Ensemble Deep Learning Object Detection Fusion for Cell Tracking, Mitosis, and Lineage

Imad Eddine Toubal, Noor Al-Shakarji, D. D. W. Cornelison, and Kannappan Palaniappan

***Abstract*—Cell tracking and motility analysis are essential for understanding multicellular processes, automated quantification in biomedical experiments, and medical diagnosis and treatment. However, manual tracking is labor-intensive, tedious, and prone to selection bias and errors. Building upon our previous work, we propose a new deep learning-based method, EDNet, for cell detection, tracking, and motility analysis that is more robust to shape across different cell lines, and models cell lineage and proliferation. EDNet uses an ensemble approach for 2D cell detection that is deep-architecture-agnostic and achieves state-of-the-art performance surpassing single-model YOLO and FasterRCNN convolutional neural networks. EDNet detections are used in our M2Track multiobject tracking algorithm for tracking cells, detecting cell mitosis (cell division) events, and cell lineage graphs. Our methods produce state-of-the-art performance on the Cell Tracking and Mitosis (CTMCv1) dataset with a Multiple Object Tracking Accuracy (MOTA) score of 50.6% and tracking lineage graph edit (TRA) score of 52.5%. Additionally, we compare our detection and tracking methods to human performance on external data in studying the motility of muscle stem cells with different physiological and molecular stimuli. We believe that our method has the potential to improve the accuracy and efficiency of cell tracking and motility analysis. This could lead to significant advances in biomedical research and medical diagnosis. Our code is made publicly available on GitHub.**

***Index Terms*—Cell tracking, deep learning, detection, ensemble, multiobject tracking, deformable object tracking.**

***Impact Statement*— This work provides a robust state-of-the-art framework for the automatic detection and tracking of cell image sequences. This enables a more accurate computation analysis for quantifying cell behaviors including proliferation, interaction, and motility to draw biological conclusions.**

I. Introduction

EXPERIMENTS with live-cell imaging have provided a fascinating glimpse into the behavior of biological systems. Despite the fact that these trials can yield huge volumes of data, it is difficult to analyze these datasets computationally. Cells must be precisely detected in each image and then tracked over time in order to comprehensively capture the temporal evolution of cellular phenotypes. A robust cell tracking system should be able to track cells that are partially overlapping or touching in addition to well-separated cells, and should be able to identify and quantify potential heterogeneity among the cell population. Early work on cell tracking uses classical unsupervised methods to detect or segment cells [11], [12], [53], [54]. In order to exploit abundant cell imaging data to train deep learning models, data often needs to be annotated by experts. Hence, the difficulty of automatically performing single and multi-cell analysis.

In theory, given enough data, deep neural networks are able to achieve human-like performance on numerous vision tasks. This has been proven in natural images with the developments of massive and carefully annotated vision datasets [30], [34]. However, lack of large annotated biomedical datasets introduces challenges in advancing the research in generalized models for medical vision tasks. Numerous attempts for creating large-scale microscopy image data for cells have been made. The Cell Tracking Challenge (CTC) [46], [67] offers a dataset of different cell lines using different imaging modalities. This dataset offers “Gold Truth” annotations that are manually made by experts for a subsample of images; and “Silver Truth” that is semi-automatically generated and less reliable but provided for all images. CTC provides two benchmarks: Segmentation and Tracking; this means that exact instance segmentation masks are provided. In contrast, the CTMCv1 dataset [3] provides fully manual annotations in the format of bounding boxes that are persistently tracked over time. Additionally, CTMCv1 dataset features a single imaging modality with different cell lines. Despite the large scale of these datasets, the low diversity in data (in comparison with natural images) causes overfitting when training complex deep networks.

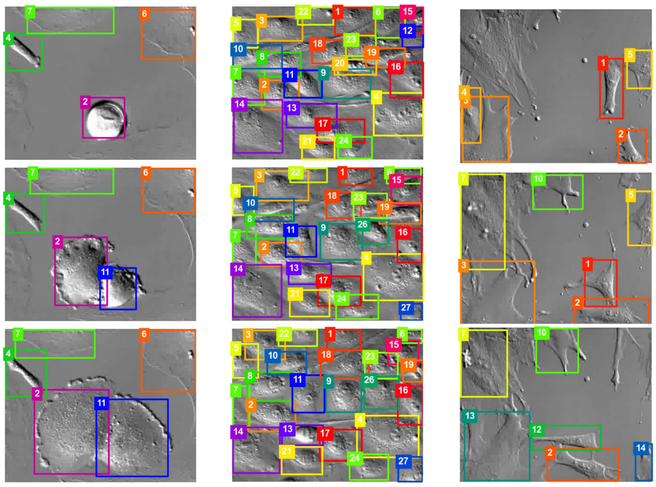
Manuscript received 4 January 2023; revised 10 April 2023 and 13 June 2023; accepted 13 June 2023. Date of publication 21 June 2023; date of current version 11 June 2024. This work was supported in part by the U.S. NIH National Institute of Neurological Disorders and Stroke under Grant R01NS110915, in part by the U.S. Army Research Laboratory project under Grant W911NF-18-20285 to KP, and in part by the NIH National Institute of Arthritis and Musculoskeletal Diseases under Grant AR062836 to DDWC. The review of this paper was arranged by Editor Y. Zhang. *(Corresponding author: Kannappan Palaniappan.)*

Imad Eddine Toubal, Noor Al-Shakarji, and Kannappan Palaniappan are with the Department of Electrical Engineering and Computer Science, University of Missouri, Columbia, MO 65211 USA (e-mail: itdfh@umsystem.edu; nmahyd@mail.missouri.edu; pal@missouri.edu).

D. D. W. Cornelison is with Christopher S. Bond Life Sciences Center, University of Missouri, Columbia, MO 65211 USA (e-mail: cornelisond@missouri.edu).

This article has supplementary downloadable material available at https://doi.org/10.1109/OJEMB.2023.3288470, provided by the authors.

Digital Object Identifier 10.1109/OJEMB.2023.3288470



(a) MDOK-run02 (b) BPAE-run02 (c) APM-run06

Fig. 1. Three different cell line DIC video microscopy examples showing variability in cell appearance, number, density and shape. Each column shows temporal cell tracking with colored bounding boxes and cell IDs for Frame 0 (Row 1), 100 (Row 2), and 200 (Row 3). The range in complexity varies from: (a) MDOK (Col 1) an *easy* sequence with few cells to detect and track; (b) BPAE (Col 2) a *difficult* sequence with many smaller cells at high density; and (c) APM (Col 3) a difficult sequence due to *large shape deformation*, though few cells. Mitosis or cell division events are identified and handled during tracking by passing the parent ID to one of the daughter and sibling cells as shown in MDOK-run01 (Col 1).

Several cell tracking methods have been proposed in the literature. The task of tracking is often composed of two subtasks: (a) detection [3] or segmentation [4], [27], [48], [59], [67] and (b) tracking [3], [4], [62], [67]. Ever since U-Net made a breakthrough in biomedical image segmentation [59], deep learning has been adopted as the most prominent method in both detection and segmentation. Most methods used in cell tracking can fall into the following two categories: (i) tracking-by-detection either using graph-based tracking [15], [17], [61], label propagation using overlap or distance [9], or nearest neighbor linking [45]; (ii) tracking-by-model evolution which includes an initialization step to locate the cells of interest on the first frame followed by a per cell model evolution in time by using deformable models such as active contours [19] or level sets [23], [24]. The baseline method on the CTMC dataset uses FasterRCNN [57] for detection and Tracktor [6] or DeepSORT [72] for tracking.

Tracking-by-detection [4], [38], [44], [55] is the most popular approach for automatic cell tracking. This method needs to associate pre-located cells from either segmentation [32], [38], [50] or detection [14], [66] approaches. The association process links the detected cells in time to produce cell trajectories. Tracking-by-detection can be implemented in two ways: (i) online (short-term) by associating and linking detections and tracks between consecutive frames [36]; or (ii) offline (long-term) tracking that uses information from the complete sequence of in the time-lapse [43]. Online tracking tends to be sensitive to detection errors, producing short, fragmented trajectories. An offline tracker often produces longer and more reliable trajectories but is more computationally expensive.

TABLE I
STATISTIC OF EACH CELL LINE IN THE CTMCv1 DATASET

		Number of frames			
Cell line	# Seqs.	Min.	Med.	Max.	Total
3T3	5	1,770	2,039	2,114	9,916
A-10	4	1,571	1,930	2,462	7,893
A-549	1	1,592	1,592	1,592	1,592
APM	3	1,458	1,794	2,006	5,258
BPAE	4	1,568	1,789	2,160	7,306
CRE-BAG2	2	1,652	1,792	1,931	3,583
CV-1	2	1,613	1,728	1,842	3,455
LLC-MK2	5	1,050	1,866	2,187	9,004
MDBK	5	368	1,500	2,195	6,667
MDOK	5	763	1,972	2,100	8,802
OK	4	1,200	1,350	1,620	5,520
PL1Ut	3	1,470	1,731	1,848	5,049
RK-13	2	900	1,200	1,500	2,400
U2O-S	2	1,972	1,972	1,972	3,944

This shows the variability in the number of sequences across cell lines as well as the number of frames across sequences. We note that thevariation in the number of frames across sequences is especially high in cell lines mdbk and mdok.

The contribution of this article is a detection and tracking framework that produces state-of-the-art performance on cell tracking on the CTMCv1 dataset. Our framework can be summarized in two-fold:

- First, we introduce a Model-Agnostic Detection Ensemble method that shows great robustness to over-fitting, and test our method using the YOLOv3 architecture as a base model;
- Second, we employ a robust tracking algorithm (M2Track) to link cell detections and track cell lineage.

II. Related Work

A. Scale-Invariant Feature Transform for Cell Tracking

Some earlier work on tracking cells in DIC time-lapse videos uses classical feature-matching methods. [28] used Scale-Invariant Feature Transform (SIFT) to detect key points inside the cells in each frame. The tracking of the SIFT feature points along succeeding frames of cell sequences is proposed utilizing a structural locality preservation (SLP) approach based on the Laplacian Eigenmap. This approach is completely unsupervised, thus reducing any human labor associated with annotating cells. However, this approach is only able to correctly match cells around 30% of the time on BPAE, MDBK, U2OS, APM, 3T3, Pl1Tu cell lines. Furthermore, this method does not provide explicit mitosis detection or lineage tracking.

