Title: SenID - Cell Senescence Classification

Who: Anthony Agudelo, Matt Murakami, Nikolai Stambler

Introduction:

Cellular senescence is characterized by cell cycle arrest and the transition to a non-proliferating state. Senescence is a hallmark of aging and is associated with the pathogenesis of many age-related diseases, such as Alzheimer's and dementia (Micco et al., 2021). Currently, the only way to identify senescent cells from proliferating cells is through a three-step process that involves the detection of SA-B-gal, co-staining of additional markers such as p16 and p21, and Lamin B1, as well as staining for markers to determine the specific type of senescence such as SASP, DNA damage/DDR, or PI3K/FOXO/mTOR (Gorgoulis et al., 2019). Here we propose a method for identifying senescent cells based on their nuclear morphology using CNNs: SenID. Furthermore, we trained and tested our model on cells from mice and humans to identify the features associated with senescence across cells from different species. In addition, to understand how our model works, we used LIME to help add interpretability to our model. Finally, we attempted to train a DNN model that would use SHAP to take in descriptive features such as area, max_intensity, and eccentricity and rank them by feature importance.

Methodology:

The overall process of our project is as follows:

1. The overall process of our project is as follows:
2. Cell Segmentation of the Nucleus
3. Cropping Images
4. Cropped Image Transformations
5. Training on 80% of image corpus
6. Perform a grid search to find optimal parameters
7. Testing on the remaining 20% of the image corpus
8. Use LIME and to interpret the model

