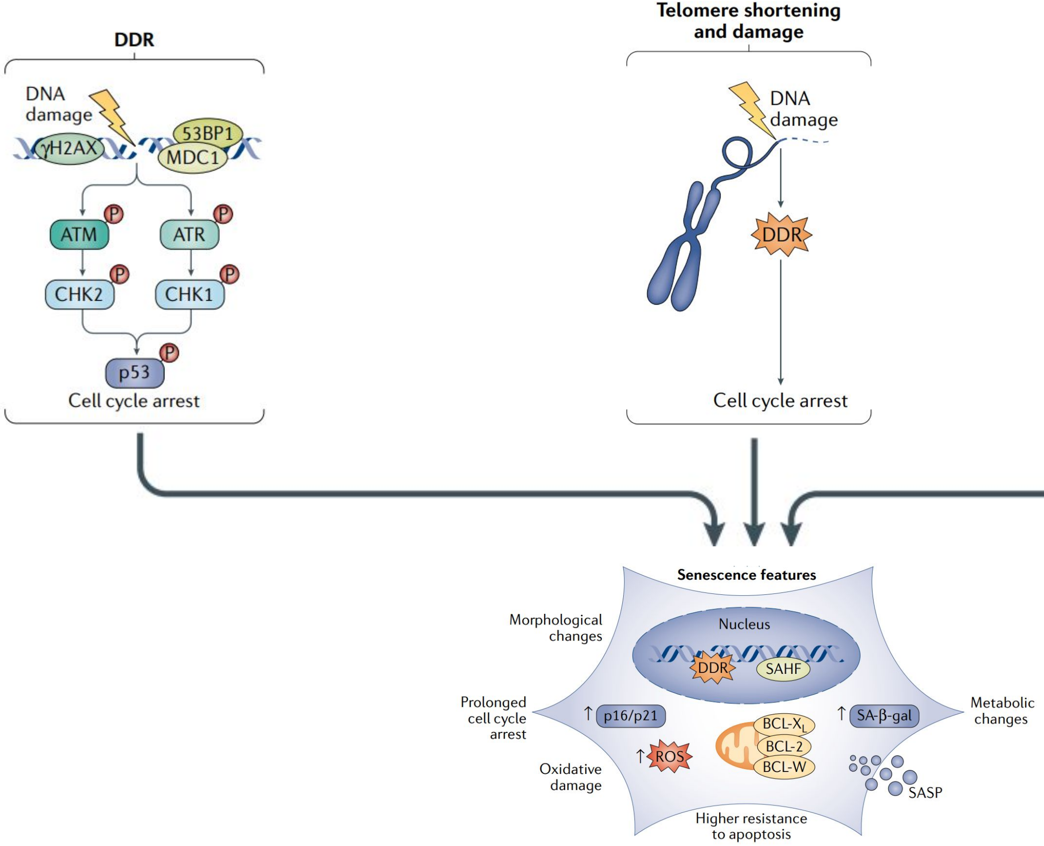
SenID: Identifying Senescent Cells Based on their Nuclear Morphology

Anthony Agudelo

Matthew Murakami

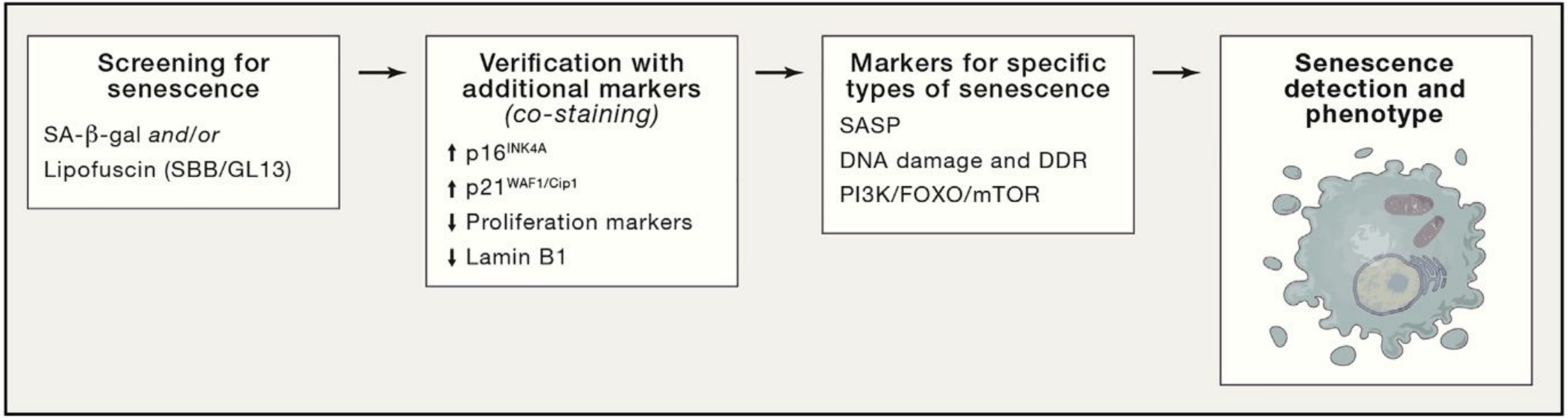
Nikolai Stambler

What is Senescence?



Micco et al. 2021

Identifying Senescence Cells



Gorgoulis et al. 2019

Morphology by Deep Learning

nature COMMUNICATIONS

ARTICLE

https://doi.org/10.1038/s41467-020-20213-0 OPEN

Anti-senescent drug screening by deep learning-based morphology senescence

Dai Kusumoto[1,2], Tomohisa Seki[3], Hiromune Sawada[1], Akira Kunitomi[4], Toshiomi Katsu
Shogo Ito[1], Jin Komuro[1], Hisayuki Hashimoto[1,2], Keiichi Fukuda[1] & Shinsuke Yuasa

TECHNICAL REPORT

https://doi.org/10.1038/s43587-022-00263-3

nature aging

Check for updates

OPEN

Nuclear morphology is a deep learning biomarker of cellular senescence

Indra Heckenbach[1,2,3], Garik V. Mkrtchyan[1], Michael Ben Ezra[1,4], Daniela Bakula[1], Jakob Sture Madsen[1], Malte Hasle Nielsen[5], Denise Oró[5], Brenna Osborne[1], Anthony J Covarrubias[6,7], M. Laura Idda[8,9], Myriam Gorospe[8], Laust Mortensen[4,10], Eric Verdin[2], Rudi Westendorp[4,10] and Morten Scheibye-Knudsen[1,3] ✉

