

**Jeffery Dirden, Jazmine Brown, Jovan Darby,
Katherine Stanton**

ITAI 2277

October 2, 2025

Phase 02 Report

In this assignment, we created a modest prototype for TriagePal, an AI-powered tool that analyzes eye photos and helps with preliminary triage for illnesses like conjunctivitis. Our goal was to create a complete data pipeline that included data gathering, preprocessing, model development, training, and testing, while keeping the prototype simple and controllable within Google Colab. We started by loading all of the packages required for image processing, data management, and deep learning. This featured OpenCV for picture reading and scaling, NumPy and Pandas for dataset management, Matplotlib for visualizations, and TensorFlow/Keras for convolutional neural networks (CNNs). We used KaggleHub to download a collection of conjunctivitis photos, which comprised both healthy and diseased eyes. We first looked over the dataset to confirm the classes and the quantity of photos available.

We next preprocessed the photos, shrinking them to 128x128 pixels and normalized the pixel values to a range of [0,1]. The labels were numerically encoded and divided into training and testing sets in an 80/20 ratio. We also generated class weights to balance the contributions of healthy and diseased photos, which is critical in small datasets to avoid bias toward one class. For the model, we used transfer learning with MobileNetV2 as a feature extractor, with a few dense layers on top to classify photos as healthy or sick. Initially, we trained the model for only 5-10 epochs to reduce runtime. However, predictions for test photos were incorrect, and images from the dataset were misclassified. This was to be expected given that the network had not yet learnt enough differentiating features after only a few training rounds. To overcome this, we

extended the training period to 30+ epochs, employed data augmentation to boost variety in the training data, and used class weights to handle imbalance. With these modifications, the model began to correctly identify both healthy and diseased eye pictures. This demonstrated the significance of sufficient training iterations and preprocessing consistency, even for a small prototype dataset.

Finally, we ran the model across two uploaded photos from our dataset. The images were scaled and normalized in the same way as the training data. The model accurately predicted one healthy image and one infected image, and the findings were shown alongside the photographs for visualization. This demonstrated that our modest prototype pipeline, which included data gathering, preprocessing, model training, and prediction, was viable and could be used as a foundation for future enhancements. Overall, this project allowed us to build and test a full image classification pipeline in Google Colab. While it is a simple prototype using a small dataset and a limited number of classes, it demonstrates the core workflow of an AI-based triage tool. Moving forward, increasing dataset size, adding more classes, and integrating text-based symptom analysis would enhance the system's accuracy and real-world applicability.