2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

N-QR: Natural Quick Response Codes for Multi-Robot Instance Correspondence

Nathaniel Moore Glaser[1,2], Rajashree Ravi[2], and Zsolt Kira[1]
[1]Georgia Institute of Technology
[2]Bowery Farming
{nglaser,zkira}@gatech.edu

***Abstract*— Image correspondence serves as the backbone for many tasks in robotics, such as visual fusion, localization, and mapping. However, existing correspondence methods do not scale to large multi-robot systems, and they struggle when image features are weak, ambiguous, or evolving. In response, we propose *Natural Quick Response* codes, or *N-QR*, which enables *rapid* and *reliable* correspondence between large-scale teams of heterogeneous robots. Our method works like a QR code, using keypoint-based alignment, rapid encoding, and error correction via ensembles of image patches of natural patterns. We deploy our algorithm in a production-scale robotic farm, where groups of growing plants must be matched across many robots. We demonstrate superior performance compared to several baselines, obtaining a retrieval accuracy of 88.2%. Our method generalizes to a farm with 100 robots, achieving a 12.5x reduction in bandwidth and a 20.5x speedup. We leverage our method to correspond 700k plants and confirm a link between a robotic seeding policy and germination.**

I. Introduction

Many robotic tasks, such as visual localization and mapping, rely on matching the same features across image views. This process, commonly referred to as image correspondence, often requires prominent, static features to perform the matching (e.g. the rigid corners of buildings). However, these types of features are not guaranteed, especially as robots venture into environments that have non-rigid features.

In this paper, we address one such environment—robotic agriculture. In our setting, plants are grown by moving them through a sequence of specialized robot stations, in a process similar to a factory assembly line. Ultimately, we seek to use the cameras of each robot station to track every single plant throughout its lifecycle. By monitoring individual plants from seed to harvest, farmers can make key decisions about adjusting seeding patterns, water, light, and nutrients.

However, due to system limitations, our plants are shuffled between stations, and plant-level tracking must be performed without external markers (e.g. QR tags). Instead, we must rely on the plants themselves as the unique identifiers. This constraint makes tracking challenging, as plants are non-rigid, growing objects with ambiguous features.

To address these challenges, we propose **Natural Quick Response**, a learned approach for performing high-volume, ambiguity-prone correspondence between bandwidth-limited robots. **N-QR** aligns a candidate object to a uniform representation where it then ensembles and encodes image patches into compact, robust features for cross-robot comparison.

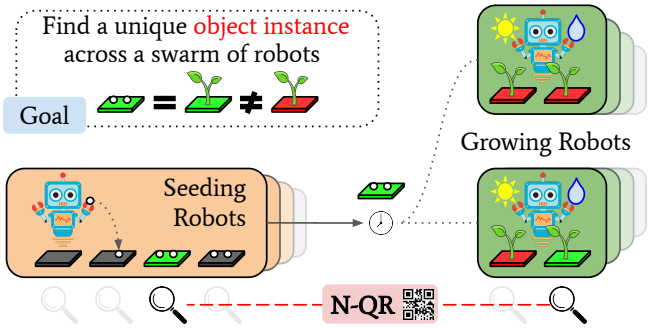

Fig. 1: **N-QR** uses naturally-occurring patterns and rapid encoding to help large teams of robots find a unique object. This correspondence allows robotic farmers to track the same plant, from seed to harvest, without extra tagging hardware.

Our approach expands the operational domain of image correspondence along three dimensions: (1) ***multi-robot scale***, (2) ***viewpoint heterogeneity***, and (3) ***visual ambiguity***. First, our algorithm scales to a farm that has thousands of communicating robots, each with their own sensors, actuators, and compute. Second, it performs matching despite significant ***visual dissimilarity*** between (a) cameras of different resolutions, lighting, and positioning and (b) objects that have changing visual features. Third, it performs matching despite misleading ***visual similarity***, as the subjects that we are imaging (i.e. a grid of plants) have strong, ambiguous features, but weaker unique features.

We summarize the contributions of our paper as follows:

- We address the task of ***large-scale***, ***multi-robot*** **instance correspondence** in an unprecedented research setting—a **production robotic farm** with *thousands* of robots.
- Our method, **N-QR**, achieves a state-of-the-art image retrieval accuracy of ***88.2%*** via a novel, multi-tiered ensembling scheme. This approach matches the same physical object *despite drastic appearance changes*.
- Our bandwidth-efficient transmission policy allows each robot to iteratively describe its observations via a scheduled transmission of low-dimensional embeddings. We leverage decentralized compute and reduce bandwidth by ***12.5x*** and computation latency by ***20.5x***.
- Finally, we deploy this matching pipeline for multi-view agricultural insights. Our method finds a link between our robotic seeding policy and resultant plant growth.

