Predicting DNA Damage in Fluorescent Imaging

Daniel McDonough and Surya Vadivazhagu
Worcester Polytechnic Institute
May 2020

Abstract — There are many methods to analyze cell nuclei, but few can detect the presence of DNA damage, without labeling proteins. We have developed a process to detect and classify cell nuclei in the presence of DNA damage. This method uses the morphology of DNA-associated histone proteins, without proteins that specifically label or localize to sites of DNA damage. We discuss how and why we address this problem, followed by the specifics of the program pipeline, then our results and how it compares to other methods, and concluding about future additions.

Keywords — Nuclei Detection; Foci Detection, Feature Detection, Random Forest, Convolutional Neural Networks;

Introduction

Researchers interested in the dynamics of DNA, tend to look at the response and efficiency of repair factors in the presence of damage. These researchers use a mix of manual and software assisted analysis. All known openly available software options fail to detect cell nuclei and foci in conjunction with classifying nuclei [1]. This has limited the utility of these software in the study of DNA damage acquisition and repair. Processing individual cells in high fidelity is a computationally expensive and time consuming process. This project has two goals: to enhance the ability to accurately identify damage foci in labeled immunofluorescent images, and identify morphological features in organization of DNA that indicate, without a priori knowledge of DNA damage status, which cells contain DNA damage. Successfully enhancing the detection and classification process will provide insight into possible correlation between morphological features from fluorescence images and important regulators of the process that may be altered in disease contexts such as cancer.

Using varied libraries for image processing and artificial intelligence, we developed a tool for the ingestion, analysis, and classification of these cells. Our pipeline processes microscopic images of cell cultures whose DNA and damaged sites are labeled by proteins. This pipeline can process samples of varying fluorescence, and is extremely modular in practice, thus making it extensible for any use case between segmentation and unsupervised classification. Hot-swapping modules dependent on the use case is very feasible and easily achievable across different colored nuclei strains and feature vector additions.

Background

The field of computer-aided microscopy has undergone much innovation due to advancements in imaging technology, and robust libraries for automated image analysis. Over the past few decades, researchers have utilized deep learning algorithms able to classify cells and nuclei based on descriptive features [2,3]. It is possible to differentiate cell type and determine cycle stage by shape and texture characteristics of cell nuclei [4,5]. These classifications focus on high phenotypic classifications that can also be detected with the human eye. Our research hopes to use nuclear morphology to classify damaged nuclei. Identifying such damage is not easily detectable to the human eye without labeling proteins.

With regards to cell damage, biologists use computer aided diagnostics to process many images using tools such as FoCo and CellProfiler. These tools have built-in algorithms for the segmentation of large cultures of cells into individual images of cells for examining, as well as options to highlight and count marked foci inside the cells. However, these tools can be cumbersome when working with a large amount of images. We identified this issue and researched the currently used applications with the goal of identifying their flaws with respect to the datasets that will be examined.

A popular open-source tool used is CellProfiler [6], a “cell image analysis software” developed in 2003 by the Broad Institute of Harvard and MIT. While the software does provide updates compared to previous tools with respect to user interface, its core engine for doing image analysis proved ineffective for our dataset. Images would have dense amounts of cells clustered next to each other with each having their own foci, and CellProfiler proved unreliable in detecting these.

To improve upon CellProfiler, FoCo [7] was developed to improve the nuclei detection and added the ability to outline foci. FoCo’s strengths lie in how versatile it is at identifying foci in cells with varying image quality, background noise, and fluorescence levels. FoCo’s processes became the basis for our automatic labeling system. Because of this, we were able to generate much more “training data”, which was used for training various neural network models for the classification of these nuclei.

Methodology

After observing the procedures developed through FoCo and CellProfiler, as well as conducting extensive research on recently published solutions for cell segmentation and analysis. In this section we will elaborate upon each one of these steps, as shown in Figure 1. One of the major factors that we wanted to keep in consideration was time to process. We identified from testing CellProfiler and FoCo that on average, CellProfiler took 15 minutes to load the image, run its algorithms, and output results in a table format. FoCo on the other hand, needed only 8 minutes to process the image and output results. The need for a program to automate detection of foci and classify the cells is apparent, as manual detection and labeling of the cells’ classes took hours of labor. Using our pipeline we were able to obtain a dataset of nuclei from a single frame in an average of 30 seconds.

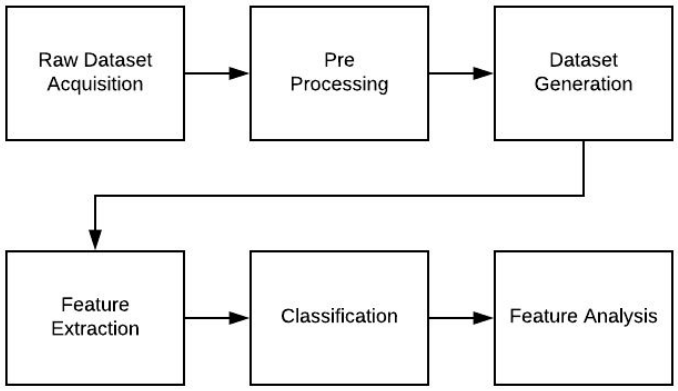

Figure 1: This figure shows the general process used in our research. Starting with obtaining the nuclei videos, automatically generating a dataset, obtaining features, classifying the nuclei to finally analyzing the feature importance and relevance.

Raw Data Acquisition

Our dataset included 3 unique ND2 movies of fixed cells, named “Movie1”, “Movie2” and “Simple”. Simple was given its name due to the low amount of nuclei (about 5 - 40) in each image, allowing it to be a basis for manual labelling and cross-checking. The Simple dataset includes 24 frames from footage of cell nuclei over 2 hours with 5 minutes between each frame. The cells are filmed from 16 different wells, and the images were split into 8 different XY coordinates. Movie1 and Movie2 contain only one well and were not split into different coordinates, resulting in a much larger amount of nuclei as shown in Figure 2. For each of these movies, RFP was used to label nuclear DNA and GFP to indicate DNA damage and captured with a Zyla sCMOS camera mounted on a Nikon Ti-E microscope with a 20x CFI Plan Fluor objective such that each pixel is about .67um. The cultures shown in the movies have been tested to have uniform GFP presence however, there are intrinsic limits to detecting GFP indicative of DNA damage.

