BME 590 – Medical Software Design

Instructors: Dr. Mark Palmeri, Mr. Suyash Kumar

Melanoma Classification Project

Final Report

14 December 2017

Inje Lee, Jing-Rui Li, Niranjana Shashikumar

Final project requirement:
https://github.com/mlp6/Medical-Software-Design/blob/master/Assignments/final_project_fall_17/final_project_f17.pdf

Final RFC Document:
https://docs.google.com/document/d/10zwrlYD09Ox9IoRAPnOqVUvyigVB17V5fiFTYvP5VgI/edit?usp=sharing

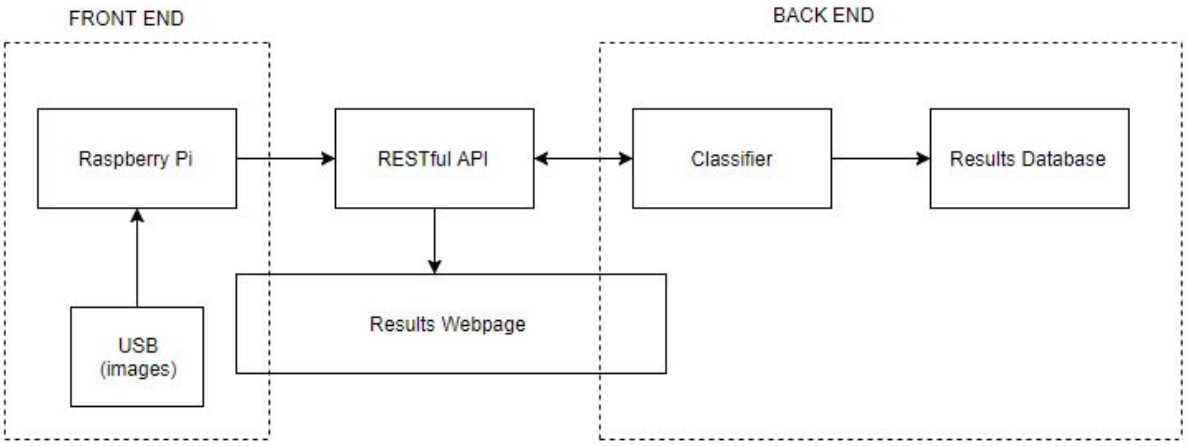
Introduction

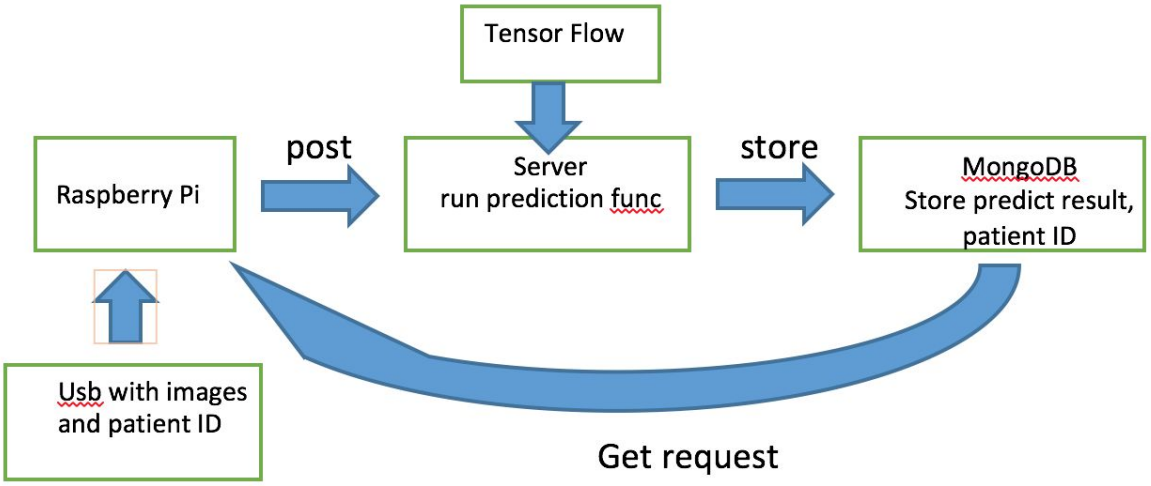
Melanoma is one of the most dangerous forms of skin cancer in which triggered mutations cause skin cells to multiply rapidly and form malignant tumors. Annually, Melanoma kills an estimated 10,130 people in the US alone. However, if Melanoma is diagnosed and treated early, it is almost curable, with a 98% 5-year survival rate when treated with simple excision.

The Melanoma classification project consists of a system computes the likelihood that a lesion is malignant by analyzing skin images, in order to facilitate early diagnosis of melanoma. A Raspberry Pi accesses images stored on a USB and sends them to a cloud-base web service that hosts a trained classifier. The results of the classification can then be accessed by visiting a web page.