979-8-3503-8457-4/24/$31.00 ©2024 IEEE

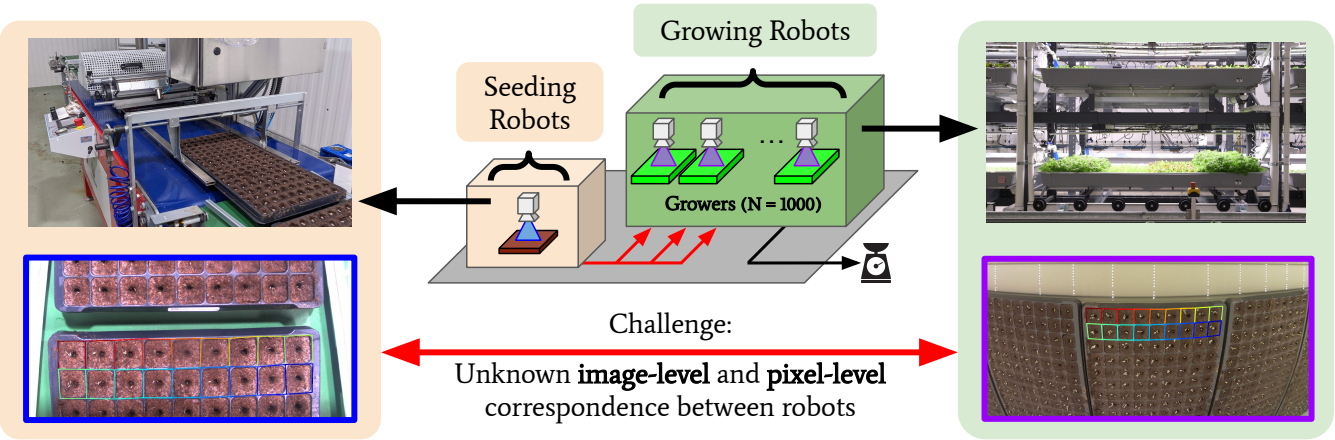

Fig. 2: **Robotic farming system and image matching challenge.** A seeding robot (left) continuously plants seeds into "rafts" on a moving conveyor belt. After a germination period, these rafts are transferred to a stationary growing robot (right), who supports and monitors growth for the remaining duration of the plant lifecycle. ***Image-level, raft-level, and pixel-level correspondences are unknown between the seeding and growing robots.***

II. Related Work

Several domains relate to the task of **multi-robot instance correspondence** and **multi-view growth analysis**.

Image correspondence methods [17, 12, 14, 2, 18, 11, 23, 22, 3, 4] seek to match features or perform dense alignments between two images, on a pixel level. **Traditional sparse correspondence** methods [17, 12, 14, 2] rely on strong corners and geometrically-consistent features to compute and confirm matching keypoints between different images. **Learned sparse correspondence** methods [18, 11] leverage learned feature descriptors, semantics, and relationships to match across broader featureless regions. **Dense correspondence** methods [23, 22] compute a dense pixel warping grid between images. **Multi-robot dense correspondence** [3, 4] methods address the added challenge of corresponding across several agents, often while paying heed to bandwidth and computational constraints. However, these methods completely fail in our setting (see Sec. IV-B). In response to these failures, we decompose the instance correspondence problem into two subproblems: (1) aligning the pair of images to a normalized representation and (2) performing discrete image (block) matching.

Keypoint detection works [9, 5] are useful for our alignment problem. These works use CNN architectures to generate a heatmap mask from which keypoins are extracted. We leverage similar techniques to detect keypoints within our scene, such as seeding tray corners and vertices, which we then use for image warping.

Discrete image matching methods [19, 8] seek to find matches on an image level. **Metric learning** [19] approaches learn image-level descriptors such that metric distances between similar images are low compared to distances between dissimilar images. One popular instance of this approach is the **Siamese Network** [8], which uses a shared neural network encoder to produce these image-level descriptors. Our work similarly uses a metric learning objective. However, unlike these prior works, our approach leverages multiple tiers of patch ensembling to overcome noisy and misleading inputs.

Content Based Instance Retrieval (CBIR) methods [1, 25, 27, 21, 20, 15] aim to improve accuracy by extracting distinctive features and reducing the impact of image clutter when efficiently querying a large database. While our approach addresses similar challenges, our dataset presents a higher level of complexity, characterized by minimal scene variation and a notable degree of visual similarity among instances, distinguishing it from commonly used datasets such as **GLDv2** [25]. Moreover, previous works have focused on addressing search efficiency concerns using methods like **deep hashing** [27, 21, 20] and inverted files [15]. Our method employs a bandwidth efficient iterative transmission policy with increasing feature sizes to minimize the total number of packets required for accurate matching.

Multiple instance learning looks at multiple instances to determine an overall classification. Several works [10, 6] consider multiple image patches of a cancer cell before rendering a final verdict, which is especially useful when individual patches are noisy or misleading. Our approach extends this idea from a classification setting to a metric learning setting, especially for robust image matching.

Yield estimation methods use plant phenotypes [13] or overhead camera information [7, 26] to predict the final harvest mass of a crop. Similarly, we evaluate crop yield in our system by calculating leaf area from overhead camera images, a metric strongly correlated with harvest yield, as demonstrated in prior studies.

Unlike prior research [16], which performs **multi-view yield estimation** by capturing the same plant from different angles, our approach employs multiple views differently. We gather complementary information from heterogeneous sensors observing various stages of plant growth, thus enriching our analytical insights.

