



Early-Stage Well Image Recognition for Cancer Identification

Fahim Tajwar, Sandhini Agarwal, Darian Martos

CS 231n: Computer Vision, Spring 2019, Stanford University

Motivation

Cancer cell recognition has been a core application of deep learning thanks to the advancement of CNNs in computer vision. Using well images provided by Alexandra Adams Sockell from the Stanford Department of Genetics, we aim to characterize cancer growth through various deep learning models. Using basic CNNs and logistic regression, we establish a baseline to explore other models their effects. We ultimately explore VGG, 3D CNNs, and a mixed CNN + LSTM model to best capture the time-sequenced data of cancer well images. We conduct this prediction during early-stage cells, to see whether early stage detection leads to cancer diagnosis.

Dataset

- Our dataset during our milestone consisted of images for 4800 single gastric cancer cells. We used only 350 of these as they were the only ones labelled.
- Each of these cells had 14 corresponding black- and-white images with each image corresponding to one day in a two-week time-frame.
- Each cell had one of 3 labels - "cell dies," "grows sparse," and "grows dense".
- We used an indirect data augmentation method, where we keep our original dataset, but during training, each image goes through a series of transformations, which include random flips, rotations, cropping, normalization.



Figure: "Grows Dense" Example

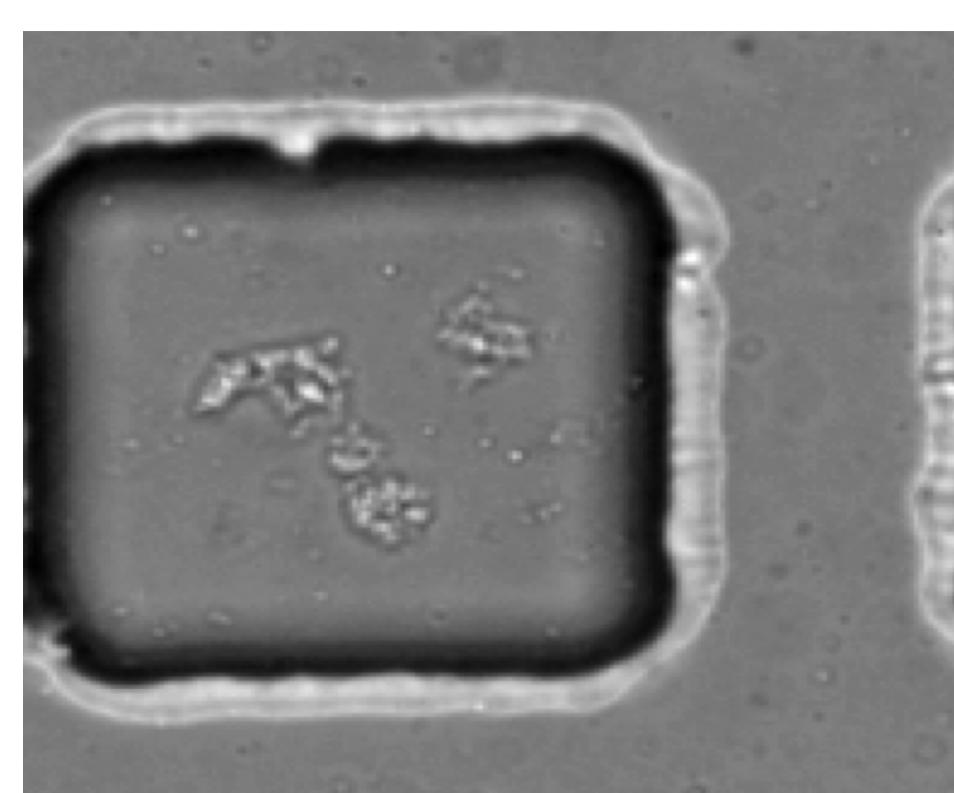


Figure : "Grows Sparse" Example

Method 1: CNN Models

- We built a CNN (CNN1) which took in Day 5 image data with the following architecture: (Conv → ReLU → Max Pool) × 3 → (Affine) → (Weighted Cross Entropy Loss)
- We built another CNN (CNN2) of the architecture (Conv → ReLU → Conv → ReLU → Max Pool) × 3 → (Affine) → (Weighted Cross Entropy Loss)

Method 2: 3D CNN

- With a 3D-CNN, we used 3D-Kernels in CNNs, which are common in analyzing video data.
- The stacked Pytorch tensor had shape $C \times T \times W \times H$, where T is the number of days, C is the number of channels, W and H are the height and width. For our dataset, a particular well would generate a torch tensor of shape (3, 5, 224, 224).