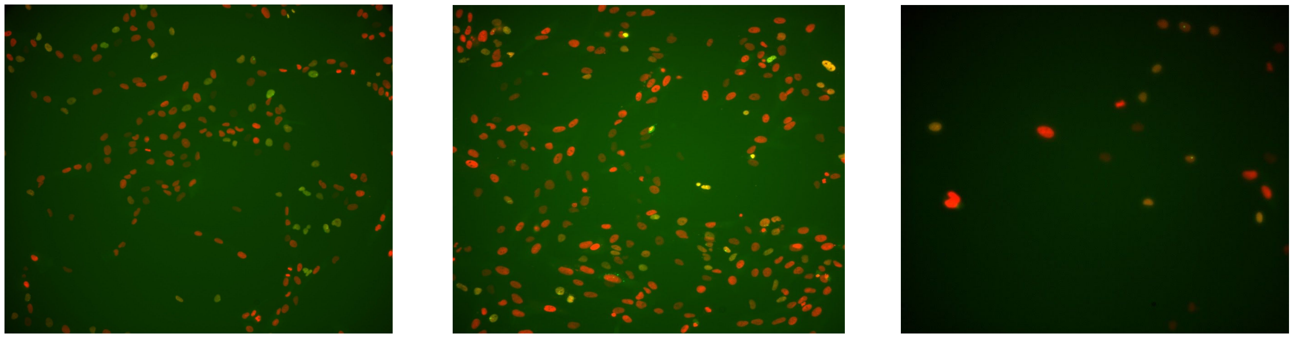

Figure 2: This figure shows the first time step for each dataset, Movie1 [left], Movie2 [middle], and Simple [right]. Here it can be seen how Simple is split into wells based on the resolution of the frame and how it contains less nuclei in a single frame than the other movies. Additionally you can see the variety in haze among the datasets.

PreProcessing

From the given ND2 files when processed are compiled into separate TIFF images for each frame and for each color channel. As each image has empty values for the blue channels, storing the images in this method is very spatially intensive. To save space in memory, we merged the red and green channels of each corresponding image, and saved the resulting image with no additional compression to prevent data loss.

A major issue that came up during the detection of foci was the presence of background fluorescence. The background noise made it difficult to autonomously pinpoint the foci locations due to the overwhelming presence of the green channel. We consulted using an inverse point spread function, in addition to global and local thresholding to reduce haze [8]. Taking time and complexity into consideration, we used mean global thresholding as it was easy to compute and only needed to be done once per frame. This allows the GFP within the nuclei to be normalized such that only nuclei with high enough GFP expression can be seen, as shown in Figure 3. Additionally, as the amount of fluorescence varies between nuclei we used gamma correction (1) to allow easier detection of the darker nuclei.

$$(Pixel/255)^{1/} * 255 \;\; ; \gamma = 1.5 \qquad (1)$$

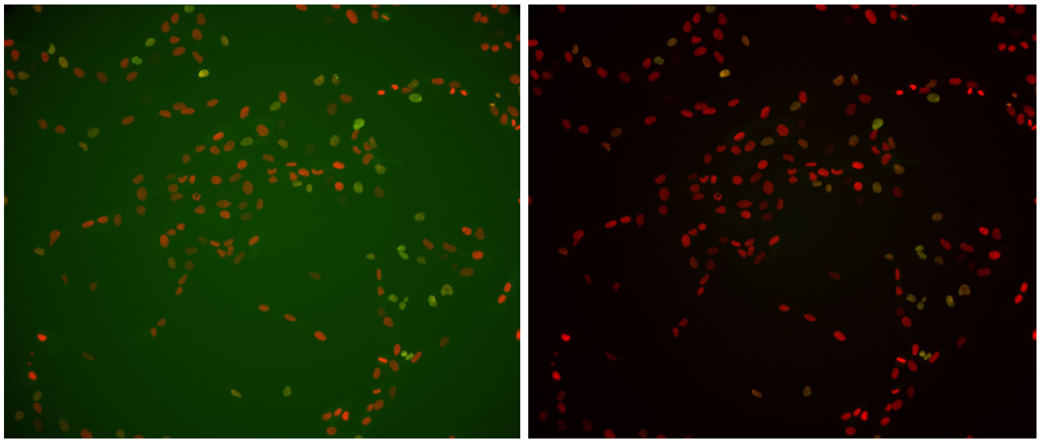

Figure 3: Before [left] and after [right] of global average de-hazing. Note that the background green 'haze' is removed, and that nuclei with GFP presence are still visible.

Dataset Generation

From the Simple movie we produced a labeled non time dependent dataset. The dataset included 20 frames from random wells and coordinates. To label the dataset we manually segmented each nucleus by looking at the overlaid frames to be ground truth of the edge of the nucleus. After manual and automated segmentation, we analyzed the nuclei to detect foci and GFP diffusion using FoCo. We labeled pixels that we deemed to be part of the nucleus red (FF000) if it contained no foci, green (00FF00) if it contained foci as shown in Figure 4.

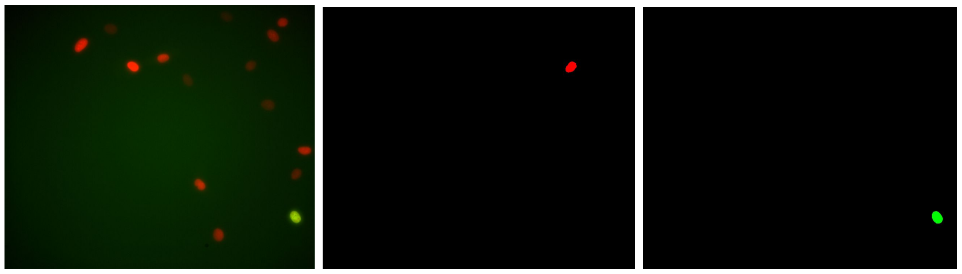

Figure 4: Original Image [Left], Single healthy nuclei mask [Middle], Single damaged nuclei mask [Right]. This figure shows masks for a damaged and undamaged nucleus that has been segmented and labeled from the original image. These masks are a single instance chosen from the masks generated from the original image.