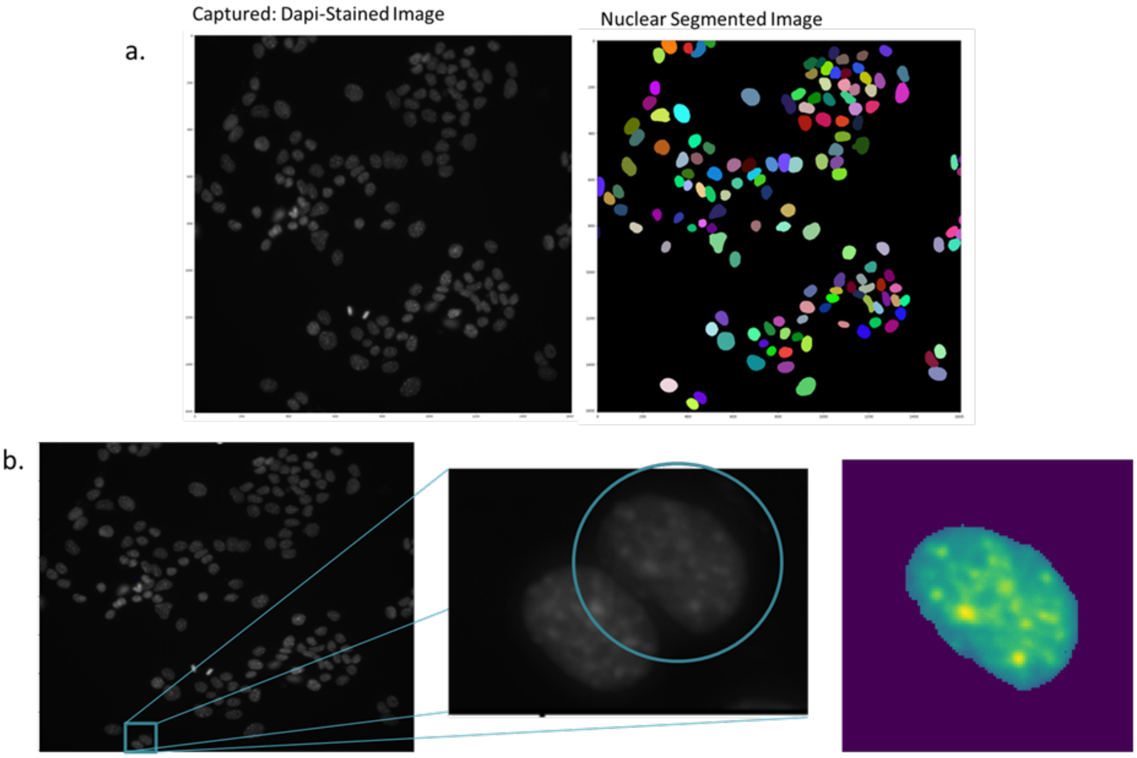


Fig 1. Segmentation and Splitting of Nuclei via Cellpose. **a** Captured Dapi-stained images (left) were segmented, and the segmented labeled image is shown (right). **b** Using the labeled image nuclei locations were used to grab isolate nuclei and center them on a 200x200 pixel grid resulting image was saved to a separate jpeg as a grayscale image.

The project required images of senescent and proliferating cells stained with markers for the nucleus. Cells were collected in-house by us and split into two groups, one cultured and fed proper growth factors to keep proliferating, and one treated with etoposide (a drug used to cause DNA damage) over two days to induce senescence. Samples were fixed and stained with a fluorescent marker of DNA (and, as a result, the nucleus), Dapi, and a short z-stacked image of the cells were captured using multiple channels on a fluorescent microscope.

Images were collected as .nd2, and the z-stack was compressed using a max-intensity projection and converted to a .tif in ImageJ. Next, the multi-channel .tif was converted to a single-channel image by selecting the first channel containing the Dapi signal. Images contained 10s to 1000s of cells, so a cell segmentation algorithm was utilized to isolate individual nuclei. For the cell segmentation algorithm, we used the Python library, Cellpose. The segmentation algorithm outputted one object utilized by our team, a label_image object, corresponding to an array of the same shape as the inputted single-channel image. This label_image 2D array indexed each nuclei in the image with a unique identifier and ascribed every pixel associated with that nuclei with that unique identifier. We could use this identifier to select the pixels associated with each unique nuclei in the single-channel image and populate the center of a 200 by 200 empty array with the corresponding pixel intensities. These arrays were then saved as grayscale .jpeg files in a new directory for processed samples. We then further cropped them by implementing an image cropping class that finds the largest contour of each image. We did this after realizing that a large portion of the image was dark space. After we then enlarged the images to 224 x 224 x 3 (repeating the greyscale channel to create an imitation of a RGB image). We did this as ResNet50 requires RGB images in this size.

