance where a given query image is used to search a large gallery of images potentially containing the same person. As gallery images are usually taken from different cameras at different points in time, the system must

Figure 1: The proposed classifier successfully learns a metric representation space that is robust to articulation, lighting, and background variation. For each query image the five most similar and dissimilar images are shown.

person re-identification dataset, until recently only a limited amount of labeled images were available. This has changed with publication of the Market 1501 [36] and MARS [35] datasets. MARS contains over one million images that have been annotated in a semi-supervised fashion. The data has been generated using a multi-target tracker that extracts short, reliable trajectory fragments that were subsequently

To fix these cases we are attempting to learn unique re-identifiable features for berries over multiple view points.

We can then formulate our cost matrix for associating tracks to detections as:

$$c(i,j) = \lambda d_1(i,j) + (1 - \lambda) d_2$$

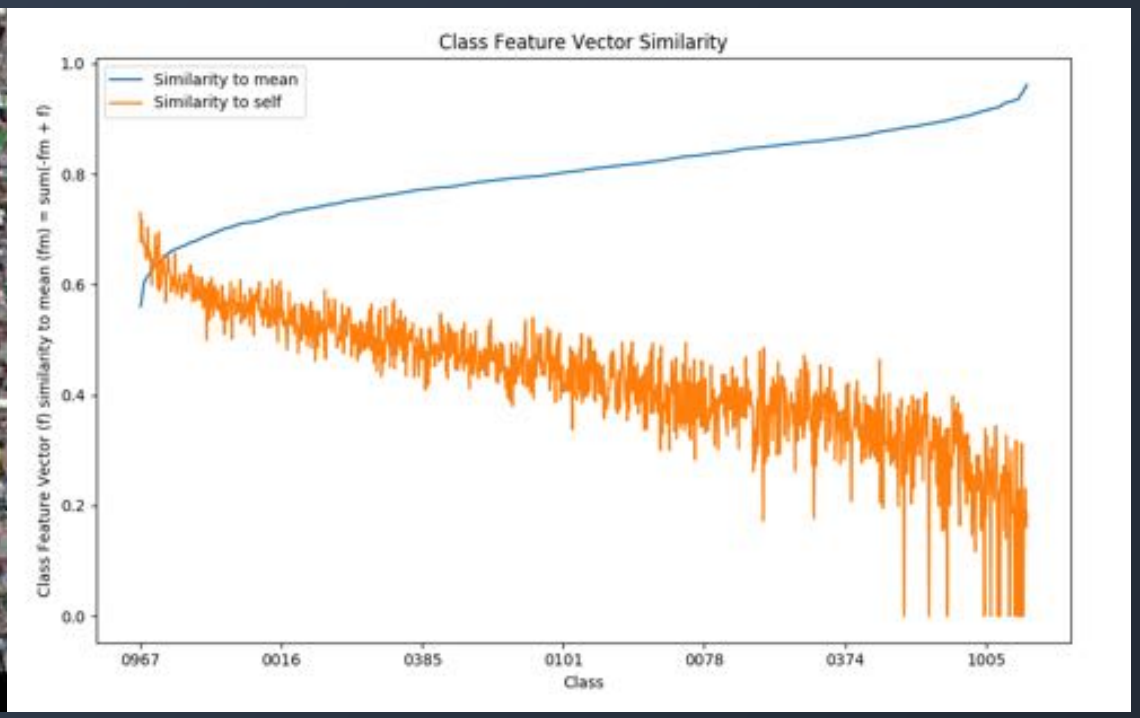
d_1 is the distance between predicted Kalman states and current detections.

d_2 is the cosine similarity between the last i^{th} track re-identification feature vectors and j^{th} detection reid features.

Re-Identification Feature Vectors for People

Simple Idea: Use deep learning to extract unique features individual detections.

Instead of just tracking using image co-ordinates (x, y) track also using a intra-class discriminatory feature vector. Good preliminary results on discriminating people even with similar appearance.