From here we sought to automate the labeling process by using various segmentation techniques, and FoCo for foci detection. To test the accuracy of an automatic labeling process, we generated a subset of the Simple set to be our training set. Our training labeled dataset included 20 unique frames for a total of 324 labeled nuclei. Of the labeled nuclei, 285 were deemed to have no DNA damage, and 41 were damaged through our analysis in FoCo. We tested four different segmentation algorithms, Mean Shift, Chan-Vese, Otsu, Triangle and K-Means. These algorithms were chosen as they are used in software such as ImageJ and have previously shown to have good detection results in fluorescence images [9]. All of these tests were followed by watershed segmentation to separate nuclei that may be too close to each other, and all detected regions that were smaller than 150 pixels in area were removed. From our testing we chose to continue with Otsu thresholding. By using Otsu we can expect segmentations of at least 80% of nuclei in a frame, about 60% of the time as shown in Table 1.

Table 1: This table shows the mean average precision of select segmentation method with a threshold of 0.8 for true positives. Meaning, that in a given frame we can expect the algorithm will have an average IOU score of at least 0.8 for all nuclei.

	Mean Shift	Chan-Vese	Otsu	K-Means	Triangle
Mean Average Precision (TP = 0.8)	0.054	0.202	0.598	0.526	0.212

To prevent classification or feature extraction algorithms from having bias towards non-biological factors such as orientation or location, we normalized each nucleus [10]. To normalize a nucleus we cropped the image to a minimum the bounding box relative to its contours, and found the angle or orientation. We then rotated each nucleus to be standardized at 0^{o} such that the nucleus is vertical along its major axis. Each crop of a nucleus was then padded with zeros uniformly to make them 150x150 in size.

Feature Extraction

For each detected nucleus region we have calculated 15 major features. More specifically, 9 features concerning the shape of the region and 6 textural features for the enclosed area. As some of our major features are a collection of features or transformed images, each pixel is defined as a feature, making a total of 91,093 features as shown in Table 2. However, the contribution of each feature is different in the categorization of the data.

Table 2: This table shows a list of features analyzed. The left column contains features that are primarily used as shape descriptors. The right column contains features that are primarily used as texture descriptors. The number next to the feature represents the size of the feature.

Shape Based Features	Texture Based Features
● Major Axis Length (1) ● Minor Axis Length (1) ● Ratio Between Axis (1) ● Eccentricity (1) ● Solidity (1) ● Roundness (1) ● Extent (1) ● Histogram of Oriented Gradients (22500) ● Hu Moments (7)	● Zern-like Moments (24) ● Haralick Texture (13) ● Linear Binary Patterns (256) ● Scale Invariant Feature Transform (768) ● Laplacian of Gaussian (22500) ● Gabor Wavelet (22500) ● Cropped Image (22500)

Shape features

As the shape of a cell's nucleus changes through the cell cycle and due to aggressive damage, we included 9 features to help describe the spatial rearrangements of chromatin. By using the major axis, minor axis and the ratio between them to define the size of the nuclei. The extent of a nuclei, or the ratio between the nuclei contour and minimum bounding box, serves as a supplement for a circularity descriptor. The roundness of nuclei [11], was chosen as it has been shown to be a structural feature to predict aggressive phenotypes. We test the eccentricity and solidity of nuclei, as they project the curvature and compactness of nuclei which also may be affected under strong phenotypes. Hu moments detect variations in scale, translation, rotation, and reflection. We chose this feature as a stabilizer to help the classifier test between similar nuclei. The use of Histogram of Oriented Gradients (HoG) allows us to obtain an approximation of the perimeter of the nuclei, as it is limited to large gradients and 8 orientations. HoG does not only detect the perimeter of nuclei, it also detects the orientation and trends throughout the whole image, and it may double as a texture descriptor depending on the harshness of the fluorescent gradients of a nuclei.

Textural features

The texture analysis of the detected regions is based on the statistical properties of the intensity histogram in the red channel. This allows local binary patterns (LBP) of the pre-interpolated image to be analyzed for all 256 bins of the histogram. For each segmented nuclei we calculated 13 Haralick texture descriptors [12]. Those being, angular second moment, contrast, correlation, sum of squares variance, inverse difference moment, sum average, sum variance, sum entropy, entropy, difference variance, difference entropy, and measure of correlations 1 and 2. Additionally, to analyze the inside of the nuclei with spatial relevance, we used Laplacian of Gaussian (LoG). The second derivative of the image obtained through a LoG filter, shows harsh drops in intensities and gives us a clear image of the rates of change in the nuclei spactialy, most notably in the perimeter. The use of Gabor Wavelets [13] have been shown to be useful in improving classifiers. Lastly we use the unique feature identifiers, Zern-like Moments and SIFT, in hopes to provide a trend between the possible identifiers and DNA damage. For visual reference, the image based features are shown in Figure 5.

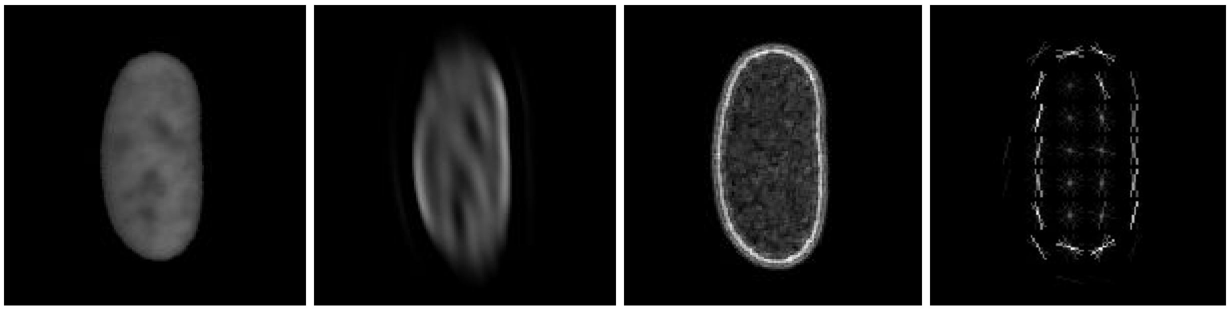

Figure 5: This figure shows the original crop or red channel of a nuclei [far left] in comparison to the gabor wavelet [middle left], LoG [middle right], and HoG [far right]. Here you can see how different filters represent the same nuclei in different ways.

Classification Methods

Much of the success behind deep learning techniques for classification comes from their use of layering techniques to learn more complex features over time across these layers. However, a very common issue in this is “overfitting”, or constraining a model too specifically to the training data that it is unable to predict for data it hasn’t seen before (i.e. testing data). There comes a trade-off then, that if a network has too many layers, it tends to overfit too much to training data. In our project, overfitting was a real concern due to the extreme homogeneity of the training and testing data. There are techniques that aim to minimize this (regularization, etc.), however the issue can still persist, as depicted in Figure 6.