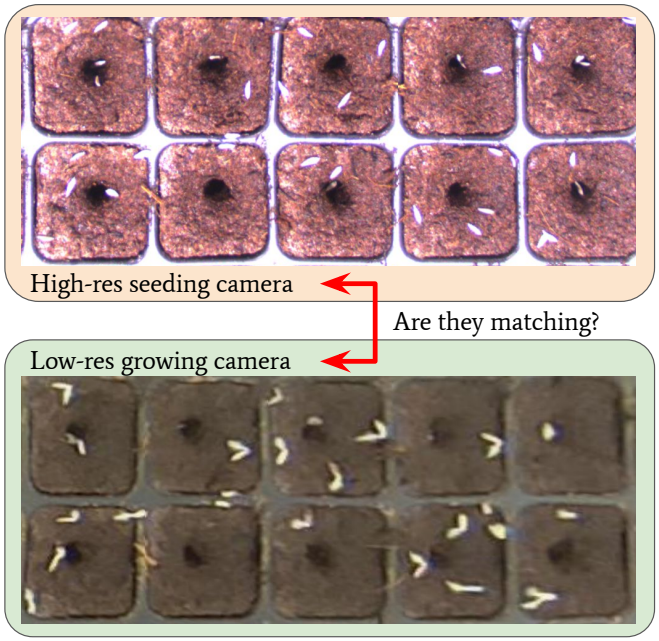

Fig. 3: **Matching difficulty**. These two normalized raft images ($\mathbf{R}$ and $\mathbf{R}'$) constitute a positive match. As an exercise for the reader, we encourage you to find the features that support and discourage this match.

III. Methodology

A. System

As depicted in fig. 2, our system is a production-scale **vertical farm**[1]. Within this system, a set of agricultural robots $\{\mathbf{S}_0, \mathbf{S}_1, \mathbf{G}_0, ..., \mathbf{G}_{3000}\}$ work together to grow a plant through different parts of its extended lifecycle.

Seeding: Each seeding robot $\mathbf{S}_i$ sits above a conveyor belt, where it drops seeds into a sequence of M rafts[2]. For each raft m, the robot captures a top-down image $\mathbf{X}_i^m$, containing one **normalized raft image** $\mathbf{R}_i^m$:

$$\mathbf{R}_i^m = \mathbb{W}_i^m(\mathbf{X}_i^m), \tag{1}$$

where $\mathbb{W}$ is an warping function that extracts a cropped, uniform grid of dirt cells $\mathbf{R}$ from $\mathbf{X}$, as in figs. 2 and 3.

Germination: Next, the rafts are moved into a chamber for germination **and are not tracked during this period**[3].

Growth: After germination, the rafts are manually placed onto benches[4]. A robot then moves each bench to a designated growing robot $\mathbf{G_j}$. This robot $\mathbf{G_j}$ supplies light and water to the plants for the remainder of the lifecycle of the plant. At regular intervals, each growing robot captures a top down image $\mathbf{Y}_j^t$, which contains 10 normalized raft images:

$$\bar{\mathbf{R}}_{j,q}^t = \mathbb{W}_{j,q}^t(\mathbf{Y}_j^t), \text{ for } q = \{0, ..., 9\}. \tag{2}$$

[1]**Vertical Farm:** A "small" footprint, indoor farm that grows crops in a space-efficient, stacked configuration

[2]**Raft:** A grid of dirt that can be irrigated by flooding it with water

[3]**Raft Tracking:** To enable raft-level tracking, we would need to retool and retrain operators. Rather than overhaul the farm with QR codes and scanners, this work explores the minimally-invasive question: **can we use the raft itself as a QR code?**

[4]**Bench:** An open container used to hold, move, and irrigate several rafts

It is important to note that each normalized raft image $\bar{\mathbf{R}}$ captured by a growing robot is a time-delayed version of a raft image $\mathbf{R}$ captured during seeding:

$$\bar{\mathbf{R}} = \mathbb{F}(\mathbf{R}, t, s, e), \tag{3}$$

where the general farming process $\mathbb{F}$ induces visual changes in $\mathbf{R}$ based on the time since seeding t, seeding configuration s, and environmental influences e such as lighting and water.

Oftentimes, the visual changes produced by $\mathbb{F}$ are so severe that traditional dense matching pipelines fail [17, 11, 23]. These changes include the following: (1) object geometry changes (since the seeds have germinated and grown), (2) minor to major positional changes (since the plants may be jostled between stages), and (3) illumination and resolution changes. A matching example of $\mathbf{R}$ and $\bar{\mathbf{R}}$ is shown in fig. 3.

B. *Task:* Multi-Robot Dense Matching

Given this farming system, we first address the task of **instance correspondence** between the raft in $\mathbf{X}$ and $\mathbf{Y}$. After substituting eq. 1 and eq. 2 into eq. 3 and setting $t = 10$ (the earliest available growing robot image), we yield an equation that summarizes our challenge:

$$\mathbb{W}_{j,q}(\mathbf{Y}_j) = \mathbb{F}(\mathbb{W}_i^m(\mathbf{X}_i^m)). \tag{4}$$

Namely, we seek to find object pixels in $\mathbf{X}_i^m$ that map to object pixels in $\mathbf{Y}_j$, despite drastic appearance changes (induced by $\mathbb{F}$), unknown robot association (i, j), multiple candidates per image (q), and unknown sub-image alignment ($\mathbb{W}_{j,q}$, $\mathbb{W}_i^m$). To tackle these challenges, we propose an **alignment** and **discrete matching** pipeline.