DeepSort Formulation

Listing 1 Matching Cascade

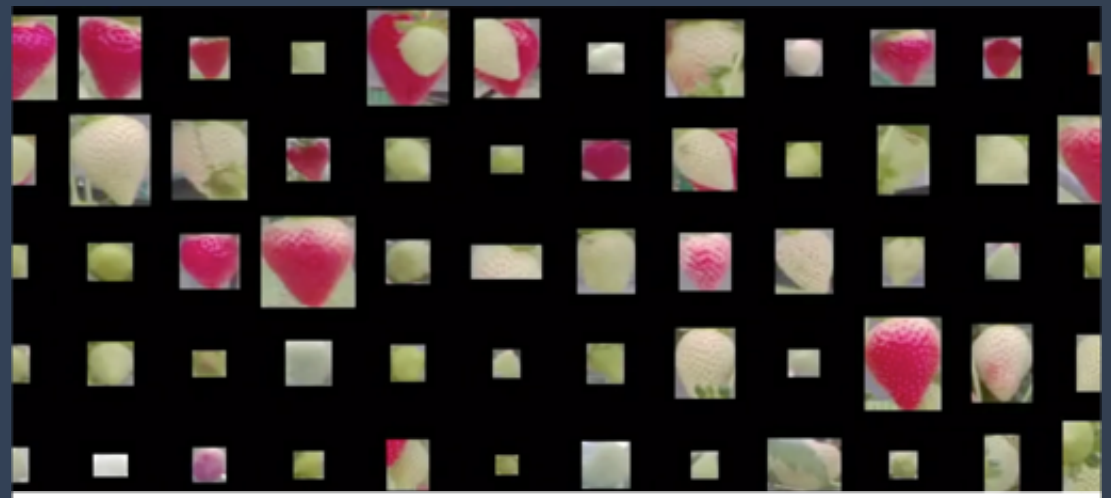
Input: Track indices $\mathcal{T} = \{1, \ldots, N\}$, Detection indices $\mathcal{D} = \{1, \ldots, M\}$, Maximum age $A_{\max}$

Compute cost matrix $\boldsymbol{C} = [c_{i,j}]$ using Eq. 5
Compute gate matrix $\boldsymbol{B} = [b_{i,j}]$ using Eq. 6
Initialize set of matches $\mathcal{M} \leftarrow \emptyset$
Initialize set of unmatched detections $\mathcal{U} \leftarrow \mathcal{D}$
for $n \in \{1, \ldots, A_{\max}\}$ **do**
  Select tracks by age $\mathcal{T}_n \leftarrow \{i \in \mathcal{T} \mid a_i = n\}$
  $[x_{i,j}] \leftarrow \text{min\_cost\_matching}(\boldsymbol{C}, \mathcal{T}_n, \mathcal{U})$
  $\mathcal{M} \leftarrow \mathcal{M} \cup \{(i,j) \mid b_{i,j} \cdot x_{i,j} > 0\}$
  $\mathcal{U} \leftarrow \mathcal{U} \setminus \{j \mid \sum_i b_{i,j} \cdot x_{i,j} > 0\}$
end for
return $\mathcal{M}, \mathcal{U}$

Eq 5. $c(i,j) = \lambda d_1(i,j) + (1 - \lambda) d_2$

1. Kalman filters with state space $(x, v_x, y, v_y, a, v_a, h, v_h)$ of detections and tracks.
2. Association matrix where $\mathrm{n}(i,j)$ contains the pair with the best similarity measure from distance metrics of Kalman predictions and current detections.
3. Data are considered candidates if they fall within gating region and 95% chi-squared distribution of gated eq.5 Otherwise new-tracks are created from unmatched sequences from the data association step falling in this region and IoU matching cascade.

Re-Identification Network



Name	Patch Size/Stride	Output Size
Conv 1	$3 \times 3/1$	$32 \times 128 \times 64$
Conv 2	$3 \times 3/1$	$32 \times 128 \times 64$
Max Pool 3	$3 \times 3/2$	$32 \times 64 \times 32$
Residual 4	$3 \times 3/1$	$32 \times 64 \times 32$
Residual 5	$3 \times 3/1$	$32 \times 64 \times 32$
Residual 6	$3 \times 3/2$	$64 \times 32 \times 16$
Residual 7	$3 \times 3/1$	$64 \times 32 \times 16$
Residual 8	$3 \times 3/2$	$128 \times 16 \times 8$
Residual 9	$3 \times 3/1$	$128 \times 16 \times 8$
Dense 10		128
Batch and ℓ_2 normalization		128