Specification and Consideration

Final Design





1. Front-end

At the front end, we have a Raspberry Pi that access images hosted on a USB drive. The detected images are encoded as unique base64 strings. Each encoded image is then posted to the web server as key-value pairs of a dictionary at the URL : https:/vcm-2217.vm.duke.edu:5000/patient_classification. Currently, the Pi will only post to the web service it detects images on the USB.

2. Web Service

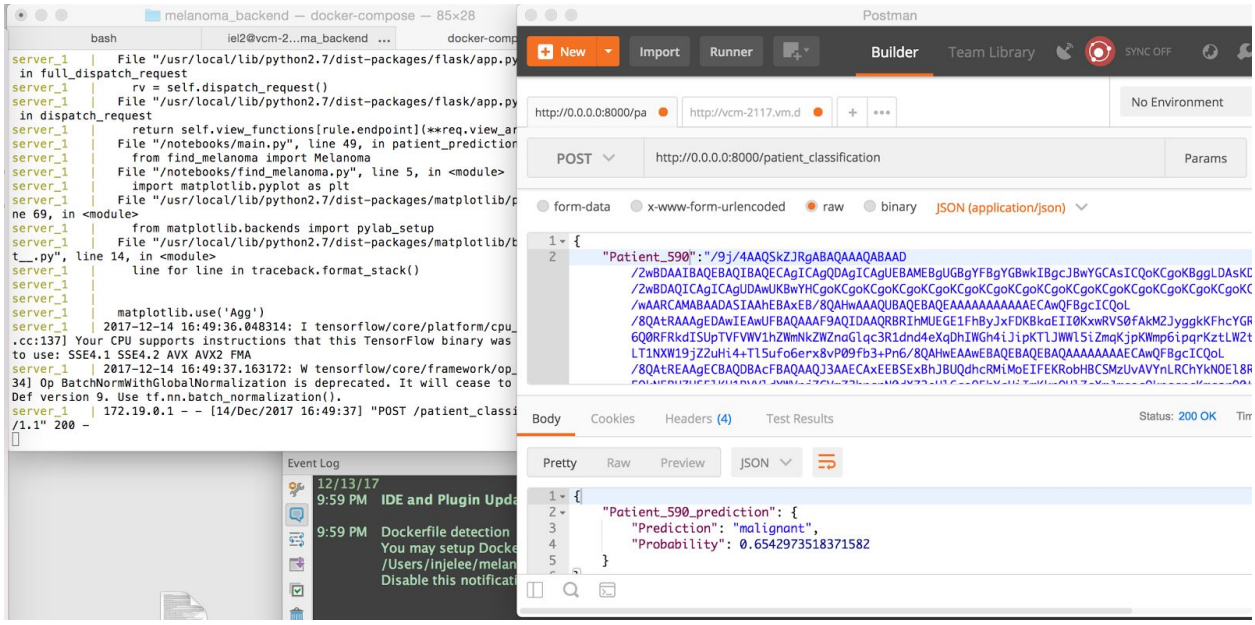
RESTful Flask API is used to connect the front and back ends together. There are two main functions in the Flask web service, the POST and GET methods. For the POST method, it receives the dictionary from raspberry pi and decodes base64 strings back to image in .jpg format. The image then becomes the input of the melanoma classification function and gives a prediction result as the output. The image ID (patient ID) and the prediction results will store in a MongoDB database. For the GET method, the user specified the patient ID and it will extract the corresponding prediction value from MongoDB database and return it to the web browser (specified web URL).