Overview of Experimental Design

Feature Based DL: Deep Neural Networks

- Nuclear Segmentation
- Feature Extraction
- Training and Testing Model

Image Based DL: CNN, ResNet50

- Nuclear Segmentation
- Image Splitting
- Training and Testing Model

Overview of Experimental Design

Feature Based DL: Deep Neural Networks

- Nuclear Segmentation
- Feature Extraction
- Training and Testing Model

Image Based DL: CNN, ResNet50

- Nuclear Segmentation
- Image Splitting
- Training and Testing Model

Description of Data Set

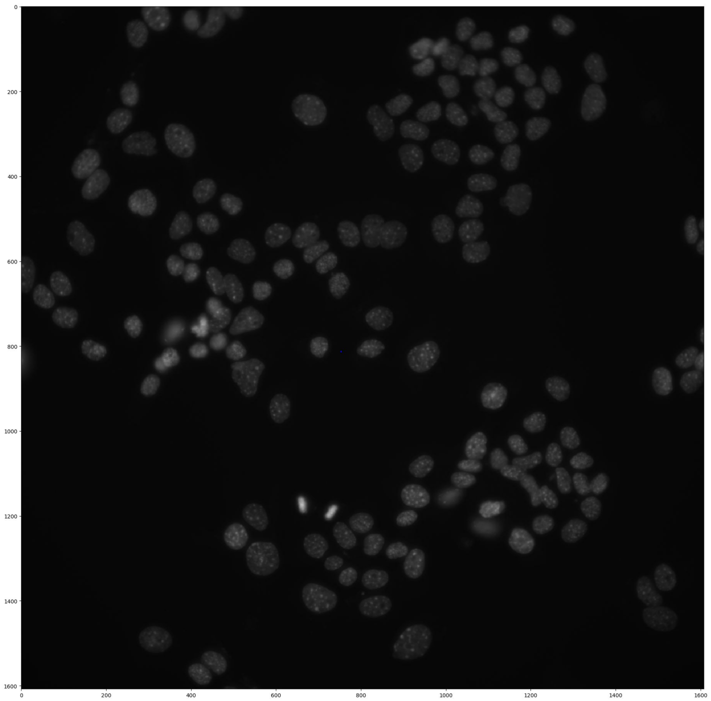
Human Set

- 515 Senescent Human Cells
- 1,093 Proliferating Human Cells

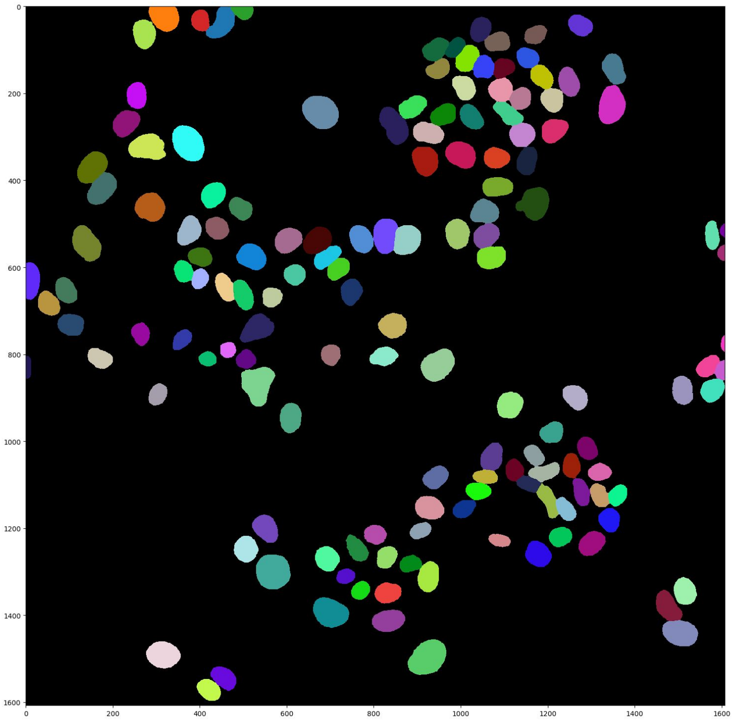
Mouse Set

- 1,732 Senescent Mouse Cells
- 38,869 Proliferating Mouse Cells

Nuclear Segmentation of Images: Proliferating

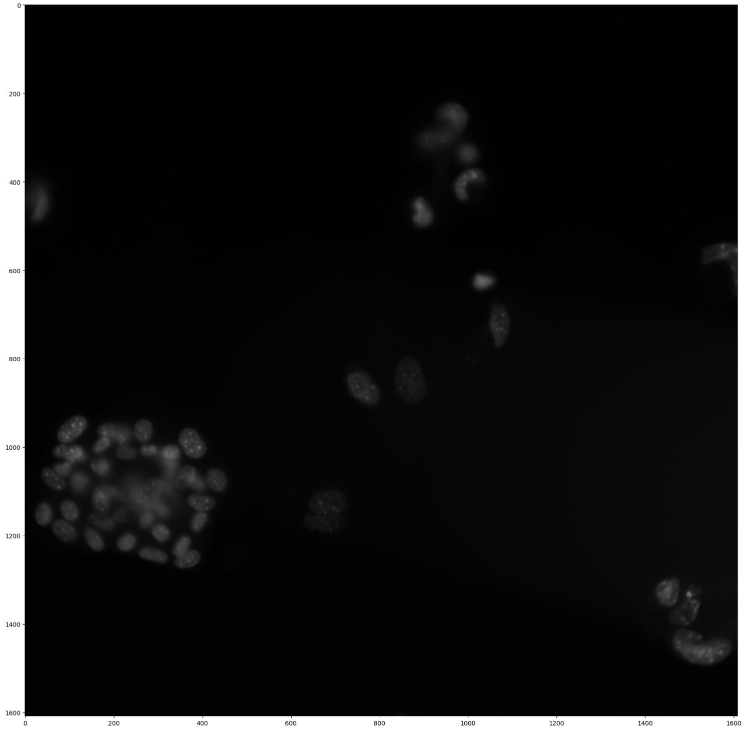


Captured: Dapi-Stained Image

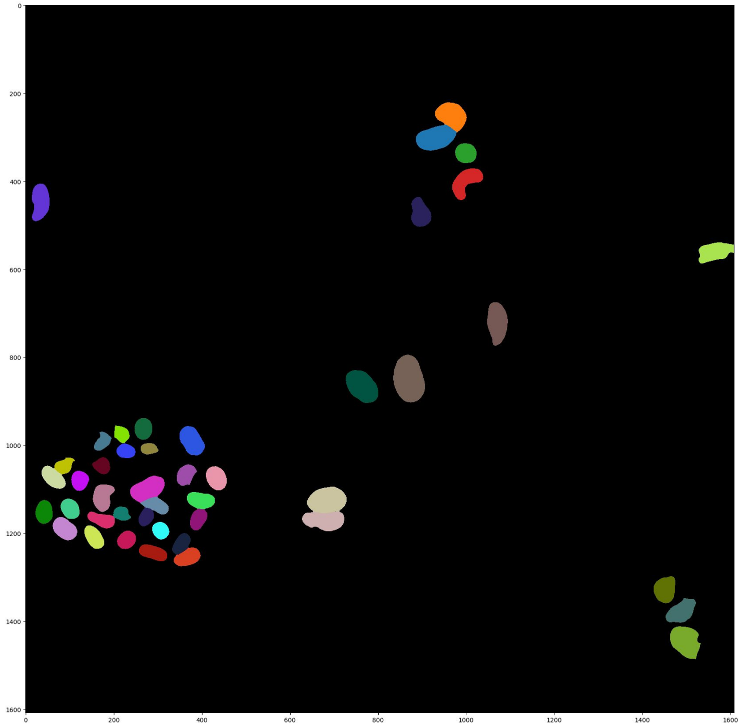


Nuclear Segmented Image

Nuclear Segmentation of Images: Senescent



Captured: Dapi-Stained Image



Nuclear Segmented Image

Feature Set for Deep Learning

Used a package called pycleperanto_prototype
to extract features from the segmented images

Images were described by **41** different features

Feature Set for Deep Learning

- mean_distance_to_centroid
- sum_distance_to_mass_center
- standard_deviation_intensity
- sum_distance_to_centroid
- bbox_min_y
- bbox_min_x
- bbox_min_z
- bbox_max_x
- bbox_max_y
- bbox_max_z
- bbox_width
- bbox_height
- min_intensity
- max_intensity
- sum_intensity
- mean_intensity
- sum_intensity_times_x
- mass_center_x
- sum_intensity_times_y
- mass_center_y
- sum_intensity_times_z
- mass_center_z
- sum_intensity_times_z
- sum_x
- centroid_x
- sum_intensity_times_z
- sum_y
- centroid_y
- sum_intensity_times_z
- sum_z
- sum_distance_to_mass_center
- mean_distance_to_mass_center
- max_distance_to_centroid
- max_distance_to_mass_center
- mean_max_distance_to_centroid_ratio
- mean_max_distance_to_mass_center_ratio
- mean_distance_to_mass_center
- centroid_z
- sum_distance_to_centroid
- mean_distance_to_centroid
- area

Feature Set for Deep Learning

- mean_distance_to_centroid
- sum_distance_to_mass_center
- standard_deviation_intensity
- sum_distance_to_centroid
- bbox_min_y
- bbox_min_x
- bbox_min_z
- bbox_max_x
- bbox_max_y
- bbox_max_z
- bbox_width
- bbox_height
- min_intensity
- **max_intensity**
- sum_intensity

- mean_intensity
- sum_intensity_times_x
- mass_center_x
- sum_intensity_times_y
- mass_center_y
- sum_intensity_times_z
- mass_center_z
- sum_intensity_times_z
- sum_x
- centroid_x
- sum_intensity_times_z
- sum_y
- centroid_y
- sum_intensity_times_z
- sum_z

- sum_distance_to_mass_center
- mean_distance_to_mass_center
- max_distance_to_centroid
- max_distance_to_mass_center
- mean_max_distance_to_centroid_ratio
- mean_max_distance_to_mass_center_ratio
- **mean_distance_to_mass_center**
- centroid_z
- sum_distance_to_centroid
- mean_distance_to_centroid
- **area**