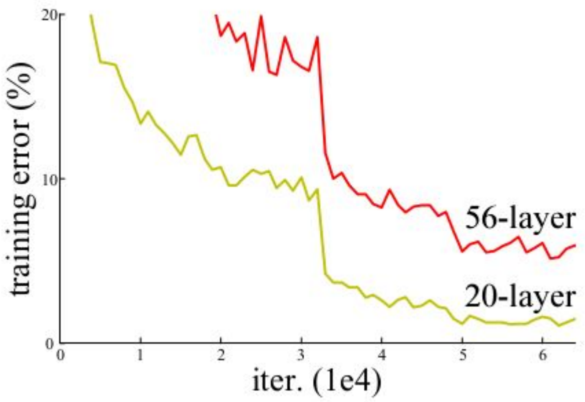

Figure 6: He et al.'s graph showing the dropoff of how more layers doesn't provide higher accuracy. The 56-layer network ends up having more errors in training [left] and testing [right] [14, Fig. 1].

He et al.'s research from 2015 showed that it's possible to design neural networks that use a variable amount of layers depending upon the use case [14]. They designed the concept of a "residual block" visualized in Fig. 7 which resolves many issues involved with training deep-layered networks.

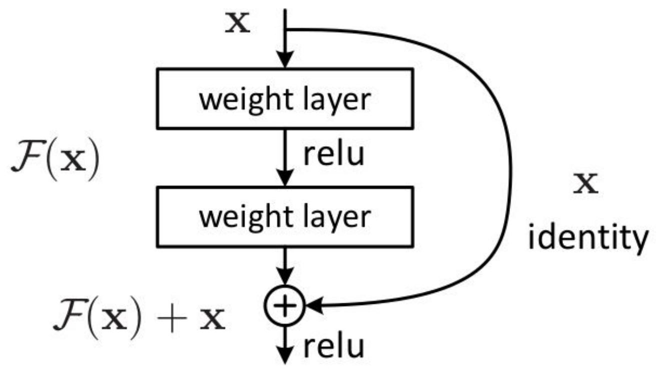

Figure 7: Model of the residual block [14, Fig. 2].

The residual block is capable of circumventing many issues with training deep networks because of how it allows the convoluted output of the previous layer to be added to the next. Subsequently, the input *x* and transformation function *F(x)* [14, eq. (2)] can be modeled as shown in (2).

$$\mathbf{y} = \mathcal{F}(\mathbf{x}, \{W_i\}) + W_s\mathbf{x}. \quad (2)$$

When implemented in convolutional networks, this allows for the output of the previous convolution to be passed along. Because of its implementation, this residual network was able to increase layer count while minimizing on error, as shown in Figure 8. Because of this residual block architecture, we are able to implement deep networks analyzing many features while reducing potential error due to the subsequent many hyperparameters.

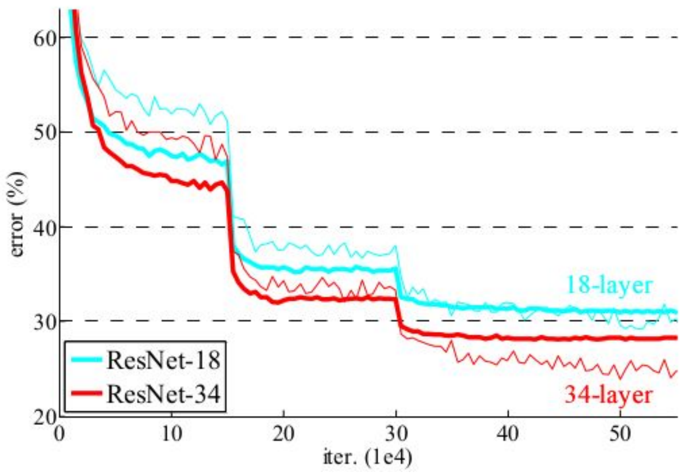

Figure 8: Thin curves denote training error, and bold curves denote validation error of the center crops. ResNets of 18 and 34 layers. [14, Fig. 4].

To analyze the image based features we used ResNet, a Semi-Supervised Convolutional Neural Network (CNN). ResNet was chosen as it and other CNN based learning algorithms have shown to have a strong use case in single-label medical image classification [15]. We used ResNet in four instances (cropped nuclei, Hog image, LoG image, and Gabor wavelet image) to produce four probabilities of a given nucleus being damaged or undamaged based on the given feature. As the amount of nuclei that have been classified as damaged is much less than the amount classified as undamaged, we weighed the damaged class training instances 3x more than those unweighted. This prevents the classifier from classifying everything as undamaged.

Another approach we used was random forest without imaged based features. Random forest is a classification algorithm consisting of many decision trees and uses feature randomness to build individual trees to try to create an uncorrelated forest of trees [16]. The random forest method utilizes bootstrap aggregation/bagging as an ensemble algorithm to randomly select data and create new training samples (bags) with which it feeds into the trees. After training, the random forest algorithm calculates predictions on all testing/unseen data *x'* by averaging all *b* predictions given by the individual trees as in equation (3).

$$\hat{f} = \frac{1}{B}\sum_{b=1}^{B} f_b(x') \tag{3}$$

This allows prediction by a series of trees which is more accurate than that of an individual tree. We used 100 estimators with max depth of 10 to prevent overfitting. To measure the purity of our input in the random forest we used Gini impurity (4) measure to analyze the degree in which a feature best quantifies DNA damage, and optimize the random forest accordingly. The Gini impurity [of a set *n*] is the probability of a randomly selected sample being classed incorrectly if that sample was labeled by the general distribution of samples inside that set. Equation 4 [16, eq. (1)] shows the Gini impurity of a set with *J* number classes on the percentage of each class p_i, squared.

$$I_G(n) = 1 - \sum_{i=1}^{J} (p_i)^2 \tag{4}$$

Interpreting the Gini impurity leads to insights regarding data misclassification. With a Gini value of 0, we are confident that each set/bag is "pure" in that there is no misclassification happening. This has tradeoffs in interpretation as it could mean that our model is overfitting to the training data. We want to have a low Gini impurity as it helps us decide how many decision trees are needed in the random forest.

As the amount of features is too high in comparison to the size of our dataset, we do not use the cropped nuclei, HoG, LoG, or Gabor Wavelets for Random Forest alone. Instead, we use an ensemble of ResNets to quantify each of these features to a likelihood. This reduces the size of each of these features to 1. As all of the features are reduced to a single description vector, we can then use mRMR to remove the redundant and irrelevant features from our feature space. The resulting features are then used for the final classification detailed in Figure 9.