B. The Cell Tracking Challenge

The Cell Tracking Challenge [46], [67] is a benchmark for segmenting and tracking moving cells in time-lapse video sequences. The challenge provides public datasets consisting of 2D and 3D microscopy videos of a wide range of cell lines imaged in different modalities. The quality of sequences vary

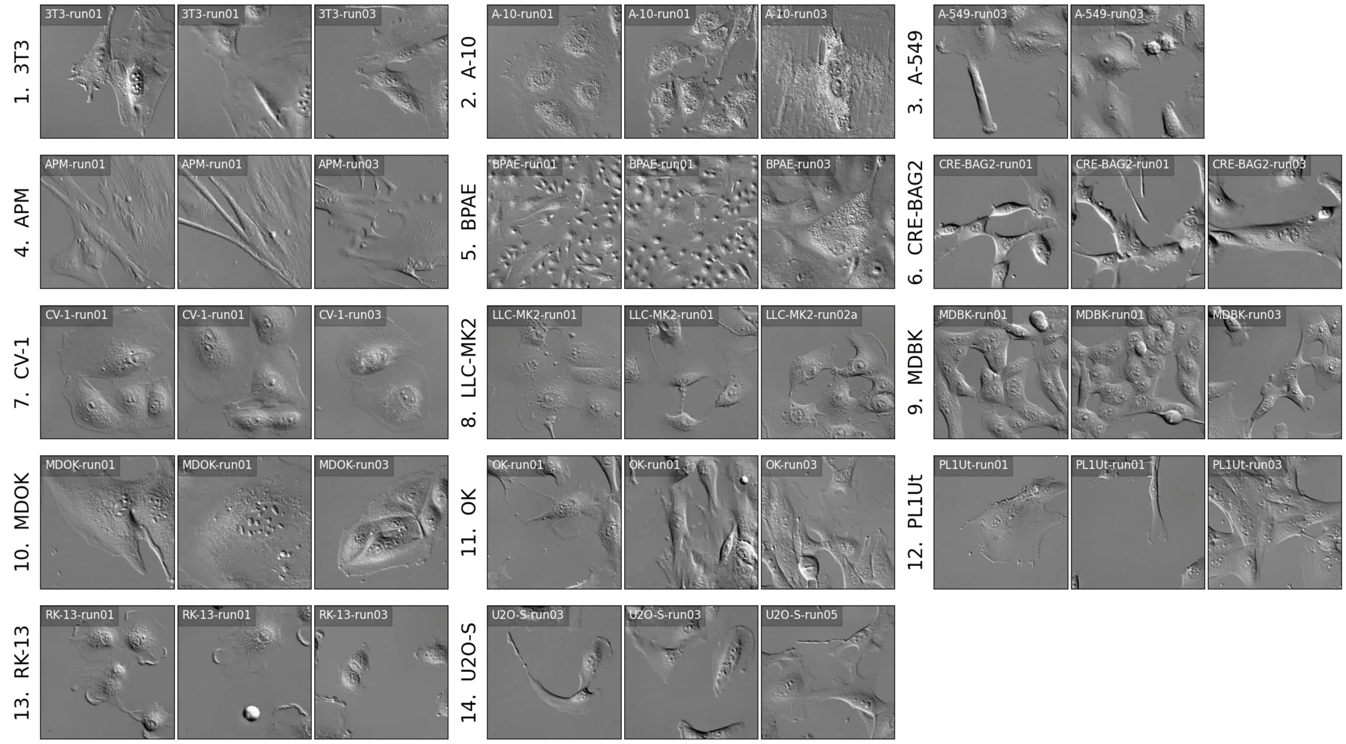


Fig. 2. Three sample frames from the training set of CTMCv1 for each of fourteen cell lines. The first two frames are taken at different time-frames from a single video sequence while the third frame is from a different sequence of the same cell type. Each cell line is given a numerical id $i \in \{1, 2, \ldots, 14\}$ so they can be easily identified. For cell line A-549, a single sequence for training is provided, hence, only two frames are shown from that sequence. The full name and cell type associated with each cell line is provided in the Supplementary Materials.

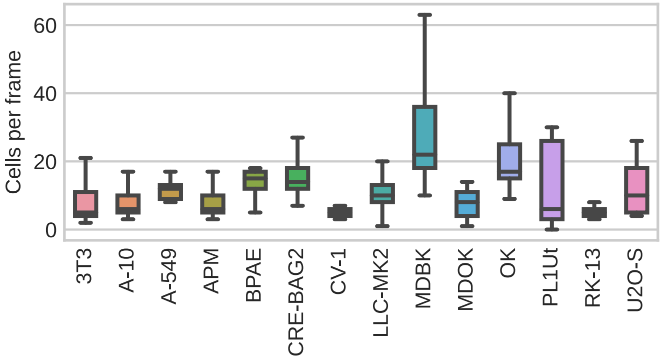


Fig. 3. A box plot showing the distribution of the number of cells per frame for each of the videos in each of the 14 cell lines in the public CTMCv1 video microscopy dataset. The box plot does not show the outlier cell counts for clarity; the upper and lower tick mark range is 5% to 95% in matplotlib.

due to difference in spatial and temporal resolution, noise levels, etc.

For the DIC-Hela[1] sequences in the Cell Tracking Challenge, the state-of-the-art method (user MU-Lux-CZ [41]) uses a pipeline that consists of pre-processing, segmentation, post-processing, and tracking. Pre-processing steps include histogram equalization or median normalization. For segmentation, a U-Net was trained to predict binary cell masks as well as cell markers. The cell markers are generated from the ground truth cell masks by eroding each cell. A pixel-wise weighted loss function is employed during training where pixels closer to cells are assigned higher weights. In the post-processing step, a watershed algorithm is used to transform output markers and masks into an instance segmentation map.

Interestingly, the tracking algorithm is rather simple: for each frame, if a cell has an overlap of at least 15% with a cell in the previous frame, the ID from the previous frame is then carried to the cell in the current frame. If two cells satisfy the overlap threshold, both cells are assigned new cells and are considered daughter cells. Cells that do not satisfy the overlap threshold are assigned new IDs.

C. Dual-Stream Marker-Guided Network (DMNet)

DMNet [4] proposes a cell segmentation tracking algorithm for DIC sequences in CTC among other cell lines and imaging modalities. DMNet solves the problem of semantic segmentation using a dual-stream segmentation network. Their architecture concurrently generates a binary semantic segmentation of the cells and a marker mask for cell centroids. The marker mask provides sufficient cell separability to apply a connected components algorithm to mark each cell with a unique identifier. A watershed algorithm is then applied on the marker mask and cell mask to create a full instance segmentation mask. This method uses a complex deep architecture HRNet [69], therefore requiring more data size and diversity with accurate cell outline

[1] Details on DIC-HeLa dataset are reported in the Cell Tracking Challenge website http://celltrackingchallenge.net/2d-datasets/

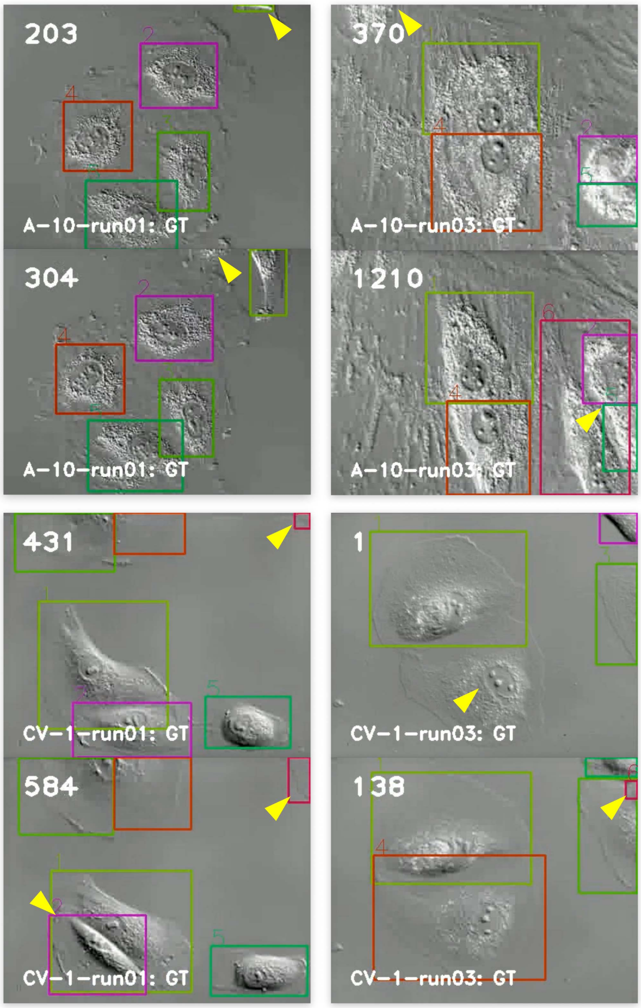


Fig. 4. Notable imperfections on the CTMC dataset ground-truth. Although the dataset provides mostly perfect ground truth. There are inconsistencies regarding cells around edges where they are sometimes annotated and sometimes not. A number of cell detections were missing in a few frames (CV-1-run03, cell id 4). The bounding box of some large cells sometimes fully or partially contains/encloses other cells. This is a limitation of using bounding box annotations which is faster but makes training difficult in comparison to instance segmentation.

annotation that is currently severely limited publicly in the DIC imaging space.

D. Mainstream Deep Tracking Algorithms

Up until the rise of deep learning, tracking has been predominantly approached using a variety of linear assignment algorithms such as the Hungarian algorithm [31]. Multi-object tracking was achieved by solving for two-frame assignment that minimizes a cost matrix. The cost is computed using spatial features (i.e., bounding box intersection over union (IoU), distance between centroids, etc.) and/or appearance features. Appearance features in the classical literature consisted of unsupervised hand-crafted features such as SIFT [37], SURF [5], ORB [60], DCT [20] or others. Alternatively, learned features can be used for appearance-based tracking [6], [52], [72]. The Multi-Domain Network (MDNet) [52] is one of the earlier works on Deep tracking and was motivated by the success of Convolutional Neural Networks (CNNs) in many vision domains. MDNet employs a simple CNN backbone for general visual representation, followed by a domain-specific layer for classification. Multi-domain learning is applied online during tracking for each sequence to adapt to the new video. Deep Simple Online and Realtime Tracking (DeepSORT) [72] works differently in that it uses learned appearance features to track multiple objects. To achieve this, a discriminator is trained offline for visual recognition. The network is suited for tracking context because it was trained on a sizable person re-identification dataset. For tracking, it uses the classic Hungarian algorithm for association considering a cost matrix derived from the appearance features. Tracktor [6] (tracked without bells and whistles) is fundamentally different than linear assignment methods. Tracktor turns a detector into a tracker using the bounding box regression of an object detector to predict the position of that object in the following frame.

TABLE II
Characteristics of the CTMCv1 Videos for Cell Tracking and Mitosis Challenge Benchmark Across Different Cell Lines and Cell Cycle Dynamics

Imaging modality	Differential Interference Contrast (DIC) microscopy
Number of sequences	86 with 49 for training and 37 for testing; 4 to 10 sequences per cell line in total.
Number of cell lines	14 lines covering 3 cell types (fibroblasts, myoblasts, epithelial) from 5 species (human, mice, rat, raccoon, rabbit).
Number of cells (tracks)	2,900 in total; the number of cells per video varies from 1 to 185.
Number of frames	152,584 total; number of frames per sequence varies from 294 to 4,438; 320×400 pixels, 30sec sampling.
Number of detections	2,045,834 in total; the number of detections per sequence varies from 1,196 to 172,074.
Cell density	13 cells per frame on average. Density by cell line ranges from 3.42 to 27.88.
Mitosis events	457 mitosis events; with 2 to 115 cell divisions by cell line.

E. Ensemble Methods for Computer Vision

Ensemble methods aim to combine the predictions of multiple individual models to achieve better results than any single model alone [70]. Ensemble methods in deep learning, particularly in computer vision tasks such as classification, segmentation, and detection, have proven to be highly effective in improving overall performance and robustness [30]. In classification tasks, ensemble methods can be employed by training multiple deep

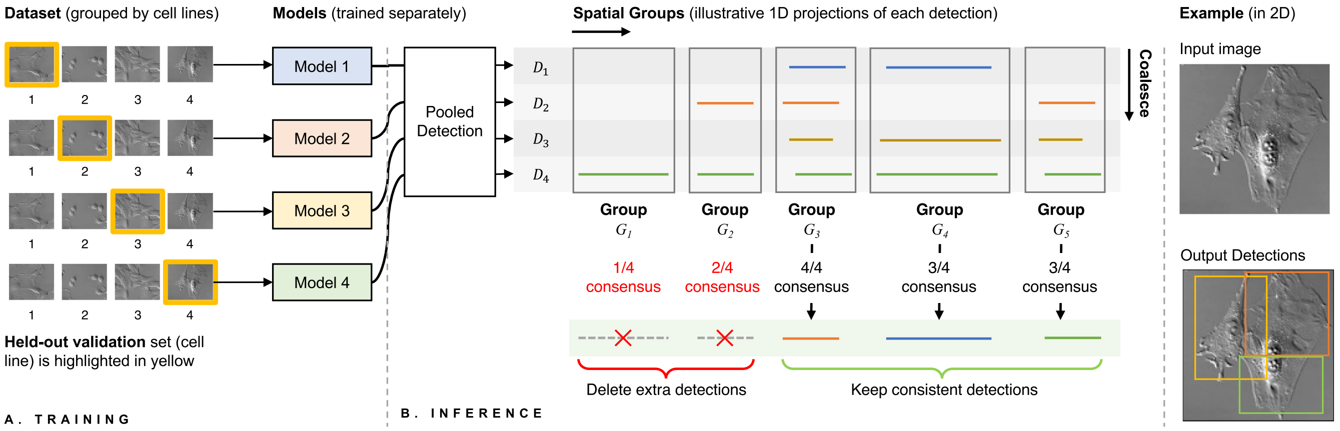


Fig. 5. General overview of the proposed ensemble detection network (EDNet). k detection sets D_i are generated from k detection models trained on different subsets of the training dataset ($k = 4$ in the diagram shown). These sets are pooled together in a single set D, then grouped together such that for each pair of detections in the same group $d_i, d_j \in G_k$, $\mathbf{IoU}(d_i, d_j) > 0.5$. Groups with fewer detections than half the number of models (i.e., 2) are discarded. The remaining groups are coalesced into a single detection by picking the detection with the highest confidence score.

neural networks with different initializations or architectures and then combining their predictions through voting or averaging [30] or more sophisticated methods like boosting [42]. This approach helps mitigate overfitting and reduces the impact of model biases. For image segmentation, ensemble methods can be applied by training multiple models on different subsets of the training data or using different architectures, and combining their predictions either by averaging the pixel-wise probabilities or by integrating the outputs using methods like conditional random fields [73]. In object detection, ensembles can be constructed by training multiple detection models, each with its own set of parameters or network architectures, and combining their outputs to improve both accuracy and robustness [13]. These ensemble methods leverage the diversity of multiple models to enhance performance, increase generalization capability, and address the inherent uncertainties and complexities of computer vision tasks. Although ensemble methods significantly increase computational cost, there has been ongoing research in improving their performance using multistage filtering [70].

