Intro page

This page is notes & should be removed in the final copy

* Introduction - Clearly state the problem you wanted to solve.
  + Be persuasive, and convince your readers that this problem is important.
* Related Work - List 2 or 3 related projects and how they dealt with the same or a similar problem
  + It is even better to choose peer-reviewed articles from journals or conferences related to your topic.
* Methodology - Outline the methods you took to deal with the problem you laid out in the introduction section.
  + This section must include:
    - At least one figure that illustrates your workflow.
    - Your figure may detail how data was collected and processed or how your methods were implemented and applied.
* Result - Outline your major results.
  + Tell your readers about the impact or usefulness of what you discovered.
* Conclusion - Summarize your project.
  + Then, point out the future directions or things can be improved or expanded upon in the future.

Introduction

* Ophthalmologist are rare. Very few exist
* Many (# here) diseases exist that can be detected through the eyes.
* Amount of eye known diseases 50-100 years ago
  + Unknown amount of diseases that could be tracked still
* Simple tests that could be less expensive

Related Work:

* 1 article that I used before in CSML
* The github repo for the 89% accuracy using CNN
  + <https://github.com/bsdr18/Image-Classification-on-Eye-Disease-Dataset>
* Article on how transformers outperform CNN

Methodology:

* Data
  + Dataset Used
    - <https://www.kaggle.com/datasets/gunavenkatdoddi/eye-diseases-classification?select=dataset>
    - Eye\_diseases\_classification
    - Contains (~1000 images of each):
      * Normal
      * Diabetic Retinopathy
      * Cataract
      * Glaucoma
  + Resizing + reason for resizing
    - The images were resized to (224x224) ??to accommodate the model being used??
  + Augmentation
  + Tensor conversion
  + RGB vs BW
    - RGB was used over black
* Training
  + Transformer architecture
    - ViT
  + Pre-trained model
  + Fine-tuning
    - Freezing layers
      * All = 48% accuracy
      * 2 layers = 94%
      * 5 layers =
* Testing
  + Base pretrained model results
  + Fine-tuned results