Mini Project 2: Where Will Dicty Meet?

Author: Dasha Lykova
Affiliation: Columbia University
Dataset: Dictyostelium discoideum aggregation movies (mixin44, mixin57, mixin64)
GitHub (public): https://github.com/daria-ly/ML-for-Slime---Where-Will-Dicty-Meet-.git
Colab (runnable):
https://colab.research.google.com/drive/1VbgP5fwj0zHrJArdeDGIQx5jbemUbecU?usp=sharing

Approach

In this project, we investigate whether it is possible to predict where Dictyostelium discoideum (Dicty) will eventually aggregate using only the early frames of each time-lapse movie. Since the data does not include explicit annotations of aggregation centers, we estimate ground truth by averaging the final frames of each movie and computing the center of mass of the resulting intensity field. The key question of the project is how much information about the final aggregation place is already embedded in the first few frames, and how different modeling approaches use these early cues.

To address this, we train several models to predict the future aggregation center. Three of the models directly regress an (x, y) coordinate. The fourth model outputs a spatial probability heatmap whose argmax is taken as the predicted center and whose shape provides additional interpretability. We evaluate all models on accuracy, reproducibility across seeds, and robustness to downsampling, and we examine how accuracy changes with the number of available early frames.

Methods

Data description

We analyze three experimental movies stored in Zarr format: mixin44, mixin57, and mixin64. Each movie is a curated "Dicty disco show" of cropped fields of view that capture different stages of aggregation and streaming as Dictyostelium cells move around. The data consist of time-lapse stacks with T timepoints (20–40 depending on the movie), one fluorescence channel, 16 z-slices, and a spatial resolution of 256×256 pixels. These stacks show cells as they begin to self-organize and migrate, so the pixel-level changes across frames can in principle be used to estimate local motion and vector flow fields. No masks, labels, or manual annotations are provided; all supervision derives from computing the aggregation center directly from the final frames.

Preprocessing

Each movie is processed as follows:

1**. Z-projection (mean over Z):**
The z-stack mostly reflects slight changes in focus rather than real 3D structure in these Dicty movies. Since aggregation basically happens in the xy plane, and the course starter code already assumes 2D input, it made sense to collapse the z-dimension. Using a mean projection (`mv = arr.mean(axis=1)`) gives us a cleaner, more high-contrast (i.e. important structures stand out more clearly from the background) version of each frame, which is what one can see in the background of the heatmap figure. Averaging across all 16

z-slices reduces the random slice-to-slice noise and makes the actual cell-dense areas more visible. The cleaner input makes the training process more stable and helps the models pick up the relevant patterns more reliably.

2. **Windowing for aggregation-center prediction:**

For the aggregation prediction task, we use two slightly different windowing strategies for training and testing. During training, we create a larger dataset by taking all possible sliding windows of length *K* from each movie. Each window contains K consecutive early frames, and all windows from the same movie share the same target aggregation center, which is computed from the final frames. This is implemented in our `SlidingWindowDataset`:

```
for start in range(0, T-K):

    x = mv[start:start+K]

    samples.append((x, target))
```

This approach gives the model many examples to learn from.

For testing, the setup is more restrictive. We only provide the model with the very first K frames of each movie, since this matches the actual prediction task: forecasting where Dicty will aggregate based solely on the earliest frames. This is handled by the `AggregationCenterDataset`:

```
x = mv[:K]

y = true_center
```

Together, these two datasets allow the model to train on a richer set of early-frame windows while ensuring that evaluation follows the intended "predict from the first K frames only" structure.

3.**No registration, normalization, denoising, or masking:**
We did not perform registration because the movies do not show any lateral drift or frame-to-frame misalignment. The field of view stays fixed across time, so there was nothing for a registration step to correct. We also did not apply denoising. After collapsing the 16 z-slices using a simple mean projection, the resulting frames were already smooth and stable enough for the downstream models to learn from, and there were no slice-level artifacts that required additional filtering.

We did not normalize the Part B data either. The z-projected frames already had consistent intensity ranges across all three movies, and the center-prediction models trained stably without rescaling. In addition, the ConvLSTM used in Part B never sees raw pixel values directly. Each frame is first passed through a convolutional encoder, the feature maps are flattened and explicitly scaled (`z=z/10`), and only then fed into the LSTM. Because the LSTM operates on these small, compressed embeddings rather than full-intensity images, the gates do not saturate and normalization provides no benefit here.

Finally, we did not apply masking. The dataset does not include segmentation masks, and the prediction task is defined at the whole-frame level rather than around individual cells. Masking would therefore not support the task and might even remove contextual information. Given how the data load and how the models use these inputs, adding any of these preprocessing steps would not have meaningfully improved performance or changed the prediction target.