III. Datasets

A. Cell Tracking and Mitosis Challenge (CTMC) Dataset

CTMCv1 dataset introduces a human-annotated live-cell video larger and more diverse than prior cell tracking datasets. The dataset consists of 86 live-cell imaging videos that represent 14 different cell lines of various shapes and sizes, with each cell annotated using a bounding box. Table II summarizes the statistics of the dataset and Fig. 2 shows examples of each cell line in the given data. Fig. 4 shows challenges associated with annotating this dataset by human annotators. Additionally, Fig. 3 is provided to show the cell count distribution across different cell lines; and Table I shows statistics of the number of frames for different sequences in each cell line. This dataset is part of the Multi-Cell Tracking with Mitosis Challenge. The CTMCv1 dataset poses different challenges. Namely, low contrast, low resolution (400×320) pixels, unclear cell edges, and MPEG compression artifacts. Additionally, each cell line provides an average of two sequences, both of which contain a significant number of similar frames. This proposes a poorly distributed dataset despite its large scale, which, in turn, is a major cause of model over-fitting. Additional tracking challenges are caused by cell overlapping and cell mitosis (cell division) events. Ideally, the two resulting cells should have two new distinct labels that point to their parent cell. Alternatively, this is sometimes handled by passing the parent ID to one of the two daughter cells (our approach), or simply left unhandled.

TABLE III
Number of Total Cells, Average Cells per Frame, and Number of Mitosis Events for Both Sequences in the MSC Dataset

	#Cells	Avg. #Cells/frame	#Mitosis events
MM1.1	45	29.06	2
CME1.1	128	72.73	12

B. Muscle Stem Cell (MSC) Brightfield Microscopy Videos

In order to evaluate the generalization and transfer learning performance of the proposed EDNet we use two live cell brightfield microscopy videos of skeletal muscle stem cells (also known as muscle satellite cells) [16], [40], [63]. These muscle stem cell (MSC) videos for *external* evaluation are from a different cell type and different imaging modality compared to the CTMC dataset. The MSC live cell imaging data were collected as part of an experiment to identify protein components in crushed muscle extract (CME) that are motogenic and/or chemotactic for satellite cells [47]; CME is a complex mixture of proteins and is a well-recognized physiological stimulus for these activities [8]. In this work, we use a total of 6 videos collected from two different *wells*. Each well contains 3 videos that correspond to 3 different *fields*. The first well features muscle stem cells grown in minimal media (MSC-MM), while the second well contains the same type of cells grown in minimal media supplemented with Crushed Muscle Extract (MSC-CME). We, therefore, refer to the 6 videos as: MM1.1, MM1.2, and MM1.3 for the different fields in the MM well; and CME1.1, CME1.2, and CME1.3 for the different fields in the CME well. Each timelapse video was imaged at 5-minute intervals for 12 hours for a total of 145 frames. It should be noted that both sequences contain both viable and dead cells throughout the entire sequence or part of the sequence. CME well is expected to contain a greater number of cells, owing to the better cell growth environment provided by the media compared to MM. Healthy cells typically exhibit halos, smooth boundaries, regular appearance, and motion, while dead cells may display an irregular and rough appearance, a bumpy surface, lack of motion, or may get swept up by motile live cells. The initial and emerging heterogeneity among the primary cells in both viability and motility highlights challenges in detection and segmentation as well as opportunities for identifying biological relevance in investigator-supplied datasets.

Two of the MSC sequences (MSC-MM1.1 and MSC-CME1.1) were annotated using v7Labs[2] where a bounding box was manually marked for each cell in every frame of the video. Table III summarizes statistics associated with the two labeled sequences of this dataset. The exact cell boundary (instance segmentation mask) was automatically extracted using the bounding box input and an object-agnostic segmentation model [26]. Fig. 6 shows an example annotated sequence (MSC-CME1.1) and highlights the differences in zoom level, resolution, and illumination, compared to the CTMC dataset the task of predicting cells in this environment can be challenging when training using CTMCv1 dataset that has vastly different properties.

Additionally, for each of the 6 videos centroids for 4-5 cells were tracked manually by a human annotator in order to quantify cell motility between the different experiments (MM vs. CME).

IV. Methods

We describe our method as two-fold. Given a sequence of images, we first detect cells in each frame independently. Second, we track the cells by linking ids of the same cell in time using spatial queues.

A. Deep Detection Network

1) Architecture: Prior to YOLOv3, advancements of this mainstream detection network have been focused on neural architectures. YOLOv4 [10] is the first YOLO architecture that was proposed by other than the original YOLO author (J. Redmon). YOLOv4's incremental improvement on the mAP score on the MS COCO dataset [34] is attributed to data augmentation and training techniques "bag of freebies" rather than architecture improvements. Minor architecture developments have also been introduced with YOLOv4 (Weighted-Residual-Connections (WRC), Cross-Stage-Partial-connections (CSP), Cross Mini-Batch Normalization (CmBN), and Mish-activation). YOLOv7 [68] built upon YOLOv4 and focused on improving inference speed while maintaining a competitive performance on MS-COCO [34]. In our experiments, we focus on comparing the neural architectures of these different iterations of models without changing the training methods. Therefore, better performance in newer architectures is not guaranteed over YOLOv3. After an ablation study on different architectures of YOLO [10], [56], [68] (Table V). We adopt an off-the-shelf YOLOv3 [56] and apply it to our framework (EDNet-YOLOv3). Our proposed methods for training a generalized detection and tracking pipeline are applicable to any detection method. Thus, this work is orthogonal to research in deep detection networks. Similar to the Single-Shot Detector (SSD) network [35], YOLO is trained to perform classification and bounding box regression simultaneously (unlike other detection networks such as R-CNN [22] and Fast R-CNN [21]). This enables YOLOv3 to run inference in real-time.

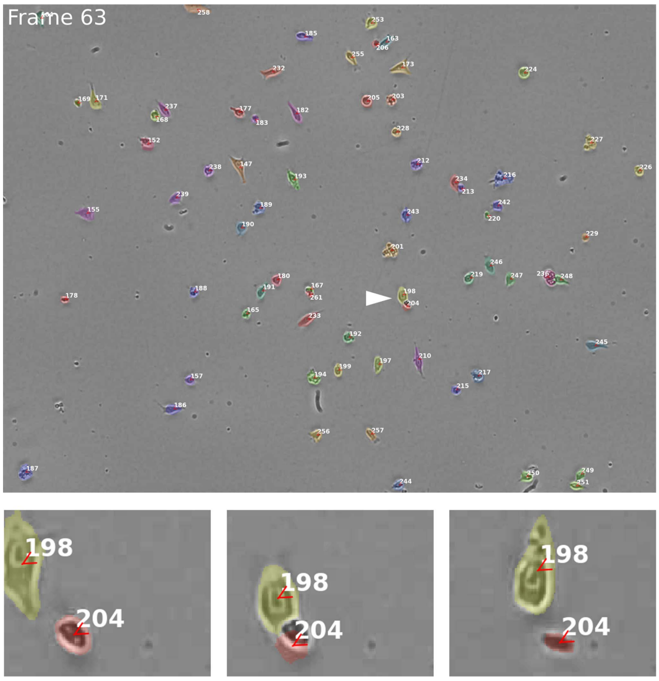


Fig. 6. A sample frame from the MSC-CME1.1 sequence showing manual cell segmentation masks. We show inset views near the white arrowhead over three time steps (Frames 60, 63, 72). Cells may die during the experiment but retain their shape and these apoptetic cells would need to be filtered out for velocity estimation. For example, Cell 204 is a dead cell but has some motion due to cell 198 sweeping it along as it moves.

2) Training Configuration: The utilized YOLOv3 model is initialized with pretrained weights from the ImageNet [30] dataset. The final prediction layer is modified for a single "cell" class, therefore it is initialized randomly using Kaiming He initialization [25]. During training, we minimize a sum of squared error loss function for bounding box regression, and a cross entropy loss for bounding box confidence. All training in this work uses the same hyper-parameters: mini-batch size of 48, an input resolution of 416×416, a learning rate of 10^{-3}. We allow the training to run for a maximum of 500,000 iterations and we multiply the learning rate by 0.1 at step 400,000 and once more at step 450,000.

[2] v7Labs Darwin tool: https://darwin.v7labs.com

B. Training for Generalization

Like most deep tracking network architectures, YOLOv3 is quite complex and requires extensive and well-distributed datasets to train. As such, deep detectors can easily fit the CTMCv1 data. Hence, we introduce a cross-cell-type validation method. We train 14 folds; in each fold, we leave one cell line out for validation and train on the remaining 13 cell lines. By excluding a whole cell line from training and using it for validation, we train our base models to generalize to new cell lines. Furthermore, to address the limited size of the data, we apply random online augmentation during training. The utilized image augmentation techniques include random rotation, flipping, translation, scaling, illumination, and sharpening.

C. Few-Shot Learning for Generalization Across Imaging Modalities

In order to test EDNet's generalizability across datasets, we perform zero-shot, few-shot and full fine-tuning experiments and evaluations on the MSC dataset. For zero-shot learning, we run inference using the model trained in Section IV-B directly on the MSC-MM1.1 test sequence. For few-shot learning, we train on a few frames from MSC-CME1.1 and run inference on MSC-MM1.1 sequence. We demonstrate that our model can generate reliable detections based on training on a few frames of an external dataset. Finally, we run an experiment where we train on a full sequence (MSC-CME1.1) of 145 frames. Performing zero-shot, few-shot, and full fine-tuning experiments on a new dataset like MSC allows us to investigate the generalizability of the EDNet model and gain insights into the amount of data required for effective fine-tuning on datasets with different modalities and resolutions.