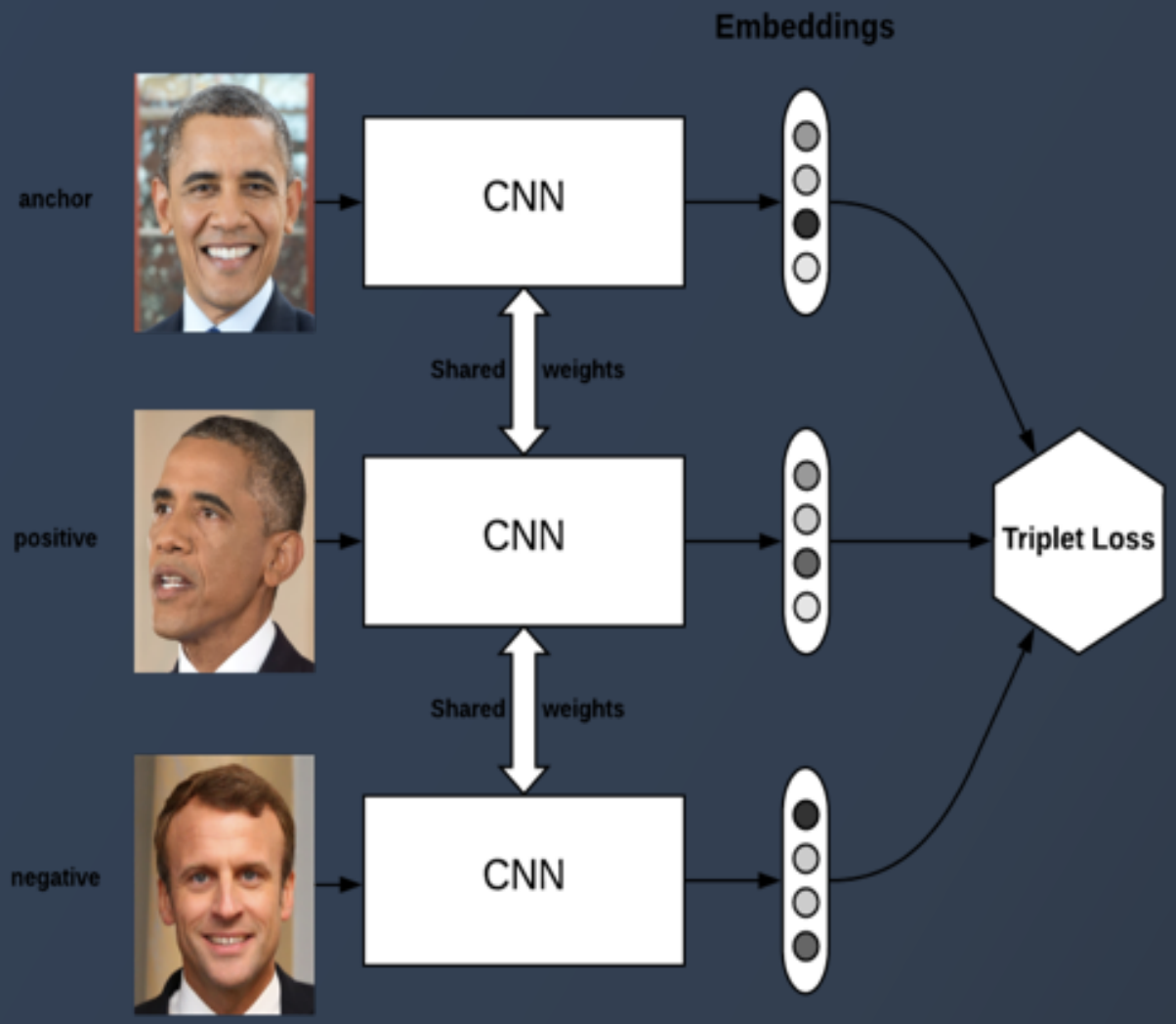
We utilise the Bayesian 3D tracker to generate ID data for strawberries and prune the generated tracks to only allow strong tracks