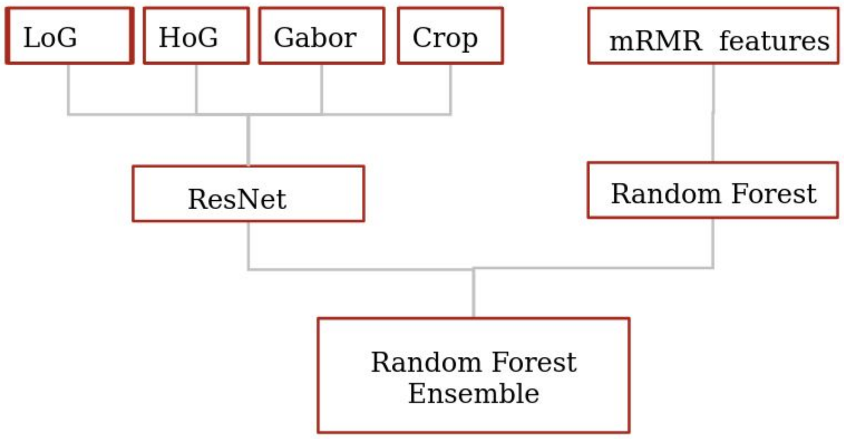

Figure 9: This figure shows what features were used with what classification methods. The non-image based beatures that were selected from mRMR were used as a random forest. Whereas the image based features were used in ResNet and returned as four features dictating the probability of damage. These features were then analyzed through random forest again to produce a final classification.

Assessment Protocols

Measuring our segmentation performance, we made use of the Average Precision (AP) confidence measure [17]. From calculating the Jaccard Index between nuclei (5) we computed the precision (6) of a given frame then the average precision of all the frames within our segmentation dataset. This can be seen in pseudo code in Figure 10.

$$IOU = \frac{Intersection\ of\ Ground\ Truth\ and\ Predicted}{Union\ of\ Ground\ Truth\ and\ Predicted} \quad (5)$$

$$\begin{aligned} TP &= True\ Positive \\ FP &= False\ Positive \\ FN &= False\ Negative \\ TN &= True\ Negative \end{aligned}$$

$$Precision = \frac{TP}{TP + FP} \quad (6)$$

```
Pseudocode for Our implementation of AP:
Input: Set of predicted and ground truth masks
Output: Average Precision of an algorithm over a whole dataset.
For each frame:
    For each nuclei in a frame
        Compute Precision where:
            $TP = IOU \geq 0.8$
            $FP = IOU < 0.8$
    Precision = Precision from all nuclei.
AP = Average Precision of all frames
```

Figure 10: This figure shows the pseudo code for AP. In our test we used a constant 0.8 threshold to ensure that our resulting segmentation algorithm of choice, produced a high frequency of accurate segmentations.

For the selection of the most discriminative features, a feature selection technique is employed which is based on the minimal-Redundancy–Maximum-Relevance (mRMR) criteria [18]. The mRMR criteria allows the features chosen to be ranked in a range beginning from the most powerful discriminative feature to the feature with the least discriminative power. This will allow us to determine the maximum relevancy (7) of a feature [18, eq. (4)]. As the Max-Relevance can include redundant features, or features that rely on each other. Using Minimum Redundancy (8) we can select mutually exclusive features [18, eq. (5)]. By optimizing these two parameters simultaneously (9) we can obtain an optimal feature set [18, eq. (6)]. To rank the features, mRMR discretizes each feature into 3 positions, the mean value of the feature ± its standard deviation. This method does assume that our features follow a unimodal-like distribution.

$$\max D(S,c), \quad D = \frac{1}{|S|}\sum_{x_i \in S} I(x_i; c). \tag{7}$$

$$\min R(S), \quad R = \frac{1}{|S|^2}\sum_{x_i, x_j \in S} I(x_i, x_j). \tag{8}$$

$$\max \Phi(D, R), \Phi = D - R. \quad (9)$$

To measure the rigidity and thoroughness of our classification techniques we used the k-fold cross validation [19]. Cross validation is a set of techniques used in machine learning to estimate its accuracy on new data. It is widely used in the field due to how it minimizes bias and variance. The dataset is split into *k* subsets For our experiment we chose k = 10 as it is a standard metric. This technique was used to get a more accurate estimate of our classifier's performance, as we want to prevent overfitting our classifier to the particular data being tested. Thus, shuffling the data amongst 10 sets and training it separately would give a more realistic answer into the accuracy when applied to new data. The pseudocode for k-fold cross validation can be seen in Figure 11.

```
Pseudocode for k-fold cross validation:
Input: Shuffled Dataset with validation labels
Output: Average test scores of an algorithm over a whole dataset.
K =10
For ki in K folds:
        Set fold Ki as test set
        Train model on remaining K-Ki
        Evaluate model performance on fold Ki
Calculate average performance over K folds
```

Figure 11: Pseudocode for k-fold cross validation. Our dataset is shuffled and separated into k equal subsets. By using k-1 of those sets to train our classifiers, we then use the remaining part for testing. This is repeated k times using a different testing part each time.

We took into account the average False Positive Rates (10), False Negative Rates (11), Specificity (12), Accuracy (13), Recall (14), Precision (6) and F1-score (15) of all k tests, to measure the classification algorithms' performance [20]. Due to the greater amount of non damaged nuclei to damaged nuclei in our dataset we wanted to prevent the classifiers from just detecting them all as non-damaged. F1-score allows us to focus on the classifiers accuracy as it is the harmonic mean of precision and recall. When classifying a dataset of this nature, it is important to weigh false negatives highly as it is a greatly unwanted result due to false relations that may be drawn. However it is also possible, although unlikely, that the nuclei in the dataset was not was not provided enough GFP to indicate damage. False positives may be passable due to weak phenotypes that may not be visible to a human. As such we look at the averages of these metrics to rate our classifiers.

$$False\ Positive\ Rate\ = \frac{FP}{(FP+TN)} \tag{10}$$

$$False\ Negative\ Rate\ = \frac{FN}{FN+TP} \tag{11}$$

$$Specificity\ = 1\ -\ False\ Positive\ Rate \tag{12}$$

$$Accuracy\ = \frac{(TP\ +\ TN)}{(TP+TN+FP+FN)} \tag{13}$$

$$Recall = \frac{TP}{(TP\ + TN)} \tag{14}$$

$$F1\ = \frac{2 * Precision * Recall}{(Precision + Recall)} \tag{15}$$

Results

Using our nuclei detection pipeline, we generated a dataset of cropped nuclei. This dataset included 1036 nuclei, 285 of which were considered to have DNA damage, 751 of which were considered to have no DNA damage. These nuclei were cropped from non time dependent frames such between the 3 movie datasets, such that no nuclei is represented more than once. Although some nuclei are still cropped close together, or cut in half due to watershed, these crops were kept as part of our analysis, as we did not have a lot of data points in comparison to the number of features.