D. Model-Agnostic Ensemble Detection Network (EDNet)

For inference, we take inspiration from an ensemble technique from [13] to design an ensemble detection network (EDNet) that is independent of the base model as described in Fig. 5. For a given frame, let us consider each set of detection outputs $D = \{D_i | i = 1, 2, \ldots 14\}$ of different lengths $m_i = |D_i|$ produced by each of the 14 models. Each output $D_i = \{d_{ij} | j = 1, 2, \ldots, m_i\}$, is a group of bounding box detections with d_{ij} representing a single (j^{th}) detection from the i^{th} model. We first flatten the outputs as:

$$D = \left\{ d_k \mid k = 1, 2, \ldots, \sum_i m_i \right\}$$

Then, we group bounding boxes together based on their **IoU** defined as the area of intersection over the area of union as follows:

$$\mathbf{IoU}(d_i, d_j) = \frac{\mathbf{area}(d_i \cap d_j)}{\mathbf{area}(d_i \cup d_j)} \tag{1}$$

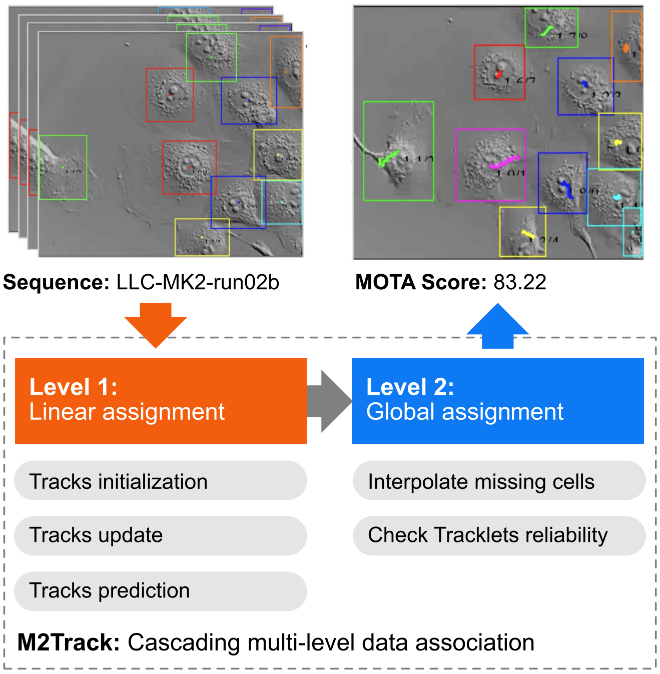


Fig. 7. An overview diagram of M2Track: the used tracking algorithm. Level 2 adds an additional layer of robustness and further improves detection through interpolating missed detections and handling mitosis.

The output of the grouping will be a list of detection groups $G = \{G_1, G_2, \ldots, G_n\}$ with a single detection group $G_i = \{d_{ij} | j = 1, 2, \ldots, |G_i|\}$. Each pair (d_{ij}, d_{ik}) inside a given detection group G_i satisfies:

$$\mathbf{IoU}(d_{ij}, d_{ik}) \geq \tau$$

where τ is the ensemble grouping threshold.

Finally, we use a consensus technique to only keep groups containing 50% the number of detector models we employ in our ensemble system (that is 7). We then pick the detection with the highest confidence score from each group; the final set of detections is $F = \{d_i | i = 1, 2, \ldots, n\}$ where:

$$d_i = \underset{d_{ij}}{\mathbf{argmax}}\ \mathbf{conf}(d_{ij})$$

where $\mathbf{conf}(\cdot)$ gives the confidence of a given input detection.

E. M2Track for Cell Tracking

Multi-cue Multi-object Tracker (M2Track) is a tracking-by-detection module that is adapted from our work on multi-cell tracking for cell lineage tracing in biomedical timelapse videos [4], BCTracker for tracking Myxococcus xanthus bacteria during swarming behavior [65] and our computer vision work on multi-object tracking for video surveillance [1], [2]. M2Track module is used to track the cells detected by EDNet-YOLOv3 (described in Section IV-D). M2Track is multi-step cascaded data association that links the detected cells and recovers from misdetections. M2Track module can explicitly handle cells entering and exiting the field of view, touching cells, mitosis or cell division events and cell fragmentation arising from birth and death of cells.

M2Track module in Fig. 7, uses detected cells as input, then performs cascade data association in two steps:

1) Short-Term Tracking: is a frame-to-frame data association and matching using bounding box intersection over union (**IoU**) score to generate cost matrix. The **IoU** cost matrix is used to optimize the associations of current N detected cells in D^t set at frame t to the M predicted tracks in T^{t-1} set at frame $t-1$, where the set of detections, $D^t = \{d_1^t, d_2^t, \ldots., d_N^t\}$ is assigned to the previously tracked objects $T^{t-1} = \{T_1^{t-1}, T_2^{t-1}, \ldots., T_M^{t-1}\}$, and T^{t-1} is the set of predicted cell trajectories previously tracked. Kalman filter [29] with constant velocity model is used to predict the motion of the cell in each frame. The optimization is implemented by minimizing the cost matrix using Munkres Hungarian algorithm [51] as:

$$\min_{b \in B} \sum_{i=1}^{m} \sum_{j=1}^{n} c^{ij} b^{ij} \tag{2}$$

where c_t^{ij} is the ith row and jth column entry in the cost matrix representing the cost of assigning detection j to tracklet i predicted bounding box location at time t and the cost value represents the **IoU** between i and j predicted and observed bounding box detections as:

$$c_t^{ij} = \frac{|d_i \cup d_j| - |d_i \cap d_j|}{|d_i \cup d_j|} \tag{3}$$

and b is a binary assignment matrix to ensure the rows and columns of b are summed to 1, with constraints,

$$\sum_{i=1}^{m} b^{ij} = 1 \qquad j = 1, 2, \ldots., n;$$

$$\sum_{j=1}^{n} b^{ij} = 1 \qquad i = 1, 2, \ldots., m.$$

To reduce false matches, eliminate low certainty association, and reduce computation cost, we use spatial circular gating regions around the predicted track positions. Association cost matrix decision assigns one out of four status (new track, linked track, lost track, and dead track) for each individual cell. Since this step considers only information from consecutive frames, having false detections, occlusions, and matching ambiguities causes track fragmentation. Further step is important to improve the performance.

2) Long-Term Tracking: Problems during object detection or data association process result in implicit fragmentation of cells. Long-term tracking, which also called global assignment, uses information across long video segments. Increasing the number of frames or the number of cells makes this process expensive. Our approach optimizes fragmented tracklets re-linking by connecting cells at the track level rather than object level using spatial and temporal clues. Cell interpolation is used to recover missing cells due to miss-detection or occlusion events [1], [2], [4].

3) Lineage and Mitosis Tracking: Cell mitosis (cell division) event introduces a challenge during tracking. During lineage tracking, it is important to record the cell division events. During a mitosis of a given cell, the two resulting cells should be assigned new distinct labels. Both labels should point to the label of the parent cell that has undergone mitosis. To reduce the complexity, our pipeline handles cell mitosis during the tracking phase rather than incorporating additional steps to detect mitosis during the detection steps. During the association step, any new cell that appears in the scene assigns as a new track with a new ID. There are three scenarios for the appearance of new cells in the scene: (i) a new cell entering the science which is usually happened from the edges of the frames. The short-term tracking step handles this part by assigning a new ID to the new cell entering the scene; (ii) when occlusion events are ended and the cells usually re-appear again in the scene, the long-term tracking step handles this part by reconnecting the cell after occlusion event; (iii) and during cell mitosis event.

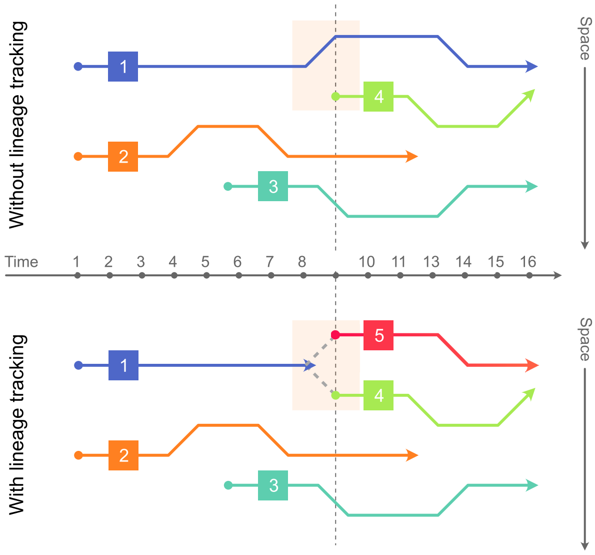


Fig. 8. Illustrative example of M2Track performance with (top) and without (bottom) cell-lineage tracking in generalized coordinates where the x-axis represents time and the y-axis is arbitrary projected spatial units. A mitosis event is highlighted in both scenarios. For simplicity, we assume a 1-D+time space where the horizontal axis represents time and the vertical axis represents space. When lineage-tracking is turned on, a mitosis event is detected at time $t = 9$, and cell #1 is corrected as a new cell from that point forward rather than cell #1 persisting over time. For a real example, see Fig. 12.

For detecting cell mitosis events, the appearance of new cells needs to be distinguished from the cells that enter the scene, which usually happened from the edge of the frames. This is described in Algorithm 1 and visually explained in Fig. 8. Additionally, Fig. 12 shows an example of using mitosis detection tracking on a real sequence from the test set (A-10-run02).

V. Experimental Results

A. Evaluation Metrics

Different evaluation metrics have been proposed for multi-object tracking [64], [71]. To report the results of our pipeline, we adopted Multi-Object Tracking (MOT) challenge evaluation metrics summarized below and described in [39], [64]. The toolkit for MOT benchmark evaluation provided in [18] was used to evaluate tracking results for training; the public benchmark was used to evaluate performance on the hidden test set. Below

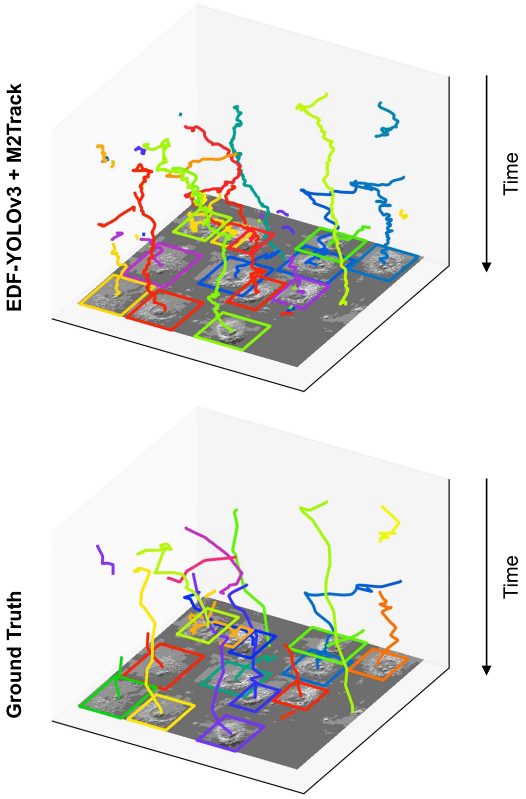


Fig. 9. A complete space-time track representation using the output of our EDNet-YOLOv3+M2Track cell tracking pipeline on the LLC-MK2-run03 sequence. Ground-truth tracks look particularly smoother due to interpolation applied during annotation to reduce the number of manually annotated frames.

is a brief description of the MOT evaluation metrics used in this work:

1) **MOTA** Multiple Object Tracking Accuracy: the most popular metric, is calculated as a function of false positives (FP), true positives (TP), and false negatives (FN) per frame. MOTA is reported as a percentage that takes values in $[-\infty, 100]$:

$$\textbf{MOTA} = 1 - \frac{\sum_t (FN_t + FP_t + IDSw_t)}{\sum_t GT_t} \tag{4}$$

Where t is the frame index and GT is the number of ground-truth objects. FP, TP, and FN count are determined by thresholding the intersection over union (**IoU**) between predicted and target bounding boxes using a threshold $d = 50\%$.

2) **ID Sw** Identity Switches: counts the number of times that the matched identity of a tracked trajectory changes.

3) **Frag** Fragmentation metric: is the number of times that trajectories are fragmented.

4) **MT** Mostly Tracked metric: computes percentage of trajectories (with respect to the number of ground-truth trajectories) tracked accurately for more than 80% of the trajectory duration.

5) **ML** Mostly Lost metric: computes the percentage of trajectories tracked accurately for less than 20% of the trajectory duration.