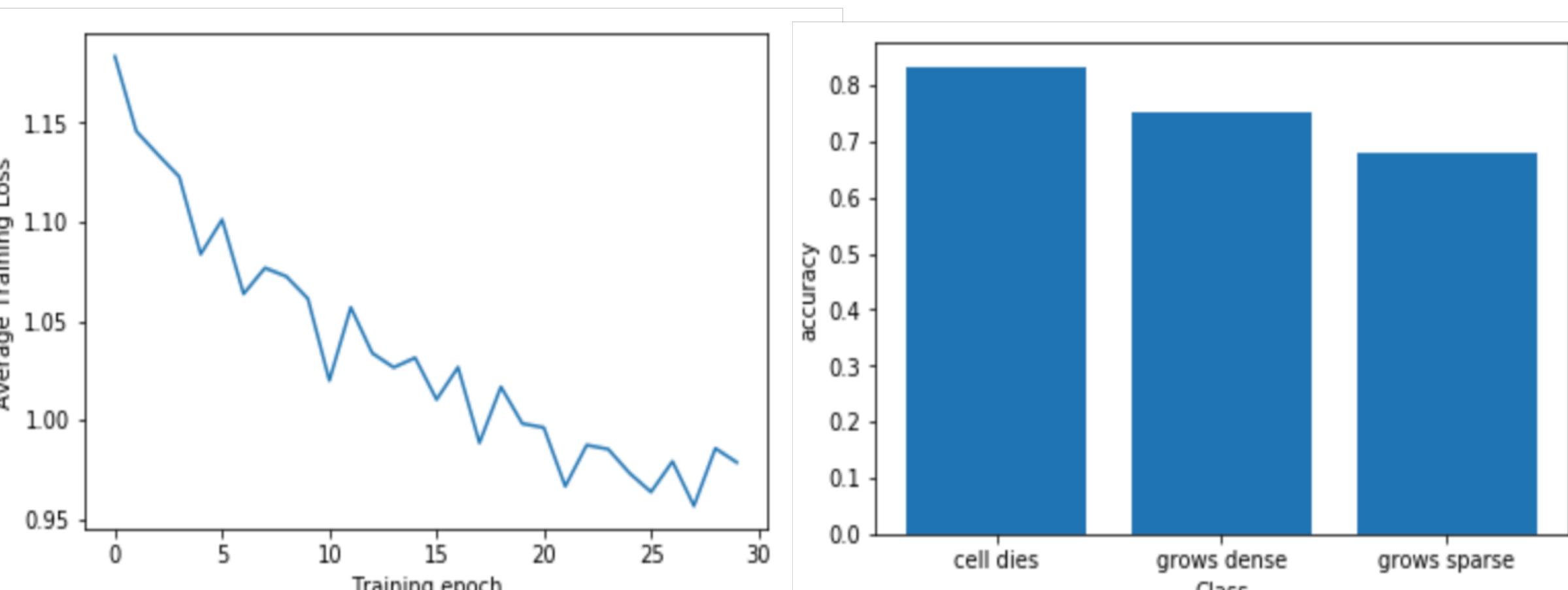
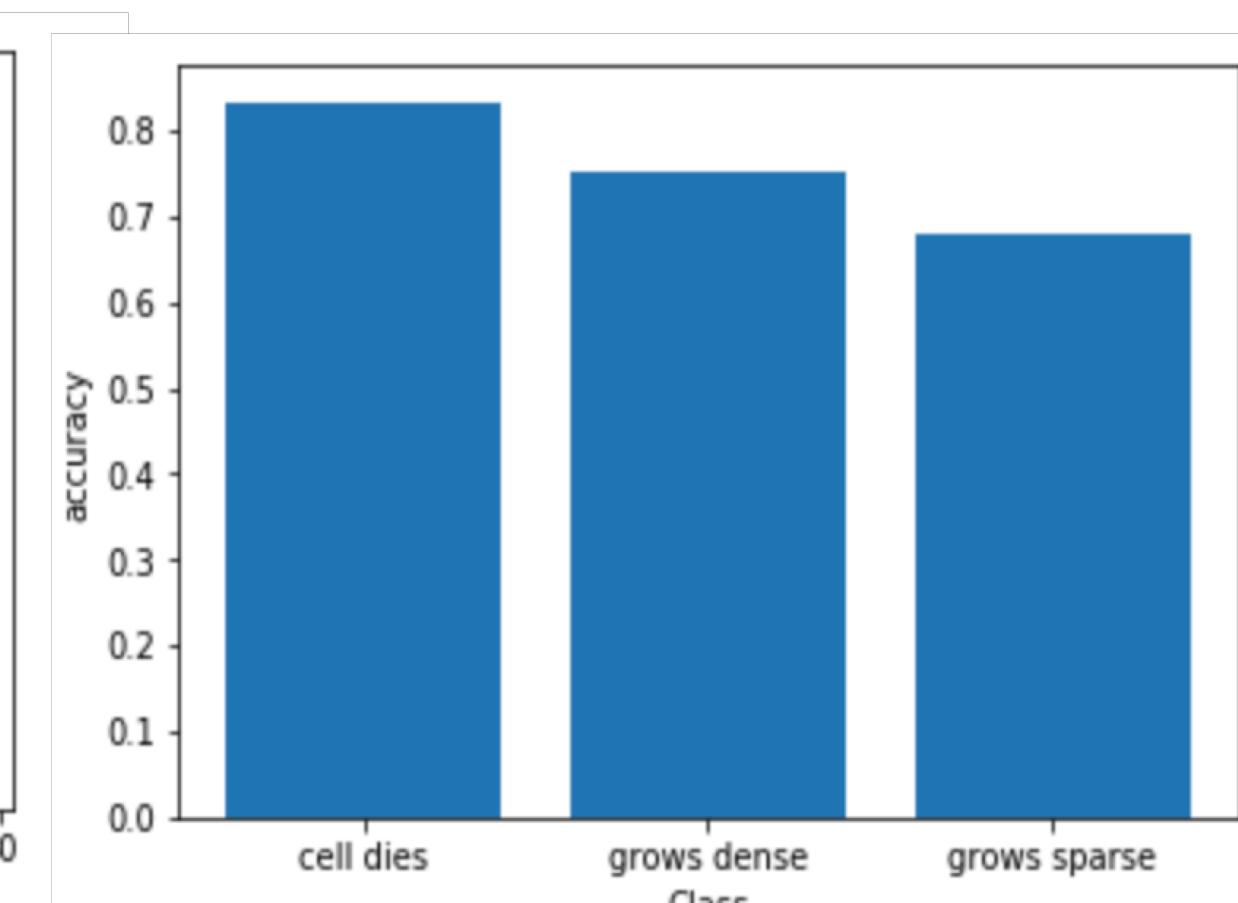