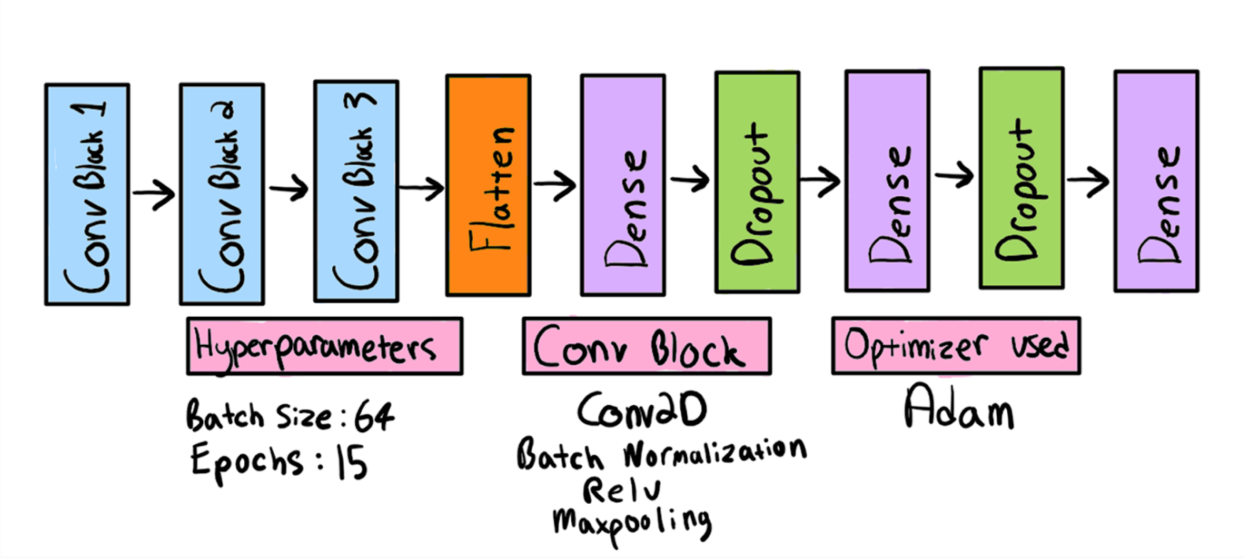


Fig 2. Custom Convolutional Neural Network Architecture

Our team utilized two CNN models to detect senescent morphology features effectively. The first model, a custom CNN, was developed by our team. Its architecture was derived from the model we used for a homework assignment, comprising three blocks. Each block consisted of a convolutional layer, batch normalization, ReLU activation, and max-pooling. Following these blocks, a flatten layer was applied, followed by dense and dropout layers (Figure 2). The second model we employed was ResNet-50, a well-established model for image classification tasks. To enhance its performance, we fine-tuned the model by training it further on our specific cell dataset. In order to retain important learned features, we froze certain layers during the fine-tuning process (Figure 3). In addition, our team endeavored to create a DNN capable of classifying cells based solely on a specific set of features extracted from each cell. We trained our CNN models using separate training and testing datasets to observe the impact of excluding or including certain cells on the model's accuracy – we made sure that all classes ad species in the test and training sets were equally balanced through custom stratifying functions and train-test-split stratify. Since our data was very imbalanced it was important we did this. Firstly, we trained the models on one species and tested them on both species. Subsequently, we trained the models on both species and tested them on the same species. To further optimize the performance of our models, we implemented a Search to identify optimal hyperparameters. Upon completing the training process for both CNN models, we successfully identified and interpreted senescent morphology features accurately. We employed saliency maps based on LIME to gain insights into the importance of different features in the classification process.

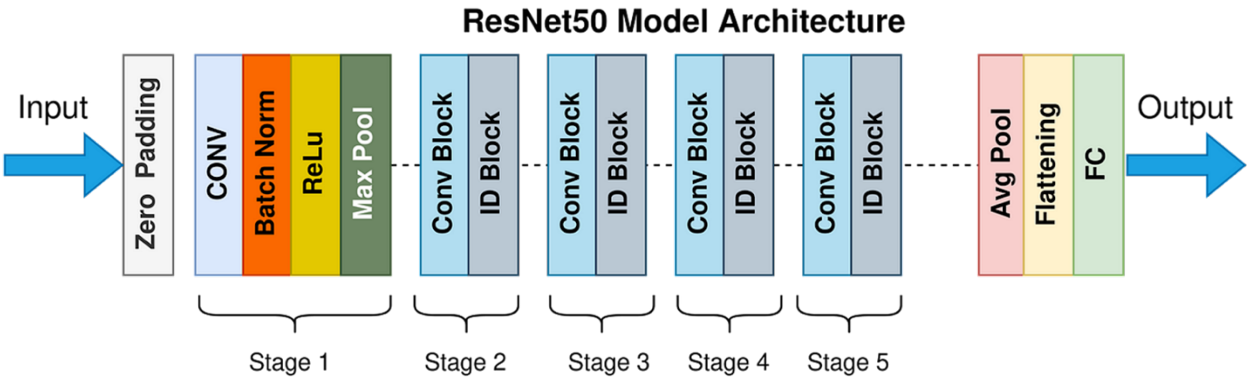
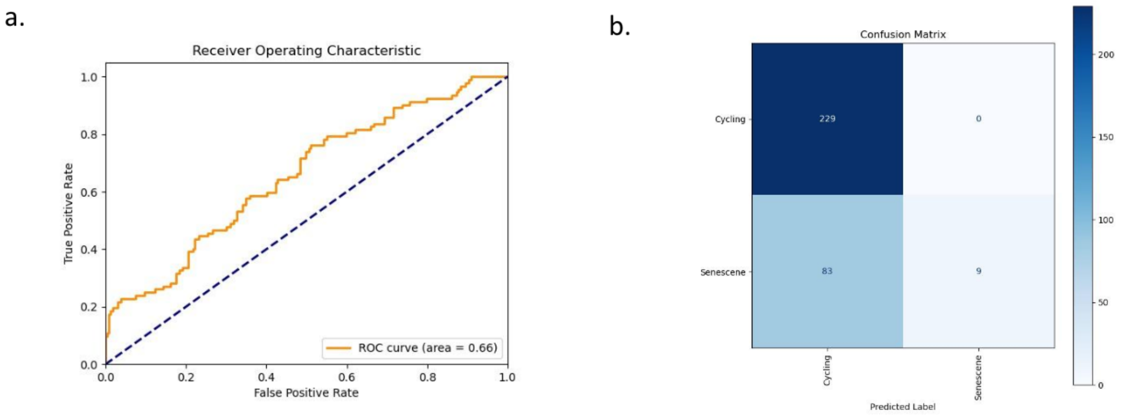