Feature Set for Deep Learning

We ran into issues utilizing pyclesperanto_prototype to gather feature data

We are working to address these issues

Overview of Experimental Design

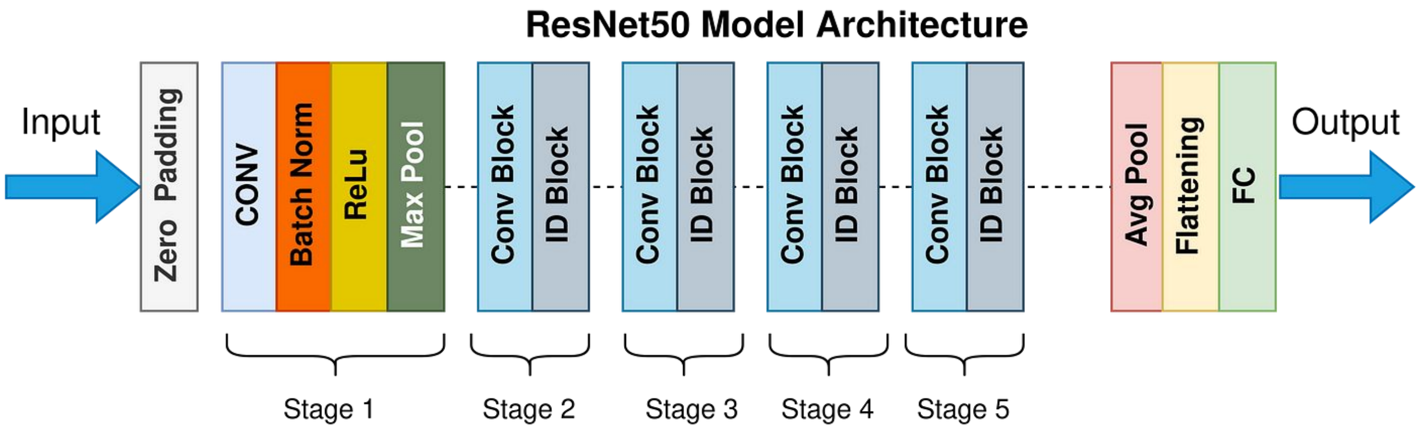
Feature Based DL: Dense Neural Network

- Nuclear Segmentation
- Feature Extraction
- Training and Testing Model

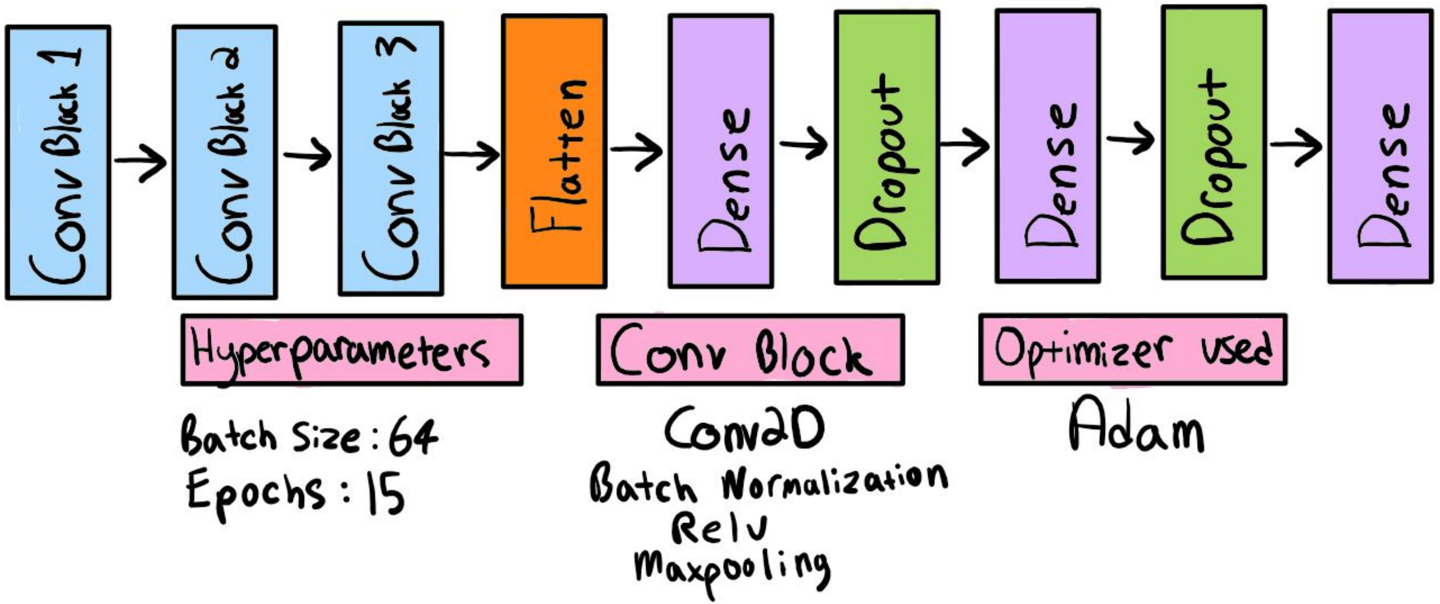
Image Based DL: CNN, ResNet50

- Nuclear Segmentation
- Image Splitting
- Training and Testing Model

ResNet50 Architecture

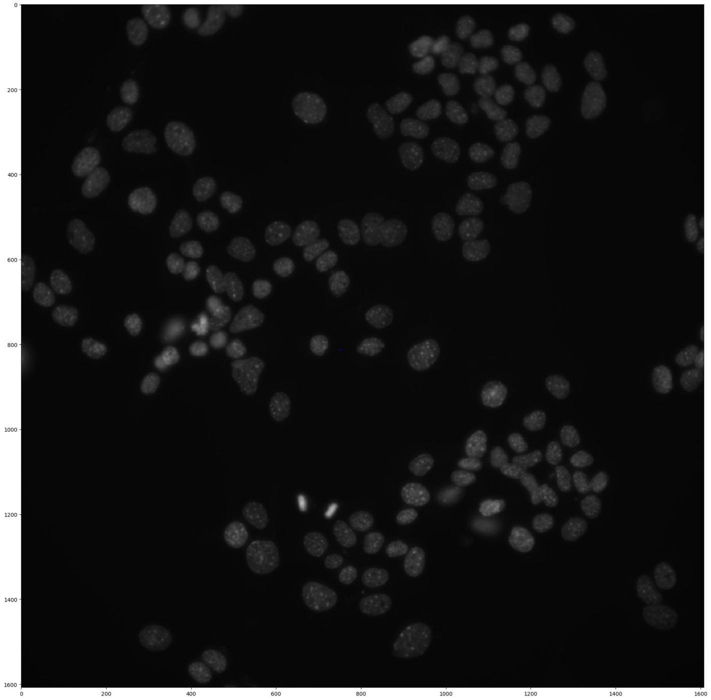


Custom CNN

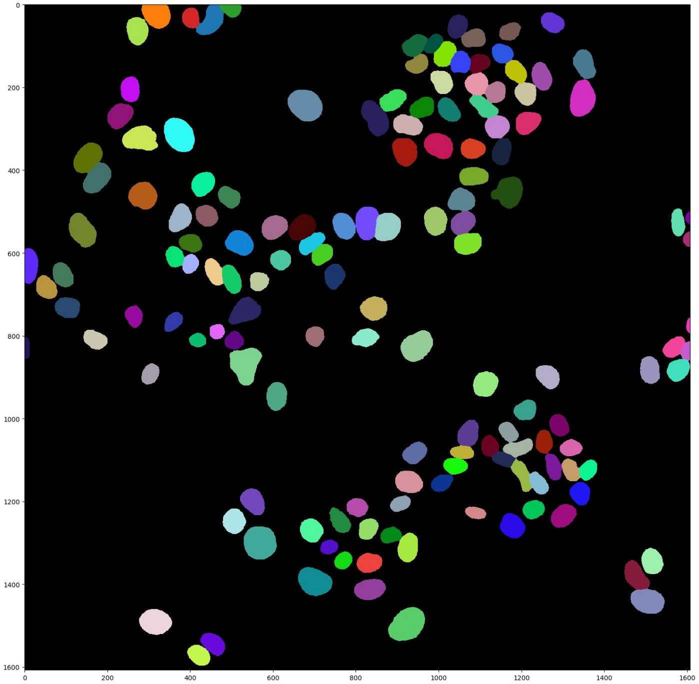


Conv Block 1
Conv Block 2
Conv Block 3
Flatten
Dense
Dropout
Dense
Dropout
Dense
Hyperparameters
Batch Size : 64
Epochs : 15
Conv Block
Conv2D
Batch Normalization
Relu
Maxpooling
Optimizer used
Adam