Algorithm 1: Mitosis Lineage Detection.

```
Input: Tracking information:
  $Q$: set of detected tracks,
  $G$: cell-acyclic-graph represented as a list of track
       summaries $\{id, frame\_start, frame\_end, parent\_id\}$,
  $H$: tracks history, a list of tracks' spatial information in
       each frame $\{frame, id, x, y, w, h\}$
  $\sigma$: distance threshold,
  $\beta$: frame margin
Output: Updated $G$, updated $H$
// Get tracks that enter the scene from
   known spatial entering points
$E \leftarrow \{Tr \mid Tr \in Q$ and $(Tr_x, Tr_y)$ within $\beta\}$
// Remove all tracks $\in E$ from $Q$
$M \leftarrow Q - E$
// Check whether each track is a result of
   mitosis
for $ids \in M$ do
    // The set of tracks that has distance
       less than $\sigma$ with current track $ids$ in
       set $M$
    $L \leftarrow \{ Tr \mid Tr \in Q$ and $\|ids_{x,y} - Tr_{x,y}\| < \sigma$ and
      $ids_{frame\_start} = Tr_{frame} \}$
    if $L \neq \{\}$ then
        // first daughter
        $d_1 \leftarrow ids$
        // Second track involves in mitosis
           event need to be fragmented
        $R \leftarrow$ closest_tracklet$(L)$
        // Divide track $R$ to be the parent and
           the second daughter and take the
           first part of $R$ as a parent
        $p, d_2 \leftarrow$ split_track_by_frame$(R, frame =$
          $ids_{frame\_start})$
    Update $G$
    Update $H$
```

6) **IDF1** Identification F1 metric: measures the ratio of correctly identified detections over the average number of ground-truth and computed detections using (5). It is expressed in terms of $IDTP$ True Positive ID, $IDFP$ False Positive ID, $IDFN$ False Negative ID. This metric ranges between 0 and 100%,

$$\textbf{IDF1} = \frac{2IDTP}{2IDTP + IDFP + IDFN}. \tag{5}$$

7) **TRA**: Tracking or lineage metric as described by Cell Tracking Challenge [67] calculates the number of graph edit operations required to match the predicted trajectory graphs with the human ground-truth,

$$\textbf{TRA} = 1 - \frac{\textbf{min}(AOGM, AOGM_0)}{AOGM_0} \tag{6}$$

where $AOGM$ is the edit cost for transforming predicted trajectory graph to ground-truth graph, and $AOGM_0$

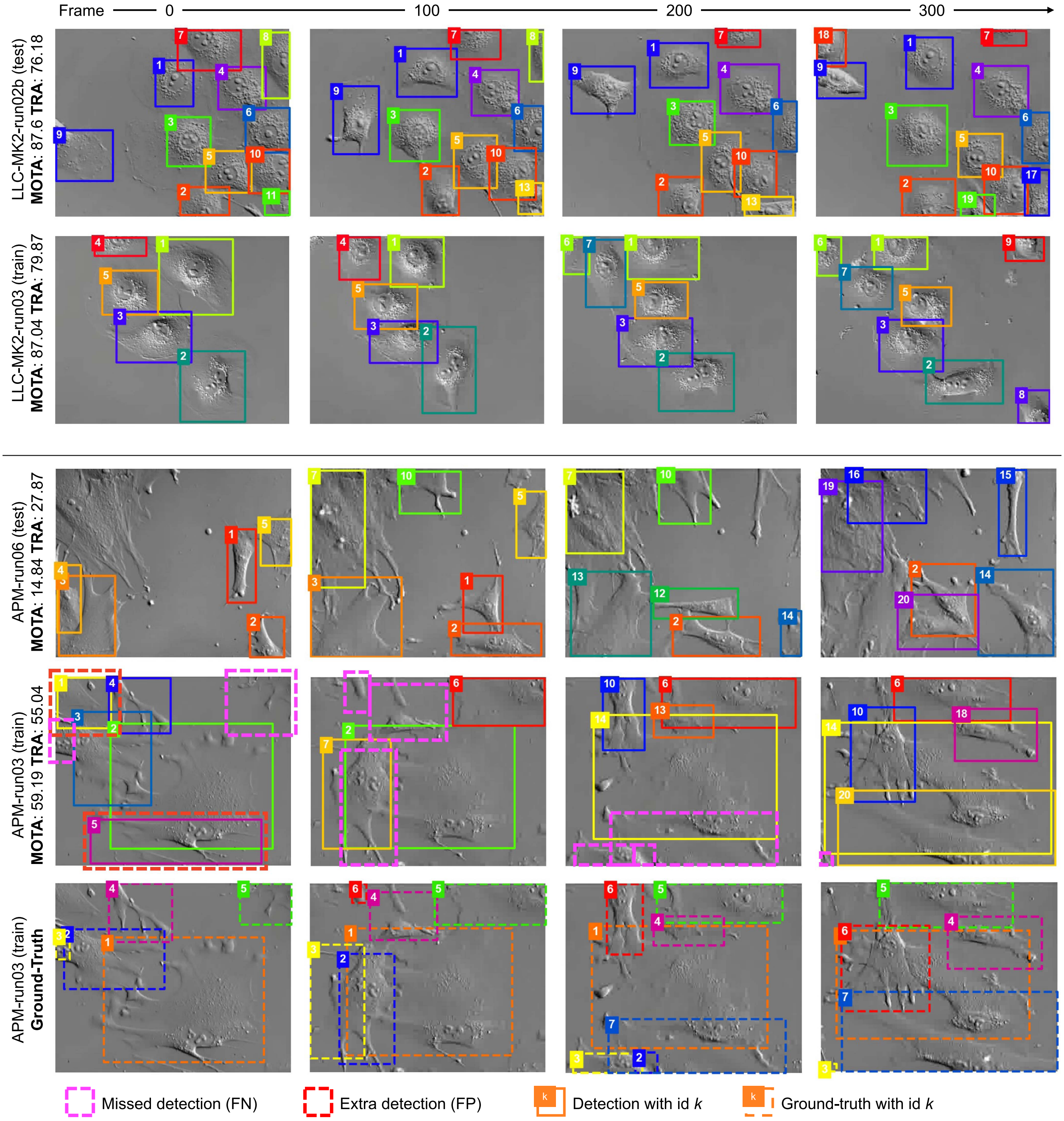


Fig. 10. Qualitative results of our method (EDNet-YOLOv3 + M2Track). Results of LLC-MK2 (easy) and APM (challenging) cell lines for samples in a testing sequence (top) and a training sequence (bottom). We demonstrate how effective our approach is in generalizing across training and testing (unseen data). The proposed method has no errors for LLC-MK2 in the shown frames (top two rows). APM cells show more variance in appearance across frames due to their deformable shape dynamics that changes more significantly than LLC-MK2 cells. Missed detections and extra detections are highlighted for the training set (where ground truth is known). For better clarity in the APM-run03 sequence, we add a third row showing just the ground-truth annotations.

is the cost for creating predicted trajectory graph from scratch (i.e., assuming empty tracking results). Fewer graph edit operations result in higher values of the TRA score which indicates better lineage tracking performance.

It is worth noting that compared with the $MOTA$ score, the TRA score is more sensitive to “mitosis” or cell division detection and cell lineage tracing. This is due to the fact that in order to score perfectly in TRA, parent-daughter relationships resulting from cell divisions need to be captured corrected in the cell IDs and lineage graphs. Whereas in the $MOTA$ score, mitosis parent cell and daughter cells are treated independently.

B. Evaluation Results

Figs. 1, 10, and 11 present a selection of sample results. Specifically, Fig. 10 offers an in-depth examination of the

TABLE IV
SUMMARY RESULT TABLE SHOWING THE MOTA SCORE FOR DIFFERENT PROPOSED METHODS COMPARED TO THE BASELINE

	Avg. per sequence		**Avg. per cell line**		**Avg. per cell**	
Method	Train	Test	Train	Test	Train	Test
Baseline (FasterRCNN + Tracktor)	-	23.85	-	22.24	-	35.10
Cell-type specific YOLOv3	**90.37**	22.92	**89.98**	19.67	**88.92**	24.16
YOLOv3 Model I	73.10	34.04	75.60	33.70	75.30	37.48
YOLOv3 Model II	89.96	38.92	89.80	37.53	88.86	44.72
YOLOv3 Model III	63.73	41.95	66.33	42.45	65.64	42.34
EDNet-YOLOv3 (ours)	81.40	**48.08**	80.83	**48.60**	77.74	**50.55**

YOLOv3 Model I: Pool data together and select randomly 80% for training and 20% for validation (no ensemble). YOLOv3 Model II: Leave one sequence out from each cell line for validation (no ensemble). YOLOv3 Model III: Leave one cell line out for validation (no ensemble); in our experiments, the cell line that was left is 3T3. All of the methods below (excluding the baseline) use M2Track for tracking.

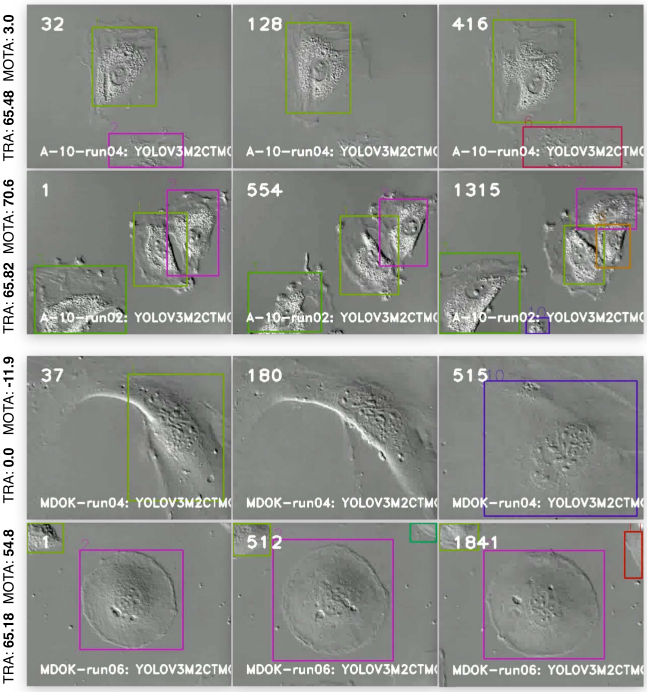


Fig. 11. Qualitative EDNet+M2Track tracking results for two difficult cell lines (test sequences) using MOTA and TRA scores. In Row 1, for cell line A-10 (run04): extra cell detections reduce the MOTA score but TRA is still above average due to tracking accuracy. In Row 2, for A10-run02 both EDNet detections and M2Track cell lineage have high accuracy with three cells resulting in high MOTA and TRA. In Row 3, for cell line MDOK (run04): missing detections due to unusual cell shape and appearance lowers MOTA and TRA since there is only one cell to track. In Row 4, for cell line MDOK (run06): the three cells in the shown frames are detected and tracked accurately over a large number of frames so the MOTA and TRA scores are high.

qualitative results obtained from using the EDNet+M2Track method on both training and testing sets. This figure is crucial in demonstrating the ability of our method to generalize from the training to the testing set. Fig. 11 showcases the challenging sequences where our tracker did not perform as expected, highlighting its limitations. Additionally, Fig. 9 provides a valuable perspective by representing our tracking results on select sequences through a space-time visualization.

To illustrate the difficulty of generalization, we report the results for the different methods on both the training and testing sets in Table IV. The results for the hidden test sets were generated using the public benchmark [7], [33], [58], [67]. As can be seen in Table IV, a cell-specific model – that is trained on specific cell lines – performs best on training data and worst on testing data. This is a clear indication that training a different model for each cell line yields over-fitting. In contrast, a general model (Model I, where all data was pooled together to train a single detector) consistently results in a worse performance in training but a superior performance for the hidden test data.

TABLE V
AN ABLATION STUDY ON DIFFERENT YOLO ARCHITECTURES (V3, V4, AND V7)

Detector	**Strategy**	**MOTA**	**TRA**	**IDF1**
YOLOv3	Single	42.34	48.25	50.75
	EDNet	**50.55**	**52.51**	**56.72**
YOLOv4	Single	32.03	41.76	42.86
	EDNet	**40.69**	**46.97**	**46.33**
YOLOv7	Single	37.93	47.65	50.13
	EDNet	**43.06**	**50.29**	**45.42**

The results shown are on the hidden challenge test dataset of CTMCv1. YOLOv3 outperforms YOLOv4 and YOLOv7 architectures. EDNet consistently improves all detection models tested.