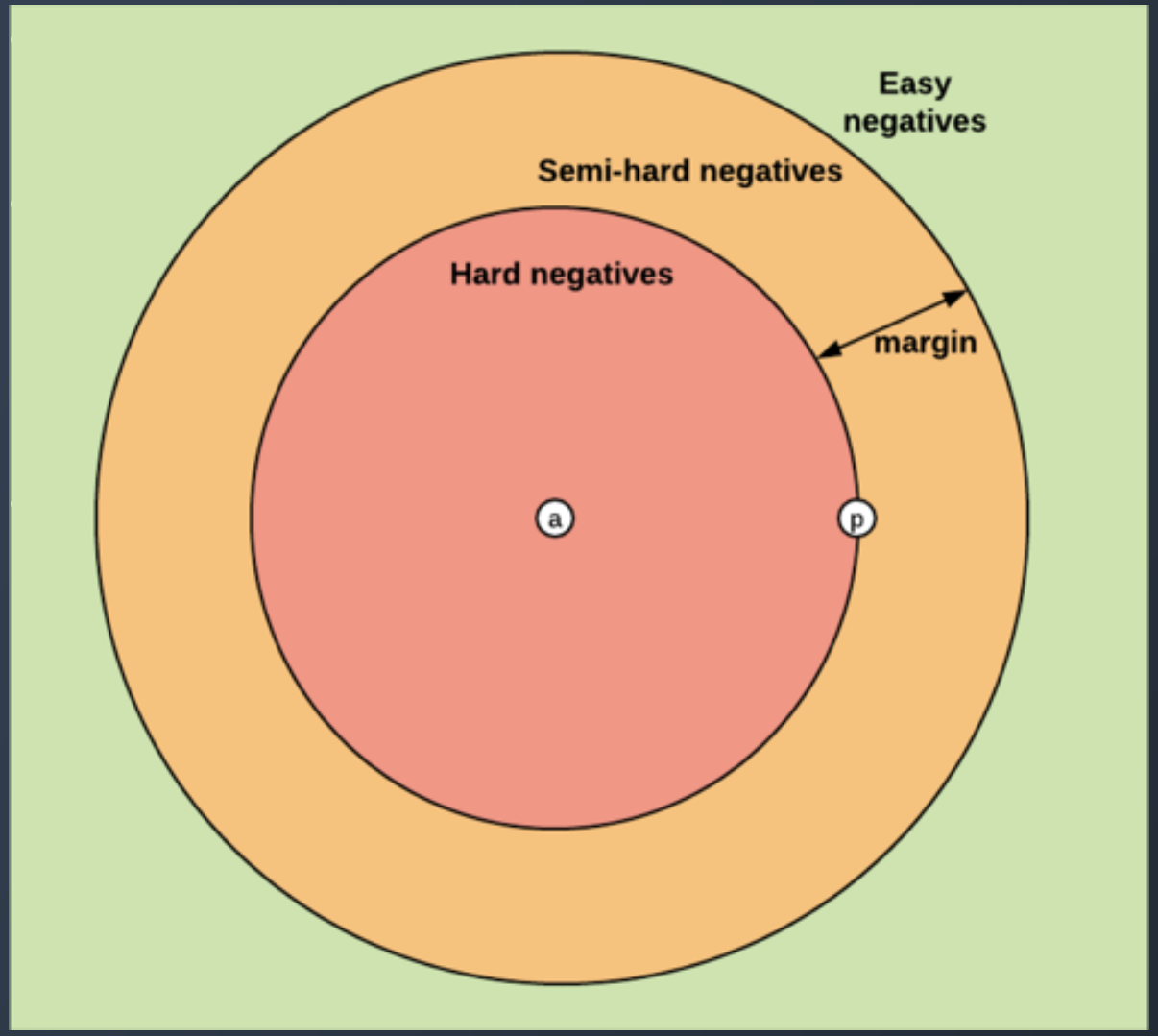
We train the following network as a unsupervised classification problem to reidentify each berry.

97.63% Reidentification Rate on validation sets containing unseen views of each of the 270 different berries.

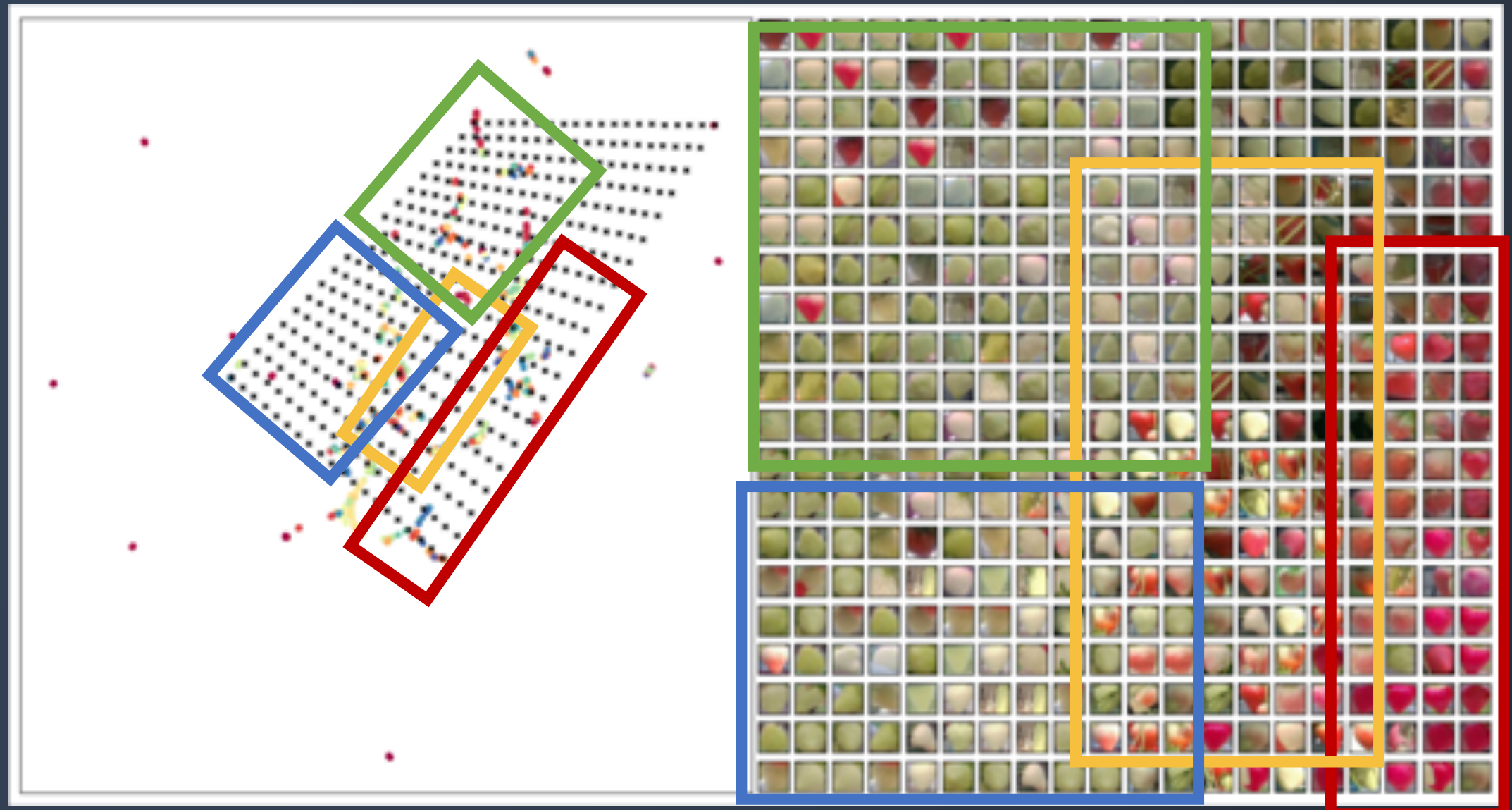
Re-Identification Feature Vectors for Berries