Nuclear Segmentation of Images: Proliferating

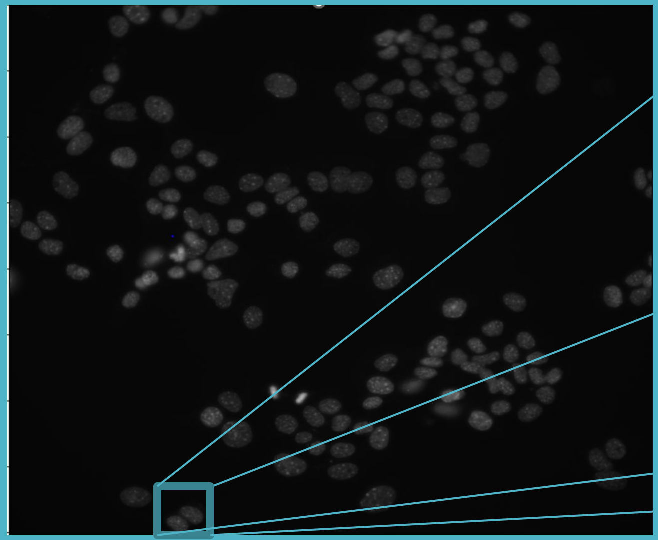


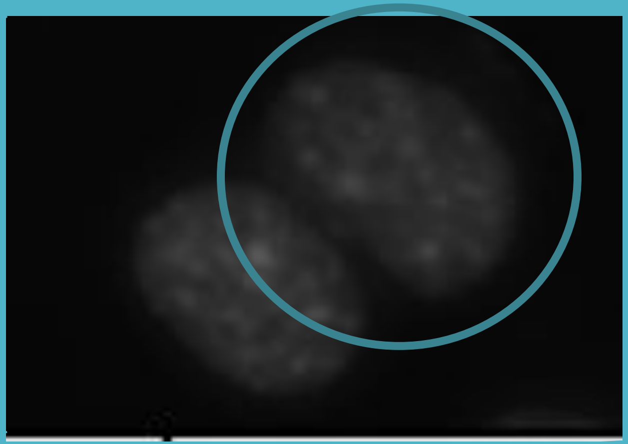
Captured: Dapi-Stained Image



Nuclear Segmented Image

Image Splitting





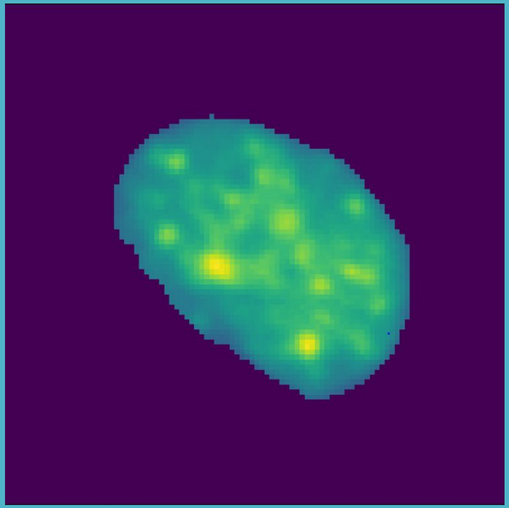
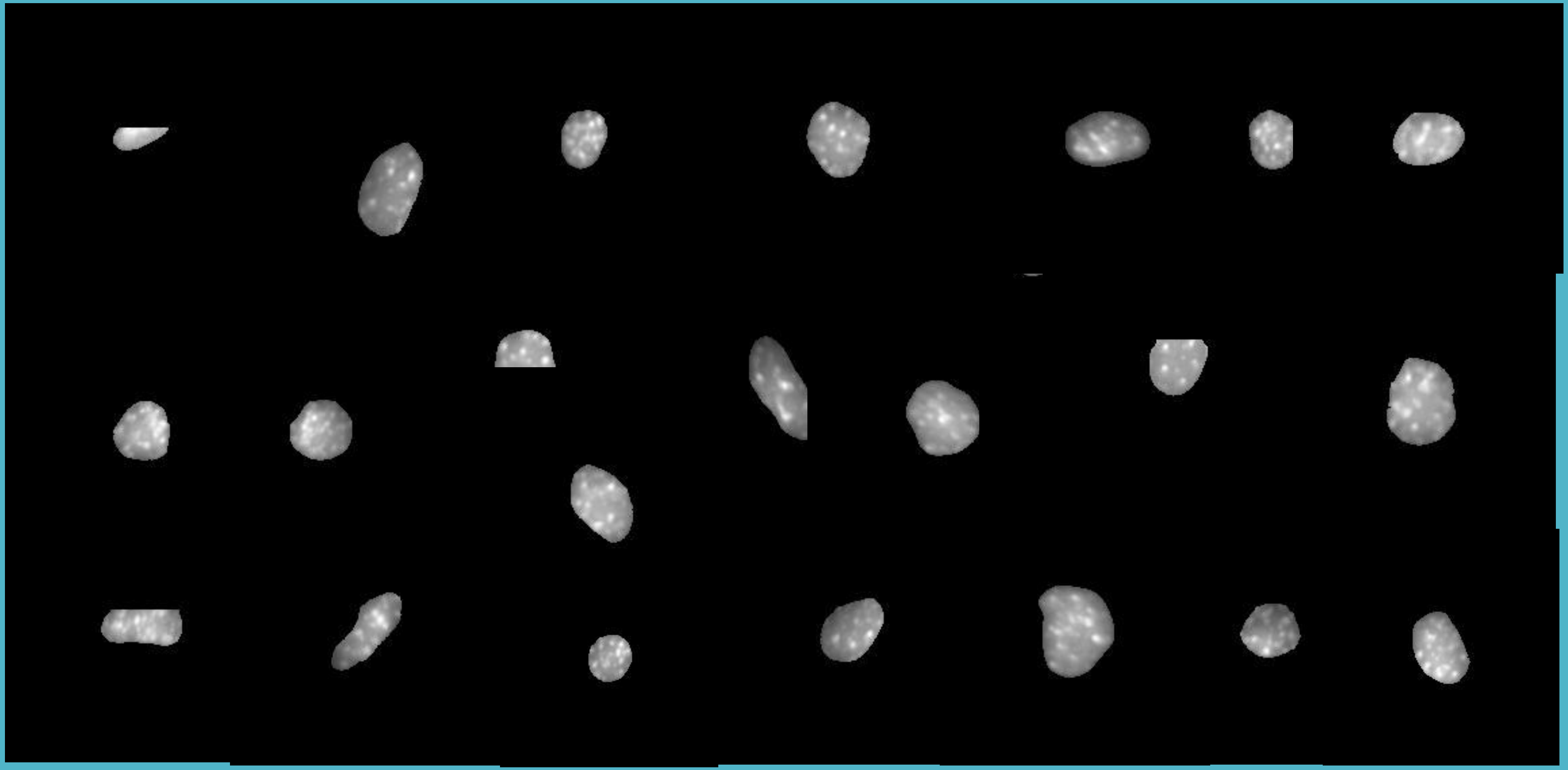
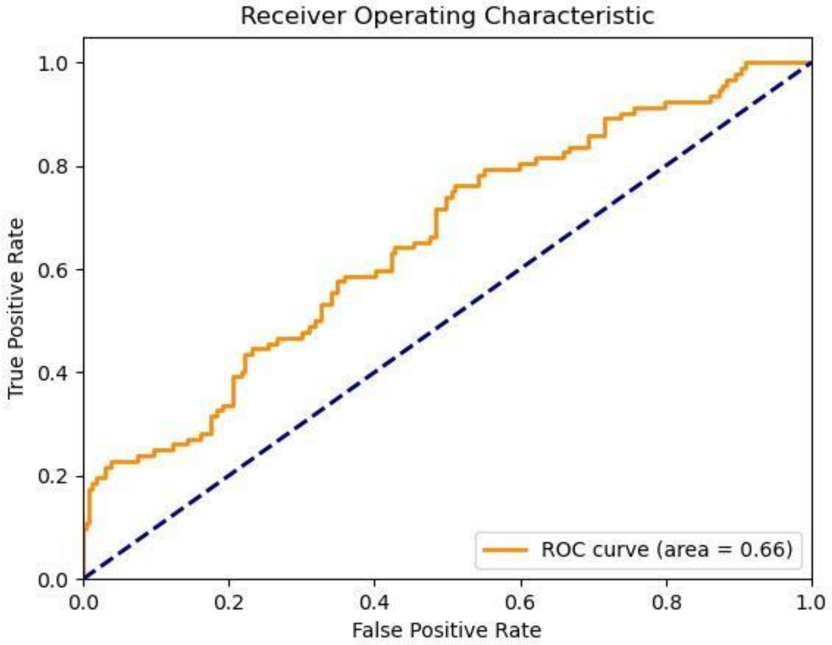
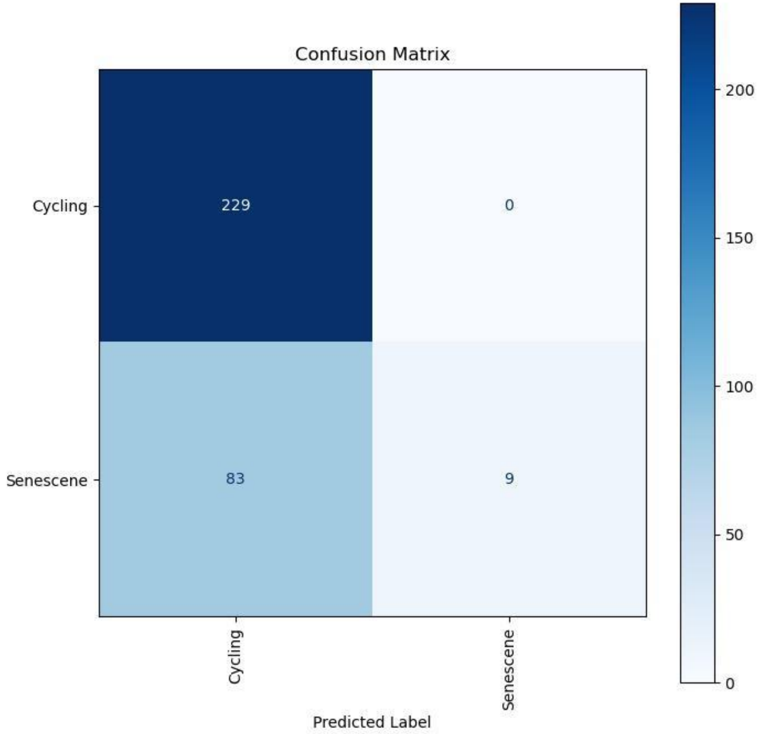