It can also be observed how deep detection networks (in our case YOLOv3) are sensitive to the way data is split into training and validation. We can see that random splitting (YOLOv3 Model I) yields a baseline score of 37.50. Leaving one sequence out from each cell line for validation (YOLOv3 Model II) greatly improves generalizability; giving a MOTA score of 44.72. Similarly, leaving-out an entire cell line for validation (YOLOv3 Model III) can also yield a comparable improvement, giving a MOTA score of 42.34. Finally, our EDNet applied to YOLOv3 (EDNet-YOLOv3) produces a state-of-the-art MOTA score on the public leader board with a MOTA score of 50.55.

We conclude that leaving a cell line for validation to train a deep detection model can greatly boost the model's generalizability. Furthermore, applying this method 14 times with each of the cell lines and combining the resulting models into one using our EDNet can produce the best cell detection framework on the CTMCv1 dataset.

Table VII shows the TRA performance on each cell line for our method in comparison to two state-of-the-art cell tracking algorithms: Viterbi [43] and DeepCell [49]. We show that our best method (EDNet-YOLOv3 + M2Track) outperforms these methods significantly. However, for cell lines with very few

TABLE VI
MOTA SCORES ON THE *TESTING SET PER CELL LINE* FOR OUR DETECTION METHOD EDNET-YOLOV3 WITH THREE DIFFERENT TRACKING ALGORITHMS TRACKTOR, DEEPSORT AND M2TRACK (OURS), COMPARED TO BASELINE CELL DETECTOR FASTERRCNN WITH TWO DIFFERENT CELL TRACKING METHODS TRACKTOR AND DEEPSORT

	FasterRCNN		**EDNet-YOLOv3**		
Cell Line (#Cells)	**Tracktor** [6]	**DeepSORT** [72]	**Tracktor** [6]	**DeepSORT** [72]	**M2Track (ours)**
3T3 (124)	34.33	-33.40	51.93	51.79	52.48
A-10 (14)	-18.37	-151.33	38.08	38.54	38.10
A-549 (82)	35.50	2.70	52.22	52.79	53.01
APM (90)	-31.00	-132.80	27.89	27.80	28.36
BPAE (114)	34.47	0.87	57.43	57.46	57.86
CRE-BAG2 (229)	25.75	-35.80	36.42	37.79	38.59
CV-1 (4)	-14.60	-135.90	41.02	41.44	40.50
LLC-MK2 (56)	57.00	28.67	64.32	64.62	64.18
MDBK (205)	65.04	40.00	67.32	67.14	67.88
MDOK (27)	-38.88	-151.05	20.40	20.25	20.45
OK (111)	33.20	-13.93	46.56	46.28	47.39
PL1Ut (71)	25.15	-14.15	39.40	39.33	40.26
RK-13 (20)	64.20	36.70	70.62	70.36	70.99
U2O-S (137)	58.50	45.65	59.66	60.10	60.34
Average MOTA	23.59	-36.70	48.09	48.26	**48.60**

Tracktor exploits the regression head of the Faster-RCNN network by providing region proposals based on detections from a previous frame. since YOLOv3 is a single-shot detector, tracktor cannot be easily integrated in YOLOv3. therefore, Ednet-YOLOv3 + tracktor experiment only uses the appearance and re-identification models as well as data association from tracktor [6]. using EDnet-YOLOv3, the average mota scores on the training set for tracktor, deepsort, and M2track are 77.44, 77.30, and 80.83 respectively

cells (A-10, and CV-1), Viterbi and DeepCell outperform our methods. Fig. 13 shows a comparison of how the performance of each method scales with the density of a given sequence. We note that, while Viterbi and DeepCell perform well on low-density sequences, they scale poorly with higher-density sequences. On the other hand, our method shows great robustness against cell density, and its performance is not correlated with the number of cells.

Table VI shows the MOTA metric for different mainstream detection and tracking algorithms in comparison to ours. FasterRCNN experiments were performed in the original work of the CTMC dataset [3]. We observe that using FasterRCNN architecture for detection, Tracktor is more effective than DeepSORT. This has been consistent with our experiments using EDNet-YOLOv3. Tracktor exploits the regression head of the Faster-RCNN network by providing region proposals based on detections from a previous frame. Since YOLOv3 is a single-shot detector, Tracktor cannot be trivially integrated in YOLOv3. In our experiments, EDNet-YOLOv3 + Tracktor experiment only uses the appearance and re-identification models as well as the data association algorithm from Tracktor. M2Track consistently outperforms Tracktor and DeepSORT despite being completely unsupervised and not using appearance information. Furthermore, Table VI shows that our detection method produces a significantly higher MOTA score using the same tracking algorithms (DeepSORT and Tracktor).

Tracking performance is affected by the quality of the object detection algorithm used. Table VIII shows tracking performance as reported on Cell Tracking and Mitosis Detection Challenge website.[3] Our EDNet-YOLOv3 method for detection combined with M2Track tracker outperforms in most of multi-object tracking metrics **MOTA**, **IDF1**, **TRA**, **MT**, **FN**, **ID Sw**, and **Frag**. As an ablation study, we substitute M2Track with DeepSORT (row 2 of Table VIII). M2Track provides global assignment and explicit lineage detection. As a result,

TABLE VII
COMPARING THE TRA TRACKING LINEAGE PERFORMANCE SCORES ON THE *TESTING SET PER CELL LINE* (USING HIDDEN TESTING SEQUENCES) OF OUR CELL TRACKING ALGORITHM (EDNET-YOLOV3+M2TRACK) WITH MU-LUX-CZ, THE BEST DIC CELL SEGMENTATION ALGORITHM IN THE CTC BENCHMARK. MU-LUX-CZ USES A CUSTOM U-NET FOR DETECTION WHICH IS USED IN COMBINATION WITH VITERBI AND DEEPCELL TRACKING ALGORITHMS IN THE ORIGINAL BASELINE PAPER [3]

Cell line	Mu-Lux-CZ		EDNet
(#Runs; #Cells)	Viterbi [43]	DeepCell [49]	M2Track
3T3 (4; 124)	0.38	0.15	0.57
A-10 (3; 14)	0.83	0.82	0.60
A-549 (2; 82)	0.12	0.00	0.58
APM (3; 90)	0.33	0.00	0.34
BPAE (3; 114)	0.30	0.46	0.58
CRE-BAG2 (2; 229)	0.07	0.00	0.49
CV-1 (2; 4)	0.89	0.86	0.52
LLC-MK2 (3; 56)	0.59	0.44	0.63
MDBK (5; 205)	0.38	0.48	0.63
MDOK (5; 27)	0.51	0.36	0.32
OK (3; 111)	0.25	0.00	0.49
PL1Ut (2; 71)	0.21	0.26	0.41
RK-13 (1; 20)	0.53	0.60	0.65
U2O-S (2; 137)	0.04	0.00	0.59
Average TRA	0.39	0.32	**0.53**

Number in brackets is number of cells across all runs. In comparison, our method significantly outperforms both methods with the exception of A-10, CV-1 and MDOK cell lines. these results were evaluated on testing sets. The training set average TRA for our method is 0.71.

[3] https://motchallenge.net/results/CTMC-v1/

TABLE VIII
TRACKER PERFORMANCE COMPARISON AS REPORTED ON CELL TRACKING WITH MITOSIS DETECTION CHALLENGE WEBSITE ON THE TEST DATA

Tracker	Cell Detection	MOTA↑	IDF1↑	TRA↑	MT↑	ML↓	FP↓	FN↓	ID Sw↓	Frag↓
M2Track	EDNet-YOLOv3	**50.6**	56.7	52.51	428	174	158,593	309,015	2,118	10,453
M2Track-M	EDNet-YOLOv3	49.6	53.4	**52.83**	**438**	166	172,538	**303,212**	2,630	**8,219**
DeepSORT	EDNet-YOLOv3	50.1	**58.5**	52.41	412	693	160,423	312,645	**1,406**	8,350
Tracktor	EDNet-YOLOv3	49.6	52.7	22.88	428	174	157,379	309,693	11,400	10,437
M2Track	DMNet [4]	39.0	41.0	36.95	219	317	**109,689**	466,667	3,142	20,829
Tracktor	FasterRCNN [57]	35.1	50.5	51.95	427	**126**	299,941	312,368	4,318	25,992

The TRA score is an average over all cells and shown as a percentage; note that in Tables VI and VII the averages are over cell lines . M2Track-M is a variation of M2Track that uses mitosis detection mechanism. Tracktor + fasterRCNN results are from the CTMC benchmark and DeepSORT + FasterRCNN results are not available.

TABLE IX
TRACKING PERFORMANCE ON THE MSE-MM1.1 TEST SET [40]

Method	MOTA↑	IDF1↑	TRA↑	MT↑	ML↓	FP↓	FN↓	ID Sw↓	Frag↓
Zero-shot	-14.143	25.106	18.775	2	17	1,951	3,329	86	395
Few-shot	11.652	35.697	27.355	4	13	1429	2,988	88	257
Full fine-tuning	15.781	41.896	34.984	5	9	1,658	2,658	93	266

Zero-shot, few-shot, and full fine-tuning results are reported.

we can see that M2Track outperforms DeepSORT in **MOTA** and **TRA** scores.

Overall, our method achieved the top one rank on the Computer Vision for Microscopy Image Analysis (CVMI) workshop at CVPR 2021.

C. Performance on External Data

We evaluate of our Ensemble Detection Network (EDNet) on the MSC dataset as external data. This data uses brightfield instead of DIC and was taken at a different zoom level and sampling rate with smaller cell sizes, making it challenging for benchmarking our model's generalizability. We use one sequence (MSC-CME) for training and one sequence (MSC-MM) for testing. We report the performance of zero-shot, few-shot, and fully fine-tuned EDNet-YOLOv3 on the testing set in Table IX. Fig. 15 shows a qualitative performance of EDNet-YOLOv3 on external MSC data. Despite the differences in imaging modality and sampling rate, our approach shows promising results for accurately detecting and tracking cells in this external dataset even with few-shot fine-tuning. This highlights the potential for our approach to be applicable in a range of cell tracking scenarios.

Fig. 14 shows a quantitative comparison of muscle satellite cell (MSC) velocities in two conditions: minimal media (MM) which would not be expected to stimulate motility, and crushed muscle extract (CME) which is a well-recognized physiological stimulus for satellite cell motility [8]. Human tracking of selected cells does not adequately differentiate between the two conditions, while the automated EDNet correctly identifies both greater total motility and a broader range of cell velocities in CME, as would be expected.

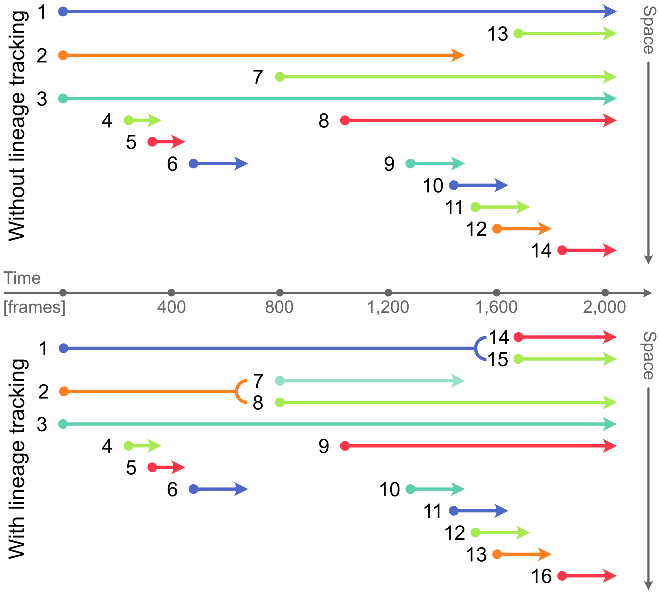


Fig. 12. Lineage and mitosis tracking for sequence A-10-run02 from the test set. The upper box shows the cell lineage tracking results *without* using mitosis detection, Track #1 was under mitosis event on frame 1605 but the tracking pipeline does not detect the mitosis, hence keeping #1 with the same ID and assigning a new ID (#13) to the divided cell (one of the daughter cells). The bottom box shows the lineage cell tracking results with the mitosis detection applied. In this case, at frame 1605, Track #1 is divided into #14 (first daughter) and #15 (second daughter) and #1 is assigned as a parent for both #14 and #15. The x-axis is time and y-axis is a graphical representation of the distance between cells not actual positions.