From our tests using Resnet with each image based feature, it is important to note that although the use of a HOG did provide greater average F-Score, it is in no way the best metric. Our implementation of ResNet given the HoG feature tended to classify all nuclei as Healthy as it was the class with the most data points, despite us valuing damaged nuclei as 3 times more during training. Additionally, all the ResNet Classifiers are about 50% accurate as shown in Table 3.

Table 3: This table shows the F-score, accuracy, specificity, false negative rate, and false positive rates for all classifiers mentioned. Here true positive indicates that an undamaged nuclei was accurately predicted to not have damage.

Classifier / Metric	ResNet (Crop)	ResNet (HOG)	ResNet (LoG)	ResNet (Gabor)	Random Forest	Ensemble	Ensemble (+mRMR)
F-Score	0.262	0.468	0.340	0.298	0.511	0.540	0.591
Accuracy	0.596	0.330	0.527	0.509	0.727	0.740	0.759
Specificity	0.775	0	0.643	0.574	0.871	0.876	0.861
False Negative Rate	0.658	0.670	0.683	0.688	0.377	0.347	0.349
False Positive Rate	0.753	0	0.624	0.608	0.558	0.530	0.452

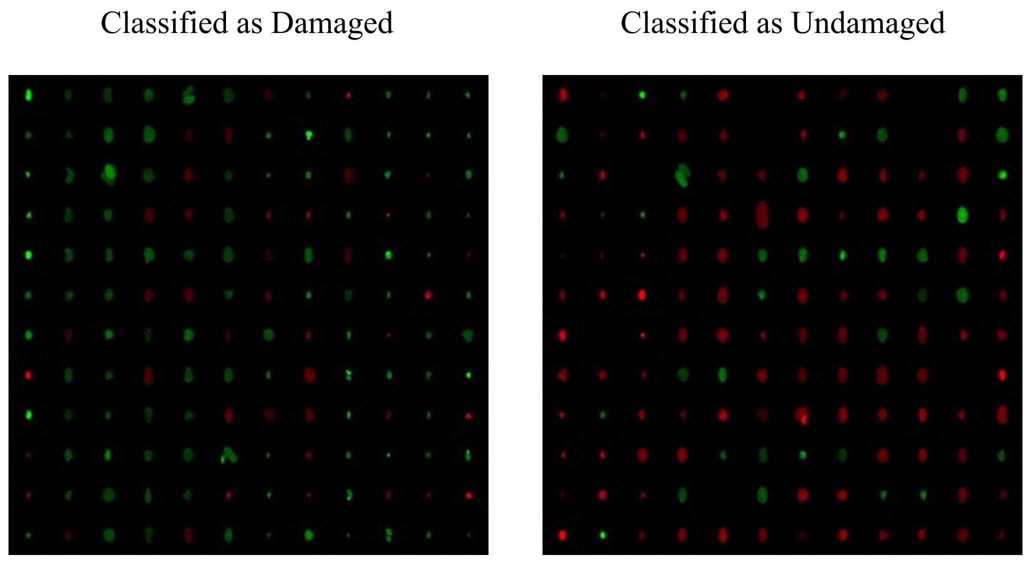

Figure 12: This shows a sample of 288 instances of nuclei that have been classified by our ensemble + mRMR as damaged [left] and undamaged [right]. All images shown are of only the original red channel. The nuclei color shifted to green are instances that have foci, but not the foci themselves. Ideally the left image would be of only green nuclei and the right image would have red nuclei. Here you can see examples of nuclei that were wrongly classified by our classifier.

Using an ensemble of Random Forest and ResNet, we were able to classify between the two classes with an average accuracy of 74% which is better than that of any feature alone. This is better visualized as shown in Figure 12. Expanding upon this, by removing unnecessary features (keeping the top 20) with mRMR we were able to reduce false positive rate by 8%. When constructing the Ensemble tree we looked at GINI purity features to show what features were most indicative of damage. In Table 4, you can see that the Haralick textures and Gabor Wavelets produce high factors (in comparison to others) to distinguish between classes. Although it is important to know that these features are not the only ones as features importances are sensitive to biases and features that still correlate receive lower scores.

Table 4: This figure shows the top 5 features for distinguishing between the classes based on their GINI scores, and the top 5 features deemed relevant by mRMR. In both cases Haralick features make a great showing providing evidence for its importance in detecting damage.

GINI Feature	Score	mRMR Feature	Score
Contrast	0.168	Sum Variance	0.585
Diff. Variance	0.105	HoG	0.066
Gabor Wavelet	0.081	Entropy	0.098
Correlation	0.066	Sum of Squares: Variance	0.121
Extent	0.046	Sum entropy	0.071

To expand upon the feature's distinguishability, we put the features through Primary Component Analysis as shown in Figure 13. To do this, all features were scaled between 0 and 1 to prevent variance maximization errors. We do not use this method for segmentation, we use it as a visualization of our results through the random forest ensemble. Through these figures you can see that the misinterpretation of the ground truth happens uniformly through the dataset, and that there is a slight distinction between the two classes. This leads us to believe that there may be other features to classify these nuclei more appropriately or our ensemble may have underfit our classification methods.