Image Splitting



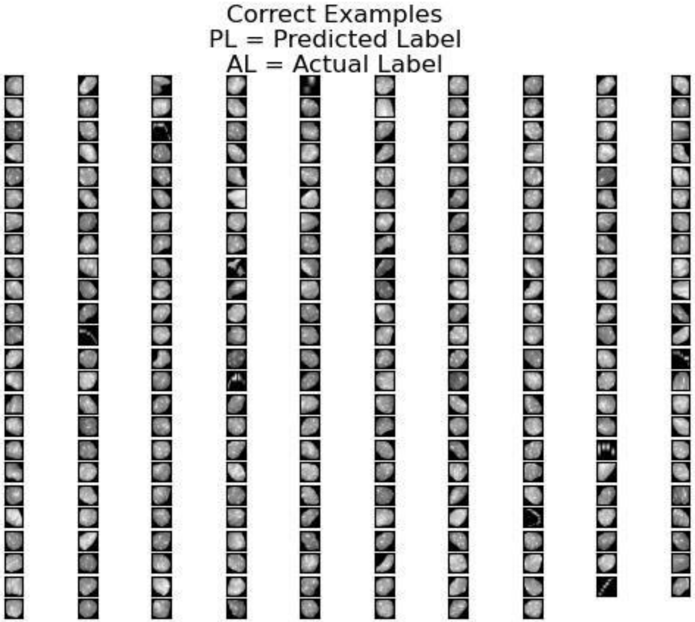
Custom CNN: Training and Testing on Human Cells

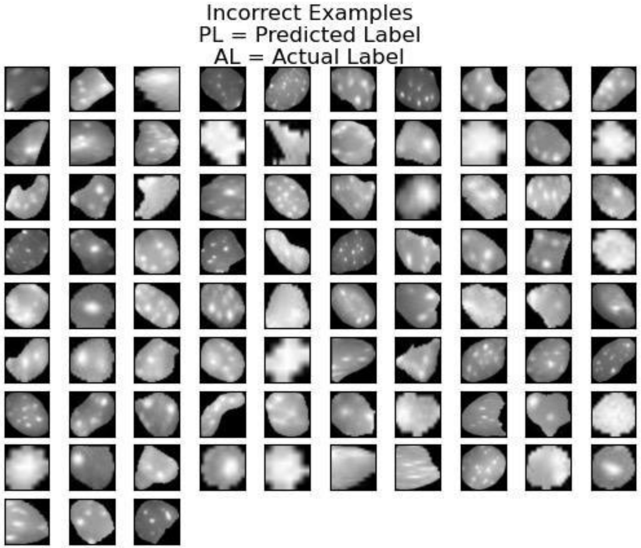




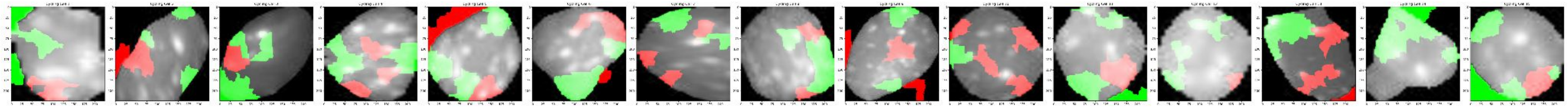
Test Accuracy: 74.13 Baseline Accuracy 71.34

