Previous Goals:

Based on our previous progress, we had three main goals: finalizing the user interface prototype and gaining feedback from our project sponsor, continuing with image labeling and differentiation, and improving the performance of our neural net operator and trainer. For the user interface, we planned to complete the multiple branches based on potential user inputs so that this prototype could be exported from the prototype application into our diagnostic application once we begin to integrate all individual portions of our project together. Our team also wanted to work on classifying more images to be used within the neural net trainer to improve accuracy. Once this portion was done, the neural net would have improved accuracy in differentiation performance. These were our goals prior to switching our team focus to full project integration.

We have completed a first approximation of what we want our user interface to look like, and we currently have a functional model into which we will begin to integrate both the neural network and camera applications. Our sponsor, Dr. Kumar, has gone through our application and he has determined that besides small changes our user interface appears to meet the requirements to be used in the Burn ICU. Now with this working model we can begin the integration process while also updating the user interface to streamline the process and improve efficacy of the application. Additionally, we have increased the number of images to train our neural network and we have updated the code in order to correctly save the model after it has been trained.

Progress Since Last Time:

The Convolutional Neural Network trainer version 2 and predictor version 3 are up and running. Changes made in the trainer include increased training subimages and a different saving protocol. The previous save protocol saved in checkpoint format and did not save the full model. This was useful locally and allowed for the first two predictor versions. The new save protocol saves in a format that is compatible with Tensorflow mobile and the full model is saved in a .pb file for easy transfer between devices. The predictor now can consistently predict all 50x50 subimages within a larger image and overlay and display the original image with colors corresponding to the classification. The predictor shows between 7% and 17% error when given control images taken of group members' healthy skin.

We have completed the user interface using Builderx! After meeting with our advisor, we received edits and corrections which were implemented. All slides in the application, all buttons, and all visual pieces of user interface are now complete. Users should now be able to navigate the application. We are currently in the process of putting this user interface on a phone and integrating it with the smartphone camera.

Challenges Faced:

This week we moved into the subsystem integration phase of the project. The UI code generated by the BuilderX software, neural network model, TBSA calculation algorithms, burn resuscitation treatment algorithms, and other features of the project must all culminate in a single Android application. This process has presented two challenges over the past week.

First, the UI code generated by BuilderX is an older version of javascript which has slightly different syntax from that of the React Native libraries we will use for the project. We attempted to switch the libraries and React Native environment to an older version, but could find no solution that worked for all features. We have decided that the best solution is to perform a visual inspection on each of the UI files for syntax discrepancies and to build the UI code with the latest version of javascript that supports React Native. Second, the image processing and neural net features included in the TBSA calculation require functionality that is not yet present in current React Native libraries. Our solution is to write the algorithm in Java, a different programming language, and to write a javascript function wrapper such that the Java code can be called and executed from the javascript code run by our Android application.

Next Steps/Moving Forward

Moving forward, our goals include adding burn degree to the neural net, finishing the user interface prototype, and working to fully integrate into our front end design. Currently, our neural net is able to differentiate between healthy skin, burned skin, and background, but we hope to add an additional parameter, burn degree, by prompting the user to conduct a blanch test, which is the ICU convention used to determine burn degree. We are continuing to verify and revise our user interface design as we receive review android developer guidelines and feedback from our sponsor and, potentially, nurses from the Vanderbilt burn center. Nurse's feedback will be especially helpful to test the application's usability and functioning. Our most immediate objective is to completely integrate the tensorflow and flowchart functions into our user interface design, giving us a fully operating application. This also includes completing algorithms for calculations. Once we acquire the mobile phone on which to test with, we will deploy our application on the device in order to start test runs.