

Final Projects

Final project DUE: --/-- (when our final exam block time would end)

RFC Draft DUE: 11/21/2017 11:59PM EST

Overview

The final project in this class will require your team to leverage the industry-standard skills you've learned over this semester to design and implement a software system to acquire a signal, classify it using a cloud-based web service that you build (using deep learning or other machine learning approaches), and display the performance and outcomes of your system. This final project is somewhat open-ended to allow groups to tailor this to their areas of interest, however recommended datasets and project requirements are provided below.

If you plan to stray away from the recommended projects and datasets, please submit a one-page project proposal to Dr. Palmeri and Mr. Kumar by **11/17/17** for evaluation to ensure the proposed project meets the requirements for the class. Be sure to include motivations, technologies, functional specifications, and anticipated deliverables.

It is expected that your team will follow proper professional software development and design conventions taught in this class (git feature-branch workflow, continuous integration, unit testing, PEP8, etc. as detailed in sections below). At minimum, your system will be comprised of:

- A client device (the Raspberry Pi or equivalent) that will acquire a signal (image, biometric, etc), perform necessary preprocessing steps, and then issue a RESTful API request to your cloud service for further processing.
- A cloud-based web service that exposes a well-crafted RESTful API that will implement a statistical or machine learning model to classify the signal into a labeled bucket (for example, high/low cancer risk). If the project merits it, a database should be used for storing persistent application data.

Deliverables

- Final 1-2 page writeup describing the final performance and state of your project (including and statistical measurements of model accuracy)
- Demonstration video
- All project code (in the form of a tagged github repository)
- Link to deployed web service
- Any hardware block diagrams, circuit schematics, etc.
- Final RFC

Engineering Design Document

As discussed in class, before you start implementing your project, you and your team will be expected to flesh out a request for comments (RFC) document that will document the design, architecture, APIs, and edge cases for your system in advance. This document is traditionally reviewed (and commented on) by your colleagues in industry, and will be thoroughly reviewed by course instructors and TAs in this class.

You and your team should also thoroughly review the RFC to ensure everyone is on the same page about what is going to be built (and in what way using what technologies, etc).

You can find the Engineering RFC template at rfc.suyash.io.

Recommended Projects & Datasets

Melanoma or Not: Classification of Skin Lesions

Melanomas are readily treatable if caught early, with a 98% 5-year survival rate when treated with simple excision. The goal of this project is to develop a device that captures an image of a skin lesion and determines the likelihood that the lesion is melanocytic (and potentially whether it is malignant). Over 13000 annotated skin lesion images are available from the International Skin Imaging Collaboration (ISIC) project that can be used to develop machine learning models to classify new images. This dataset can be accessed [here](#). A zip of all annotated images can be downloaded by navigating to the Gallery and then clicking "Download as Zip" in the upper right hand corner. All data can also be accessed through a RESTful API provided by the ISIC.

For this project, you may wish to entertain certain image preprocessing tasks such as lesion image segmentation, edge detection, and other signal processing techniques you believe may enhance your signal.

ECG Analysis

PhysioNet hosts a multitude of annotated [ECG recordings](#). Your Raspberry Pi should be able to either receive analog input from an actual ECG circuit you build or from files on a USB drive. The backend web service should predict some annotation (for example, presence of a specific arrhythmia) given an input signal. For example, with the following [dataset](#), you may be able to diagnose sleep apnea given an input ECG trace.

Other Datasets

1. http://www.vision.caltech.edu/Image_Datasets/Caltech101/
2. <https://www.cs.toronto.edu/~kriz/cifar.html>
3. <https://github.com/beamandrew/medical-data>

Grading

You should approach this final project as an opportunity to show a potential future employer an example of your software development skills.

- RFC Document
- Git Repository
 - Issues/Milestones
 - Commits are discrete, logical changesets
 - Feature-branch workflow
- Software best practices

- Modularity of software code
 - Handling and raising exceptions
 - Language convention and style (PEP8)
 - Sphinx documentation for all modules/functions
- Testing and CI
 - Unit test coverage of all functions (except Flask handler)
 - Travis CI passing build
- Cloud-based Web Service
 - RESTful API Design
 - Validation Logic
 - Robust deployment on Virtual Machine
 - Machine learning functionality
- Client software and hardware functionality (running on a Raspberry Pi or equivalent)
- Demo of the final working project