In Fig. 16 we compare human versus EDNet+M2Track performance by checking the track accuracy of the same small subset of cells tracked by the human in the three fields from each well. We first applied a track-to-track association step between human-generated tracks and EDNet-generated tracks, to find the algorithm-tracked cells associated with human-annotated cells. Overall there is good agreement between the human ground truth and our EDNet+M2Track algorithm and most of the average cell velocities in each field are within one standard error. The exception is MM1.2 where EDNet was not able to detect all

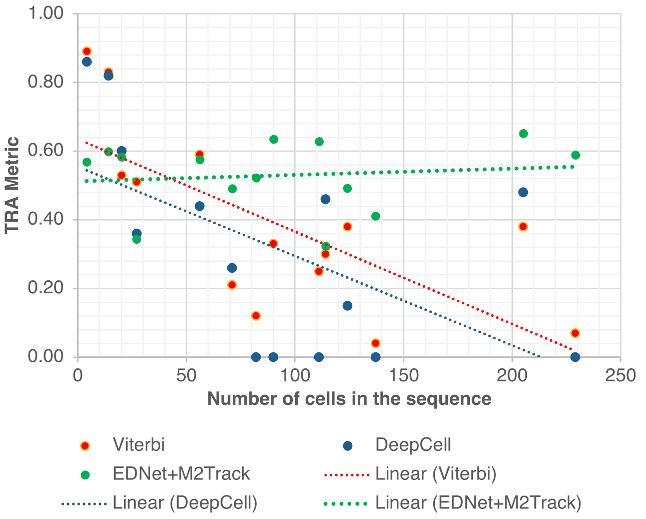
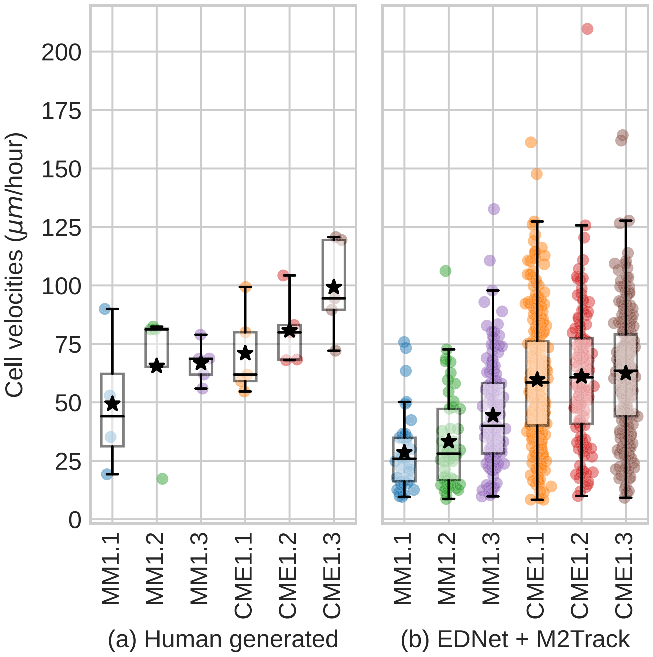


Fig. 13. A comparison of the TRA performance on CTMCv1 dataset for different cell densities (number of cells per sequence). The plot of Viterbi (blue dots) and DeepCell (orange dots) tracking results shows the decreasing and highly variable performance with increasing number of cells in a given sequence. Our EDNet method (yellow dots) shows robust and consistent performance across increasing number and density of cells in each sequence (black dotted line).



(a) Human generated (b) EDNet + M2Track

Fig. 14. A comparison of cell velocities at the well and image field level, between MSC-CME vs MSC-MM calculated using (a) Human-generated labels for select cells, and (b) algorithm-generated labels (using EDNet + M2Track). Human tracking consists of 4 or 5 cells per sequence, for EDNet, we track all cells detected (A total of 225 cells were detected in MM and 604 cells in CME). Each data point is taken as the mean velocity for a tracked cell.

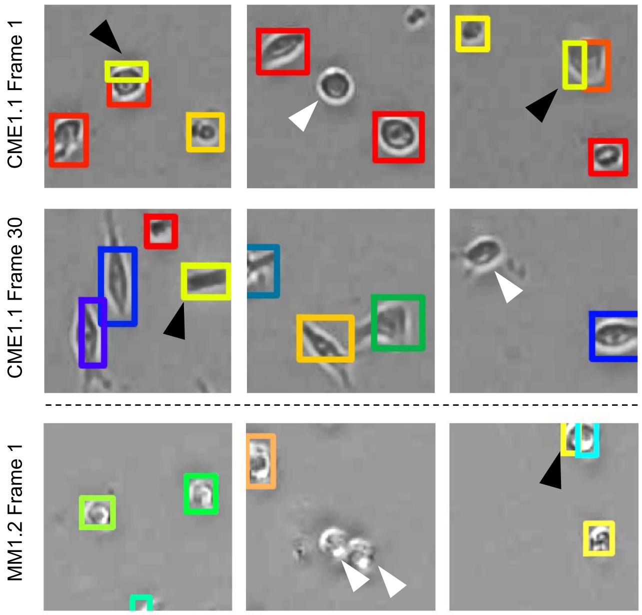


Fig. 15. Qualitative performance on external data in different zoomed regions of interest for two frames from the MSC-CME1.1 video and one frame from MSC-MM1.2 video (challenging case). Rows represent different points in time in this microscopy time-lapse sequence. Fine-tuned EDNet-YOLOv3 using few-shot learning was used to improve detection performance. Colored boxes are cell detections, black arrowheads highlight false positives and white arrowheads false negatives.

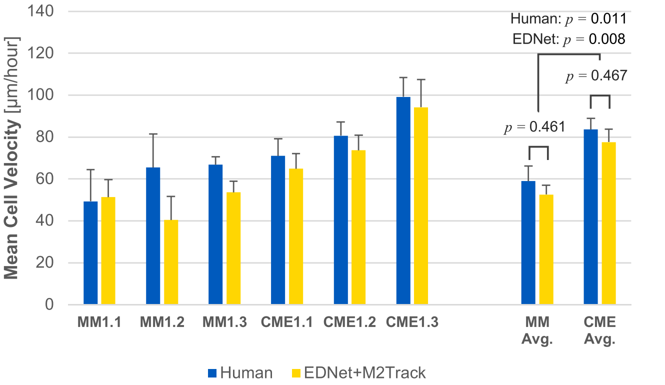


Fig. 16. A comparison of mean cell velocities of the cells tracked by human annotators. We use association based on track proximity to find all fragmented tracks from EDNet+M2Track that correspond to the same cells tracked manually. We were not able to track the same cells in MM1.2 due to missed detection of cells. This can be attributed to the fact that EDNet was trained on CME only. MM1.2 is therefore not used in the average computation. A t-test between the algorithm and human in cell velocities yields a p-value of 0.461 (MM) and 0.467 (CME). This indicates no significant difference in the distribution of velocity estimation in human vs algorithm.

cells in the field and so tracks became fragmented which led to ID switches, as EDNet was only trained on CME images to evaluate generalization performance. The variability between fields within the same well (where all conditions are the same) is due to a combination of cell selection bias by the human tracker as not all cells in the field are tracked and heterogeneity in cell behavior exists within the same population which is evident in Fig. 14. For operational use, the performance of EDNet would be improved by training on a large amount of MSC data in different media to improve detection performance when deployed in the laboratory. In terms of speed, EDNet+M2Track is able to track a full sequence in 30 minutes (unoptimized sequential version), enabling more high-throughput studies with statistical significance compared to current manual tracking which can take a week or longer.

VI. Future Work

As it stands, EDNet requires training multiple (m) networks, raising the computational complexity by a factor of m in comparison to base detection models. This can be addressed in future work by exploring weight sharing between networks. This can reduce the scaling from a linear function $\mathcal{O}(m)$ to a near constant function $\mathcal{O}(1)$. Furthermore, deep features can be added to the M2Track association step to take advantage of appearance during cell tracking. This will facilitate the analysis of primary cells or otherwise heterogeneous cell populations, which will be necessary to generate biologically relevant data.

VII. Conclusion

In this work, we propose a two-step detection and tracking pipeline that uses a model-agnostic EDNet for detection followed by a two-stage tracking algorithm (M2Track). Our detection framework can overcome over-fitting and achieve better generalization compared with other cell detection models including the best-performing model in the Cell Tracking Challenge DIC dataset. Furthermore, M2Track tracker outperforms the mainstream tracking algorithm DeepSORT as well as cell tracking algorithms (Viterbi and DeepCell) only using location features. It is also effective in analyzing a heterogeneous population of primary cells (skeletal muscle satellite cells) presented with different physiologically relevant stimuli, supporting its utility in biological research. As future work, utilizing deep appearance features is expected to further improve tracking using M2Track.

VII. Code Availability

The code is availaby on Github at https://github.com/CIVA-Lab/EDNet-YOLO.

Acknowledgment

The authors appreciate the correspondence with the organizers of the CTMC dataset (Samreen Anjum) in clarifying questions about the dataset. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the U. S. Government or agency thereof.

References

[1] N. M. Al-Shakarji, F. Bunyak, G. Seetharaman, and K. Palaniappan, “Multi-object tracking cascade with multi-step data association and occlusion handling,” in *Proc. IEEE 15th Int. Conf. Adv. Video Signal Based Surveill.*, 2018, pp. 1–6.

[2] N. M. Al-Shakarji, G. Seetharaman, F. Bunyak, and K. Palaniappan, “Robust multi-object tracking with semantic color correlation,” in *Proc. IEEE 14th Int. Conf. Adv. Video Signal Based Surveill.*, 2017, pp. 1–7.

[3] S. Anjum and D. Gurari, “CTMC: Cell tracking with mitosis detection dataset challenge,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, 2020, pp. 4228–4237.

[4] R. Bao, N. M. Al-Shakarji, F. Bunyak, and K. Palaniappan, “DMNet: Dual-stream marker guided deep network for dense cell segmentation and lineage tracking,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops*, 2021, pp. 3354–3363.

[5] H. Bay, T. Tuytelaars, and L. Van Gool, “SURF: Speeded up robust features,” in *Proc. Eur. Conf. Comput. Vis.*, 2006, pp. 404–417.

[6] P. Bergmann, T. Meinhardt, and L. Leal-Taixe, “Tracking without bells and whistles,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 941–951.

[7] K. Bernardin and R. Stiefelhagen, “Evaluating multiple object tracking performance: The CLEAR MOT metrics,” *EURASIP J. Image Video Process.*, vol. 2008, pp. 1–10, 2008.

[8] R. Bischoff, “Chemotaxis of skeletal muscle satellite cells,” *Develop. Dyn.: Official Pub. Amer. Assoc. Anatomists*, vol. 208, no. 4, pp. 505–515, 1997.

[9] E. Bochinski, T. Senst, and T. Sikora, “Extending IOU based multi-object tracking by visual information,” in *Proc. IEEE 15th Int. Conf. Adv. Video Signal Based Surveill.*, 2018, pp. 1–6.

[10] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, “YOLOv4: Optimal speed and accuracy of object detection,” 2020, *arXiv:2004.10934*.

[11] F. Bunyak, K. Palaniappan, V. Chagin, and M. C. Cardoso, “Cell segmentation in time-lapse fluorescence microscopy with temporally varying sub-cellular fusion protein patterns,” in *Proc. IEEE Annu. Int. Conf. Eng. Med. Biol. Soc.*, 2009, pp. 1424–1428.

[12] F. Bunyak, K. Palaniappan, S. K. Nath, T. L. Baskin, and G. Dong, “Quantitative cell motility for in vitro wound healing using level set-based active contour tracking,” in *Proc. IEEE 3rd Int. Symp. Biomed. Imag.: Nano Macro*, 2006, pp. 1040–1043.

[13] Á. Casado-García and J. Heras, “Ensemble methods for object detection,” in *Proc. Eur. Conf. Artif. Intell.*, 2020, pp. 2688–2695.