Fig 3. Architecture of ResNet50 model. Our model was fine-tuned by training it further on our specific cell dataset. In order to retain important learned features, we froze certain layers during the fine-tuning process

Results:

* Note on Current Results: Owing to modifications in our data splitting methodology, incorporating stratification, there is a slight discrepancy between the current results and those presented on DL Day. We have refined our splitting procedure to be more user-friendly and accurate over the long run, anticipating superior outcomes as we amass a more extensive dataset in the future. Therefore, while we will showcase the DL day results (with the exception of mouse training and testing with ResNet50 as this is our best model yet), please know that differing outcomes can be found in the visualizations folder. This divergence primarily stems from a considerable reduction in our training set size, notably in human data, leading to sub-optimal performance. Conversely, our results from mouse data have shown marked improvement, with an AUC of 96% and a Precision-Recall score of 97%. Consequently, we are confident that this shift in performance is predominantly attributable to the volume of training data.

Custom CNN: Training and Testing on Human Cells



Test Accuracy: 74.13 Baseline Accuracy 71.34

Fig 4. Model Performance of Custom CNN model both Tested and Trained on Human. **a** AUC of the Receiver Operator Characteristic. **B** Confusion matrix of the model shown.

As observed, this model behaves akin to a majority classifier, predicting cycling cells for almost every occurrence. As a result, the AUC is 0.66, indicating a modest improvement over a random/majority classifier but nothing substantial 71.34% baseline to 74.13% model accuracy. This is significantly different from our ResNet50 model, which demonstrate a substantial enhancement in performance, achieving an AUC of 95% and an accuracy of around 89%. This is illustrated in the subsequent figure.