Models
We train four models that reflect increasing representational capacity:

1. **Simple CNN** (coordinate regression): a lightweight convolutional baseline that outputs a predicted (x, y) coordinate.
2. **TinyCenterCNN** (coordinate regression): a slightly deeper CNN with pooling layers and a small MLP head.
3. **ConvLSTMCenterModel** (coordinate regression): a spatio-temporal model that encodes each frame with a CNN, passes frame embeddings to an LSTM, and maps the final hidden state to an (x, y) prediction.
4. **HeatmapModel** (heatmap prediction): a fully convolutional model that outputs a spatial probability map; the predicted center is the heatmap argmax. This model is less accurate numerically but emphasizes interpretability. Instead of outputting a single (x, y) coordinate, it produces a full spatial probability map that shows which regions of the frame the model considers plausible aggregation sites. The heatmap lets us see how the model distributes its attention across the image, where it assigns higher probability, and where it gets distracted by bright or noisy regions. This makes it easier to understand why the model's prediction ends up where it does. Even when the predicted center is off, the heatmap gives us a sense of the model's internal reasoning, which is something the coordinate-based models cannot provide.

Coordinate models (1-3) are trained with mean squared error. The heatmap model uses BCEWithLogitsLoss with a Gaussian target map centered at the ground-truth point. For all models, we fix random seeds, enable deterministic cuDNN behavior, and repeat training 10 times to compute confidence intervals and robustness scores (Mean Drop (%) in table below).

Code Availability

A runnable Colab notebook reproduces the analysis. Package versions are pinned in requirements.txt which can be found on GitHub:

GitHub: https://github.com/daria-ly/ML-for-Slime---Where-Will-Dicty-Meet-.git
Colab: https://colab.research.google.com/drive/1VbgP5fwj0zHrJArdeDGIQx5jbemUbecU?usp=sharing

Results

Early-frame hotspot prediction (overlay figure)
The overlay figure shows how the heatmap model interprets the early frame. The true aggregation center (cyan) lies inside the brighter region of the image, while the predicted center (yellow) is shifted upward toward a slightly brighter local patch. The heatmap itself forms a broad, fuzzy region of elevated probability instead of a single concentrated peak. This tells us two things: first, the model is able to detect part of the correct spatial structure, since the predicted point is not randomly placed, and second, its confidence is spread over a fairly large area, which explains why the heatmap model's center-error is much higher than that of the coordinate-regression models. The figure also makes clear that the model is influenced by scattered bright spots elsewhere in the frame, which likely contribute to its occasional large errors.

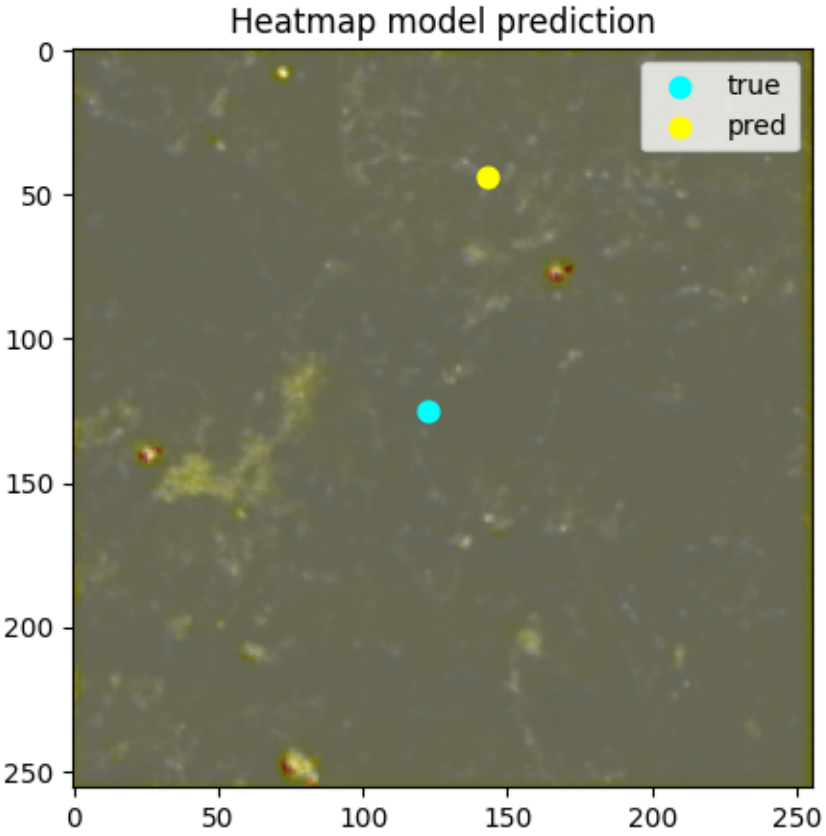

Figure 1. Heatmap-based aggregation prediction.
The predicted center (yellow) is derived from the peak of the model's heatmap, shown over an early movie frame, and compared to the true center (cyan).