Figure: Loss curve for CNN1



Results (on Test Set)

Method	Total Accuracy	"Cell Dies" Accuracy	"Grows Sparse" Accuracy	"Grows Dense" Accuracy
Logistic Regression	52%	80%	31%	44%
CNN1	75%	83%	68%	75%
CNN2	38%	50%	33%	33%
VGG	25%	0%	25%	50%
CNN+LSTM	36%	8%	0%	100%
3D-CNN	54%	77%	52%	33%

- Our best model turns out to be the CNN 1, which is not that deep. The reason is, we hypothesize, the lack of data. There is not enough data to train the deeper, more complex networks, where we easily get model overfitting and bad results.

Error Analysis: Visualizing Misclassification

We were having the highest number of errors for the grows sparse class which were being classified as grows dense since at early stage there isn't much difference between them. Also some growth change trajectory, and from an early dense growth, can later change into a sparse growth.

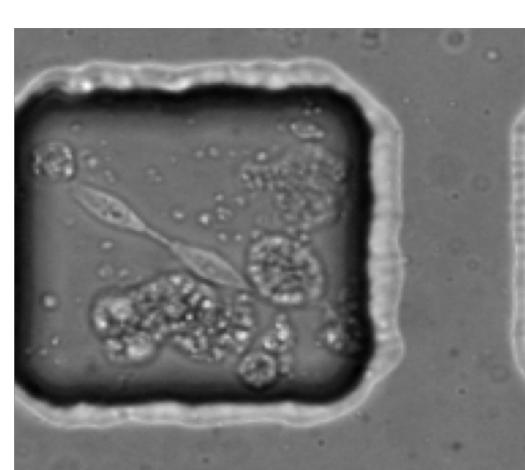
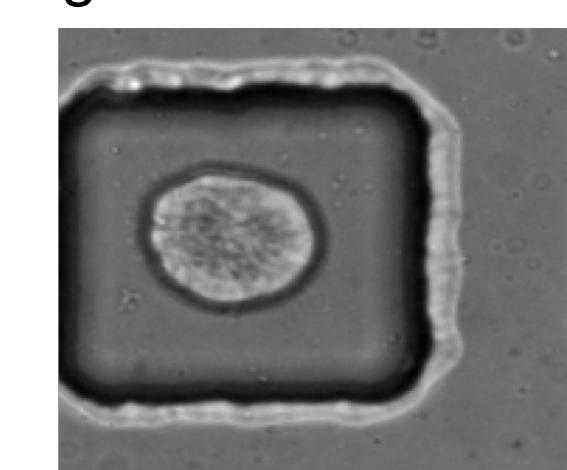


Figure: On left, well 1037 day 5, and on right day 12 data. We see a dense structure disintegrated into a sparse one, making it hard to predict. We ran a binary classification on two classes, 'cell growth' and 'cell death', which gave better results. Total accuracy jumped to ~85%, showing these two classes are more predictable from early stage images.

Method 3: CNN + LSTM

- We tried a CNN followed by an LSTM network. This is because we hoped that the CNN would help us with feature extraction across the images and the LSTM would help support sequence prediction for the time-series data.
- This again had tensors of shape $C \times T \times W \times H$,
- Our CNN architecture was (Conv → Batch Normalization → ReLU → MaxPool) × 3 and the features from this were fed into a 5 layer LSTM.