Learning more separable features: Triplet Loss





Re-Identification Feature Vectors for Berries



Open Questions/Future Work

Questions?

Contact: rkirk@lincoln.ac.uk

Dataset Availability, Access and Evaluation Methodology

Summer Vision Workshop 2020

Afternoon Session

Strawberry Dataset Availability

Name	Object Detection	Semantic Segmentation	Instance Segmentation	Key Point Detection	Registered Depth	Multi-modal (IR, UV etc)	Multiple viewpoints
China Complex				*			
China Simple				*			
Riseholme 2018 Simple				*			
Riseholme 2018				*			
Norway 2018				*			
Riseholme Pico Multi-View				*			
Riseholme 2019				*			

Challenge paper to be published late 2020 posing these datasets for strawberry detectors (Raymond Kirk, Adrian Salazar).

* Key points provided as approximations of bounding box/segmentation centroids.

Dataset Access Future and Current

All of the current datasets can be obtained from myself (rkirk@lincoln.ac.uk) in a compressed format containing the images with pre-computed all, training and testing splits in COCO format.

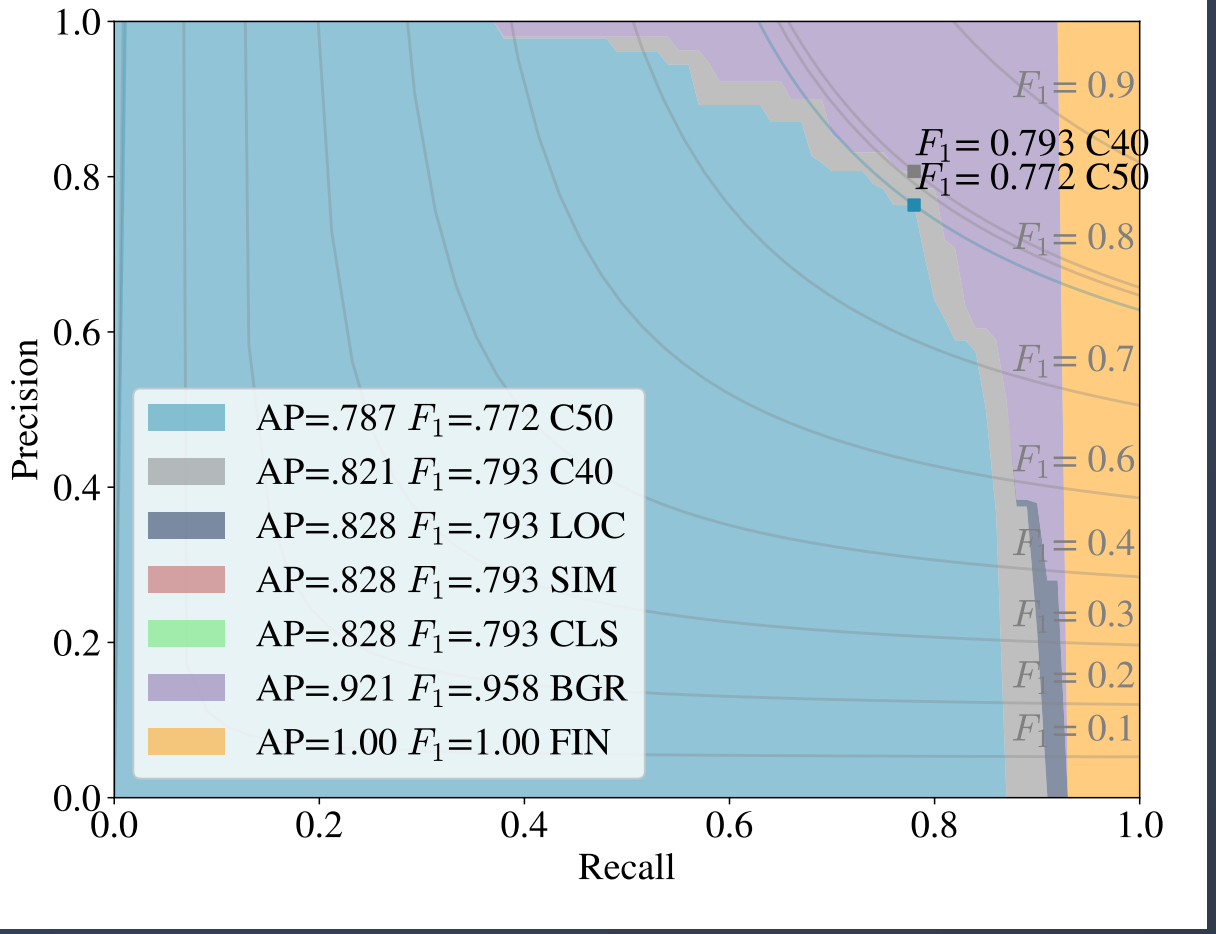
We are in the process of porting everything over to the L-CAS database so the data is centralised and accessed on a per-user basis (also contains localisation and environment info captured from Thorvald).