Custom CNN: Training and Testing on Human Cells

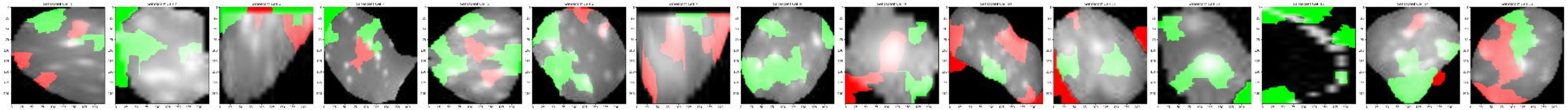




Cycling



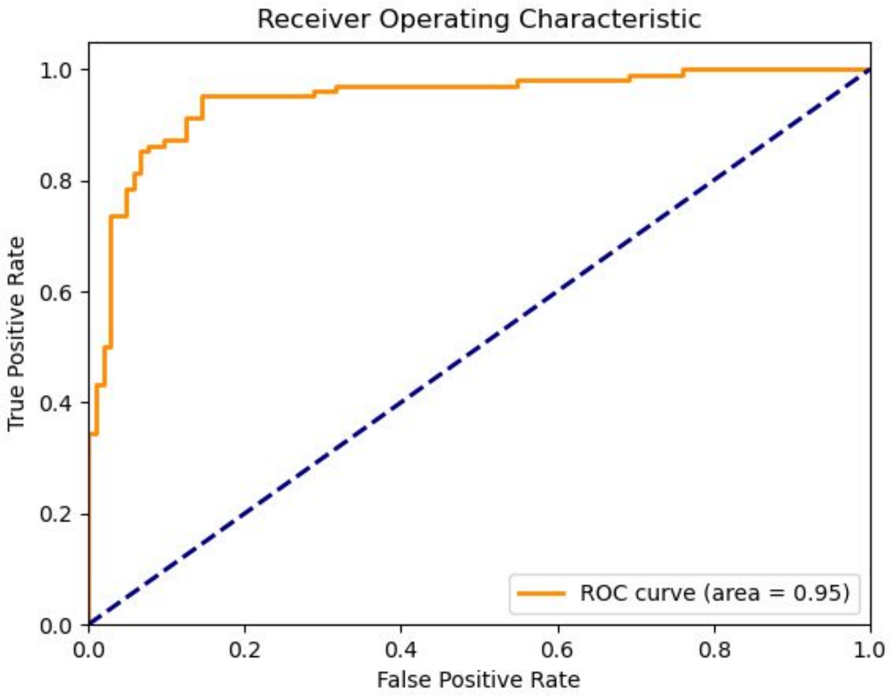
Senescent

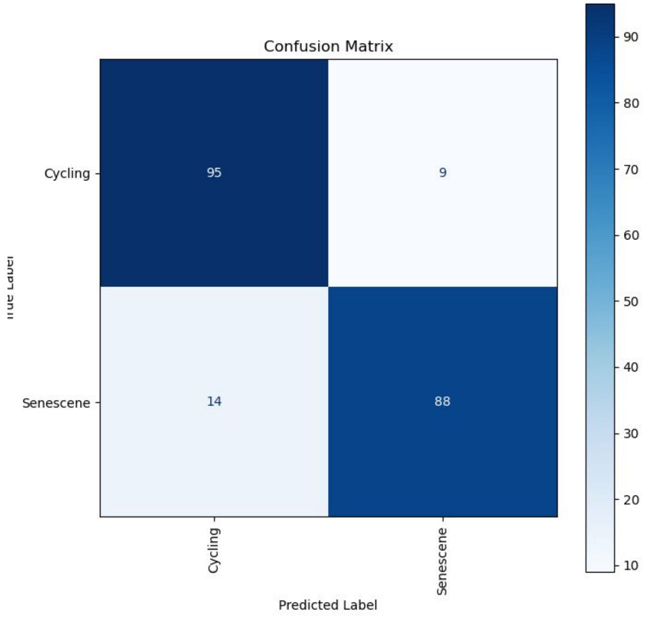


ResNet50: Training and Testing on Human Cells

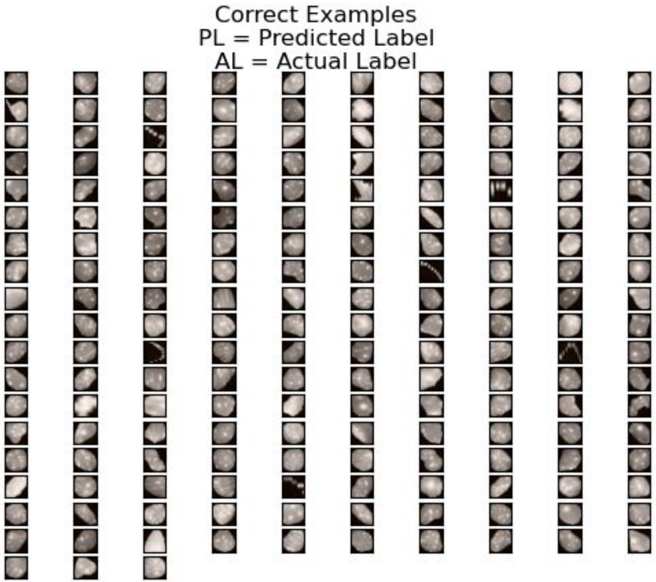
- Test Accuracy: 88.83
- Test Precision: 88.94
- Test Recall: 88.81
- Test F1: 88.82
- Test AUC: 94.65
- Size (~1000, 206)
- Baseline Accuracy: 50.45%

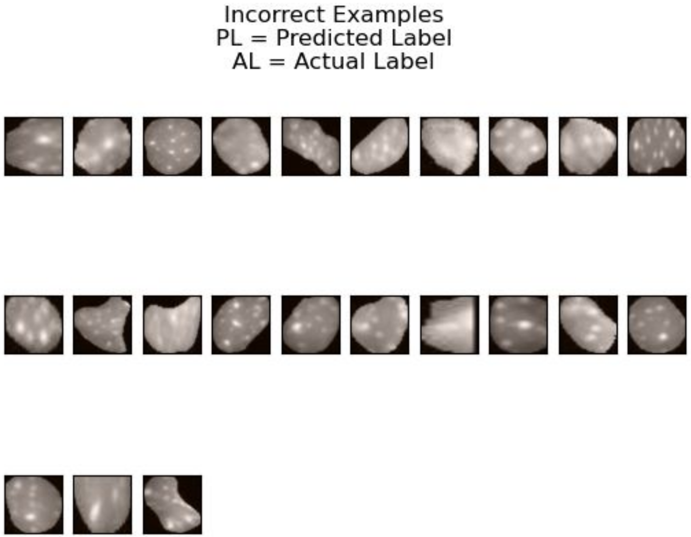
ResNet50: Training and Testing on Human Cells



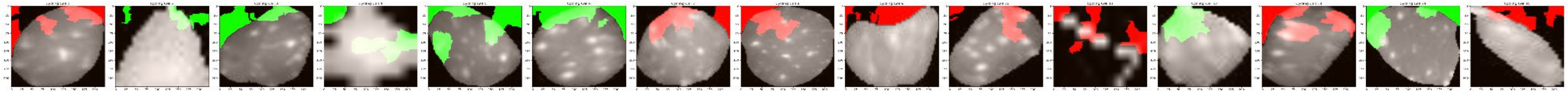


ResNet50: Training and Testing on Human Cells

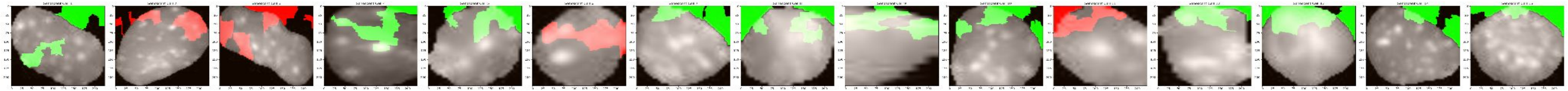




Cycling



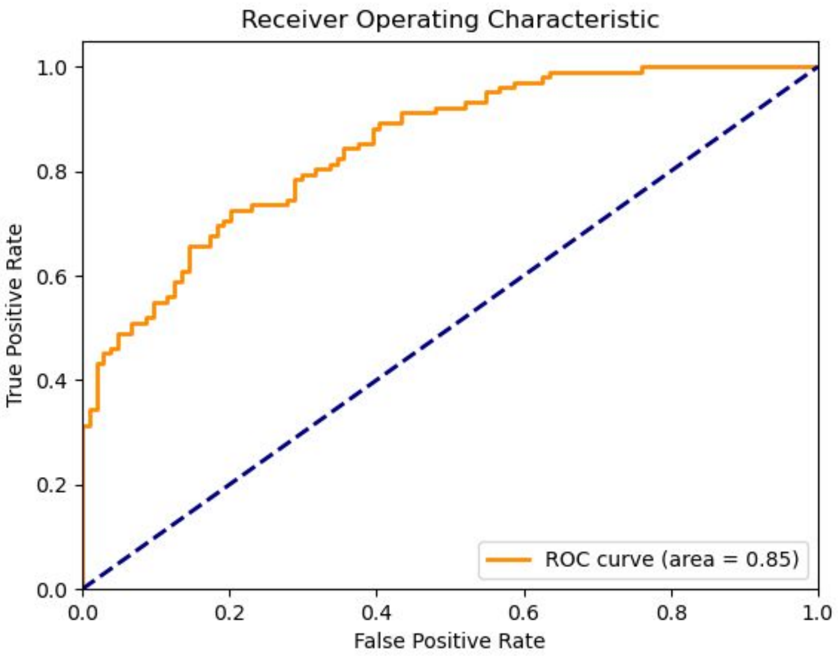
Senescent

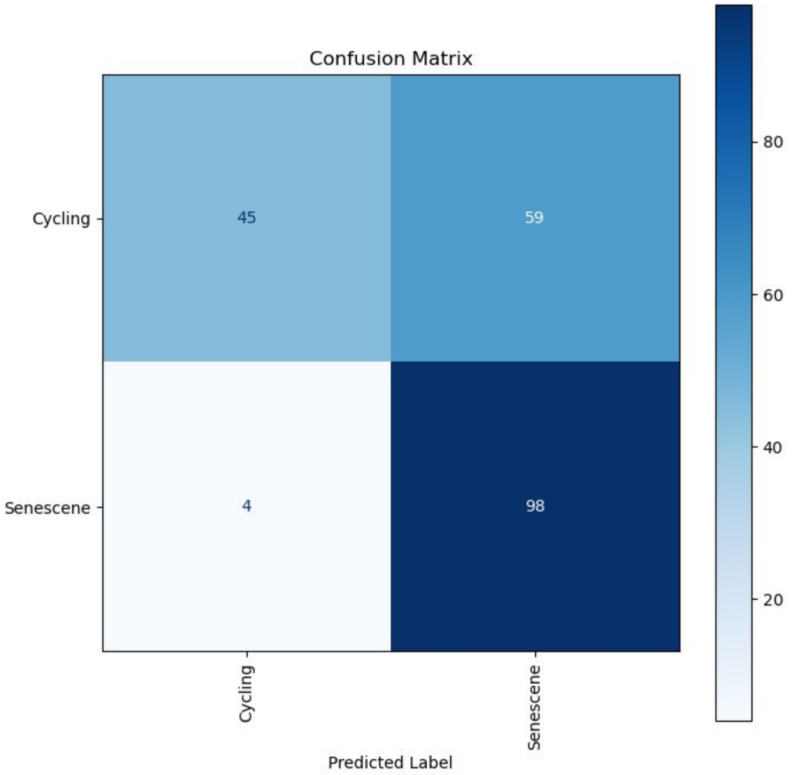


ResNet50: Training on Mice Cells and Testing on Human Cells

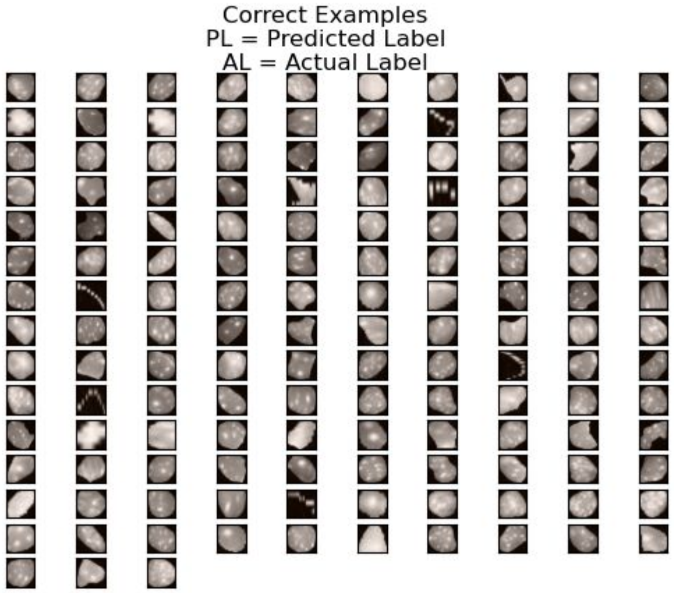
- Test Accuracy: 69.92
- Test Precision: 77.13
- Test Recall: 69.67
- Test F1: 67.25
- Test AUC: 84.79
- Size (2760, 206)
- Baseline Accuracy: 50.59%