ResNet50: Training and Testing on Human Cells

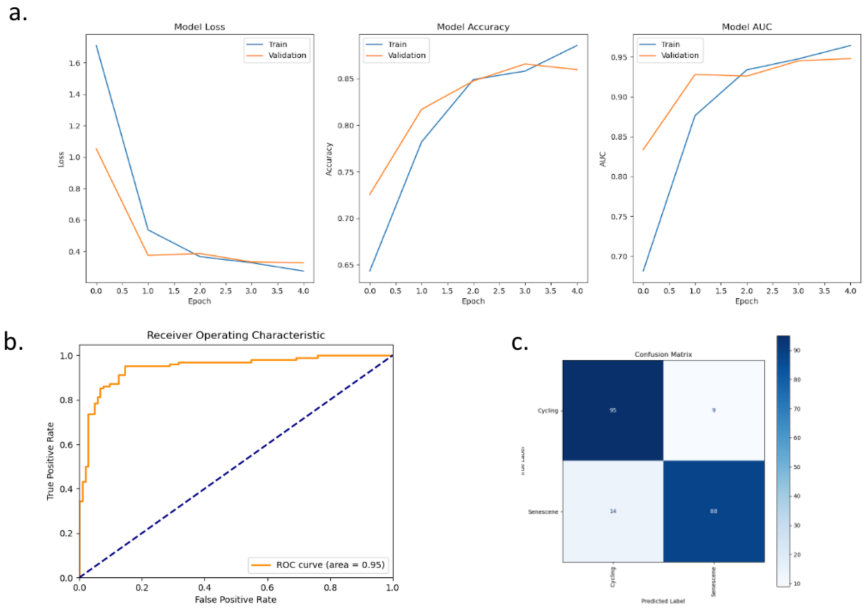


Fig 5. Model Performance of ResNet50 model Trained on Human Tested on Human. **a** Validation set model loss, accuracy and AUC over the 5 epochs are shown. **b** AUC of the Receiver Operator Characteristic. **c** Confusion matrix of the model shown.

This model was trained and tested on human cells and achieved a test accuracy of 88.83% and baseline accuracy of 50.45%, precision of 88.94%, and recall of 88.81% - indicating a well-balanced model. The F1 score was 88.82%, suggesting an excellent balance between precision and recall. The AUC of this model was 94.65%. This high AUC indicates that the model has an excellent measure of separability and is very effective at distinguishing between classes.

ResNet50: Training on Both and Testing on Human Cells

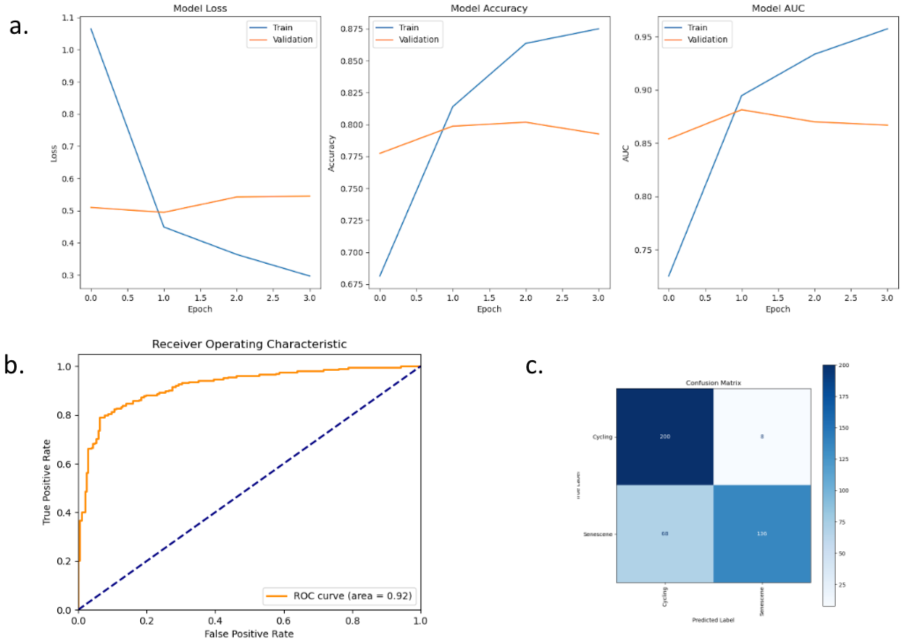


Fig 6. Model Performance of ResNet50 model Trained on Both Tested on Human. **a** Validation set model loss, accuracy and AUC over the 5 epochs are shown. **b** AUC of the Receiver Operator Characteristic. **c** Confusion matrix of the model shown.

This model was trained on human and mouse data and tested on human data. The model achieved impressive performance, with an accuracy of 85.92% - which is significantly higher than the baseline accuracy of 50.59%, precision of 86.47%, and recall of 85.98% - indicating a balanced performance in identifying true positives and avoiding false positives. The F1 score, a harmonic mean of precision and recall, was 85.88%, suggesting a strong balance between precision and recall. The AUC (Area Under the Curve) was 91.88%, significantly higher than the baseline accuracy of 50.59%, indicating that the model has a good measure of separability and can distinguish between classes.