3. Backend

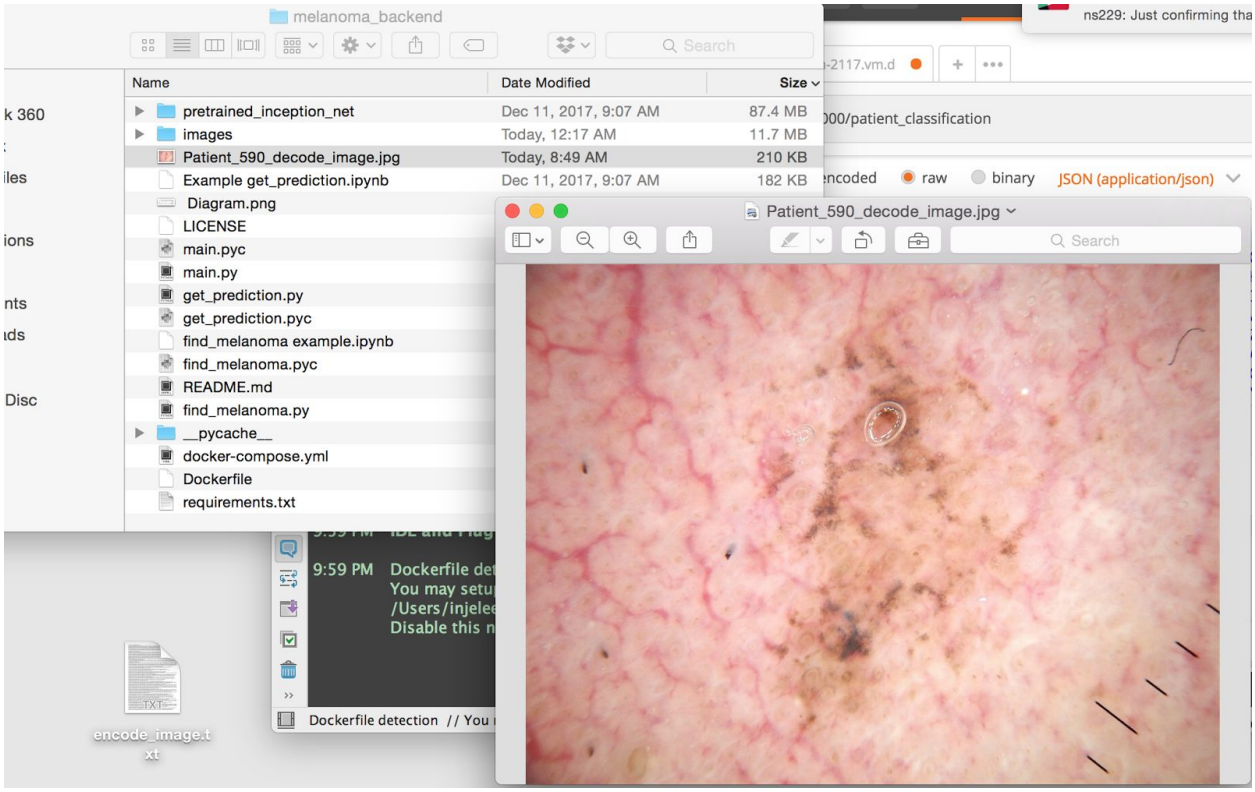
The MongoDB side of things is a work in progress. There have unfortunately been too many bugs and issues with the actual processing, which didn't afford Inje the full time to delve into the Mongo side of the repo. As of 12/14/17, the problem is a runtime error with the .save() mongo function, which doesn't allow for saving the structured dictionary into the created mongo database on our VM.

Performance

We can post a base64 image string to our VM and have it return predictions, should the input be in the correct format and not empty.

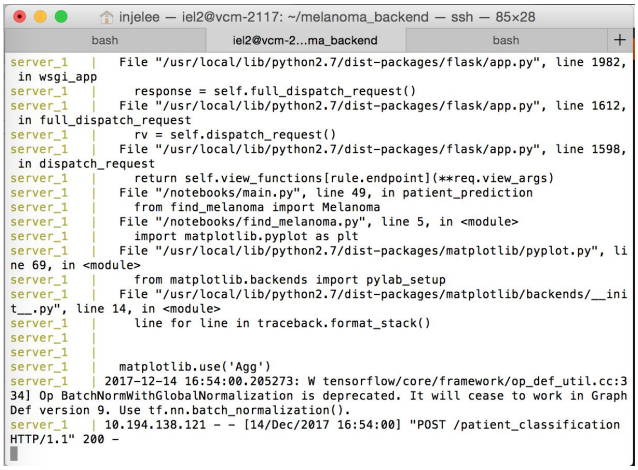
1) 

2) Successful image file save:



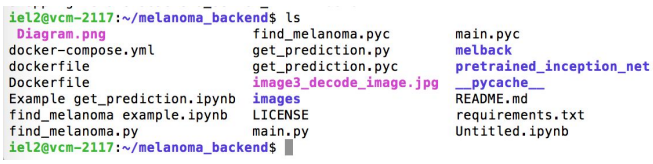
3) Full Stack Integration - Jing plugs in the Raspberry Pi at Duke, Niranjana posts up the encoded string from India, and Inje sees the response on his VM at Duke in California.

a)



b) Note the RPI IP address accessing the VM

c)



i) File saved as image3_decode_image.jpg on VM

Design Iteration

If there is no image at the front end, we return a response with error message from the POST and store it in the Raspberry Pi. However, the user needs to get access to the Raspberry Pi to be aware of that. Since there is a LED on the Raspberry Pi, we are thinking to report the status of the image with the light. If there is input image to the Raspberry Pi, the light is off (the state of the LED is LOW). If there is no image to the Raspberry Pi, the light is on (the state of the LED is HIGH). Currently, the melanoma classification function has only taken images as one feature to train. It does not consider other unique and relevant features such as statistical measures (mean, standard deviation and covariance) of the RGB color space and shape information (area, diameter, compactness and asymmetry). This data will be used in future classification training for the already existing classification model by utilizing either an averaging or logistic regression modification method with more features considered.

Citation

Suyashi BME590 Melanoma Detection: https://github.com/suyashkumar/bme590_melanoma_detection