COCO Format and Evaluation

We use the standard pipeline for evaluating object detections, semantic segmentations and instance segmentation networks improved on from Pascal VOC in the MS COCO dataset challenge.

```
Average Precision (AP):
  AP                % AP at IoU=.50:.05:.95 (primary challenge metric)
  AP^IoU=.50        % AP at IoU=.50 (PASCAL VOC metric)
  AP^IoU=.75        % AP at IoU=.75 (strict metric)
AP Across Scales:
  AP^small          % AP for small objects: area < 32^2
  AP^medium         % AP for medium objects: 32^2 < area < 96^2
  AP^large          % AP for large objects: area > 96^2
Average Recall (AR):
  AR^max=1          % AR given 1 detection per image
  AR^max=10         % AR given 10 detections per image
  AR^max=100        % AR given 100 detections per image
AR Across Scales:
  AR^small          % AR for small objects: area < 32^2
  AR^medium         % AR for medium objects: 32^2 < area < 96^2
  AR^large          % AR for large objects: area > 96^2
```

Evaluation Error Analysis



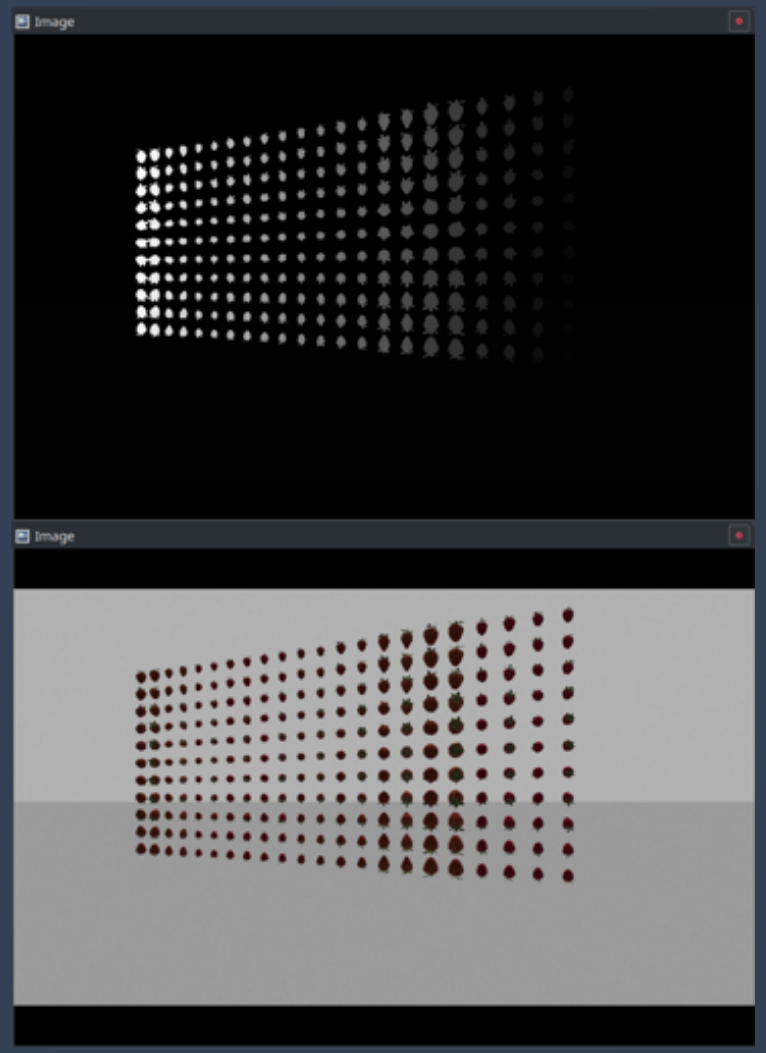
We utilise the ‘Diagnose Error in Object Detectors’ approach analysis by Derek Hoiem.

