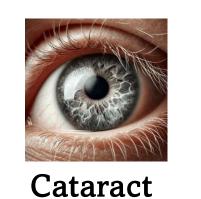# EyeNet

AI-Driven Early Eye Conditions Detection using Video Analysis

Eyes are a sensitive organ and exposed to infections and bacteria.
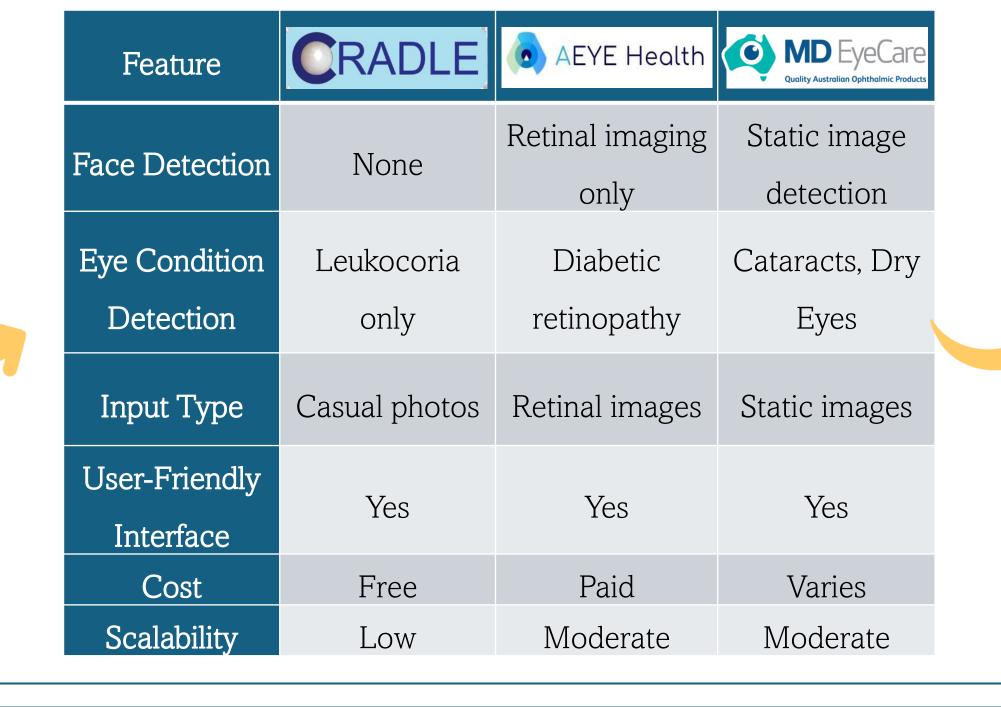This leads to external eye diseases like:

Cataract Conjunctivitis Stye

Problems with current situation

- ❌ Diagnosis is slow and expensive
- ❌ Requires specialist equipment
- ❌ Inaccessible in remote areas
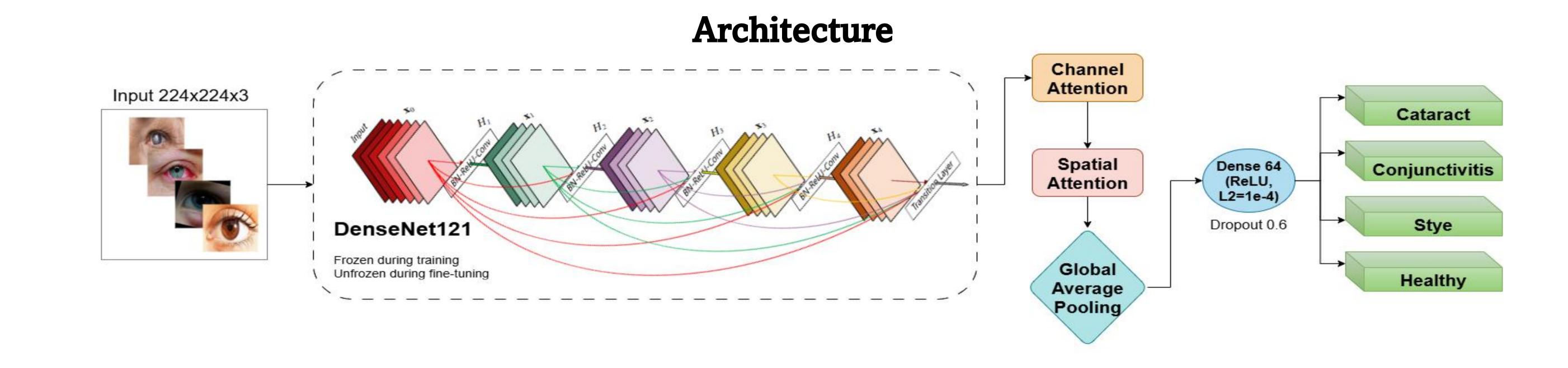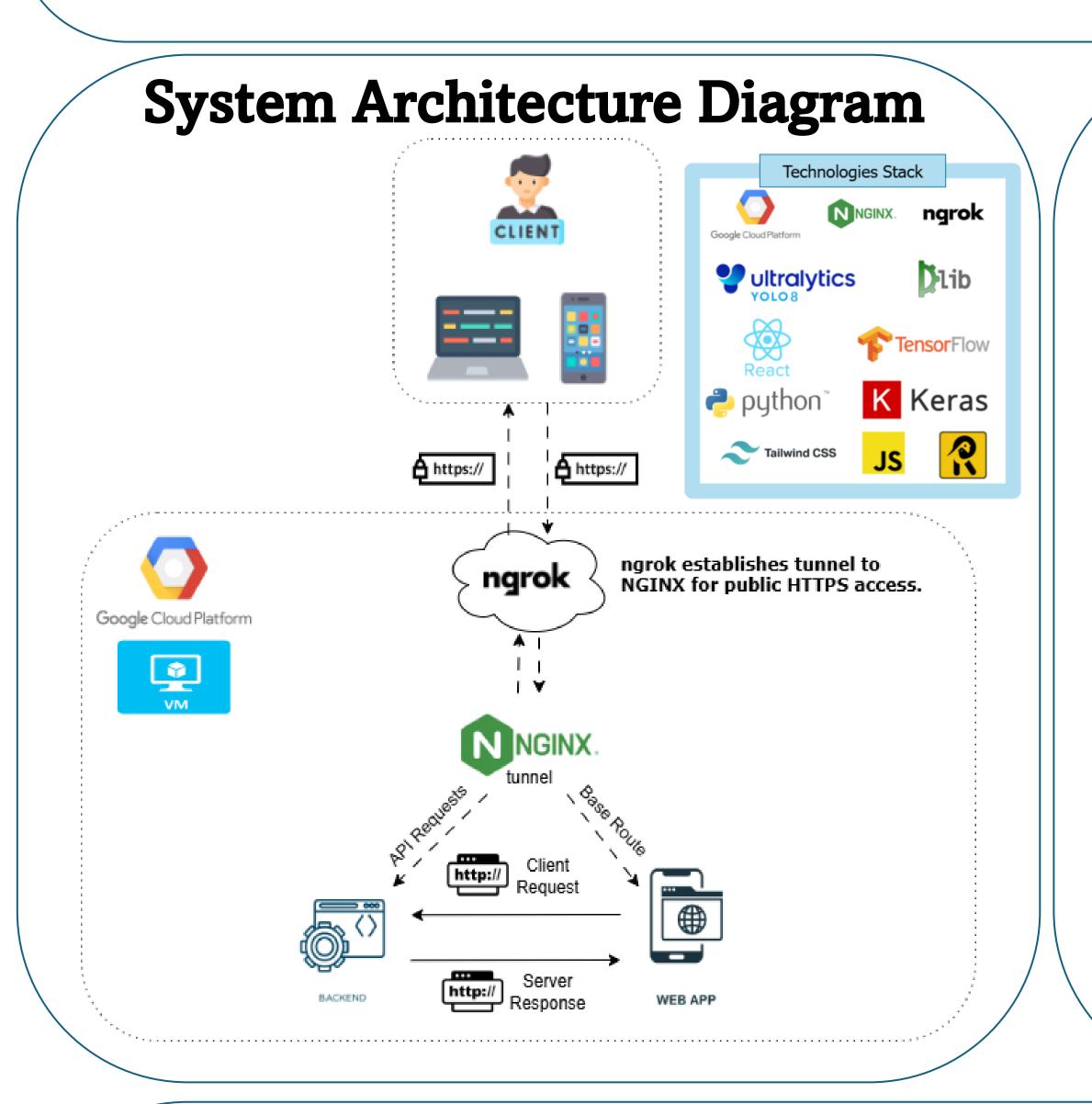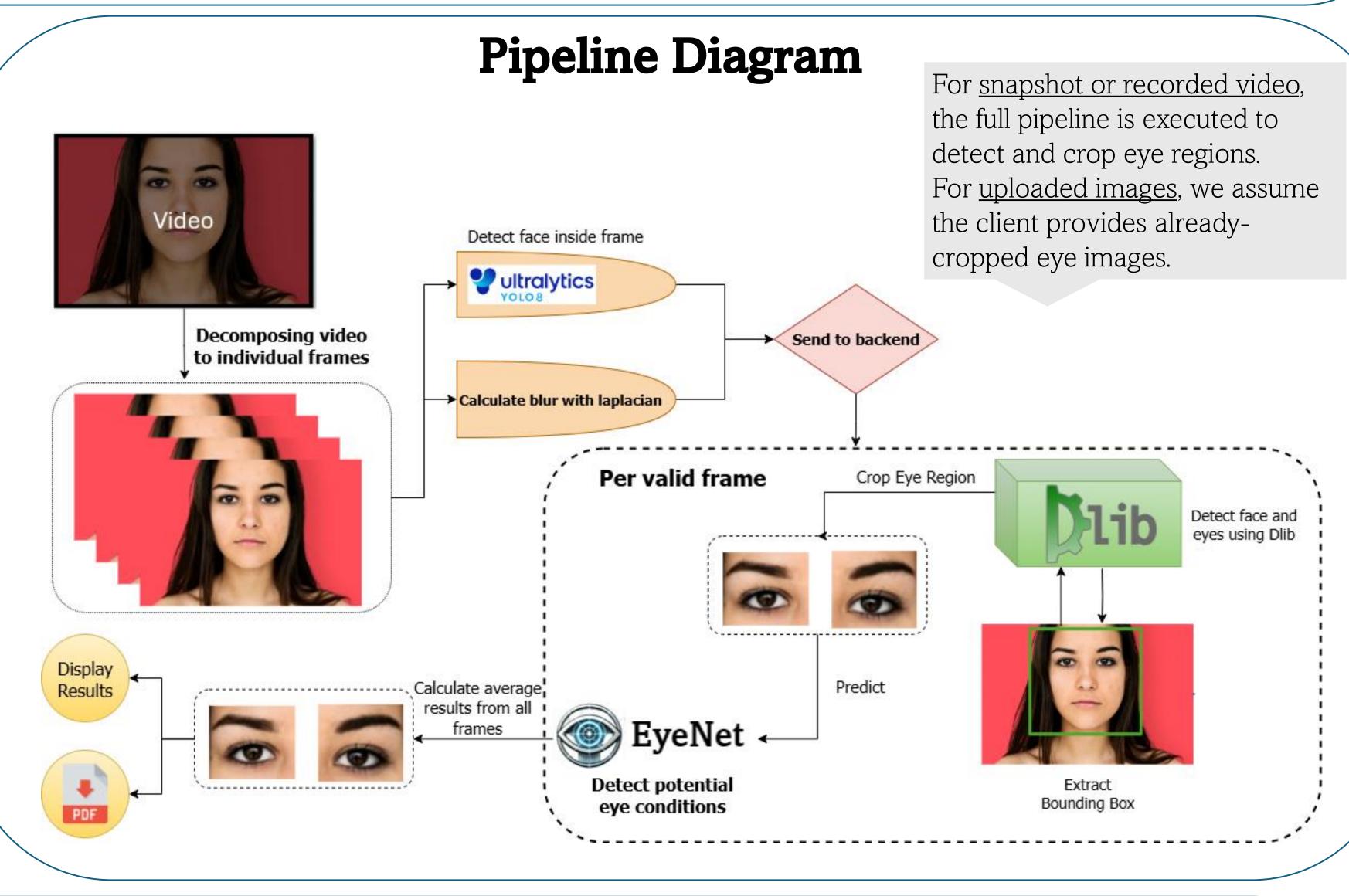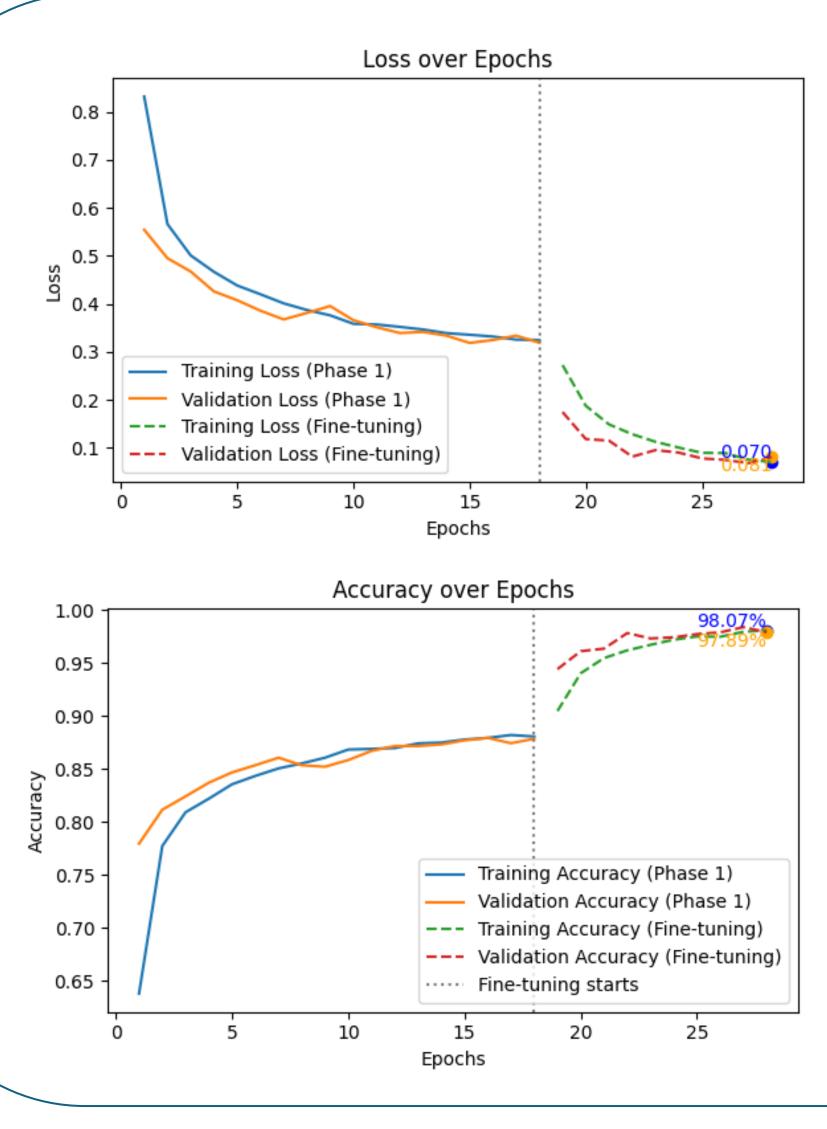- ❌ Patients often diagnosed too late