C. Alignment

To compute the normalized raft images as described in eqs. 1 and 2, we perform the following:

1) **Raft BBox NN**[5]**:** For each image, use a keypoint NN to identify $Q \times 2$ corner points for each raft, then crop the raft.
2) **Raft Vertex NN:** For each raft image, use a second keypoint NN to extract $H \times W$ grid vertices. Group points to represent the four corners of each dirt cell.
3) **Patch Extraction:** For each set of vertex corners, warp the source image into a square target image. Recombine these cells to form a **normalized raft image**, as in fig. 3.

D. Discrete Matching

The objective of the discrete matching pipeline is to satisfy eq. 3, given the discrete choices of $\{\mathbf{R}_i^m\}$ and $\{\bar{\mathbf{R}}_{j,q}\}$, as generated by the warping procedure:

$$\bar{\mathbf{R}}_{j,q} = \mathbb{F}(\mathbf{R}_i^m) \tag{5}$$

In other words, we want to find the correct choice of $\hat{m}$, $\hat{i}$, $\hat{j}$, and $\hat{q}$ out of all potential choices $(|G| \times Q) \times (|S| \times M)$. The total number of pairwise choices for the entire farm is approximately 10^6.

[5]**BBox NN:** bounding box neural network

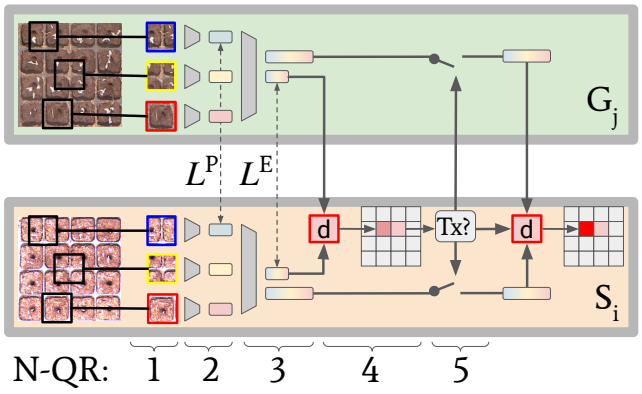

Fig. 4: **N-QR** iteratively transmits feature embeddings until a minimum distance threshold is met.

To address these challenges, we propose a metric learning approach that uses a decentralized, bandwidth-efficient feature extractor to generate $\mathbb{F}$ invariant embeddings. Namely, we propose two parallel neural networks $\mathbf{\Phi}_S$ and $\mathbf{\Phi}_G$ to compute embeddings f and a pairwise distance l_2:

$$f_i^m = \mathbf{\Phi}_S(\mathbf{R}_i^m), \qquad \bar{f}_{j,q} = \mathbf{\Phi}_G(\bar{\mathbf{R}}_{j,q})$$
$$l_2(\mathbf{R}, \bar{\mathbf{R}}) = \|\bar{f}_{j,q} - f_i^m\|_2 \tag{6}$$

To satisfy eq. 5, we want the distance for a positive match $l_2(\mathbf{R}, \bar{\mathbf{R}}^+)$ to be smaller than the distance of a negative match $l_2(\mathbf{R}, \bar{\mathbf{R}}^-)$ by some margin α. We adopt the triplet loss objective [19]:

$$\mathcal{L}(\mathbf{R}, \bar{\mathbf{R}}^+, \bar{\mathbf{R}}^-) = \max(l_2(\mathbf{R}, \bar{\mathbf{R}}^+) - l_2(\mathbf{R}, \bar{\mathbf{R}}^-) + \alpha, 0) \tag{7}$$

1-vs-1 Raft Matching: In order to make $\mathbf{\Phi}_S$ and $\mathbf{\Phi}_G$ invariant to the extreme influences of $\mathbb{F}$, we propose a specialized comparison network based on image patch ensembling, as shown in fig. 4. Decomposing images into patches lets us consider partial raft images, apply an intermediate training signal at the patch level, and increase our dataset size (via random permutations and augmentations for each patch). To over come insufficient or misleading information in noisy patches, our method aggregates their features in an ensemble. Our method involves randomly sampling image patches from the same locations in $\mathbf{R}$ and $\bar{\mathbf{R}}$, extracting patch-level embeddings using a ResNet extractor, and stacking them along their channel dimension using a fully connected network to generate a final image-level embedding f and $\bar{f}$. Triplet losses $\mathcal{L}^P$ and $\mathcal{L}^E$ are used to train patch-level and image-level embeddings, respectively.

Based on the training objective, the distance between the features of a matching raft should be smaller than that of a non-matching raft:

$$\|\bar{f}_{\hat{j},\hat{q}} - f_{\hat{i}}^{\hat{m}}\|_2 + \alpha < \|\bar{f}_{j,q} - f_i^m\|_2 \tag{8}$$

1-vs-Many Raft Matching: The previous procedure evaluates a single match candidate *pair* $\mathbf{R}$ and $\bar{\mathbf{R}}$. Beyond this capability, we also want to "retrieve" the right match among numerous incorrect ones. Ideally then, the smallest computed pairwise distance between $\mathbf{R}_i^m$ and all $\{\bar{\mathbf{R}}_{j,q}\}$ would correspond to the correct match out of $|G| \times Q$.