ResNet50: Training on Mice Cells and Testing on Human Cells

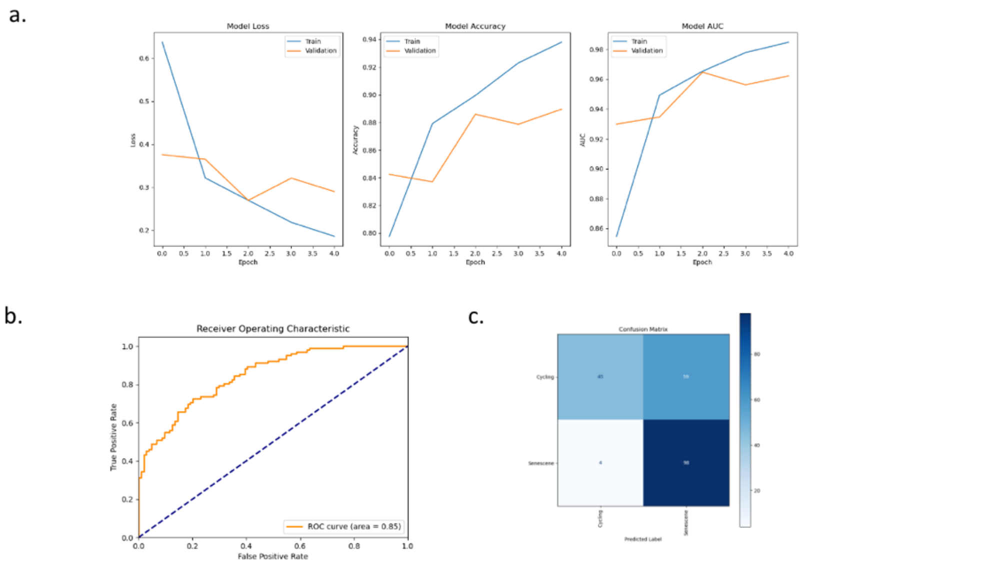


Fig 7. Model Performance of ResNet50 model Trained on Mice Tested on Human. **a** Validation set model loss, accuracy and AUC over the 5 epochs are shown. **b** AUC of the Receiver Operator Characteristic. **c** Confusion matrix of the model shown.

This model was trained on mouse cells and tested on human cells. The test accuracy was 69.92% - which is moderately higher than the baseline accuracy of 50.59%., and the precision was 77.13%, indicating that when the model predicts a cell is cycling, it is correct about 77.13% of the time. The recall of 69.67% shows that the model identified 69.67% of all actual cycling cells. The F1 score was 67.25%, showing a reasonable balance between precision and recall, though slightly leaning towards precision. The AUC was 84.79%, which, while lower than the previous model, is still significantly higher than the baseline accuracy of 50.59%.

ResNet50: Training on Both and Testing on Mice Cells (with updated data splitting)