[14] A. Chamanzar and Y. Nie, “Weakly supervised multi-task learning for cell detection and segmentation,” in *Proc. IEEE 17th Int. Symp. Biomed. Imag.*, 2020, pp. 513–516.

[15] R. Chatterjee, M. Ghosh, A. S. Chowdhury, and N. Ray, “Cell tracking in microscopic video using matching and linking of bipartite graphs,” *Comput. Methods Programs Biomed.*, vol. 112, no. 3, pp. 422–431, 2013.

[16] A. S. Chowdhury, A. Paul, F. Bunyak, D. D. W. Cornelison, and K. Palaniappan, “Semi-automated tracking of muscle satellite cells in brightfield microscopy video,” in *Proc. IEEE 19th Int. Conf. Image Process.*, 2012, pp. 2825–2828.

[17] D. Delibaltov, S. Karthikeyan, V. Jagadeesh, and B. S. Manjunath, “Robust biological image sequence analysis using graph based approaches,” in *Proc. IEEE Conf. Rec. 46th Asilomar Conf. Signals Syst. Comput.*, 2012, pp. 1588–1592.

[18] P. Dendorfer, “Official Motchallenge Evaluation Kit,” 2021.

[19] I. Ersoy, F. Bunyak, K. Palaniappan, M. Sun, and G. Forgacs, “Cell spreading analysis with directed edge profile-guided level set active contours,” in *Proc. Int. Conf. Med. Image Comput. Comput.- Assist. Interv.*, 2008, pp. 376–383.

[20] K. Gao, H. AliAkbarpour, G. Seetharaman, and K. Palaniappan, “DCT-based local descriptor for robust matching and feature tracking in wide area motion imagery,” *IEEE Geosci. Remote Sens. Lett.*, vol. 18, no. 8, pp. 1441–1445, Aug. 2021.

[21] R. Girshick, “Fast R-CNN,,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1440–1448.

[22] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 580–587.

[23] A. Hafiane, F. Bunyak, and K. Palaniappan, “Fuzzy clustering and active contours for histopathology image segmentation and nuclei detection,” in *Proc. Adv. Concepts Intell. Vis. Syst. 10th Int. Conf.*, 2008, pp. 903–914.

[24] A. Hafiane, F. Bunyak, and K. Palaniappan, “Evaluation of level set-based histology image segmentation using geometric region criteria,” in *Proc. IEEE Int. Symp. Biomed. Imag.*, 2009, pp. 1–4.

[25] K. He, X. Zhang, S. Ren, and J. Sun, “Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1026–1034.

[26] P. Hu, F. Caba, O. Wang, Z. Lin, S. Sclaroff, and F. Perazzi, “Temporally distributed networks for fast video semantic segmentation,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 8815–8824.

[27] F. Isensee, P. F. Jaeger, S. A. A. Kohl, J. Petersen, and K. H. Maier-Hein, “nnU-Net: A self-configuring method for deep learning-based biomedical image segmentation,” *Nature Methods*, vol. 18, no. 2, pp. 203–211, 2021.

[28] R. M. Jiang, D. Crookes, N. Luo, and M. W. Davidson, “Live-cell tracking using SIFT features in DIC microscopic videos,” *IEEE Trans. Biomed. Eng.*, vol. 57, no. 9, pp. 2219–2228, Sep. 2010.

[29] R. E. Kalman, “A new approach to linear filtering and prediction problems,” *J. Basic Eng.*, vol. 82, no. 1, pp. 1–35, 1960.

[30] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, vol. 25, pp. 1097–1105.

[31] H. W. Kuhn, "The Hungarian method for the assignment problem," *Nav. Res. Logistics Quart.*, vol. 2, no. 1–2, pp. 83–97, 1955.

[32] H. Li, X. Zhao, A. Su, H. Zhang, J. Liu, and G. Gu, "Color space transformation and multi-class weighted loss for adhesive white blood cell segmentation," *IEEE Access*, vol. 8, pp. 24808–24818, 2020.

[33] Y. Li, C. Huang, and R. Nevatia, "Learning to associate: HybridBoosted multi-target tracker for crowded scene," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 2953–2960.

[34] T.-Y. Lin et al., "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 740–755.

[35] W. Liu et al., "SSD: Single shot multiBox detector," in *Proc. Eur. Conf. Comput. Vis.*, 2016, vol. 9905, pp. 21–37.

[36] X. Lou and F. A. Hamprecht, "Structured learning for cell tracking," in *Proc. Adv. Neural Inf. Process. Syst.*, 2011, vol. 2, pp. 1296–1304.

[37] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–1102004.

[38] J.-B. Lugagne, H. Lin, and M. J. Dunlop, "DeLTA: Automated cell segmentation, tracking, and lineage reconstruction using deep learning," *PLoS Comput. Biol.*, vol. 16, no. 4, 2020, Art. no. e1007673.

[39] J. Luiten et al., "HOTA: A higher order metric for evaluating multi-object tracking," *Int. J. Comput. Vis.*, vol. 129, no. 2, pp. 548–578, 2021.

[40] D. K. Lund, P. McAnulty, A. L. Siegel, and D. D. W. Cornelison, "Methods for observing and quantifying muscle satellite cell motility and invasion in vitro," in *Proc. Muscle Stem Cells: Methods Protoc.*, 2017, pp. 303–315.

[41] F. Lux and P. Matula, "MU-Lux-CZ," *Cell Tracking Challenge*, 2020. [Online]. Available: https://public.celltrackingchallenge.net/participants/MU-Lux-CZ.pdf

[42] L. Lyu, I. E. Toubal, and K. Palaniappan, "Multi-expert deep networks for multi-disease detection in retinal fundus images," in *Proc. IEEE 44th Annu. Int. Conf. Eng. Med. Biol. Soc.*, 2022, pp. 1818–1822.

[43] K. E. G. Magnusson, J. Jaldén, P. M. Gilbert, and H. M. Blau, "Global linking of cell tracks using the viterbi algorithm," *IEEE Trans. Med. Imag.*, vol. 34, no. 4, pp. 911–929, Apr. 2015.

[44] M. Maška, O. Daněk, S. Garasa, A. Rouzaut, A. Muñoz-Barrutia, and C. Ortiz-de Solorzano, "Segmentation and shape tracking of whole fluorescent cells based on the Chan-Vese model," *IEEE Trans. Med. Imag.*, vol. 32, no. 6, pp. 995–1006, Jun. 2013.

[45] J. Mazzaferri, J. Roy, S. Lefrancois, and S. Costantino, "Adaptive settings for the nearest-neighbor particle tracking algorithm," *Bioinformatics*, vol. 31, no. 8, pp. 1279–1285, 2015.

[46] M. Maška et al., "The cell tracking challenge: 10 years of objective benchmarking," *Nature Methods*. pp. 1–11, 2023.

[47] P. McAnulty et al., "Wnt5b is a skeletal muscle satellite cell motogen and chemoattractant that is expressed by resident fibro-adipogenic presursor cells (FAPs)," In preparation.

[48] E. Meijering, "Cell segmentation: 50 years down the road [life sciences]," *IEEE Signal Process. Mag.*, vol. 29, no. 5, pp. 140–145, Sep. 2012.

[49] E. Moen et al., "Accurate cell tracking and lineage construction in live-cell imaging experiments with deep learning," *BioRxiv*, to be published, https://doi.org/10.1101/803205

[50] N. Moshkov, B. Mathe, A. Kertesz-Farkas, R. Hollandi, and P. Horvath, "Test-time augmentation for deep learning-based cell segmentation on microscopy images," *Sci. Rep.*, vol. 10, no. 1, 2020, Art. no. 5068.

[51] J. Munkres, "Algorithms for the assignment and transportation problems," *J. Soc. Ind. Appl. Math.*, vol. 5, no. 1, pp. 32–38, 1957.

[52] H. Nam and B. Han, "Learning multi-domain convolutional neural networks for visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 4293–4302.

[53] S. K. Nath, F. Bunyak, and K. Palaniappan, "Robust tracking of migrating cells using four-color level set segmentation," in *Proc. Int. Conf. Adv. Concepts Intell. Vis. Syst.*, 2006, pp. 920–932.

[54] S. K. Nath, K. Palaniappan, and F. Bunyak, "Accurate spatial neighborhood relationships for arbitrarily-shaped objects using Hamilton-Jacobi GVD," in *Proc. Scand. Conf. Image Anal.*, 2007, pp. 421–431.

[55] D. H. Rapoport, T. Becker, A. M. Mamlouk, S. Schicktanz, and C. Kruse, "A novel validation algorithm allows for automated cell tracking and the extraction of biologically meaningful parameters," *PLoS One*, vol. 6, no. 11, 2011, Art. no. e27315.

[56] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*.

[57] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. 28th Int. Conf. Neural Inf. Process. Syst.*, 2015, vol. 1, pp. 91–99.

[58] E. Ristani, F. Solera, R. Zou, R. Cucchiara, and C. Tomasi, "Performance measures and a data set for multi-target, multi-camera tracking," in *Proc. Eur. Conf. Comput. Vis.*, 2016, vol. 9914, pp. 17–35.

[59] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.- Assist. Interv.*, 2015, pp. 234–241.

[60] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2011, pp. 2564–2571.

[61] T. Scherr, K. Löffler, M. Böhland, and R. Mikut, "Cell segmentation and tracking using distance transform predictions and movement estimation with graph-based matching," *PLoS ONE*, vol. 15, pp. 1–22, 2020.

[62] M. Schiegg, P. Hanslovsky, C. Haubold, U. Koethe, L. Hufnagel, and F. A. Hamprecht, "Graphical model for joint segmentation and tracking of multiple dividing cells," *Bioinformatics*, vol. 31, no. 6, pp. 948–956, 2015.

[63] A. L. Siegel, K. Atchison, K. E. Fisher, G. E. Davis, and D. D. W. Cornelison, "3D timelapse analysis of muscle satellite cell motility," *Stem Cells*, vol. 27, no. 10, pp. 2527–2538, 2009.

[64] R. Stiefelhagen, K. Bernardin, R. Bowers, J. Garofolo, D. Mostefa, and P. Soundararajan, "The CLEAR 2006 evaluation," in *Proc. Int. Eval. Workshop Classification Events, Activities Relationships*, 2006, pp. 1–44.

[65] S. Thutupalli, M. Sun, F. Bunyak, K. Palaniappan, and J. W. Shaevitz, "Directional reversals enable myxococcus xanthus cells to produce collective one-dimensional streams during fruiting body formation," *J. Roy. Soc. Interface*, vol. 12, 2015, Art. no. 20150049.

[66] A. L. Tyson et al., "A deep learning algorithm for 3D cell detection in whole mouse brain image datasets," *PLoS Comput. Biol.*, vol. 17, no. 5, 2021, Art. no. e1009074.

[67] V. Ulman et al., "An objective comparison of cell-tracking algorithms," *Nature Methods*, vol. 14, no. 12, pp. 1141–1152, 2017.

[68] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," 2022, *arXiv:2207.02696*.

[69] J. Wang, et al., "Deep high-resolution representation learning for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 10, pp. 3349–3364, Oct. 2021.

[70] X. Wang, D. Kondratyuk, E. Christiansen, K. M. Kitani, Y. Movshovitz-Attias, and E. Eban, "Wisdom of committees: An overlooked approach to faster and more accurate models," in *Proc. Int. Conf. Learn. Representations*, 2022. [Online]. Available: https://openreview.net/forum?id=MvO2t0vbs4-

[71] L. Wen et al., "UA-DETRAC: A new benchmark and protocol for multi-object detection and tracking," *Comput. Vis. Image Understanding*, vol. 193, 2020, Art. no. 102907.

[72] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *Proc. IEEE Int. Conf. Image Process.*, 2017, pp. 3645–3649.

[73] X.-S. Xu, Y. Jiang, L. Peng, X. Xue, and Z.-H. Zhou, "Ensemble approach based on conditional random field for multi-label image and video annotation," in *Proc. 19th ACM Int. Conf. Multimedia*, 2011, pp. 1377–1380.