Multi-Pass 1-vs-Many Raft Matching: To enhance accuracy over our base approach, we run our pairwise matching network with random patch samples in multiple passes and then average the pairwise distances across these batches.

Decentralized Processing: Our system comprises of heterogeneous robots with varying compute capabilities: the many growing robots have cost-effective, weaker CPUs, while the relatively few seeding robots have desktop processors. Instead of broadcasting all raw images to a centralized processor, we use the natural parallelism of our cluster of growing robots. Each robot performs its own warping and feature extraction, with $\mathbf{\Phi}_S$ and $\mathbf{\Phi}_G$ and features $\bar{f}_{j,q}$ are then broadcast to each seeding robot for $|G| \times Q$ pairwise comparisons.

Bandwidth Efficient Transmission Policy: We further conserve bandwidth via an iterative transmission policy, as shown in fig. 4. Since our system generates a significant amount of network chatter, we want to minimize the cumulative packet size required to perform accurate matching. Therefore, we propose to iteratively broadcast denser and denser feature representations $\bar{f}_{\text{tx}}$, based on the distance between the smallest and second smallest pairwise distances $\delta = d_1 - d_0$ relative to margin α:

$$\bar{f}_{\text{tx}}(\bar{\mathbf{R}}, \delta, c) = \begin{cases} \mathbf{\Phi}_G^{16}(\bar{\mathbf{R}}), & \text{if } \delta < \alpha \wedge c < 10 \\ \mathbf{\Phi}_G^{128}(\bar{\mathbf{R}}), & \text{if } \delta < \alpha \wedge c \geq 10 \\ \text{stop transmit}, & \text{if } \delta \geq \alpha \vee c \geq 20 \end{cases} \tag{9}$$

where c is the index of the transmission and $\mathbf{\Phi}_G^{16}$, $\mathbf{\Phi}_G^{128}$ are networks that extract features at sizes of 16 and 128, respectively. We incorporate this transmission policy into our ensembling scheme—whereby each transmitted feature is used to compute a pairwise distance, which we combine into a running average distance matrix.

*E. **Task:** Multi-View Seed-Growth Analysis*

Ideally, the seeding robot plants all seeds into the central hole within each dirt cell. This recessed area provides an ideal seed germination environment with its darkness and moisture. In practice, however, seeds often stray from this desired location, yet they still manage to grow. We seek to answer the question: how crucial is it for our robot to plant seeds in the hole? Should farms invest in optimizing seed placement, or is the current system sufficient? We hypothesize that seeds in the hole have a higher germination rate based on intuition.

To test our hypothesis, we extract all grid cell patches $\mathbf{P}$ and $\bar{\mathbf{P}}$ for each raft image $\bar{\mathbf{R}}$ and $\mathbf{R}$. Next, we want to determine how a seeding pattern s observed in a dirt cell $\mathbf{P}$ influences growth h observed in $\bar{\mathbf{P}}^t$ over growing time t:

$$s = \mathbf{\Psi}(\mathbf{P}), \qquad h(t) = \mathbb{H}(\{\bar{\mathbf{P}}^0, ..., \bar{\mathbf{P}}^t\}) \tag{10}$$

where $\mathbf{\Psi}$ is a keypoint detection network used for predicting seed locations and $\mathbb{H}$ is a procedure for measuring growth over time.

Plant Growth: We measure plant growth over time with a Mixture of Gaussians background subtraction module

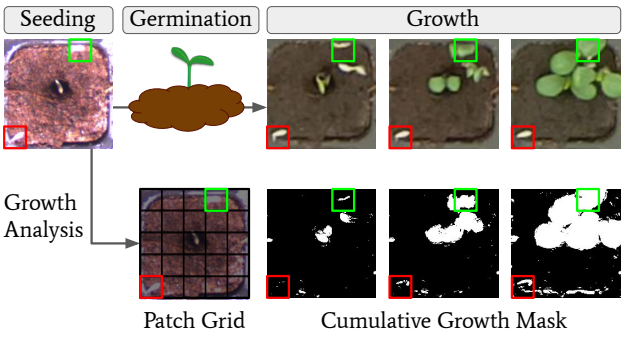

Fig. 5: **Plant lifecycle** with patch-based analysis of growth.

(**MOG2** [28, 29]). As shown in fig. 5, this module produces a sequence of binary masks for the foreground pixels in an input sequence of images: $\mathbf{I}^t = \mathbf{MOG2}(\bar{\mathbf{P}}^t, \bar{\mathbf{P}}^{t-1}, \mathbf{I}^{t-1})$. We then compute a cumulative mask by accumulating each foreground increment: $C(t) = \sum\{\mathbf{I}^0, ...\mathbf{I}^t\}$. Our growth metric is a spatial average of the binary cumulative mask at each time step: $h(t) = \text{avg}_{[x,y]}(C(t))$.