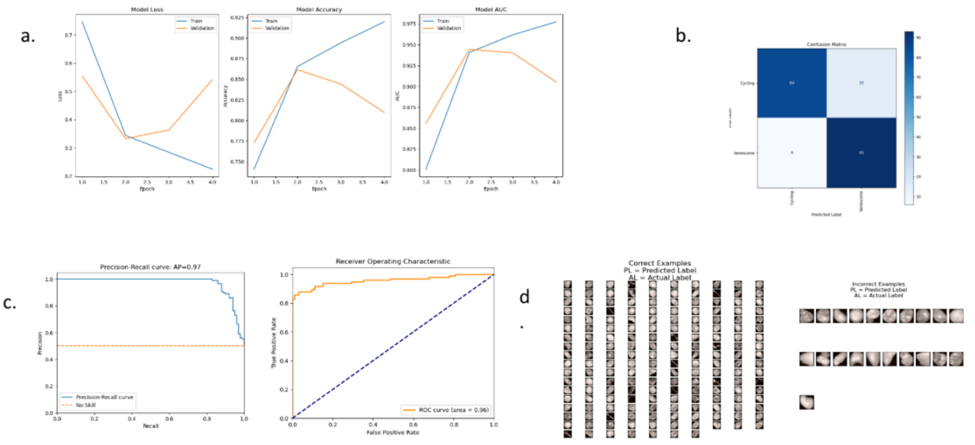


Fig 8. Model Performance of ResNet50 model Trained on Mice Tested on Mice. **a** Validation set model loss, accuracy and AUC over the 5 epochs are shown. **b** Confusion matrix of the model shown. **c** AUC of the Receiver Operator Characteristic and Precision-Recall curve **d** Images of correctly classified and incorrectly classified images are shown.

This figure presents our enhanced performance resulting from data stratification for long-term use. This method leads to our best performance to date with a AUC of 96%, a precision-recall score of 97%. Additionally we had an accuracy of 89%, precision: 89%, recall of 89%, F1 of 89%, with a baseline accuracy of 50% which instills confidence in the overall reliability of our model.

As you can see, each of these models performed better than the baseline, especially the ResNet50 model, regardless of the training and testing scenario, and offers an improvement over a random or majority classifier.

In conclusion, our research introduces SenID, which leverages CNN to identify senescent cells precisely by analyzing their nuclear morphology. Here we show that SenID can perform as well as other senescent cell classifiers in the field utilizing a fraction of the samples (Kusumoto et al., 2021). With around 1000 training examples, we achieved an accuracy of around 89% and an AUC of the ROC of around 95%. LIME saliency maps showed that the perimeter of cells contributed the most toward accurate classification. When evaluating our model's cross-species performance, we found that training on one species and testing on the other significantly reduced model performance. LIME saliency maps suggest that the perimeter of cells that had helped the model classify cells in single species classification contributed to the misclassification of senescent. This is likely due to the morphological differences between the different cell types, particularly cell size. We increased cross-species testing metrics by training on both species and subsequent testing on one.

Challenges:

Our primary challenge was collecting enough data, specifically senescent cell data, to perform our analysis. We could generate roughly 40,000 cycling cells but only 2,000 senescence cells. Since we wanted to have balanced data, this limited the number of cycling cells we could use and was the main bottleneck of our project, as the models do much better with higher amounts of training data. We also had trouble smoothing out some of the kinks that come with analyzing our images using attention maps in LIME. Further, issues arose with translating the raw images into tabular statistical data to use with our DNN. This meant we were unable to get our feature-based DNN working. With a few minor changes to the code, this model should be ready to implement. The named features outputted by pycelspranto_prototype will also allow us to understand better what features are being utilized by the model. Additionally, we are having minor issues with environment setup, such as utilizing GPU power via TensorFlow.

Reflection:

- How do you feel your project ultimately turned out? How did you do relative to your base/target/stretch goals?

Ultimately, our project exceeded our expectations, improving upon any previous related work and achieving scores and results that we did not think possible with the amount of data we collected. When training and testing on mouse datasets, we achieved an AUC value of 0.9. Our initial objective was to yield outcomes slightly worse than the presently published findings but incorporating cross-species identification. Our ultimate aim was to replicate these outcomes while utilizing data from different species, and our goal was to improve them. Our research represents a noteworthy advancement over previous studies, mainly due to our utilization of cross-species data and training techniques.

- Did your model work out the way you expected it to?

The performance of our model met, and at times even surpassed our expectations. However, we faced a considerable obstacle when generating tabular data for our DNN. Consequently, we could not incorporate the DNN into our framework, which was indeed disappointing. We were particularly interested in investigating the influence of numerical features on predictions. Regardless of the setback with the DNN, our custom CNN presented intriguing results. When pitted against the pre-trained ResNet50, it became clear that ResNet50 significantly outperformed our custom model. This result was not unexpected considering the pre-training advantage ResNet50 possesses. Despite falling short of ResNet50's performance, our custom CNN demonstrated learning aptitude, as indicated by its highest score AUC of 78% on testing both and training on both (not shown in results). The incorporation of cross-species data and specialized training methodologies played a substantial role in the success of our model. This clearly underscores the importance and efficacy of these techniques in our research.