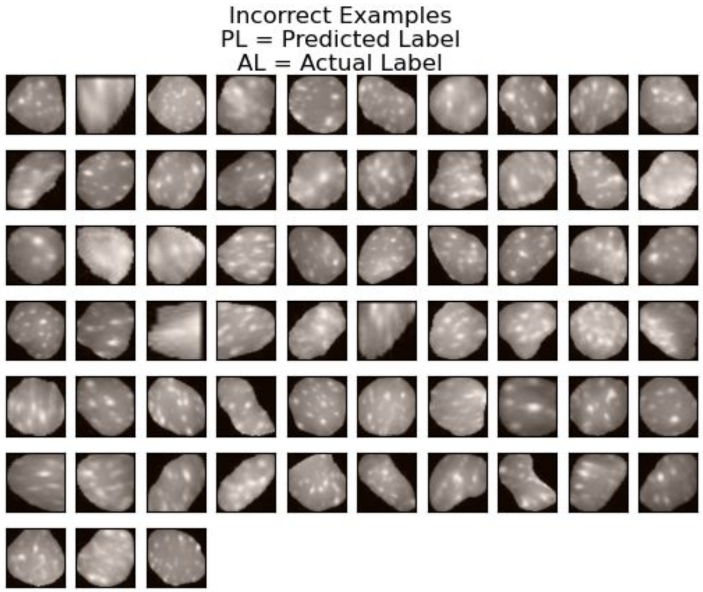
ResNet50: Training on Mice Cells and Testing on Human Cells



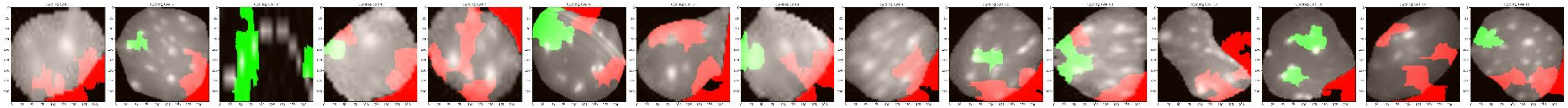


ResNet50: Training on Mice Cells and Testing on Human Cells

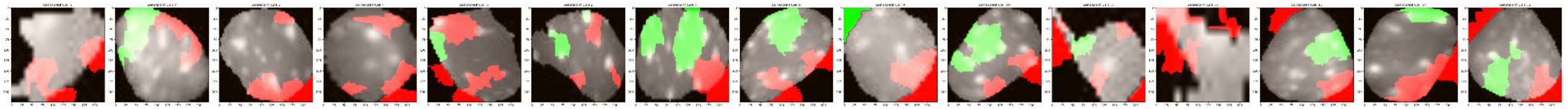




Cycling



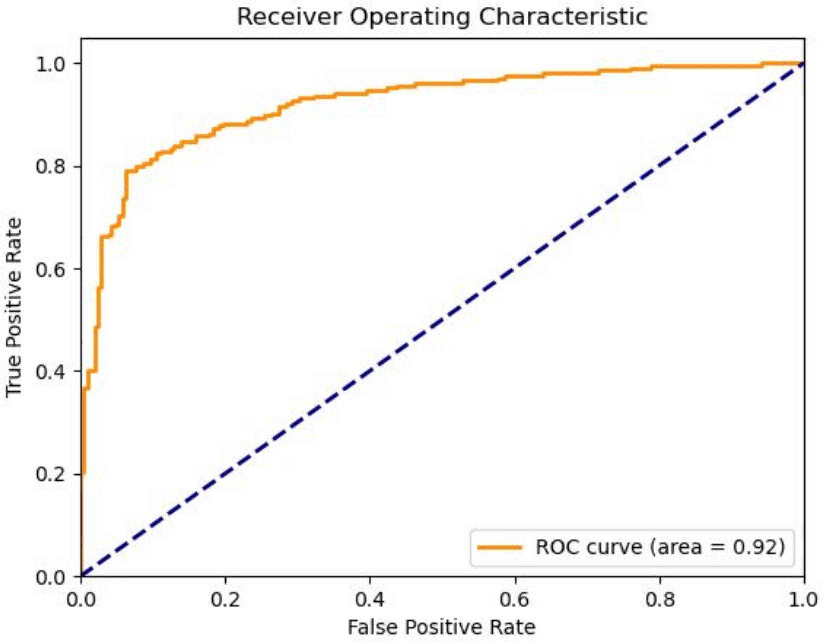
Senescent

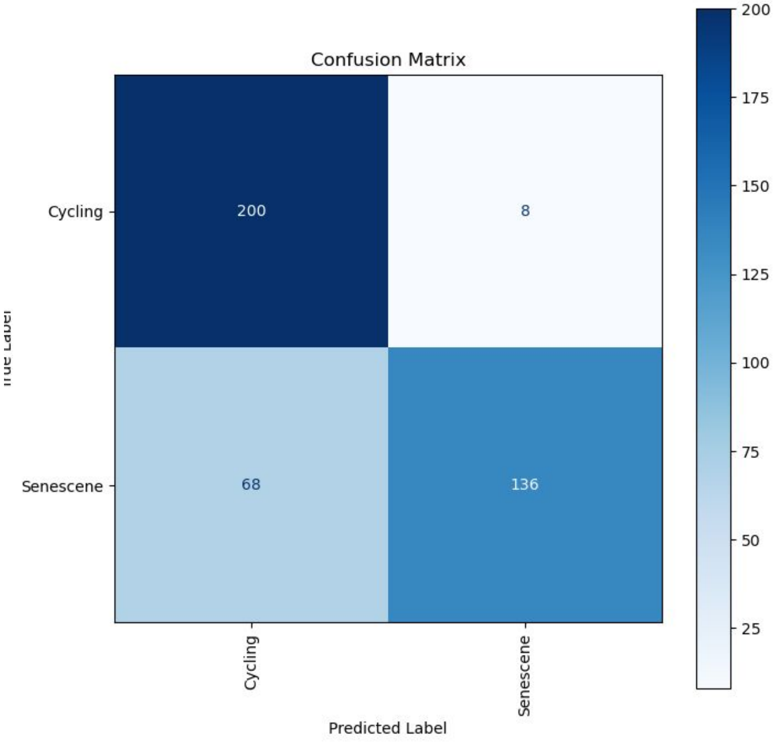


ResNet50: Training on Cells From Both Species Testing on Human

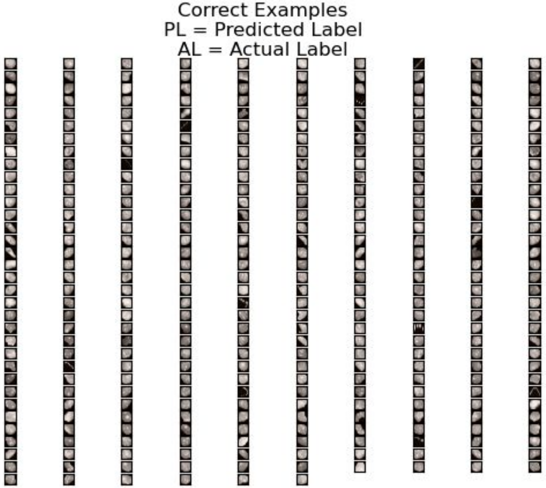
- Test Accuracy: 85.92
- Test Precision: 86.47
- Test Recall: 85.98
- Test F1: 85.88
- Test AUC: 91.88
- Size (1640, 206)
- Baseline Accuracy: 50.59%

