# NutriSnap


## Hubert Liang, Elijah Davis

Project Sponsor: Lydia Sollis, Information & Computer Sciences Department


ICS 496

Spring 2024

## Introduction:

**NutriSnap** is an app that lets you take a picture of your meal, the app identifies the food, calculates how much is on the plate, and provides nutritional information. The process involves building a machine learning model that processes these images and uses APIs to give detailed calorie counts and food quality ratings based on the NOVA classification. The main goal of NutriSnap is to create an enjoyable and easy-to-use tool that helps people maintain healthy eating habits.

## Accomplishments:

**Model Learning Model Development & Integration**

- Our primary achievement was training and developing a food recognition model to classify consumables
- The model architect we followed was the InceptionV3 image classifier, and we also converted the model to TensorFlow Lite to be more optimized and compatible with mobile devices, reducing the need for constant internet access and enhancing privacy.
- Trained with the food-101 dataset, totalling over 101,000 images categorized of 101 categories with 1,000 images per category
- Metrics and validation include a top-5 accuracy result with addition of achieving visualization of different convolution and pooling layers

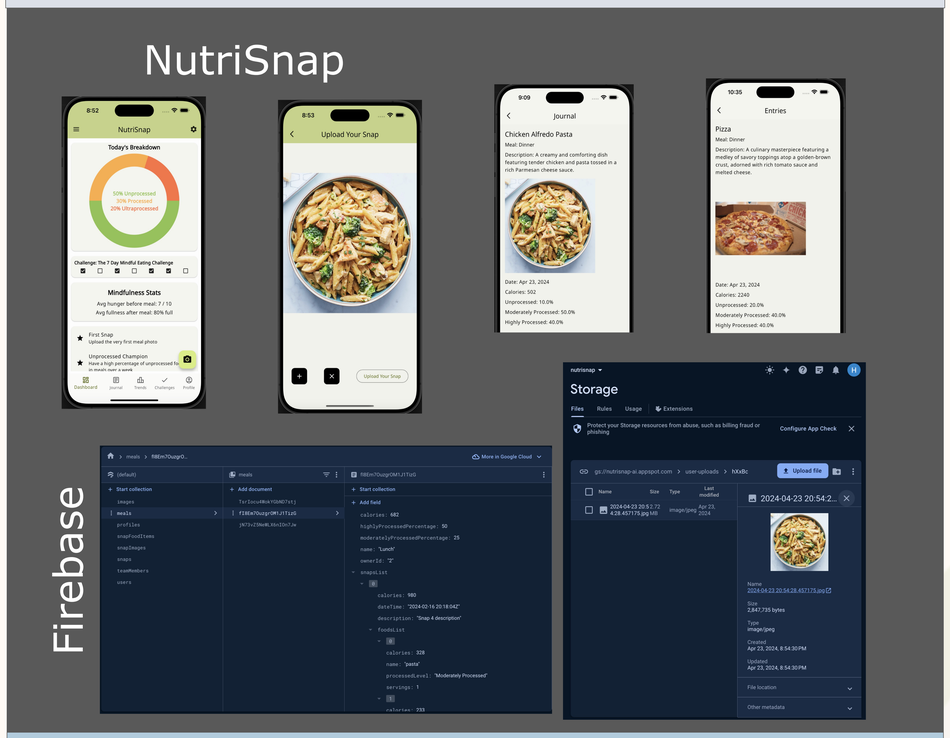
**Firebase Integration**

- Firebase Storage was utilized to store and manage user-uploaded images securely in addition to handling data retrieval, large files, and machine learning backend analysis
- Firebase Authentication allowed for seamless and secure sign-up and sign-in with email validation

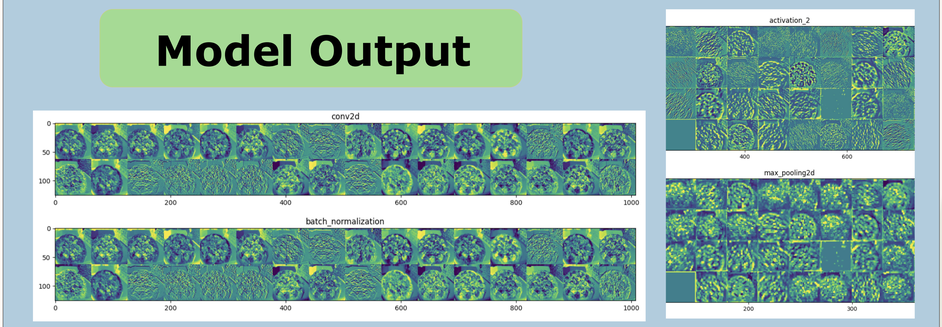
**Cross-Platform Adaptation**

- Flutter App development allowed functionality on iOS, Android, and MacOS devices without separate codebases

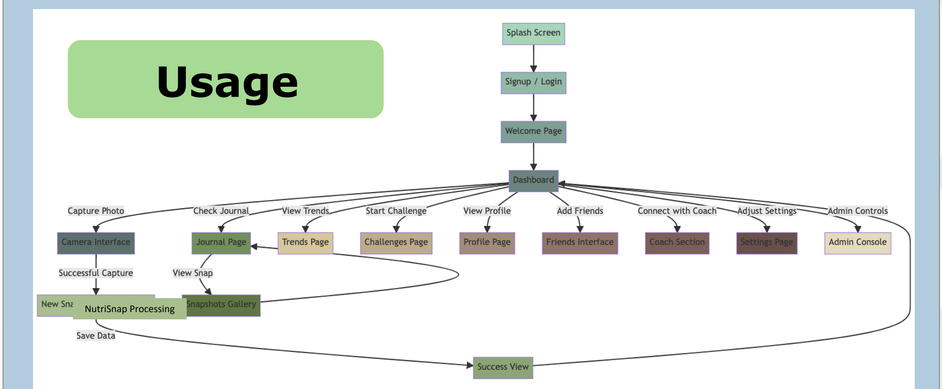
## Application




## Model Output




## Usage




## Technology Stack:

**Frontend:**

- **Flutter**: Cross-platform UI for Android, iOS, and MacOS devices
- **Dart**: Handles the primary functionality of the app, API calls, and renders the dynamic frontend user interface

**Backend:**

- **Firebase**: NutriSnap authentication, hosting, security, storage, and BaaS

**Mobile Platform Specific:**

- **Swift:** Helped to write platform-specific code and integrate with libraries that require native iOS code
- **Android SDK**: Necessary for certain Android-specific functionalities

**Build and Dependency Management:**

- **CMake**: Helped to build, test, and package software

## Issues:

**Model Optimization and Integration**

- The initial run for our model took over 998 minutes (trained on over 101,000 images) running on TensorFlow. Since then we adapted to TensorFlow Lite and integrated model optimizations with better batching, model freezing, regularization, and an optimized optimizer (via parameters) dropping the total runtime to approximately 232 minutes
- Integrating our model with Firebase took further steps we were unfamiliar with, such as saving and loading the model from Firebase storage to then perform inference on our app

## Next Steps:

**Extensive Nutritional Analysis**

- Further than attributing food items, NutriSnap should provide in-depth nutritional analysis through well-defined macros and more accurate processed values

**Testflight and Open Deployment**

- Allowing NutriSnap to be accessible for a wider audience and letting it shift over to the testing phase would allow invaluable feedback of functionality, usability, and performance