- How did your approach change over time? What kind of pivots did you make, if any? Would you have done differently if you could do your project over again?

Initially, we intended to utilize only a custom CNN, however, researching pre-trained models led us to choose ResNet50. Furthermore, we continuously incorporated various features into our custom CNN such as grid search and initially included k-fold cross-validation. However, the latter was ultimately discarded due to its inadequate performance. This was also were we did most of our ablation studies, creating a config function where we could change the parameters of our models instantly and testing different parameters as well as removing certain features and seeing how performance changed. We found our current implementation to be the strongest with a batch_size of 64, 5 epochs, and a learning rate of le-5. We also explored alternative methods such as Xplique instead of Lime – which through ablation and parameter tuning studies found 1000 samples to be optimal – and implementing a DNN. Regrettably, these endeavors did not yield successful results, necessitating our adaptation and a shift in focus toward functional enhancements. For example, if we were to redo our project, we would have placed greater emphasis on gathering senescent data and reduced the cycling process to enhance our training procedures. Additionally, a more targeted project roadmap, as opposed to a general concept, would have been beneficial. Furthermore, a significant amount of time was allocated to rectifying and modifying existing implementations that either failed to improve performance or did not function as intended. Lastly, we aspire to make our code publicly accessible to other researchers. As such, we have ensured that it is well-commented, tidy, and organized, facilitating ease of use and comprehension.

- What do you think you can further improve on if you had more time?

Given additional time, there are several areas we would enhance in our project. Firstly, we'd aim to increase the volume of training and testing data and leverage more powerful hardware to expedite the training and computation processes. Our primary enhancement would be the inclusion of a multi-model approach. This would involve integrating a fully functional DNN that processes feature-based information with a further trained, pre-existing ResNet model, using our cell image data. This fusion of textual and cell image inputs in a multi-model system is designed to boost the model's predictive accuracy by forging a link between contextual features and images. As a result, we expect to improve the prediction of images and pinpoint the associated features that informed the decision-making process. Secondly, we'd explore the use of interpretability techniques like SHAP and LIME with the multi-model architecture to gain a richer visual understanding of our model's operation. Another area of improvement would involve more frequent use of our saved models, mitigating the wait time for training each time we refine our code or introduce new features. This practice would also streamline collaboration, making it simpler to share our work and results with others in the future. Lastly, we have already implemented a loading process for functions that employ a separate test set – this works by saving the models history and the model itself during training, overwriting any existing saved model if the validation loss is lower during any epoch – and we would continue to build on this foundation.

- What are your biggest takeaways from this project/what did you learn?

Our most significant takeaways from this project have been a greater understanding of deep learning and computational biology and a greater and more granular understanding of the morphological differences between senescence and cycling clues. Specifically, we were excited about our cross-species training and testing results as this showed that this was a possible application of our research, which could open many doors for medical research in the future. Ultimately, we are pleased with how our project turned out and excited to continue working on it!

References:

1. Gorgoulis, V., et al. "Cellular Senescence: Defining a Path Forward." Cell, vol. 179, no. 4, 2019, pp. 813-827. doi:10.1016/j.cell.2019.10.005.
2. Di Micco, R., et al. "Cellular Senescence in Ageing: From Mechanisms to Therapeutic Opportunities." Nature Reviews Molecular Cell Biology, vol. 22, 2021, pp. 75-95. doi:10.1038/s41580-020-00314-w.
3. Kusumoto, D., et al. "Anti-senescent Drug Screening by Deep Learning-based Morphology Senescence Scoring." Nature Communications, vol. 12, 2021, p. 257. doi:10.1038/s41467-020-20213-0.
4. K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770-778.