Error vs. number of early frames (K-curve) (ConvLSTM model)
The error-vs-K curve shows that the center error stays in a narrow range (roughly 175–177 px) for all K values tested, from K=2 up to K=10 with step 1. The curve fluctuates a bit but shows no consistent improvement as more frames are added. This suggests that, in this lightweight training setup, early-frame appearance does not sharply differentiate between final aggregation outcomes. The overall flatness of the curve matches the visual similarities across very early frames: the movies look nearly identical at early timepoints, so supplying more of these similar frames doesn't noticeably help the model. The fact that the ConvLSTM model performs well in repeated-run evaluation but not in the quick K-curve training further supports this interpretation - this model needs deeper training and more passes through the dataset to extract subtle temporal cues.

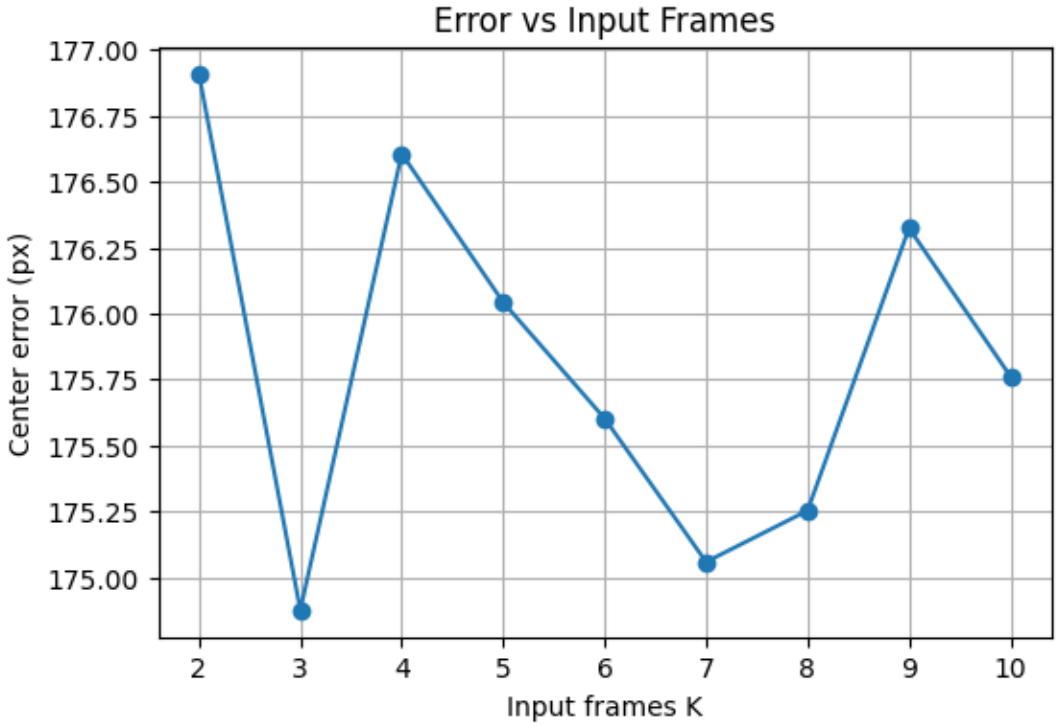

Figure 2. Error vs. number of early input frames (K) in ConvLSTM model.
Center-prediction error remains within a narrow range (approximately 175–177 px) across all tested window sizes (K=2–10, step 1). Although the curve fluctuates slightly, it shows no consistent improvement as more early frames are added, indicating that the earliest timepoints contain limited distinctive information about the eventual aggregation site

Quantitative performance and robustness

The repeated-run metrics in Table 1 show clear differences in how the models behave. The ConvLSTM model achieves by far the lowest mean error (3.31 px, CI ±0.06 px), indicating that it consistently extracts useful temporal structure across training runs. The two CNN baselines perform moderately: the simple CNN averages 21.97 px error (CI ±5.74 px), and the tiny CNN averages 29.41 px (CI ±0.95 px). Both display a predictable spatial bias toward the center of the frame, which explains their systematic under- and overshooting across movies. The heatmap model shows the highest mean error (114.10 px, CI ±16.38 px), but it is uniquely robust to resolution changes. Its low-resolution error remains almost unchanged (110.55 px), corresponding to a drop of just 0.18%. In contrast, the CNN baselines deteriorate substantially under downsampling (drops of 288.66% and 126.55%), and the ConvLSTM collapses almost completely (2606.12%). This robustness pattern matches the diffuse, low-frequency structure of the heatmap predictions, while the sharp drop in coordinate models reflects their reliance on fine spatial detail.

