Project 4 - Lightning Talk

Team Flanagan

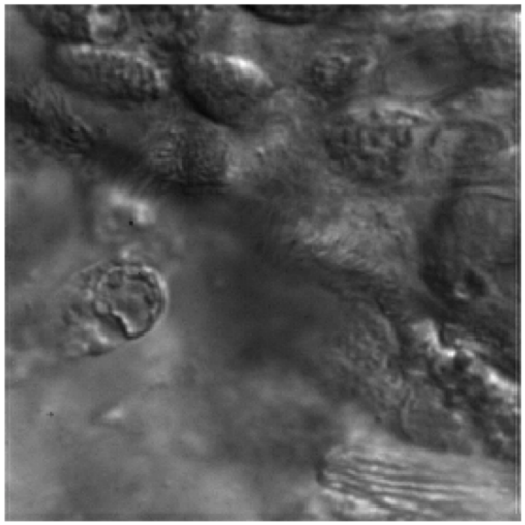
Techniques

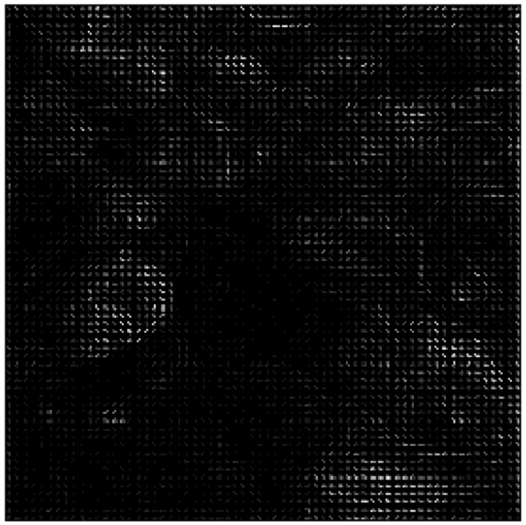
- HOG + SVM
- CNN
- Flatten Images + SVM
- Optical Flow

SVM with HOG

- Histogram of Gradient (HOG) Descriptor
- Feature descriptor used in computer vision and image processing for object detection
- Counts occurrences of gradient orientation in localized portions of an image
- Concept:
    - local object appearance and shape within an image can be described by the distribution of intensity gradients or edge directions
- **PRO**: Particularly suited for object detection
- **CON**:
    - Not able to use HOG features when dealing with pixelwise dataset
    - Cilium are too small to detect (better to detect cells)

SVM with HOG





CNN

- TensorFlow CNNs == Static Input Shape
- Minimum video dimension identified: 128
- Method One : PreChunk
  - PreChunk data into 128 X 128 size images
  - Train and Test the model
  - Problem: GPU RAM limitations

CNN

- TensorFlow CNNs == Static Input Shape
- Minimum video dimension identified: 128
- Method One : PreChunk
- Method Two : TensorFlow Data API
    - Serialize data into TFRecords
        - Problem: Preserving data dimensions
    - Stream data in through TFRecordDataset object
    - random_crop
    - random_slice
    - Train the model
    - Still have to slice & stitch the test data (not pretty...)

CNN

- TensorFlow CNNs == Static Input Shape
- Minimum video dimension identified: 128
- Method One : PreChunk
- Method Two : TensorFlow Data API
- Problems:
  - Vanishing Gradient?
  - Suspicious about my UNET implementation...

SVM with Flatten Images

- Storing N images of single video, each of dimension (X, Y), in a single huge matrix of dimension (X*Y, N)
- Each row of this matrix is labelled with respective pixel in the mask (X*Y, 1)
- Train this data on SVM model and predict on test data.
- Problems :
    - Takes a huge amount of time to perform SVM
    - 10 train samples and 2 test samples takes about 5 hours to train and 40 minutes to predict
    - Ended up with no decent output

Optical Flow

- Calculates two-frame motion estimation and computes a degree of flow for each pixel. Ends up with N-1 flow matrices for a video of N frames
- A sum of these flow matrices is found and scaled down to {0,1,2} by observing the pixels in motion
- Problems :
    - Difficult to scale the videos where even the cell has a motion. Results in larger areas of motion. This effected on overall mean IoU
- Best Score: 20.6449