

Skin cancer detector

Nitin Bhaskar

Background:

Cancer needs to be detected early for successful diagnosis. With machine learning and AI, cancer can be detected at early stage. For detection using machine learning, a large dataset of images needs to be acquired on which training needs to be performed for prediction to work on input image.

Problem statement:

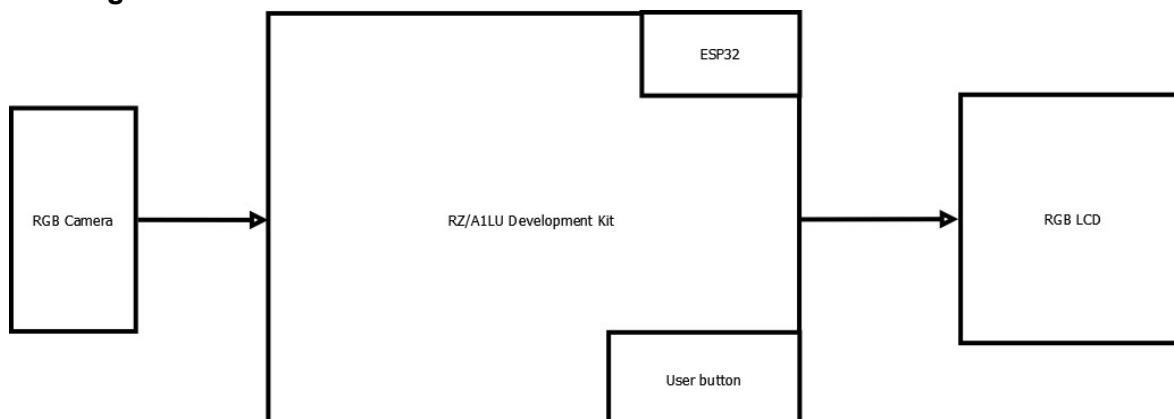
Melanoma is the deadliest form of skin cancer, responsible for an overwhelming majority of skin cancer deaths. Although the mortality is significant, when detected early, melanoma survival rate increases significantly.

Solution proposal:

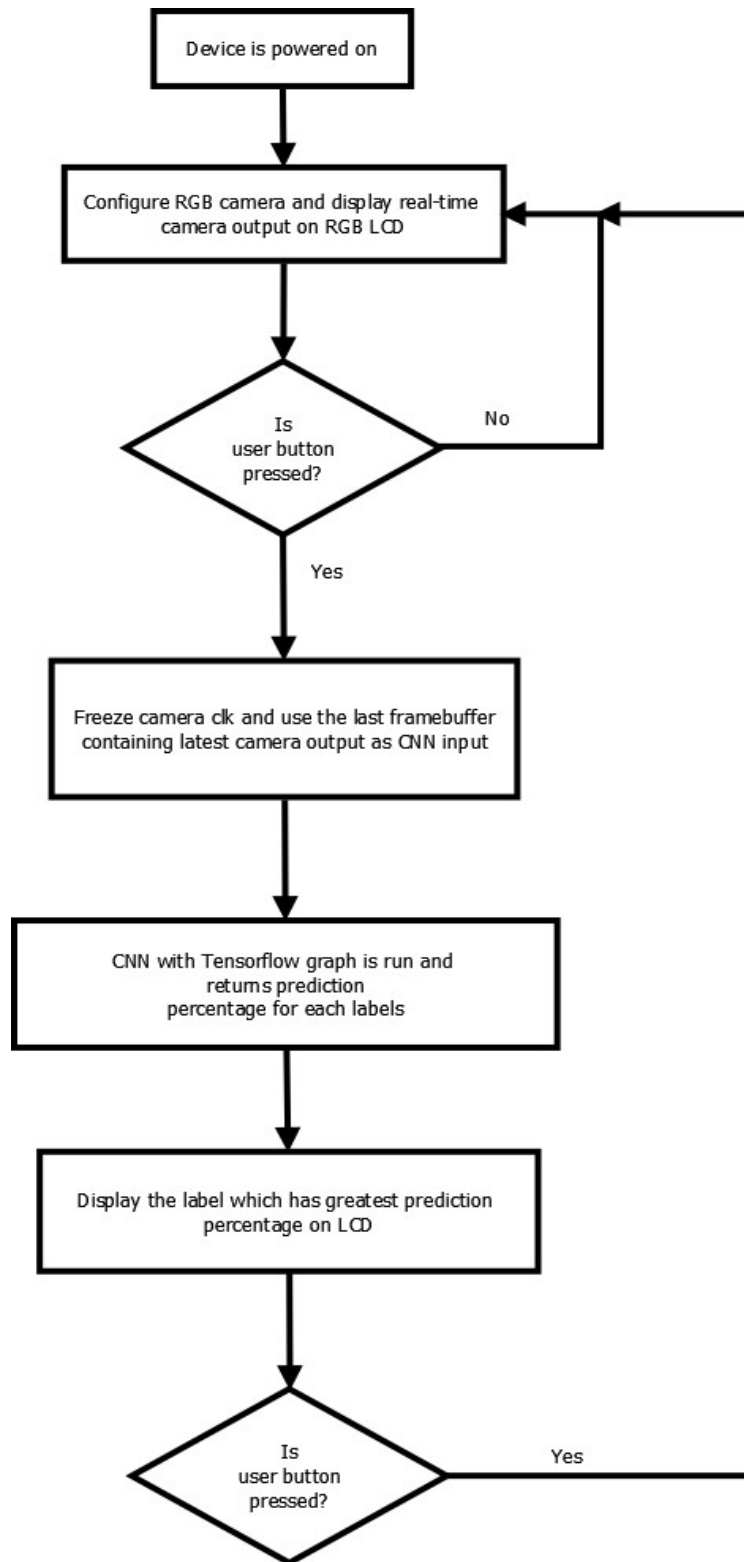
Convolutional neural network(CNN) is used on pre-trained model to predict the possibility of cancer. Image dataset for this is obtained from <https://isic-archive.com/>. There are plenty of images of affected skin available at ISIC website. All images are used to successfully detect Melanoma.

Tensorflow is used to perform training on image dataset on x86 machine with GPU. This model graph is fed to image classification program along with input image for prediction. To start with image classification program is written on Tensorflow framework and this Tensorflow session is saved on host machine and then fed as input to **e-AI translator from Renesas** (http://gadget.renesas.com/en/product/e-ai/mnist_tutorial.html). e-AI translator generates C-source and header files from Tensorflow session that is fed as input. Using these source files, image classification program is written to take input from camera and help user to select the area of interest by displaying the real-time camera input on to RGB LCD display. Once the image with affected area is captured by pressing user button, this image is fed to Convolution neural network and probability of Melanoma is displayed on the screen.

Block diagram:



Top level flowchart:



Bill of materials:

- RZ/A1LU Development Kit
- RZ/A Software Development Kit (SDK)
- IAR's Embedded Workbench for ARM
- FreeRTOS
- ESP32
- RGB LCD

Improvements:

Updating model graph over the air(OTA) using ESP32. A server is needed and RZ/A1LU Development Kit queries for model graph version number and updates its model graph stored in SD card over WiFi. Loading graph from SD card is something that e-AI does not support and this needs to be implemented.

Conclusion:

The scope of this project is very vast as this can be applied on almost all medical imaging domain and with proper training and tweaking, machine learning on RZ/A1LU Development Kit can do wonders.