False positive analysis clearly showing the error introduced from localisation failures and mis-classification intra and inter-class and background.

Quick Summary of Datasets by other members in L-CAS

Simulated Berries

(Nikolaus Wagner nwagner@lincoln.ac.uk)



RGB image with corresponding segmentation map

Annotated renders of simulated strawberries

Auto-annotated from Gazebo simulation

Contains:

RGB- & registered depth-images
Full image (per-object) segmentation masks
Per-object bounding boxes
Per-object orientation annotation (rel. to camera)
Per-object COCO-segmentation annotation
Unique object IDs used in annotations

Generated for training orientation estimator
→ expandable on demand

Standard COCO format with extra fields for depth- & segmentation-maps and orientation

Use case: Trying out algorithms where real data is not available / hard to collect

Synthetic Data, Real World

Justin Le Louedec (jlelouedec@lincoln.ac.uk)

High quality strawberries replacing captured one in point clouds

Thousands of examples (minimum as much as data collected)

Segmentation, orientation, better shape information

~100 good quality strawberries, but need to process them manually

Based from Riseholme 2018 data.

Synthetic Data, Simulation

Justin Le Louedec (jlelouedec@lincoln.ac.uk)

THE SUMMARY OF DATASETS COLLECTED.

sensor	stereo	ToF	simulation
# point clouds	64	139	134
resolution	1280x720	1280x720	530x871
range	20cm-65m	20cm-70cm	20cm-80cm
# instances	~ 1000	~1900	~1200

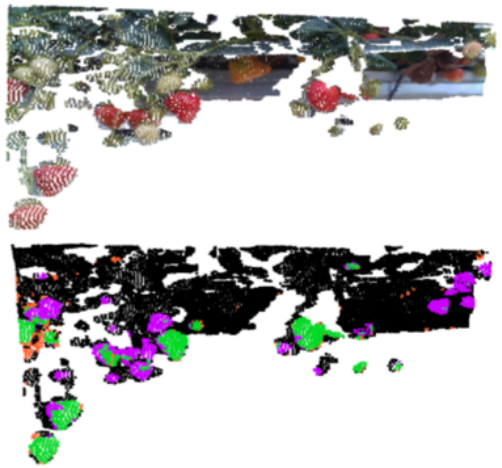


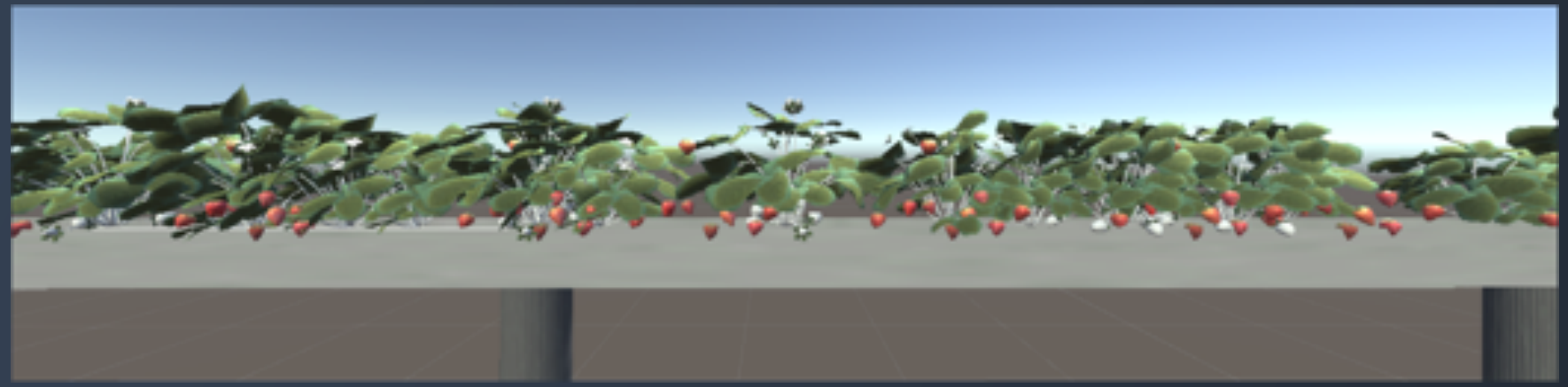
Figure 4: The segmentation results for $PNet_{colour}$ trained on data from the stereo camera: the original point cloud (top), segmentation results (bottom). The colours indicate: TP in green, FP in orange, FN in purple and TN in black.