Seed Location vs. Plant Growth: To evaluate the effect of seed location on plant growth, we subdivide each image patch into a 5×5 grid of subpatches, as shown in fig. 5. For each seed detected in one of these subpatches, we compute the corresponding $h_{x,y}(t)$ for that subpatch. Finally, we compute a **growth score** $\bar{h}$ for each seed:

$$\bar{h}_{x,y} = \text{avg}_t(h_{x,y}(t)) \tag{11}$$

IV. Results

A. Dataset

Without loss of generality, we present results for 1 seeding robot and 100 growing robots, a representative subset of the full robotic system. Our testing dataset consists of 17 unique seeding rafts, each with a unique match among 473 total growing rafts. Our training dataset includes 38 separate positive pairs. Each of the growing rafts look visually similar, with only slight natural variations. The seeding raft looks substantially different from the growing rafts. These observations cover 10 plant types and span several months.

B. Multi-Robot Dense Matching

Fully End-to-End Correspondence: We first attempted this problem with direct raft-to-raft image matching. Our initial attempts, as well as several state-of-the-art baselines [11, 23], were not successful. These failures likely arose because our object of interest has (a) weak features that sometimes shift and change over time (i.e. plants) and (b) strong features that are ambiguously tessellated (i.e. raft gridding). Our difficult agricultural setting requires methods to focus on weak features and ignore strong, ambiguous features, contrary to traditional methods.

Alignment: We evaluate keypoint detections for our alignment algorithm. The **Raft BBox NN**, **Seeding Raft Vertex NN**, **Growing Raft Vertex NN** achieve 4 pixel MSE and $\mathbf{97.9}\%$, $\mathbf{96.1}\%$, and $\mathbf{82.6}\%$ detection accuracy, respectively. The resolution difference between the seeding and growing cameras likely accounts for the drop in performance.

		Retrieval Accuracy				
		1-vs-1	1-vs-Many			
			Single		Multi-Pass	
Method			top1	top3	top1	top3
Siamese [8]		61.2	17.1	37.6	23.5	41.2
Metric Learning [19]		68.8	16.5	27.6	17.6	29.4
DeepMIL [6]		81.6	5.3	17.1	11.8	23.5
Stacked Attention* [24]		92.5	73.5	85.9	82.4	88.2
P=22, F=128	**Ours**	**99.1**	**67.1**	**90.0**	**88.2**	**100**
P: Num Patches	1	73.0	19.4	47.6	47.1	82.4
	2	85.2	24.7	56.5	47.1	70.6
	4	91.3	43.5	74.7	64.7	94.1
F: Feature Dim	1	51.8	7.1	18.8	11.8	23.5
	2	57.3	9.4	21.8	11.8	29.4
	4	75.8	11.2	31.8	41.2	70.6
	16	98.0	60.6	81.8	88.2	100

TABLE I: **Network retrieval accuracy** for correctly distinguishing between a positive and negative match (1-vs-1) as well as finding a positive match among many negatives (1-vs-Many), especially with ensemble averaging (Multi-Pass).

Discrete Matching: We show the results of our algorithm on retrieving a correct match between a query seeding raft and several candidate growing rafts, including:

- One positive and one negative match (**1-vs-1**)
- One positive and many negative matches (**1-vs-Many**)

For (**Multi-Pass 1-vs-Many**), we average the computed distance between query and all candidates over 10 different passes. We report how frequently we place the correct match in the top 1 and 3 *lowest* distances, as averaged across our 17 matching pairs. First, we present results for a **hard negative** split of the dataset, involving **10 robots**. Later, we show how our method generalizes to a **100 robot** dataset.

Hard Negative Dataset (*10 Robots*): Prior work [19] emphasizes the importance of training with hard negatives, especially to distinguish between similar-looking negative instances (our dataset contains many). For our work, we sample patch negatives from the following categories of increasing difficulty: *same raft* (40%), *same robot* $\mathbf{G}$ (40%), and *all images* (20%). Patches from the same raft and robot look the most similar and thus pose the hardest challenge.

In Tab. I, we show that our method outperforms several key baselines [6, 24, 19, 8]. The metric learning approaches send a concatenated set of input patches into dedicated [19] or shared [8] feature encoders, which are then trained via a triplet loss. Unlike our approach, these methods do not leverage intermediate patch features and feature ensembling. DeepMIL [6] and Stacked Attention [24] do use some form of feature-based ensembling, achieving improved performance. However, these methods use a classification loss instead of a triplet loss. Our method achieves the strongest performance thanks to both patch-feature ensembling, intermediate feature training, and metric learning. We show that multiple passes of our patch-ensembling method improves our **1-vs-Many** accuracy, justifying this architectural choice.

Multi-Pass Discrete Matching: In Tab. I, we also provide

Tx Policy	Total Packet Dim	Avg Num Packets	Accuracy top1 (%)	Accuracy top3 (%)
$\alpha = \infty$	1440	20.0	94.1	100
$\alpha = 2.0$	683	10.7	90.6	98.7
$\alpha = 1.0$	160	3.8	83.6	94.1
$\alpha = 0.5$	32	1.5	75.9	90.3
$\alpha = -\infty$	16	1.0	61.2	81.1

TABLE II: **Retrieval accuracy for transmission policies.**

an ablation study showing that a larger subset of patches improves our overall matching performance. Note the poor performance for a single patch, which has a **1-vs-1** accuracy of 73.0%, compared to 22 patches with 99.1%. Individual patches frequently lack distinctive features, so successful methods must consider ensembles of multiple patches. However, increasing the numbers of patches (and hence computational cost) yields diminishing accuracy improvements. We found the optimal number of patches to be **22**.