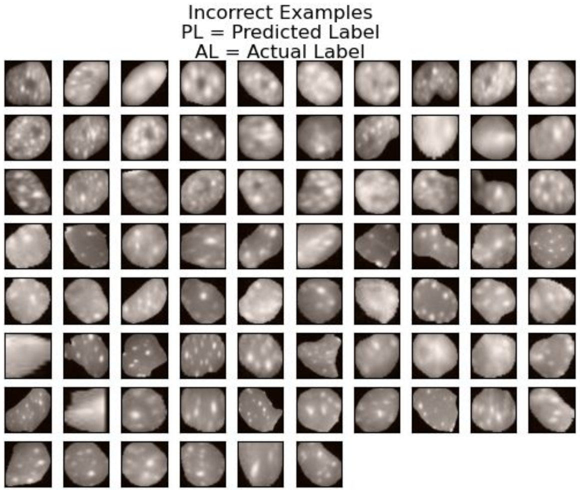
ResNet50: Training on Cells From Both Species Testing on Human



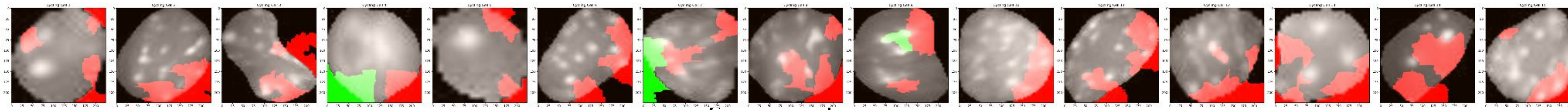


ResNet50: Training on Cells From Both Species Testing on Human

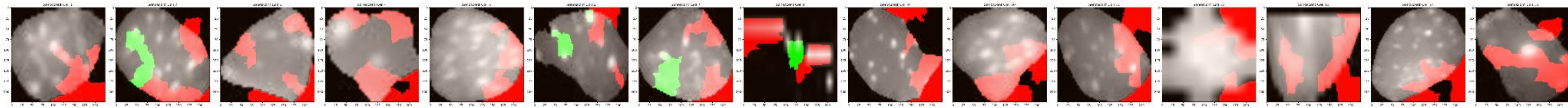




Cycling



Senescent



Conclusions

- Our Model Utilizes Reproducible/Clean Well Commented Code
- We are able to get similar accuracies as other papers in the field with a fraction of the cells
- When training on one species and testing on the other the model performs poorly this is likely due to morphological differences between the cells
- We are able to increase cross species testing metrics by training on both species
- Lime demonstrates that the perimeter of the cell is the most relevant for model performance

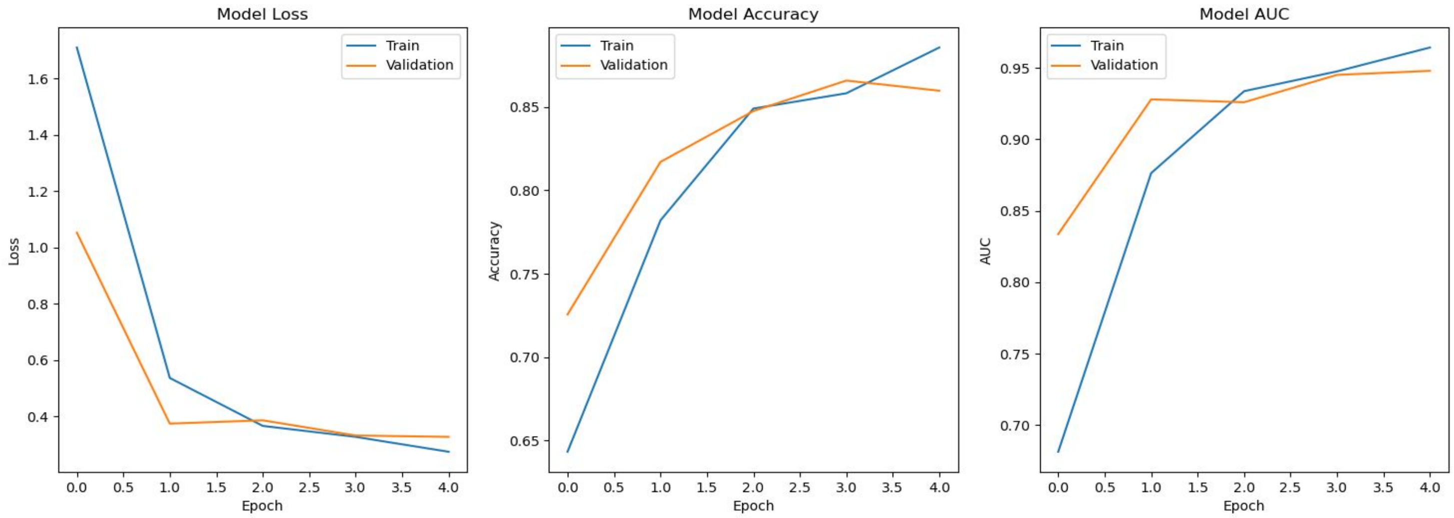
Future Directions

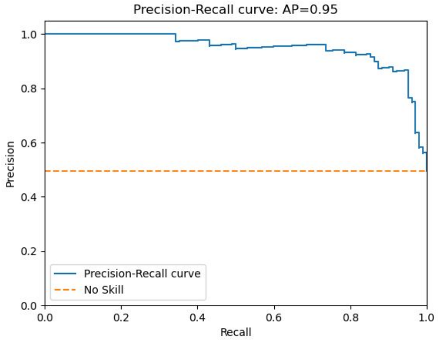
- Increase the number of Senescent cells in the dataset
- Finish the feature based DNN model
- Fine-tune hyper-parameters of the model
- Improve model interpretability
- Make our model more easily accessible to non-technical users
- Introduce multi-class classification to identify different types of senescence

Backpocket Slide

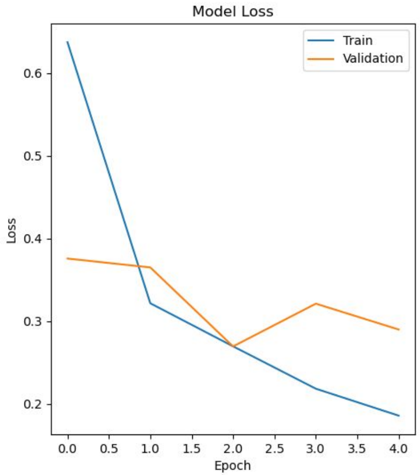
- We did a variety of tests the next few slides are for addressing other questions outside of the scope of the length of the presentation.

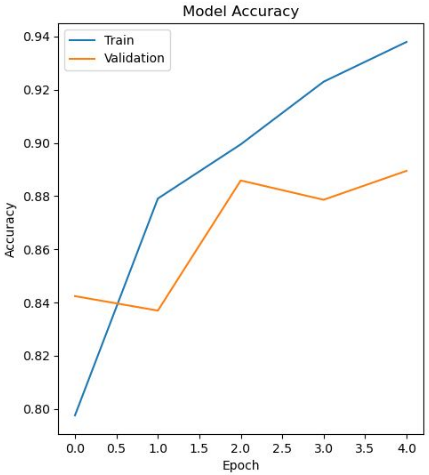
Training and Testing on Human Cells

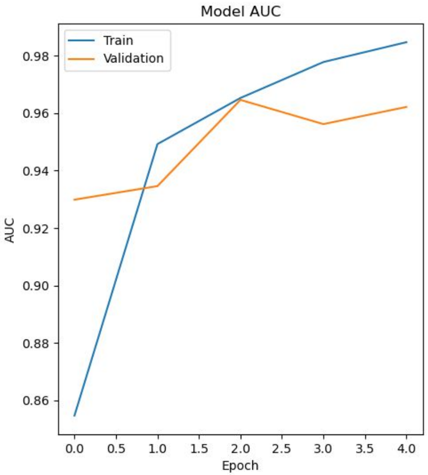


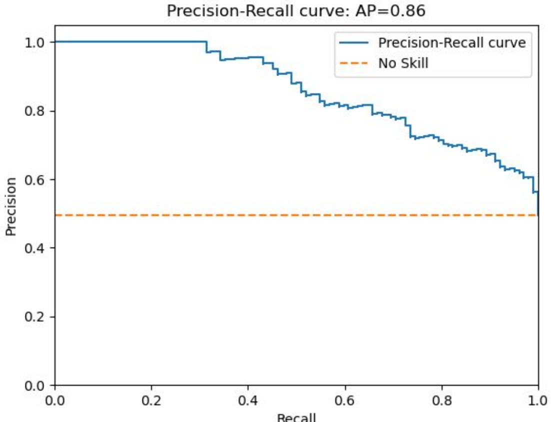


Training on Mice Cells and Testing on Human Cells









Training on Cells From Both Species Testing on Human