Limited data used for evaluation of state of the art algorithms

Semantic segmentation, and instance segmentation, various shape quality

Different sensors

Way more available but not annotated



Cool Papers (Interesting Research)

2020

Neural Ordinary Differential Equations

TLDR: “Infinite layer residual networks with vector fields instead of finite number of discrete transformations”

Neural Ordinary Differential Equations

Ricky T. Q. Chen*, Yulia Rubanova*, Jesse Bettencourt*, David Duvenaud
University of Toronto, Vector Institute
{rtqichen, rubanova, jessebett, duvenaud}@cs.toronto.edu

Abstract

We introduce a new family of deep neural network models. Instead of specifying a discrete sequence of hidden layers, we parameterize the derivative of the hidden state using a neural network. The output of the network is computed using a black-box differential equation solver. These continuous-depth models have constant memory cost, adapt their evaluation strategy to each input, and can explicitly trade numerical precision for speed. We demonstrate these properties in continuous-depth residual networks and continuous-time latent variable models. We also construct continuous normalizing flows, a generative model that can train by maximum likelihood, without partitioning or ordering the data dimensions. For training, we show how to scalably backpropagate through any ODE solver, without access to its internal operations. This allows end-to-end training of ODEs within larger models.

1 Introduction

Models such as residual networks, recurrent neural network decoders, and normalizing flows build complicated transformations by composing a sequence of transformations to a hidden state:

$$\mathbf{h}_{t+1} = \mathbf{h}_t + f(\mathbf{h}_t, \theta_t) \quad (1)$$

where $t \in \{0 \dots T\}$ and $\mathbf{h}_t \in \mathbb{R}^D$. These iterative updates can be seen as an Euler discretization of a continuous transformation (Lu et al., 2017; Haber and Ruthotto, 2017; Ruthotto and Haber, 2018).

What happens as we add more layers and take smaller steps? In the limit, we parameterize the continuous dynamics of hidden units using an ordinary differential equation (ODE) specified by a neural network:

$$\frac{d\mathbf{h}(t)}{dt} = f(\mathbf{h}(t), t, \theta) \quad (2)$$

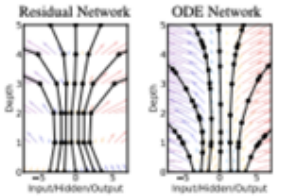


Figure 1: *Left:* A Residual network defines a discrete sequence of finite transformations. *Right:* A ODE network defines a vector field, which continuously transforms the state. *Both:* Circles represent evaluation locations.

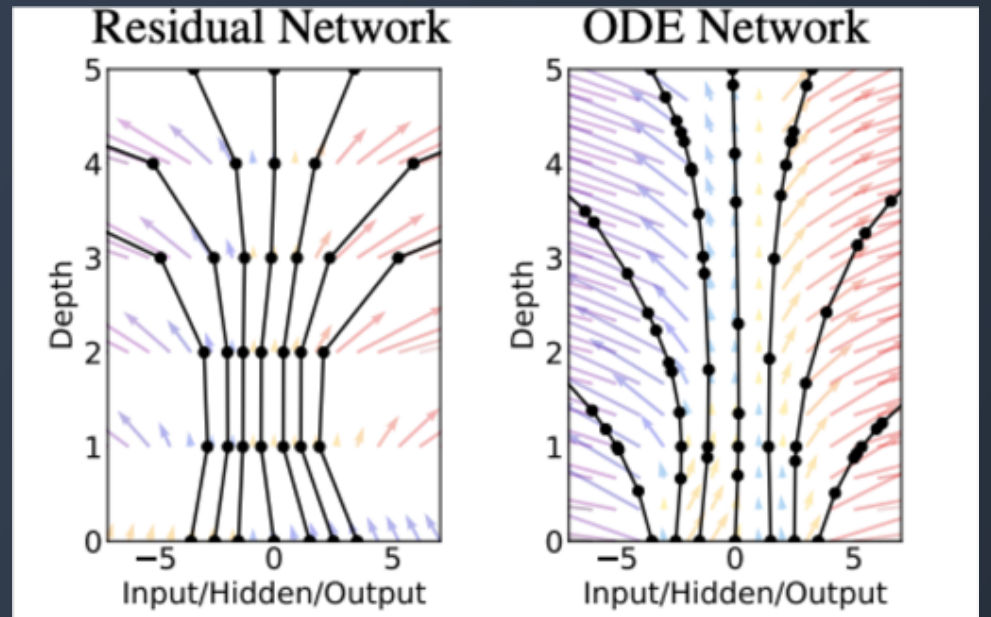


Figure 1: *Left:* A Residual network defines a discrete sequence of finite transformations. *Right:* A ODE network defines a vector field, which continuously transforms the state. *Both:* Circles represent evaluation locations.