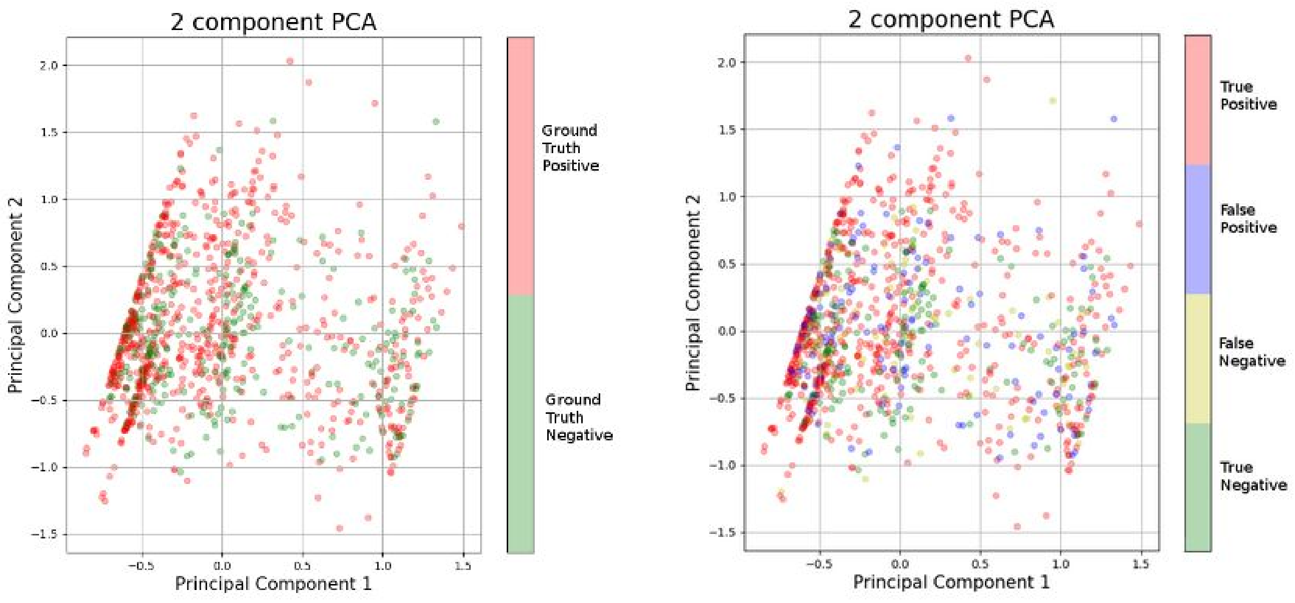

Figure 13: This figure shows the relationships between 2 primary components. These two graphs show our dataset predicted as damage or undamaged. This method gives us a visualization of our features and how they segregate nuclei. The first graph shows our ensemble attempt at segregating the nuclei where positive is healthy (red) nuclei, negative is damaged (green) nuclei, false positives are blue and false negatives are yellow. The second graph shows only the ground truths for easy comparison.

Discussion

Initially we experimented with a Bayesian approach, and found that it was an unsuitable algorithm for our use case. Mostly due to how more often than not, it would mark a cell's nucleus as unhealthy, despite them being healthy. The reasoning for this ties back into how Naive Bayes is strong at detecting variations in one feature in datasets. The fact with our dataset is that each cell, more often than not, varies from others in more than one attribute. The Bayesian classifier, while promising strong extensibility, is simply unusable due to the varying nature of cells.

We also attempted an unsupervised approach through the development of a Generative Adversarial Network (GAN). Although GAN proved to be an excellent use for image classification in bone marrow tests [21], our use case was too complex and resulted in negligible results. The approach is interesting because users don't define what makes up each label- the neural network does. This would aid us in identifying some of the key features that separate healthy nuclei from damaged ones without the need of the green channel, while providing the service of the cell classification. This approach may be worth revisiting upon obtaining a larger resolution dataset.

On the other hand, the semi-supervised learning approach through the CNN is leading promising results after being able to train on a relatively limited input dataset and differentiating the 2 classes well. The convolutional neural network structure was used as it was proven in Lundervold's experiment in 2018 [22] to be a strong classifier for damage in brain images via MRI analysis. The strength lies in the flexibility of the neural network architecture- the variable number of layers and the ReLu activation layers allows for a myriad number of input variations, and our dataset was one such example.

By the nature of our dataset there are many features that can distinguish between nuclei. Beyond the shape and texture features used, we hoped to capture features such as the current stage in the cell cycle. Although the morphology may be used to determine a nucleus's state in the cell cycle, we have not explicitly added the state as a feature. By not explicitly identifying nuclei by cell state it is impossible to obtain metrics about the distribution of visible damage between states in our dataset.

Future Work

To improve upon our work, we could have used more accurate segmentation techniques, and implement stronger textural features [23,24]. Due to the extreme lack of homogeneity in nuclear morphology, more advanced processing power leads to enhanced segmentation and cropping features, which in turn makes training classifiers all the easier of a task. Enhanced optics also means that the texture of the nuclei can be examined at a much higher degree. The improved quality of texture could be used as another feature to classify damage.

Furthermore, there is a possibility in improvements with the aid of 3-dimensional modeling [25]. These nuclei can be analyzed with more features and detail that will make the analysis much more definite. As deep learning becomes more prevalent in the medical community we hope that our insights derived through this paper can be interpreted for use across datasets of varying cell types so that chromosome damage and nuclear damage as a whole can be understood further.

The impact of time on nuclear damage was observed in the dataset. Further analysis of the tracking of cells throughout time allows more detailed insights to be drawn regarding the deterioration of nuclear morphology with respect to the passing of time. This would allow partially time based segmentation techniques such as U-NET to be utilized [26,27].

Beyond improvements in segmentation and feature selection there are many other points of interest to be extracted from our dataset. Instead of a binary (healthy/damaged) classification, future work should look towards obtaining a larger dataset to transform the labels into an interval scale based on the number of damage sites present. Recent work has shown success in separating Circulating Tumor Cells from healthy nuclei using SqueezeNet, obtaining a false positive rate of 3.270% [28]. Additionally, more features should be considered as, cell cycle stage or looking at the parent nuclei.

Ultimately, developments in optics and machine learning lead to higher degrees of feature engineering. In a system so complex as biomedical imaging analysis, more resolution creates more pixels to be analyzed, and subsequently more features to be created and sought after. Enhanced ability to detect and monitor DNA damage response and repair will greatly facilitate the identification and characterization of important regulators of the process that may be altered in disease contexts such as cancer.

Acknowledgements

We would like to acknowledge our advisors: Dr. Dmitry Korkin & Dr. Amity Manning for their guidance through the project and our family and friends for their emotional support.