Bandwidth-Efficient Matching: In Tab. I, we show the trade-off between retrieval accuracy and embedding dimension. Often, a feature size of 16, is sufficient to obtain a strong multi-pass accuracy. Alternatively, a size of 128 is ideal if there are time limitations and no bandwidth limitations. These results motivated our choice of transmission policy, which sequentially transmits 10 feature vectors of $|f| = 16$ then switches to 10 feature vectors of $|f| = 128$.

In Tab. II, we show the impact of our bandwidth-efficient transmission policy for different bandwidth preferences (α from eq. 7). A low α corresponds to a policy that prefers early termination of transmissions. This policy conserves bandwidth at the expense of accuracy. Note that our approach can still achieve a 90% retrieval accuracy with only 683 FPN, a 50% reduction from the full transmission policy and several orders of magnitude cheaper (10^4) than raw image transmission ($\approx 1MB$ aft compression). This reduced dimension saves both network bandwidth and computation.

Large-Scale Dataset (*100 Robots*): We report that our retrieval method generalizes well to large scales, achieving a pairwise matching accuracy of **99.8%**. Moreover, we attain **top1**, **top3**, **top5** retrieval accuracies of **64.7%**, **82.4%**, **100%**, respectively. On average, our choice is in the top $\mathbf{0.16}^{\text{th}}$ percentile of distances. Despite training on a much smaller dataset, our method excels at finding a matching raft in a much larger pool of 475 ambiguously similar rafts. This generalization was enabled by our novel patch extraction scheme (which expanded our dataset) and choice of triplet loss objective with hard-negative sampling.

Heterogeneous, Decentralized Compute: In Tab. III, we present a timing study for each stage of the matching pipeline. With a centralized policy, 100 growing robots transmit their images to a strong centralized computer (consuming $\sim$ **100 MB** of bandwidth), which performs the matching task in $\sim$ **21 minutes**. However, our parallelized transmission policy accomplishes the task in **64 seconds** (a **20.5x** speedup!), while consuming only **8 MB** (a **12.5x** reduction!). The results scale well for a large-scale deployment (3000 robots), obtaining a **616x** speedup and **375x** reduction in bandwidth.

[6]Intel(R) Core(TM) i7-8700K CPU @ 3.70GHz
[7]Cortex A-72 (ARM v8) 64-bit SoC @ 1.8GHz

		Strong CPU[6] (s)	Weak CPU[7] (s)
Align	Undistort	0.6	3.0
	BBox NN	0.8	9.9
	Vertex NN **G**	11.2	41.6
Match	Patch Extraction	0.2	2.6
	Matching NN	0.2	6.0
Analysis	BG Subtraction	0.2	0.9
	Total	**13.1**	**64.0**

TABLE III: **Median processing times.**

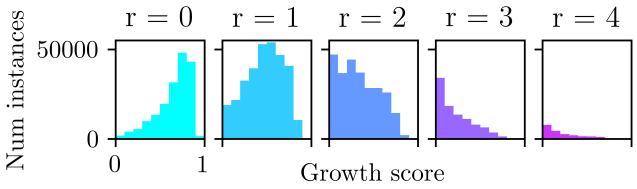

Fig. 6: **Histograms of growth scores** for seeds detected at increasing l_1 distances from the center patch ($r = 0$).

C. Multi-View Seed-Growth Analysis

The results from the preceding section allow us to monitor the same bench of plants from two specialized perspectives.

Seed Location vs. Plant Growth: We analyze the growth patterns of **712,636** individual seeds planted across **2,885** individual seeding cells. As shown in Fig. 6, we observe that seeds detected towards the center of the seeding cell have higher growth scores, as computed in eq. 11. On average, seeds in the center ($r \leq 1$) attain 52.9% growth, compared to seeds towards the edges ($r > 1$) with 29.1%. The seeds at the edge account for 46.8% of the total number of seeds but only produce 32.6% of the growth. These results confirm our hypothesis: seeds planted far from the center of the cell experience reduced average growth than those properly planted. Moreover, a corrective action is merited since a substantial fraction of seeds fall at this distance.

V. Conclusion

With **N-QR**, we tackle the task of multi-robot instance correspondence within the setting of a production-scale robotic farm. We test our approach on an unprecedented and challenging image matching dataset, full of visually similar instances with misleading features. We use novel multi-pass patch-ensembling to achieve a **top1** retrieval accuracy of **88.2%**, outperforming several key baselines. On a high-volume matching task with 100 robots, we show that our transmission policy yields a retrieval accuracy of **64.7%** (finding a single match out of 473 rafts), **12.5x** reduction in bandwidth, and a **20.5x** speedup.

Future work will explore how our approach generalizes to other settings that significantly change over time. It will also explore how our method better enables downstream robotics tasks, such as image-based fusion, localization, and mapping.

REFERENCES

[1] Wei Chen et al. "Deep learning for instance retrieval: A survey". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2022).

[2] Martin A. Fischler and Robert C. Bolles. "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography". In: *Commun. ACM* 24.6 (June 1981), pp. 381–395. ISSN: 0001-0782. DOI: 10.1145/358669.358692. URL: https://doi.org/10.1145/358669.358692.