Current Solutions

Feature	CRADLE	AEYE Health	MD EyeCare
Face Detection	None	Retinal imaging only	Static image detection
Eye Condition Detection	Leukocoria only	Diabetic retinopathy	Cataracts, Dry Eyes
Input Type	Casual photos	Retinal images	Static images
User-Friendly Interface	Yes	Yes	Yes
Cost	Free	Paid	Varies
Scalability	Low	Moderate	Moderate

EyeNet

A **deep learning-based system** for identifying and classifying external eye conditions such as, **Cataracts, Conjunctivitis, Styes**. Delivered as an **end-to-end user-friendly** web application. The core model is based on **DenseNet121**, enhanced with **attention mechanism** and trained of a dataset of over 13K images per class

- ✅ Support both static (image) and dynamic (video) analysis
- ✅ Accepts external image or video of face
- ✅ Intuitive and easy for anyone to use
- ✅ Free to use, accessible and cross-platform
- ✅ Achieved ~98% accuracy in testing
- ✅ No specialist equipment requires

Architecture

Input 224x224x3 — DenseNet121 (Frozen during training, Unfrozen during fine-tuning) — Channel Attention — Spatial Attention — Global Average Pooling — Dense 64 (ReLU, L2=1e-4) — Dropout 0.6 — Cataract / Conjunctivitis / Stye / Healthy

System Architecture Diagram

Technologies Stack: Google Cloud Platform, NGINX, ngrok, ultralytics YOLO8, Dlib, React, TensorFlow, python, Keras, Tailwind CSS, JS, R

ngrok establishes tunnel to NGINX for public HTTPS access.

Pipeline Diagram

For snapshot or recorded video, the full pipeline is executed to detect and crop eye regions.
For uploaded images, we assume the client provides already-cropped eye images.

Decomposing video to individual frames — Detect face inside frame (ultralytics YOLO) — Calculate blur with laplacian — Send to backend — Per valid frame: Crop Eye Region, Detect face and eyes using Dlib, Extract Bounding Box, Predict — EyeNet Detect potential eye conditions — Calculate average results from all frames — Display Results — PDF

Results

The model learned to focus on the most relevant eye regions, improving accuracy from 91% to **98%**. We believe a larger dataset could further enhance performance.

As we expected, **cataract** was the most challenging condition to classify, while **stye** achieved the highest and most consistent performance.

Loss over Epochs — Training Loss (Phase 1), Validation Loss (Phase 1), Training Loss (Fine-tuning), Validation Loss (Fine-tuning)

Accuracy over Epochs — Training Accuracy (Phase 1), Validation Accuracy (Phase 1), Training Accuracy (Fine-tuning), Validation Accuracy (Fine-tuning), Fine-tuning starts — 98.07%

EyeNet Performance Metrics by Class — Metric: Accuracy, Precision, Recall (TPR), F1 Score, TNR

Students: Gal Bitton, Ron Bendel

Supervisors: Mrs. Elena Kremer, Dr. Dan Lemberg

BRAUDE College of Engineering, karmiel