References

1. K. W. Eliceiri, M. R. Berthold, I. G. Goldberg, L. Ibáñez, B. S. Manjunath, M. E. Martone, R. F. Murphy, H. Peng, A. L. Plant, B. Roysam, N. Stuurman, J. R. Swedlow, P. Tomancak, and A. E. Carpenter, "Biological imaging software tools," *Nature Methods*, vol. 9, no. 7, pp. 697–710, 2012.
2. S. Liu, P. A. Mundra, and J. C. Rajapakse, "Features for cells and nuclei classification," *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2011.
3. M. Sharma, R. Singh, and M. Bhattacharya, "Classification of breast tumors as benign and malignant using textural feature descriptor," *2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 2017.
4. G. Thibault, B. Fertil, C. Navarro, S. Pereira, P. Cau, N. Levy, J. Sequeira, and J.-L. Mari, "Shape And Texture Indexes Application To Cell Nuclei Classification," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 27, no. 01, p. 1357002, 2013.
5. E. Moen, D. Bannon, T. Kudo, W. Graf, M. Covert, and D. V. Valen, "Deep learning for cellular image analysis," *Nature Methods*, vol. 16, no. 12, pp. 1233–1246, 2019.
6. M. R. Lamprecht, D. M. Sabatini, and A. E. Carpenter, "CellProfiler™: free, versatile software for automated biological image analysis," *BioTechniques*, vol. 42, no. 1, pp. 71–75, 2007.
7. A. Lapytsko, G. Kollarovic, L. Ivanova, M. Studencka, and J. Schaber, "FoCo: a simple and robust quantification algorithm of nuclear foci," *BMC Bioinformatics*, vol. 16, no. 1, 2015.
8. D. K. Samuylov, P. Purwar, G. Szekely, and G. Paul, "Modeling Point Spread Function in Fluorescence Microscopy With a Sparse Gaussian Mixture: Tradeoff Between Accuracy and Efficiency," *IEEE Transactions on Image Processing*, vol. 28, no. 8, pp. 3688–3702, 2019.

9. S. Szenasi, Z. Vamossy, and M. Kozlovszky, "Evaluation and comparison of cell nuclei detection algorithms," *2012 IEEE 16th International Conference on Intelligent Engineering Systems (INES)*, 2012.
10. A. Khalil, J. L. Grant, L. B. Caddle, E. Atzema, K. D. Mills, and A. Arneodo, "Chromosome territories have a highly nonspherical morphology and nonrandom positioning," *Chromosome Research*, vol. 15, no. 7, pp. 899–916, 2007.
11. R. W. Veltri, S. Isharwal, M. C. Miller, J. I. Epstein, and A. W. Partin, "Nuclear roundness variance predicts prostate cancer progression, metastasis, and death: A prospective evaluation with up to 25 years of follow-up after radical prostatectomy," *The Prostate*, 2010.
12. R. Haralick, "Statistical and structural approaches to texture," *Proceedings of the IEEE*, vol. 67, no. 5, pp. 786–804, 1979.
13. P. Wang, L. Wang, Y. Li, Q. Song, S. Lv, and X. Hu, "Automatic cell nuclei segmentation and classification of cervical Pap smear images," *Biomedical Signal Processing and Control*, vol. 48, pp. 93–103, 2019.
14. K He, X. Zhang, S. Ren, J. Sun, "Deep Residual Learning for Image Recognition", *Cornell University Computer Science*, 2015.
15. S. S. Yadav and S. M. Jadhav, "Deep convolutional neural network based medical image classification for disease diagnosis," *Journal of Big Data*, vol. 6, no. 1, 2019.
16. F. Xia, W. Zhang, F. Li, and Y. Yang, "Ranking with decision tree," *Knowledge and Information Systems*, vol. 17, no. 3, pp. 381–395, Aug. 2008.
17. M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal Visual Object Classes (VOC) Challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, Sep. 2009.
18. H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, pp. 1226–1238, 2005.
19. N. Cristianini, "Cross-Validation (k-Fold Validation, Leave One Out)," *Dictionary of Bioinformatics and Computational Biology*, 2004.
20. C. Goutte and E. Gaussier, "A Probabilistic Interpretation of Precision, Recall and F-Score, with Implication for Evaluation," *Lecture Notes in Computer Science Advances in Information Retrieval*, pp. 345–359, 2005.
21. *Unsupervised Learning for Cell-Level Visual Representation in Histopathology Images With Generative Adversarial Networks - IEEE Journals & Magazine*. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8402089/metrics. [Accessed: 01-May-2020].
22. A. S. Lundervold and A. Lundervold, "An overview of deep learning in medical imaging focusing on MRI," *Zeitschrift für Medizinische Physik*, vol. 29, no. 2, pp. 102–127, 2019.
23. X. Lou, M. Kang, P. Xenopoulos, S. Muñoz-Descalzo, and A.-K. Hadjantonakis, "A Rapid and Efficient 2D/3D Nuclear Segmentation Method for Analysis of Early Mouse Embryo and Stem Cell Image Data," *Stem Cell Reports*, vol. 2, no. 3, pp. 382–397, 2014.

24. L.-K. Huang and M.-J. J. Wang, “Image thresholding by minimizing the measures of fuzziness,” *Pattern Recognition*, vol. 28, no. 1, pp. 41–51, 1995.
25. “We are human. We are the next frontier.,” *Allen Institute | Understanding the complexities of bioscience*. [Online]. Available: https://alleninstitute.org/. [Accessed: 01-May-2020].
26. J. C. Caicedo, A. Goodman, K. W. Karhohs, B. A. Cimini, J. Ackerman, M. Haghighi, C. K. Heng, T. Becker, M. Doan, C. McQuin, M. Rohban, S. Singh, and A. E. Carpenter, “Nucleus segmentation across imaging experiments: the 2018 Data Science Bowl,” *Nature News*, 21-Oct-2019. [Online]. Available: https://www.nature.com/articles/s41592-019-0612-7. [Accessed: 01-May-2020].
27. “U-Net: Convolutional Networks for Biomedical Image ...” [Online]. Available: https://arxiv.org/pdf/1505.04597.pdf. [Accessed: 01-May-2020].
28. K. Nakamichi, H. Lu, H. Kim, K. Yoneda, and F. Tanaka, “Classification of Circulating Tumor Cells in Fluorescence Microscopy Images Based on SqueezeNet,” *2019 19th International Conference on Control, Automation and Systems (ICCAS)*, 2019.
29. Additional Research Database

Code Repository: www.github.com/Mcdonoughd/MQP

Datasets:

Manning_Movie1: https://www.kaggle.com/coffeeoverflow/manning-movie1-part1
Manning_Movie2: https://www.kaggle.com/coffeeoverflow/manning-movie2
Manning_Simple: https://www.kaggle.com/coffeeoverflow/manning-simple