[3] Nathaniel Glaser et al. "Enhancing Multi-Robot Perception via Learned Data Association". In: *CoRR* abs/2107.00769 (2021). arXiv: 2107.00769. URL: https://arxiv.org/abs/2107.00769.

[4] Nathaniel Glaser et al. "Overcoming obstructions via bandwidth-limited multi-agent spatial handshaking". In: *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2021, pp. 2406–2413.

[5] Kaiming He et al. "Mask r-cnn". In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2961–2969.

[6] Maximilian Ilse, Jakub M. Tomczak, and Max Welling. "Attention-based Deep Multiple Instance Learning". In: *CoRR* abs/1802.04712 (2018). arXiv: 1802.04712. URL: http://arxiv.org/abs/1802.04712.

[7] Dae-Hyun Jung et al. "Image Processing Methods for Measurement of Lettuce Fresh Weight". In: *Journal of Biosystems Engineering* 40 (Mar. 2015), pp. 89–93. DOI: 10.5307/JBE.2015.40.1.089.

[8] Gregory Koch, Richard Zemel, Ruslan Salakhutdinov, et al. "Siamese neural networks for one-shot image recognition". In: *ICML deep learning workshop*. Vol. 2. 1. Lille. 2015.

[9] Hei Law and Jia Deng. "Cornernet: Detecting objects as paired keypoints". In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, pp. 734–750.

[10] Xin-Chun Li et al. "Deep multiple instance selection". In: *Science China Information Sciences* 64 (2021), pp. 1–15.

[11] Philipp Lindenberger, Paul-Edouard Sarlin, and Marc Pollefeys. "LightGlue: Local Feature Matching at Light Speed". In: *arXiv preprint arXiv:2306.13643* (2023).

[12] D.G. Lowe. "Object recognition from local scale-invariant features". In: *Proceedings of the Seventh IEEE International Conference on Computer Vision*. Vol. 2. 1999, 1150–1157 vol.2. DOI: 10.1109/ICCV.1999.790410.

[13] Ali Mokhtar et al. "Using machine learning models to predict hydroponically grown lettuce yield". In: *Frontiers in Plant Science* 13 (2022), p. 706042.

[14] Marius Muja and David G Lowe. *Fast library for approximate nearest neighbors*. 2015.

[15] Henning Müller et al. "Efficient access methods for content-based image retrieval with inverted files". In: *Multimedia Storage and Archiving Systems IV*. Vol. 3846. SPIE. 1999, pp. 461–472.

[16] Luis G. Riera et al. "Deep Multiview Image Fusion for Soybean Yield Estimation in Breeding Applications". In: *Plant Phenomics* 2021 (2021). DOI: 10.34133/2021/9846470. eprint: https://spj.science.org/doi/pdf/10.34133/2021/9846470. URL: https://spj.science.org/doi/abs/10.34133/2021/9846470.

[17] Ethan Rublee et al. "ORB: An efficient alternative to SIFT or SURF". In: *2011 International Conference on Computer Vision*. 2011, pp. 2564–2571. DOI: 10.1109/ICCV.2011.6126544.

[18] Paul-Edouard Sarlin et al. "Superglue: Learning feature matching with graph neural networks". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 4938–4947.

[19] Florian Schroff, Dmitry Kalenichenko, and James Philbin. "FaceNet: A Unified Embedding for Face Recognition and Clustering". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2015.

[20] Jingkuan Song et al. "Binary generative adversarial networks for image retrieval". In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 32. 1. 2018.

[21] Jingkuan Song et al. "Deep region hashing for efficient large-scale instance search from images". In: *arXiv preprint arXiv:1701.07901* (2017).

[22] Prune Truong et al. "GOCor: Bringing globally optimized correspondence volumes into your neural network". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 14278–14290.

[23] Prune Truong et al. "PDC-Net+: Enhanced Probabilistic Dense Correspondence Network". In: *CoRR* abs/2109.13912 (2021). arXiv: 2109.13912. URL: https://arxiv.org/abs/2109.13912.

[24] Ashish Vaswani et al. "Attention is all you need". In: *Advances in neural information processing systems*. 2017, pp. 5998–6008.

[25] Tobias Weyand et al. "Google Landmarks Dataset v2 - A Large-Scale Benchmark for Instance-Level Recognition and Retrieval". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2020.

[26] Lingxian Zhang et al. "Growth monitoring of greenhouse lettuce based on a convolutional neural network". In: *Horticulture research* 7 (2020).

[27] Wanqing Zhao et al. "Spatial pyramid deep hashing for large-scale image retrieval". In: *Neurocomputing* 243 (2017), pp. 166–173.

[28] Z. Zivkovic. "Improved adaptive Gaussian mixture model for background subtraction". In: *Proceedings of*

the 17th International Conference on Pattern Recognition, 2004. ICPR 2004. Vol. 2. 2004, 28–31 Vol.2. DOI: 10.1109/ICPR.2004.1333992.

[29] Zoran Zivkovic and Ferdinand van der Heijden. "Efficient adaptive density estimation per image pixel for the task of background subtraction". In: *Pattern Recognition Letters* 27.7 (2006), pp. 773–780. ISSN: 0167-8655. DOI: https://doi.org/10.1016/j.patrec.2005.11.005. URL: https://www.sciencedirect.com/science/article/pii/S0167865505003521.