Model	Mean Error (px)	95% CI Error (px)	Mean Low-Res Error (px)	Mean Drop (%)	95% CI Drop (%)
simple cnn	21.97	5.74	64.07	288.66	153.19
tiny cnn	29.41	0.95	66.30	126.55	14.52
conv2d+lstm	3.31	0.06	89.61	2606.12	53.73
heatmap model	114.10	16.38	110.55	0.18	13.67

Table 1. Model accuracy and robustness metrics (rounded to 2 decimal places). Each model is evaluated over 10 random seeds, reporting mean center error, 95% CI, performance on downsampled movies, and percentage drop relative to high-resolution performance.

True vs. predicted coordinates:

The predictions in Table 2 show model behavioral patterns directly. For Movie 0, the ConvLSTM produces a coordinate very close to the true aggregation center [122.14, 124.89], while the two CNN baselines drift toward a similar central region of the frame. The simple CNN undershoots substantially, predicting [108.93, 107.09], and the tiny CNN shows a comparable bias with [111.34, 106.49]. The heatmap model selects [143.0, 44.0], locking onto a bright distractor patch in the upper part of the frame. A similar pattern appears for Movie 1: both CNN models again lean toward centrally biased coordinates, and the ConvLSTM remains close to the true center [131.32, 126.85]. The heatmap model instead predicts [42.0, 122.0], which lines up with another bright, isolated region visible in the early frames. For Movie 2, the ConvLSTM once again stays extremely close to the ground truth [126.70, 126.64], while the CNN baselines undershoot sharply ([90.43, 82.54] for the simple CNN and [84.20, 78.47] for the tiny CNN). The heatmap model predicts [218.0, 34.0], tracking a bright but irrelevant hotspot. Across all three movies, these examples illustrate why the quantitative metrics look the way they do: the ConvLSTM maintains consistently accurate predictions, the CNNs produce stable but biased outputs, and the heatmap model is easily drawn toward bright distractor regions. This qualitative behavior matches what is visible in the overlay figure and aligns with the summary statistics reported in Table 1.

Movie	True Center	Simple CNN Prediction	Tiny CNN Prediction	ConvLSTM Prediction	Heatmap Prediction
0	[122.14, 124.89]	[108.93, 107.09]	[111.34, 106.49]	[126.43, 126.18]	[143.0, 44.0]
1	[131.32, 126.85]	[133.17, 127.39]	[139.32, 131.82]	[126.43, 126.18]	[42.0, 122.0]
2	[126.70, 126.64]	[90.43, 82.54]	[84.20, 78.47]	[126.43, 126.18]	[218.0, 34.0]

Table 2. True vs. predicted aggregation centers for all models.
Each model's predicted (x,y) coordinate is shown alongside the true aggregation center for each movie. The ConvLSTM predictions stay closest to the true centers, the CNN baselines show moderate but biased accuracy, and the heatmap model exhibits larger deviations consistent with its diffuse probability maps.

Conclusion:

Taken together, our findings show that the earliest Dicty frames contain only faint traces of the eventual aggregation site, yet these weak signals are still recoverable with the right model and enough training. Among all approaches, the ConvLSTM center model was consistently the strongest. When trained thoroughly in the repeated-run setting, it reliably converged to the correct region in all three movies and achieved by far the lowest error. The CNN baselines also behaved in stable and predictable ways: they tended to fall back on a central-frame bias and did not adapt much across movies, which explains their moderate but not exceptional accuracy.

The heatmap model, although the least precise in terms of center error, played an important complementary role. Its broad, diffuse probability map provided a clear look at what the model found visually salient in the earliest frames. The same spatial smoothness that limited its accuracy also made it remarkably robust to downsampling, highlighting that it focuses on coarse, low-frequency structure rather than fine-grained detail. This interpretability was especially useful for understanding why predictions drifted toward bright distractor regions.

The error-versus-K analysis reinforces the broader story. The ConvLSTM's accuracy barely changed when we increased the number of early frames, and the overall curve remained extremely flat. This indicates that the first several frames contain limited distinctive information about the final aggregation decision. Because these early frames look almost identical visually, supplying more of them does not meaningfully alter the model's predictions. This also explains why the ConvLSTM shines only when trained for long enough: extracting meaningful temporal cues from such subtle early dynamics requires more than the lightweight K-curve setup can provide.

Overall, combining coordinate-based models with a heatmap model gave a more complete picture of the task than either approach alone. The coordinate models quantify how close we can get to the true aggregation center using early data, while the heatmap model shows why the predictions look the way they do by revealing the spatial structures the model attends to.

Looking forward, several directions could improve performance or deepen interpretability: incorporating explicit motion features such as optical flow, adding uncertainty estimates to the heatmaps, or designing models capable of predicting multiple possible aggregation centers. Attention-based architectures may also help capture the subtle, spatially distributed cues that characterize the earliest frames. Together, these extensions could make early-frame prediction of Dicty aggregation both more accurate and more biologically informative.